UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI SCIENZE STATISTICHE

Corso di Laurea in
SCIENZE STATISTICHE, DEMOGRAFICHE E SOCIALI

TESI DI LAUREA

Previsione dei consumi idrici per la città di Padova

LAUREANDO:
Roberto Nalon

RELATORE:
Chiar.mo Prof. Guido Masarotto

Anno Accademico 2008/2009

Indice

Introduzione

L’acqua è elemento vitale. Nell’antichità il problema dell’approvvigionamento idrico era affrontato a livello quasi individuale o familiare: ciascun nucleo provvedeva alle proprie necessità in modo spontaneo e diretto. Sporadici esempi di acquedotti erano stati sperimentati nel mondo greco e orientale. Una vera e propria rivoluzione venne introdotta dai Romani, i quali per la prima volta considerarono questo un problema sociale, che come tale andava affrontato e risolto: l’acqua, captata e incanalata in condotti, veniva trasportata dalle sorgenti, a volte lontane decine di chilometri, fino al cuore delle città, per essere poi distribuita in modo sempre più capillare al servizio della comunità. Nasce così l’*acquae ductus*, dietro la cui semplice definizione di *conduzione dell’acqua*, si celano diverse complesse operazioni che, con il calcolo preciso di portata, dislivelli, pendenze, distanze, rendono possibile il trasferimento di cospicue quantità d’acqua dalle sorgenti alla distribuzione. È acqua incanalata e condotta nei punti desiderati per essere direttamente utilizzata: è un vero e proprio servizio pubblico ed in quanto tale pone una serie di problemi di carattere sociale, economico, giuridico e legislativo. L’acqua è di proprietà dello stato e quindi, una volta assolte le necessità della casa imperiale, viene messa a disposizione di tutti i cittadini attraverso una serie di grandi e piccole strutture termali ed una rete capillare di fontane, distribuite quasi ad ogni incrocio stradale.
Sul modello di Roma, ogni città si dotò di uno o più acquedotti, in relazione all’estensione e alla conformazione dei centri urbani: queste strutture si possono considerare in qualche modo omogenee poiché, pur attraversando territori molto diversi tra loro, rispondono tutte alle medesime esigenze di trasporto dell’acqua da sorgenti o altri punti di presa, quali fiumi o laghi, fino alla distribuzione. Indispensabile è che il condotto mantenga sempre una determinata quota ed un’inclinazione o *declivitas* la più possibile costante, per assicurare l’agevole scorrimento dell’acqua, a pelo libero all’interno del canale, senza sbalzi, né vorticosità, né problemi di pressione.

L’approvvigionamento idrico a Patavium dalle origini ad oggi

L’antica Padova, attraversata dal fiume *Medoacus*, si trova ad una quota modesta sul livello del mare (10-12 metri) ed in un territorio ricco di falde superficiali. Inoltre le aree limitrofe sono, oggi come in passato, importanti riserve d’acqua: oltre ai vicini rilievi dei Berici e degli Euganei a ovest e sud-ovest, ricordiamo in particolare

l’ampia fascia delle risorgive a nord/nord-ovest nella pianura tra Dueville, Cittadella e Camposampiero, ad una quota sul livello del mare variabile tra 50 e 25 metri.
Le prime fonti di approvvigionamento furono sicuramente il fiume e le falde superficiali, facilmente raggiungibili mediante lo scavo di pozzi: questi furono costruiti con tecniche molto simili a quelle comunemente usate nel mondo romano e continuarono certamente ad essere in uso anche dopo la realizzazione di acquedotti. Purtroppo tutti i rinvenimenti di reperti archeologici relativi al sistema acquedottistico patavino sono avvenuti in modo occasionale e non forniscono il quadro complessivo necessario per conoscerne le origini. Sappiamo solo che la realizzazione dell’acquedotto potrebbe essere collocata tra il 45 a.C. e l’epoca augustea. Nel periodo medievale, eventi catastrofici quali il diluvio del 589 d.C. e la distruzione della città da parte del re longobardo Agilulfo nel 601-602 d.C., hanno portato la popolazione, ridotta numericamente ed impaurita, ad abbandonare ogni genere di controllo sulle opere pubbliche. Le fonti di approvvigionamento diventarono il fiume ed i pozzi. Con l’inizio dell’epoca comunale, tra il X e XI secolo, si accentuò ancor più il rapporto tra il fiume e la popolazione, sia con la sistemazione di terrapieni e fortificazioni, sia con l’apertura di nuovi corsi d’acqua, che servivano capillarmente la città e le sue attività produttive, tutte basate, sia pure in modi diversi, sullo sfruttamento dell’acqua. Tutte le necessità di approvvigionamento idrico per l’uso personale erano assolte dai pozzi, che esistevano in gran numero, pubblici e privati, e che continuarono ad essere in uso fino alla costruzione nel 1888 dell’acquedotto moderno. Solo la scoperta di nuovi materiali, in seguito alla rivoluzione industriale, portò ad una sostanziale modifica del modello allora applicato per giungere alle moderne linee di approvvigionamento. La città si apprestava ad una radicale trasformazione delle abitudini quotidiane: avere l’acqua disponibile in casa significava risolvere, con una comodità quasi insperata, molti dei problemi igienici ed alimentari. Non sarebbe stato più necessario raccoglierla, conservarne scorte e, soprattutto, sarebbe stata realmente *potabile*, cosa che numerosi tra i pozzi cittadini non potevano più garantire. Si scelse di emungere l’acqua dalle risorgive delle fonti di Dueville nel vicentino e di affidare la gestione dell’acquedotto ad una società privata con una concessione sessantennale. Tra i problemi affrontati vi era quello della grande distanza tra l’opera di presa a Dueville e la città di Padova, di circa $42\,km$. La presa fu realizzata mediante la costruzione di una galleria e due vasche connesse a due canali ai quali convergevano i tubi dei 131 pozzi infissi nel 1888. L’adduzione a Padova era progettata con una condotta in muratura, chiamata *canaletta*, funzionante, salvo pochi tratti, a *pelo libero*, ossia con occupazione parziale della sezione, lasciando quindi sul cielo della condotta un certo

volume d'aria. Il percorso complessivo fu di $42.353\,m$, con un dislivello di $25\,m$ e una pendenza media di $58\,cm/km$. L'acquedotto fu inaugurato il 13 giugno 1888 e dopo appena 5 anni venne riscattato dal Comune che provvide ad ampliarne la rete di distribuzione. Fino al 1959 la *canaletta* fu l'unica fonte di approvvigionamento della città ed è tuttora in funzione, unitamente al Nuovo Acquedotto. Nel **primo capitolo** del presente lavoro è riportata una descrizione della realizzazione del primo acquedotto moderno e degli sviluppi del sistema di adduzione fino ai giorni nostri.

Modelli di previsione dei consumi idrici

L'affinamento di metodologie e tecniche idrauliche insieme alle moderne scienze matematiche ha consentito di effettuare studi sempre più completi e complessi in relazione ai consumi idrici e alla loro previsione. Lo studio dei modelli per la previsione dei consumi idrici urbani ha di recente assunto notevole interesse per la moderna gestione delle reti di distribuzione. In questo contesto va inserito il modello realizzato nel 1982 dall'azienda municipalizzata dell'acqua di Padova AMAP, ora Acegas-Aps, che incaricò l'ing. Francesco Davanzo di analizzare i dati sui consumi idrici per costruire una previsione che consentisse di razionalizzare l'utilizzo delle risorse energetiche necessarie per l'erogazione dell'acqua alle utenze cittadine, utilizzando in modo ottimale i sistemi di pompaggio indispensabili per i processi di adduzione e distribuzione. Il modello realizzato è stato impiegato nell'esercizio degli impianti ed ha portato nell'anno 1982 ad una diminuzione dei consumi di energia elettrica impiegata dall'AMAP del 5% a fronte di un aumento nella produzione dell'acqua del 2,5%.

La bontà di questo modello è stata confermata dal lavoro dall'ing. Danillo Calaon, anch'egli della medesima azienda, che nel 2003 ha verificato il modello alla luce dei nuovi dati raccolti.

Moderne tecniche statistiche unitamente alla disponibilità di potenti elaboratori hanno consentito la realizzazione di nuovi modelli basati su reti neurali artificiali. In questo ambito si inserisce il lavoro di Campisano *et al.* che, utilizzando reti neurali stratificate di tipo *feedforward*, realizza una serie di modelli di previsione dei consumi idrici a 24 ore e a 7 giorni.

I modelli di Davanzo, la rivisitazione operata da Calaon e il modello basato su reti neurali di Campisano *et al*. verranno approfonditi nel **secondo capitolo** del presente lavoro.

A seguito di processi di trasformazione aziendale, l'AMAP è oggi parte di Acegas-Aps nella quale lo scrivente è impiegato all'interno dell'Area Sistemi

Informativi. L'Azienda ha mantenuto, nel divenire degli anni, un interesse all'ottimizzazione della gestione delle risorse consentendomi di utilizzare i dati ricavati dal sistema di telecontrollo degli impianti tecnologici. Tale sistema viene utilizzato per l'esercizio dell'erogazione dell'acqua potabile e registra in tempo reale numerosi parametri di funzionamento che sono accuratamente verificati e validati. Su autorizzazione di Acegas-Aps si dispone dei dati provenienti dal sistema di telecontrollo relativi al periodo 1995-2008, nonché dei risultati prodotti dai lavori di Davanzo e Calaon.

Nel **terzo capitolo** del presente lavoro sono riportate le elaborazioni, i risultati e le conclusioni sulle analisi operate sul *dataset* a disposizione.

Dopo aver recuperato i dati relativi ai consumi idrici registrati nel periodo 1995-2008, si è proceduto alla loro verifica, alla costruzione delle variabili necessarie all'analisi dei dati e successivamente all'elaborazione di alcune statistiche descrittive. L'insieme dei dati è stato poi suddiviso in due sottoinsiemi: il *training set*, contenente i dati dal 1995 al 2006, utilizzato per costruire i modelli di previsione e il *validation set* per verificare la bontà delle stime prodotte su un insieme di dati diverso.

Seguendo le linee guida tracciate da Davanzo sono stati costruiti i modelli Acegas-Aps, allo scopo di valutarne le prestazioni sui dati attuali.

È stata quindi costruita una serie di modelli basati su regressione lineare, su reti neurali artificiali e sull'algoritmo *Random Forests* per la previsione dei consumi giornalieri, introducendo le variabili via via considerate e confrontando i risultati ottenuti nei vari modelli.

Sono stati costruiti quindi i modelli settimanali basati su reti neurali e su *Random Forests* per la previsione dei consumi dei sette giorni seguenti.

I risultati ottenuti dai vari modelli elaborati sono stati confrontati sulla base di alcuni parametri di bontà dell'adattamento e della loro capacità di generalizzazione, utilizzando quindi rispettivamente il *training set* ed il *validation set*. In particolare, sono stati calcolati gli errori relativi delle stime prodotte.

Nel **quarto e conclusivo capitolo** sono presentati gli aspetti metodologici della teoria delle reti neurali artificiali e dell'algoritmo *Random Forests*.

Conclusioni

A seguito del lavoro svolto è possibile trarre alcune considerazioni. Tutti i modelli elaborati si adattano abbastanza bene alla realtà di interesse: non sono evidenti distanze marcate sulla capacità predittive dei modelli. Ciononostante è possibile

individuare come modelli con capacità predittiva migliore quelli basati su reti neurali, a seguire quelli basati su *Random Forests*, quindi quelli di Acegas-Aps ed infine quelli basati su regressione lineare.
Le differenze tra le famiglie di modelli sono più evidenti quando si confrontano i modelli giornalieri. In particolare i modelli giornalieri di Acegas-Aps sono penalizzati dalla loro metodologia costruttiva che parte da una previsione di stima settimanale per giungere a quella giornaliera.
È interessante osservare come, contrariamente ad ogni aspettativa, la variabile temperatura introdotta nel secondo modello Acegas-Aps diminuisca la capacità predittiva dello stesso anziché aumentarla.
Con particolare riferimento alla previsione settimanale, era giustificabile pensare che i modelli Acegas-Aps riportassero prestazioni nettamente inferiori rispetto agli altri basati su metodologie costruttive e teorie statistiche consolidate. Ciò non è avvenuto. Si osserva come i modelli settimanali di Acegas-Aps, costruiti su osservazioni empiriche ed esperienza, abbiano dimostrato un livello di adattamento ai dati appena inferiore agli altri.
Infine si ritiene importante evidenziare come i modelli di Acegas-Aps presentino il vantaggio di avere una logica costruttiva ed un comportamento noti: l'elaborazione dei dati in ingresso per produrre l'output del modello è trasparente consentendo delle analisi e delle eventuali modifiche ragionate. Tanto nei modelli basati su reti neurali quanto in quelli basati su *Random Forests* si ha per contro la tipica struttura a *black-box*: sebbene gli algoritmi costruttivi siano noti il funzionamento del modello non è trasparente consentendo all'analista solo margini di manovra ridotti per ricercare un'eventuale miglioramento delle prestazioni.

Ringraziamenti

Desidero innanzitutto ringraziare il Professor Masarotto per i preziosi e appassionati insegnamenti trasmessi nel corso degli anni di studio e per la disponibilità e cortesia dimostrata. Inoltre, ringrazio sentitamente l'azienda Acegas-Aps, in particolare il Dr. Baroncini, l'Ing. Calaon, il Dr. Pudilli, l'Ing. Degrassi, il P.I. Brazzarola e il P.I. Zuin, per avermi fornito testi, dati e consigli indispensabili per la realizzazione del presente lavoro. Ho desiderio di ringraziare con affetto i miei genitori per il sostegno ed il grande aiuto che mi hanno dato e per la pazienza dell'attesa. Un'attenzione particolare la dedico alle mie figlie Francesca e Federica e a mia moglie Emanuela per avermi rispettivamente consentito e spronato a terminare quest'impresa.

Capitolo 1
-
Il sistema acquedottistico di Padova

La città di Padova è dotata di un importante sistema acquedottistico. In questo Capitolo se ne illustrerà brevemente la sua storia, dalle origini fino ai giorni nostri.

1.1 Le origini dell'acquedotto moderno

Ogni vicenda di lungo periodo come la nascita della Padova moderna, sconta fasi di pausa e di accelerazione, legate a fattori politici, economici e demografici che ne caratterizzano fortemente l'andamento. Nell'ambito di questa cornice storica, è indubbio che l'annessione della città al regno d'Italia abbia determinato un radicale cambiamento nei modi e nelle forme della trasformazione urbana, soprattutto grazie all'avvento della borghesia imprenditoriale alla guida dell'amministrazione, ed agli strumenti d'intervento concreto che questa conseguì.
Una delle principali opere pubbliche realizzate in quella nuova stagione è stato il primo acquedotto moderno cittadino.
Le scelte operate dal Comune dopo l'annessione furono pressoché coerenti fin dai primi anni: a partire dal 1875 si delegò ad apposite commissioni lo studio delle acque potenzialmente utilizzabili e la valutazione dei progetti, la cui definizione è demandata principalmente all'iniziativa privata. Successivamente, sentiti i pareri scientifici e valutate le proposte tecniche, sarebbe rimasta a carico del Comune la scelta dell'acqua con cui alimentare l'acquedotto, del progetto da realizzare e la formula di gestione.
La scelta per le sorgenti o per le risorgive, che per la distanza delle fonti comportava la necessità di investimenti cospicui, avrebbe tuttavia posto il Comune di fronte all'alternativa tra contrarre un prestito per finanziare l'opera e le spese dei primi anni di gestione, o affidare ad un concessionario i vantaggi ed i rischi di un investimento che prometteva utili solo a medio termine.
Le decisioni definitive furono prese nel 1885, con la scelta delle fonti di Dueville e Camisino nel vicentino e la concessione sessantennale del diritto esclusivo di

fornitura d'acqua potabile alla Società Veneta per Imprese e Costruzioni Pubbliche, in cambio della costruzione dell'acquedotto. La città si apprestava ad una radicale trasformazione delle abitudini quotidiane: avere l'acqua disponibile in casa significava risolvere, con una comodità quasi insperata molti dei problemi igienici ed alimentari. Non sarebbe stato più necessario raccoglierla, conservarne scorte e, soprattutto, sarebbe stata realmente *potabile*, cosa che numerosi tra i pozzi cittadini non potevano più garantire (Maffei, 2001).

La Società Veneta si impegnava ad addurre a Padova dalle sorgenti di Dueville (Vicenza), non meno di $5.000\,m^3$ d'acqua al giorno da distribuire all'intera città con un carico minimo di 16 metri sopra la soglia del Palazzo comunale (il punto più altro della città) (Da Peppo, 2001).

Gli abitanti di Padova, nel 1886, erano circa 60.000: la dotazione (quantità di acqua giornaliera per ogni abitante) era quindi prevista di $83 l/giorno$ per abitante, valore questo valido nel caso, non reale, che tutta la popolazione venisse servita dall'acquedotto. Il Comune si impegnava a pagare un canone annuo di 25.000 lire per l'uso di $500\,m^3$ giornalieri per i pubblici servizi. La Società poteva cedere l'acqua ai privati a 25 centesimi al m^3. Il canone pagato dal Comune sarebbe diminuito se fosse incrementata la vendita ai privati finché, superati i $3.500\,m^3$ giornalieri, la Società doveva pagare al Comune 10 centesimi per ogni m^3 venduto in più.

Lo schema progettuale proposto era sostanzialmente di tipo classico: opera di presa per una portata massima di $20.000\,m^3$ al giorno; condotta di adduzione e consegna delle portate ad una capacità terminale. Vi era tuttavia un problema di notevole rilievo: la grande distanza tra l'opera di presa a Dueville, posta a nord di Vicenza, e la città di Padova, circa $42\,km$ (Figura 1.1).

Gli elementi essenziali del progetto furono i seguenti. L'alimentazione dell'acquedotto era prevista con acqua attinta da pozzi di risorgiva. La condotta d'adduzione, tradizionalmente denominata *canaletta*, si dirigeva da Dueville verso Vicenza con deflusso della portata a pelo libero secondo un tracciato parallelo alla linea ferroviaria. In prossimità di Vicenza era collocato un manufatto partitore che doveva servire per suddividere la quota parte dell'acqua da avviare a Vicenza e quella da avviare a Padova. L'acquedotto per Vicenza non fu però realizzato dalla Società Veneta.

Da Vicenza a Padova il tracciato della canaletta si doveva svolgere lungo il percorso della strada provinciale Vicenza - Padova, con deflusso a pelo libero, salvo alcuni tratti e manufatti particolari. A Padova, nella zona di fronte alla stazione ferroviaria, erano collocati due serbatoi, detti serbatoi bassi, dai quali l'acqua doveva essere

sollevata ad un serbatoio, da costruire sul bastione dell'Arena, per la distribuzione cittadina. Successivamente, in sostituzione del serbatoio dell'Arena, furono previsti due serbatoi soprelevati, uno all'Arena ed uno a Porta Saracinesca. Un'ulteriore modifica portò a realizzare il serbatoio sulla torre di Ponte Molino.

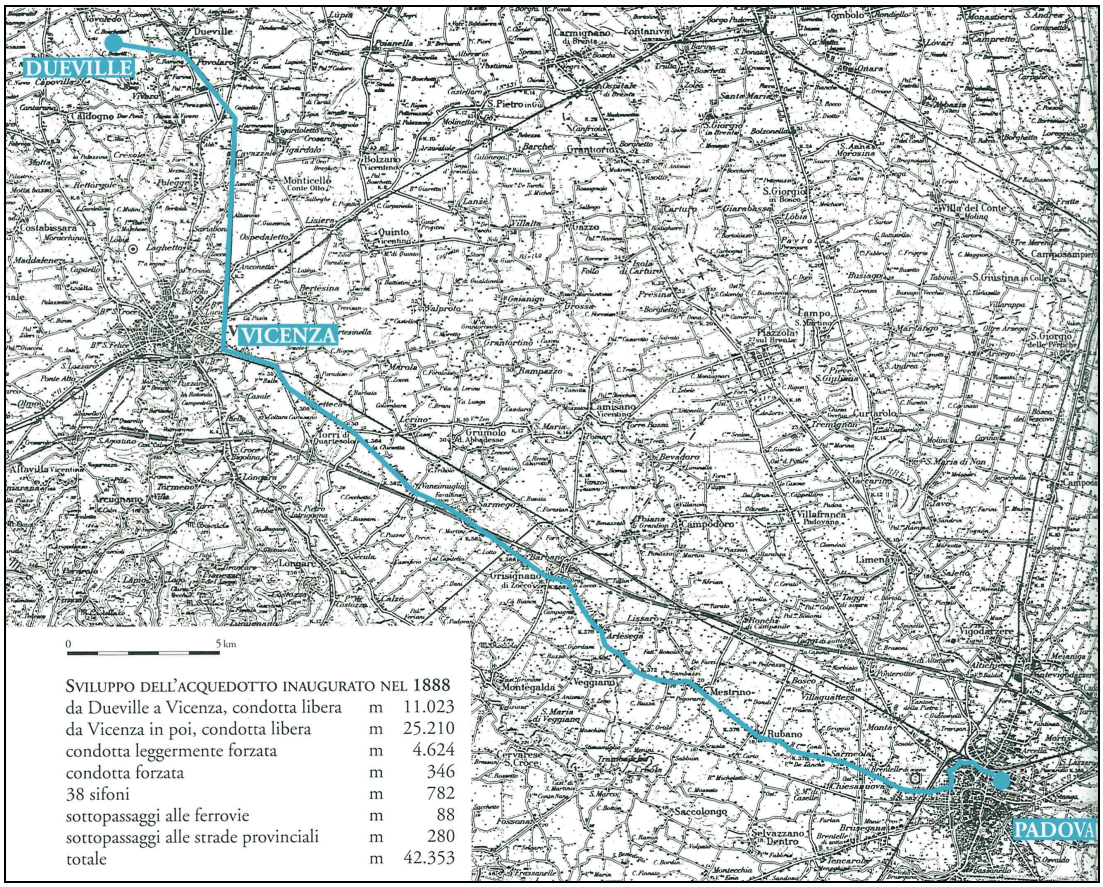


Figura 1.1 - *Sviluppo dell'acquedotto di Padova inaugurato nel 1888.*

La rete urbana comprendeva ampia parte dell'abitato entro le mura del '500. La centrale di sollevamento era posta sopra i serbatoi bassi ed utilizzava la forza motrice prodotta da una centrale da collocarsi sopra la Briglia dei Carmini: l'energia era trasmessa con dei rinvii a fune fin sopra i serbatoi bassi.

1.1.1 L'opera di presa

La *presa*[1] fu realizzata, come ricordato, a Dueville. Il fabbricato realizzato consiste in una galleria e due vasche, ciascuna di $5\,m$ di larghezza e $10\,m$ di lunghezza; la

[1] Le *opere di presa* sono serie di impianti che permettono di prelevare l'acqua dai cicli naturali. In genere tali opere si trovano lontane dai centri abitati. Tali opere rappresentano la prima parte di un acquedotto. La captazione

galleria è divisa in due canali ai quali convergono i tubi di 131 pozzi infissi nel 1888 che hanno profondità che varia da $6\,m$ a $23\,m$. L'acqua raccolta nelle due vasche passava, mediante stramazzo[2], in una galleria dalla quale poteva essere immessa nella condotta di muratura e nello scaricatore (Figura 1.2).
La temperatura dell'acqua emunta variava da $12{,}9\,°C$ in estate e $12{,}4\,°C$ in inverno, una temperatura quindi ottimale per gli usi potabili.

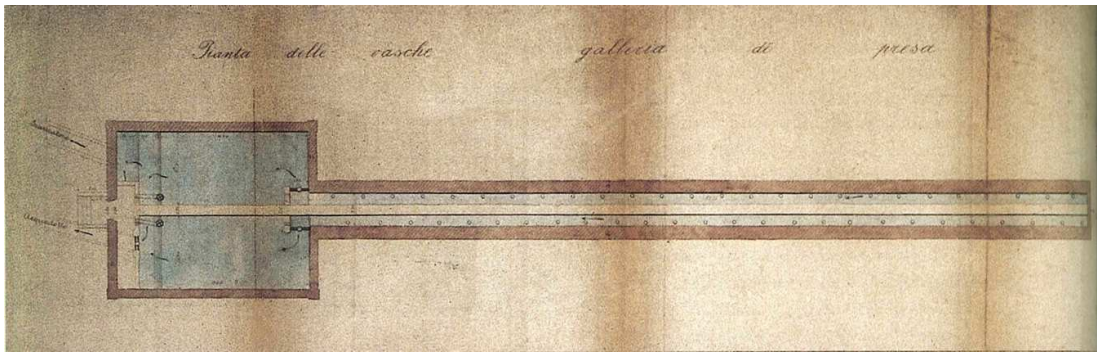


Figura 1.2 - *Opera di presa a Dueville: pianta delle vasche e galleria di presa.*

1.1.2 L'adduzione

L'adduzione[3] a Padova era progettata con una condotta in muratura funzionante, salvo pochi tratti, a *pelo libero*, ossia con occupazione parziale della sezione e lasciando quindi sul cielo della condotta un certo volume d'aria. La scelta di una condotta a *pelo libero* piuttosto che di una in *pressione* (allora detta anche condotta *forzata*), pur prevista nel progetto del 1881, fu effettuata dopo un ampio dibattito. Le ragioni principali furono le seguenti:

- la messa in pressione della condotta non avrebbe comunque eliminato il problema del sollevamento in città; infatti il dislivello disponibile avrebbe consentito di raggiungere al più di solo i piani terreni delle abitazioni;
- una condotta a pelo libero avrebbe sofferto di minori rischi di rottura, e quindi minori interruzioni di servizio, rispetto ad una condotta a pressione;
- la presenza di uno strato superficiale d'aria avrebbe consentito, secondo alcuni, una migliore coibentazione termica;

---

può essere effettuata da sorgente, da falde freatiche o artesiane, acque superficiali correnti (fiumi) o stagnanti (laghi), acque subalvee e, raramente, da acque meteoriche o piovane.

[2] Si dice *stramazzo* un'apertura in uno sbarramento di un flusso liquido, che permette il passaggio del liquido a pressione atmosferica. In altre parole uno stramazzo è il sormonto di un ostacolo da parte di una corrente liquida.

[3] *Adduzione*: nei sistemi di distribuzione idrica è il passaggio dai punti di prelievo dell'acqua potabile ai serbatoi, prima dell'immissione nella rete di distribuzione

- la maggior portata della canaletta rispetto alla condotta in pressione avrebbe consentito il soddisfacimento di altri centri oltre a Padova.

Il tracciato della canaletta da Dueville giungeva a Vicenza, ove era collocato il manufatto partitore. Il percorso verso Padova correva poi lungo la sede dell'allora strada provinciale Padova-Vicenza (l'attuale SR11), passando per Torri di Quartesolo, Grisignano di Zocco, Mestrino, Rubano, Sarmeola, per terminare a Padova nei pressi della stazione ferroviaria dove si trovavano i serbatoi bassi, della capacità di $1.000\,m^3$. Da questi le acque giungevano, con una condotta di ghisa di $700\,mm$ di diametro e lunghezza di $257{,}5\,m$, all'impianto della *Briglia delle Grate dei Carmini* (o *Briglia dei Carmini*), sul fiume Bacchiglione (Tronco Maestro), per essere sollevate ai serbatoi alti, installati all'interno della torre di Ponte Molino e da questi alla rete di distribuzione. Il percorso complessivo fu di $42.353\,m$, con un dislivello di $25\,m$, e una pendenza media di $58\,cm/km$.

La sezione corrente della canaletta fu modificata in qualche tratta. Per la maggior parte del percorso fu adottata una sezione ad U svasata verso l'alto realizzata con conglomerato cementizio non armato e coperta da una volta in mattoni. Nei tratti a debole pressione era adottata la stessa sezione ma con la volta di calcestruzzo anziché di mattoni. Nei tratti a maggiore pressione la sezione era circolare (Figura 1.3).

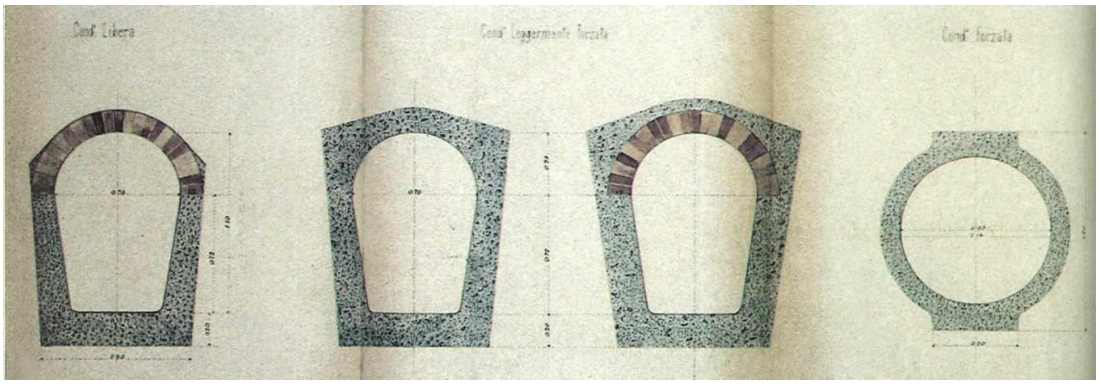


Figura 1.3 - *Sezioni della condotta (canaletta): libera, leggermente forzata in cemento, leggermente forzata in cemento e mattoni, forzata in cemento.*

Quando la condotta doveva attraversare un corso d'acqua l'attraversamento era ottenuto con un sifone subacqueo: un tratto di condotta in pressione, posto al fondo del corso d'acqua, formato da tubi di ghisa. La condotta subacquea era in genere avvolta in un getto di calcestruzzo o in una gettata di pietrame con la funzione di impedire, anche col peso della condotta, la spinta di galleggiamento a tubo vuoto. All'imbocco del sifone era collocato un pozzetto di raccordo per immettere l'acqua nel sifone, ma anche per poter scaricare lateralmente l'acqua in arrivo e quindi mettere all'asciutto il sifone stesso.

Negli attraversamenti ferroviari la condotta adottata era di ghisa con diametro di $800\,mm$, collocata in un tombino ad U chiuso da volta circolare: largo $1{,}20\,m$ e alto, all'intradosso, $1{,}50\,m$; con una volta di mattoni di tre teste[4]; e base di conglomerato cementizio dello spessore di $50\,cm$. Agli estremi del tombino erano collocati due pozzetti, uno d'ispezione e uno di raccordo, sottraendo così la condotta alle vibrazioni per il passaggio dei treni.
Nei sottopassi delle strade provinciali la volta era in mattoni di due teste.

1.1.3 I serbatoi bassi

Destinati a raccogliere le acque provenienti da Dueville, della capacità di $1.000\,m^3$, constavano di due gallerie parallele e indipendenti, divise da un muro comune: ogni galleria era lunga $32\,m$, larga $5{,}5\,m$ e alta $4{,}7\,m$; la copertura era a botte semicircolare con piano d'imposta[5] a $2\,m$ dal fondo (Figura 1.4).

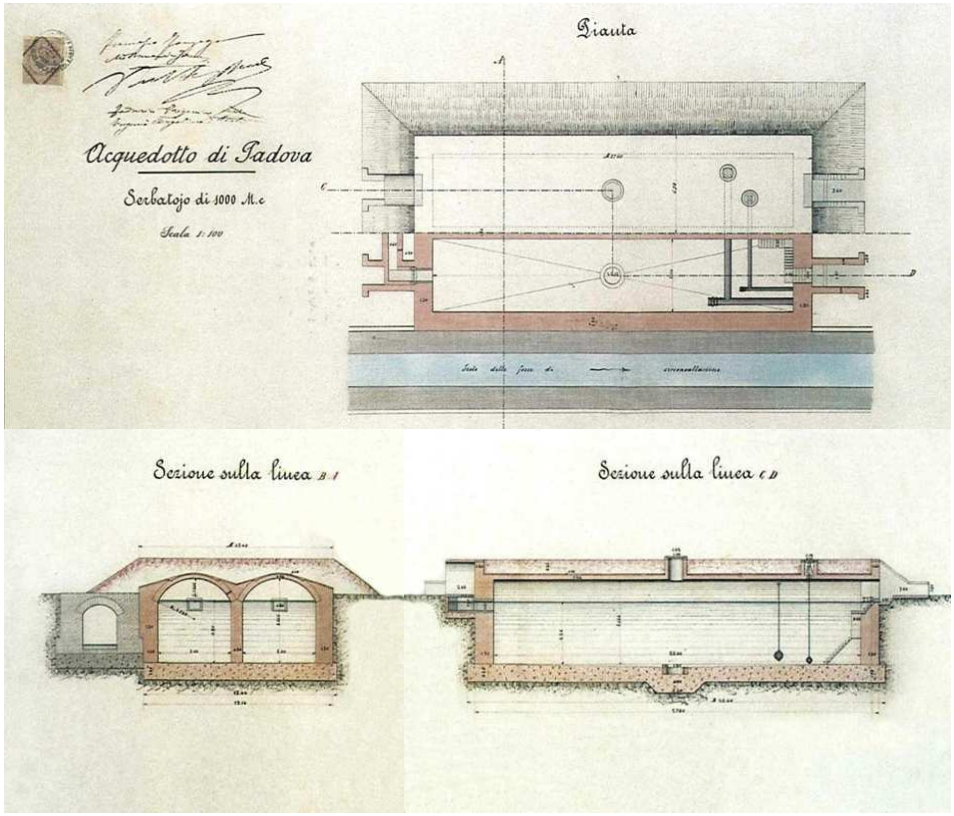


Figura 1.4 - *Serbatoi bassi.*

[4] Un muro, in questo caso la volta, si definisce a *una testa* quando lo spessore è uguale alla larghezza di un mattone, si definisce a *due teste* quando lo spessore è uguale alla lunghezza (o a due volte la larghezza) di un mattone, si definisce a *tre teste* quando lo spessore è uguale ad una lunghezza e mezzo (o a tre volte la larghezza) di un mattone. Considerate le misure del mattone unificato (5,5 cm x 12 cm x 25 cm) e lo spessore di 1 cm per la malta di separazione, si ottengono rispettivamente le larghezze pari a 12 cm, 25 cm e 38 cm.
[5] *Piano d'imposta*: il piano che passa dove terminano i sostegni (i muri laterali) e inizia la volta.

1.1.4 Il partitore a Vicenza

Poiché la Società Veneta pensava di alimentare con l'acqua di Dueville sia Vicenza che Padova, è stato realizzato a Vicenza un manufatto partitore. La società fu però costretta ad abbandonare questa idea a seguito della deliberazione del Consiglio comunale di Vicenza nel 1886 di stipulare contratto con altra azienda.
Tale manufatto consente di suddividere la portata in arrivo in due parti, secondo un rapporto prefissato. Esso consiste di una vasca d'arrivo, dalla quale si dipartono due condotte, una per Vicenza e una per Padova. Le condotte sono presidiate da paratoie per escludere la derivazione o per regolare la portata in uscita. Quando una delle due condotte deriva meno portata del previsto, la seconda può, in qualche misura derivare una maggior portata.
Il manufatto è completato da uno sfioratore di troppo pieno per l'eventuale allontanamento della portata in eccesso rispetto a quella derivata, al fine di evitare la messa in pressione della condotta.

1.1.5 La Briglia dei Carmini ed il suo utilizzo per l'acquedotto

Il sollevamento dell'acqua nei serbatoi di pressione era garantito principalmente da un sistema di pompe di sollevamento poste in un edificio realizzato su un manufatto, la Briglia del Carmine, che assolveva il duplice compito di sostenere il livello dell'acqua del Tronco Maestro del Bacchiglione, e produrre energia cinetica a vantaggio, appunto, di applicazioni industriali.
La Briglia dei Carmini fu costruita su progetto dell'ingegnere del Genio Civile Giovanni Ponti, aveva tre luci centrali di $5\,m$ di larghezza, ad ognuno dei due lati vi erano due luci larghe $4\,m$ e con la platea più alta di $1{,}5\,m$ sul fondo. L'apertura delle tre luci principali era controllata con paratoie metalliche manovrabili idraulicamente o manualmente, consentendo di sostenere il livello a monte $4{,}5\,m$ sopra la soglia e $2{,}5\,m$ circa sul livello ordinario di valle.
Le due luci minori sul lato sinistro vennero concesse alla Società Veneta per l'installazione di due turbine di tipo Jonval e di costruzione Blanchaud, che fornivano la potenza meccanica per il sollevamento dell'acqua potabile.
Sopra queste due luci, e in prolungamento della briglia, sulla sponda sinistra fu costruita la sala dove era ubicato l'impianto di sollevamento propriamente detto; la sala era lunga $40{,}5\,m$ e larga $10\,m$ (Figura 1.5). Qui furono collocati due gruppi di pompe gemelle ad effetto Girard e di fabbricazione Breda (già Società Elvetica), che per mezzo di un unico albero orizzontale e di ingranaggi venivano azionate da dette turbine (Figura 1.6), o da una motrice a vapore di riserva, tipo Cornovaglia,

necessaria specialmente in tempo di piena o morbida[6], quando le turbine non potevano funzionare per l'eccessivo rigurgito[7] delle acque a valle.

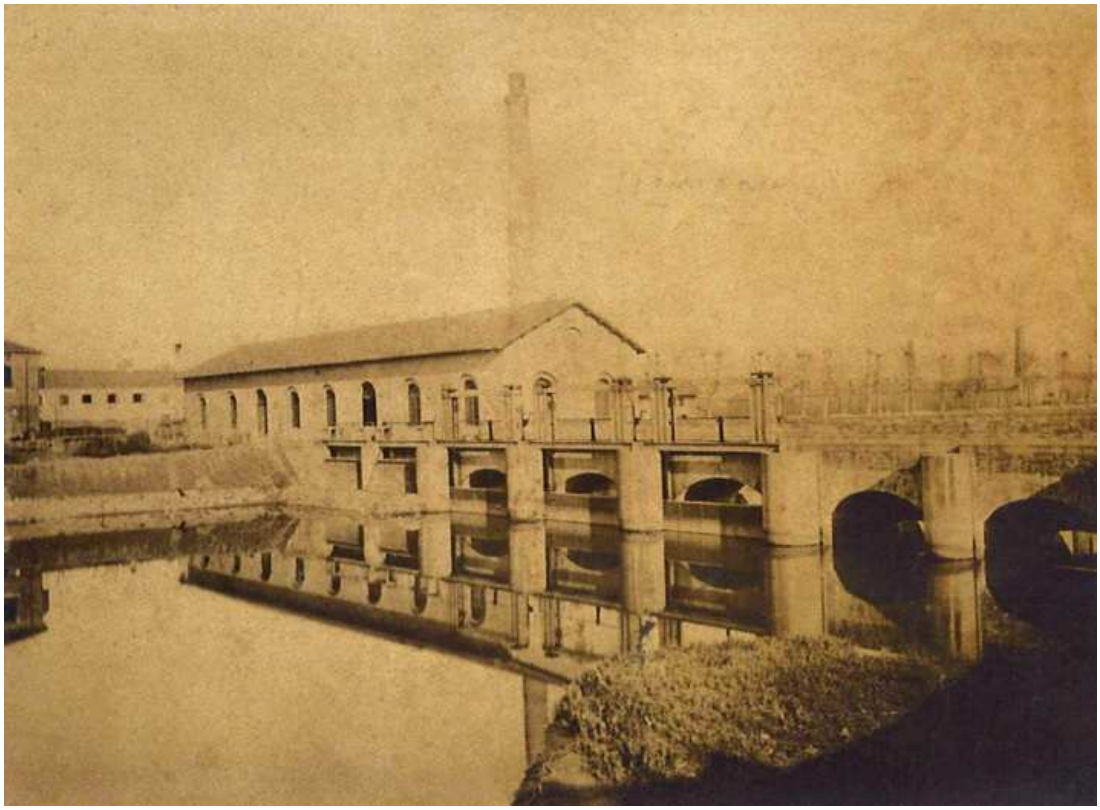


Figura 1.5 - *Briglia del Carmine: veduta da monte.*

I1 volume di acqua elaborato dai quattro gruppi fu verificato pari a $4500\,m^3$ per gruppo e per giorno, con un totale quindi di $18.000\,m^3/giorno$, la prevalenza geodetica[8] necessaria era di $29{,}10\,m$, di cui $3{,}35\,m$ per aspirazione e $25{,}75\,m$ per mandata. La macchina a vapore aveva forza di 120 cavalli, ma poteva essere spinta fino a 180 cavalli applicandovi un condensatore. In un locale attiguo erano collocate le due caldaie tipo Cornovaglia per il funzionamento della motrice; le caldaie erano timbrate a 6 atmosfere, con una superficie vaporizzante di $70\,m^2$ per caldaia. Le turbine generavano energia meccanica che veniva utilizzata per far funzionare le pompe.

[6] Il periodo di *morbida* è quello in cui nel fiume scorre abbondante acqua, quello di *piena*, quando scorre una quantità eccezionale di acqua tale da inondare aree che normalmente sono asciutte.

[7] Il *rigurgito* che consiste nell'inversione della direzione di scorrimento dell'acqua. Quando in un canale è raggiunto un livello idrico alto, queste acque possono "infilarsi" in un canale o condotta affluenti con livello più basso, e scorrere controcorrente fino ad una distanza alla quale i livelli si equilibrano.

[8] *Prevalenza geodetica*: differenza tra i livelli del liquido alla mandata e all'aspirazione, quindi la differenza tra il livello nel serbatoio sopraelevato e quello nei serbatoi bassi.

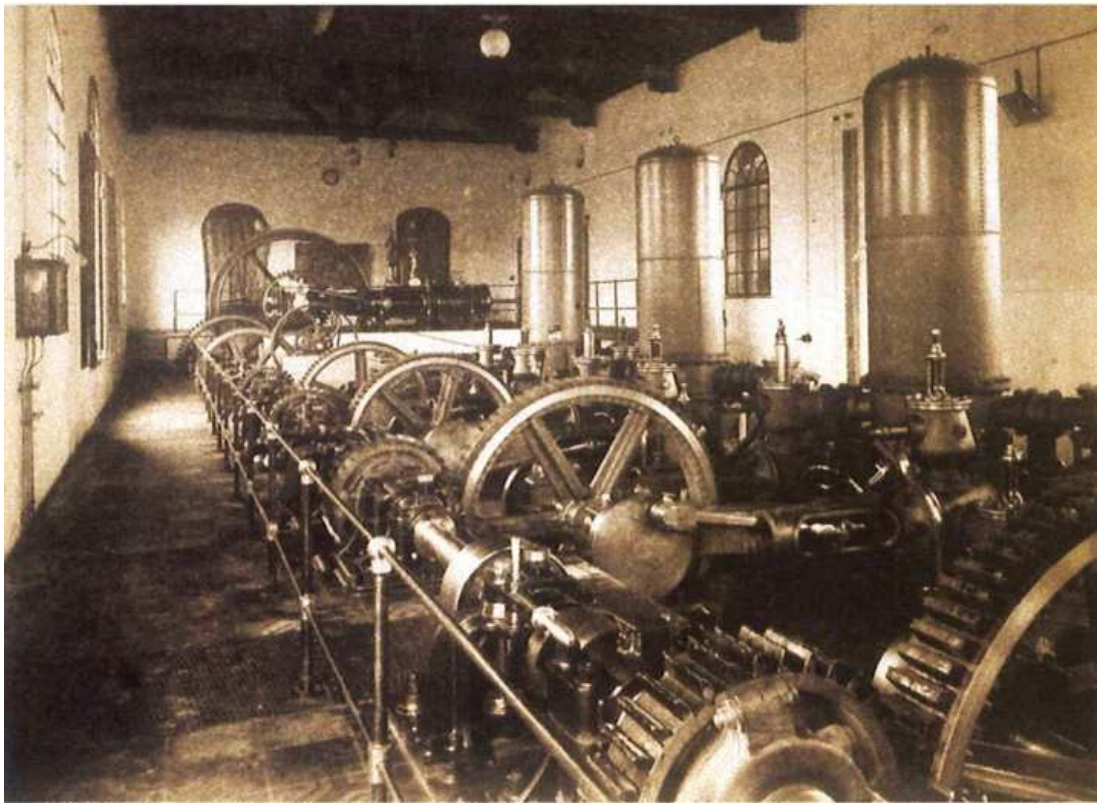


Figura 1.6 - *Briglia del Carmine: interno della sala macchine.*

1.1.6 I serbatoi sopraelevati

Collocati sulla torre medioevale di Ponte Molino (Figura 1.7) erano costituiti da vasche di acciaio con base $4m \times 4m$ ed altezza di $5\,m$. Le vasche erano poste con fondo a quota $35{,}2\,m$ sul comune marino[9]; ognuna poteva contenere $40\,m^3$ fino alla quota di sfioro posta a quota $37{,}7\,m$ ossia $20{,}33\,m$ superiore alla quota della soglia del palazzo municipale (punto più alto della città) la cui quota era $17{,}37\,m$. La condotta di mandata ai serbatoi aveva diametro di $500\,mm$, la condotta di alimentazione della rete di $400\,mm$. Il locale ove erano collocate le vasche, ottenuto con sopraelevazione della torre, era aerato e coperto con voltine in muratura.

1.1.7 La rete di distribuzione urbana

Il tracciato di progetto della rete di distribuzione si propose lo scopo di percorrere tutte le strade della città in modo da poterla approvvigionare in ogni punto. La rete era costituita da anelli e da rami aperti; i primi con lo scopo di meglio equilibrare le portate nelle varie zone anche in funzione delle variazioni di richiesta.

[9] *Comune marino*: livello medio dei massimi di marea.

Le condotte impiegate furono tutte di ghisa grigia di seconda fusione, colate verticalmente. Le condotte erano di produzione della Società alti Forni Acciaieria e Fonderia di Terni, di cui la Società Veneta deteneva la maggioranza. Le condotte venivano provate in officina alla pressione di 10 atmosfere e in opera alla pressione di 5 atmosfere, corrispondente a circa due volte quella massima d'esercizio. La profondità di posa, di circa $90\,cm$ sotto il piano stradale (Figura 1.8 (a)), garantiva all'acqua una temperatura sensibilmente costante nell'arco dell'anno. Tutte le convessità furono dotate di sfiati e le concavità di scarichi. Ogni tronco era poi isolabile con saracinesche.

Figura 1.7 - *La torre di ponte Molino, sopraelevata per inserire le vasche di pressione.*

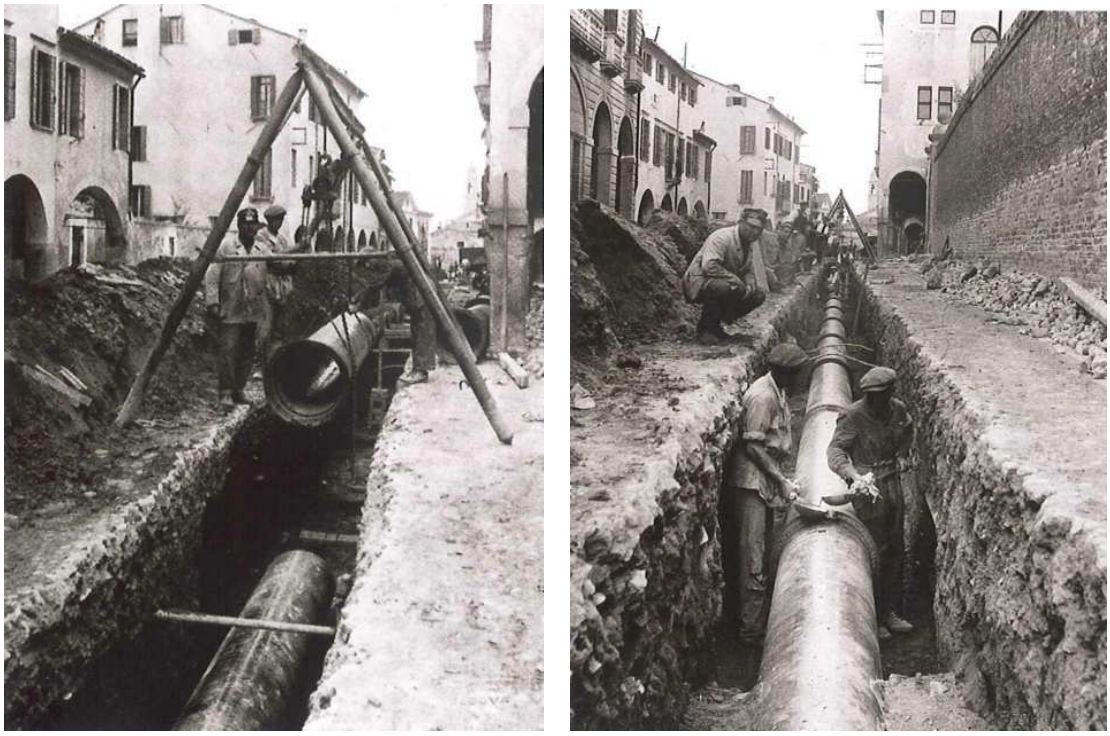
Il sistema di giuntura tra le tubazioni di ghisa detto a *guaina e cordone* o *a guaina e collare* era allora il preferito tra alcuni possibili, anche perché permetteva una certa elasticità al sistema di condotte. Ogni tubo terminava ad un'estremità con un semplice ribordo detto *cordone* che sporgeva qualche millimetro dalla superficie esterna del tubo, all'altra presentava un rigonfiamento, detta guaina, lungo parecchi centimetri e con diametro interno di qualche millimetro maggiore del diametro esterno del cordone.

Il giunto a guaina e cordone veniva eseguito inserendo l'estremità col cordone nella guaina del tubo precedente, in modo da lasciare pochi millimetri fra il cordone e la guaina per consentire eventuali dilatazioni. Nel giunto venivano inseriti, avvolti a

spirale e quindi ribattuti, alcuni giri di corda di canapa impregnata di catrame fuso od olio; la funzione principale era quella di non far entrare il piombo fuso nella condotta (Figura 1.8 (b)).

Per eseguire la piombatura veniva posto, tutt'intorno alla connessura rimasta tra maschio e guaina, un anello di creta plastica con due aperture nella parte superiore. Un'apertura serviva per il versamento del piombo fuso, l'altra fungeva da sfiato per evitare che scoppiasse l'anello di creta, con fuoriuscita di piombo fuso. Raffreddato l'anello di piombo veniva tolta la creta e ribattuto il piombo.

Lo sviluppo della rete al primo impianto, con esclusione delle diramazioni per le utenze private, era di 25.670 m , con 139 idranti, 12 fontane, 41 scarichi, 11 sfiati, 44 saracinesche d'arresto.



(a) (b)

Figura 1.8 - *Posa di tubature in ghisa per la rete di Padova: posa dell'anello diam.500, 1933. Posizionamento delle tubazioni (a) e versamento del piombo fuso per sigillare il giunto (b).*

1.1.8 La conclusione dei lavori

L'acquedotto fu inaugurato il 13 giugno 1888, ricorrenza di Sant'Antonio. La Società Veneta ripropose a Padova la stessa cerimonia usata il 23 giugno 1884 per l'inaugurazione dell'acquedotto di Venezia, anch'esso costruito dalla Veneta, per conto della Societé Generale des Eaux, quando fu realizzata una fontana in Piazza San Marco; analoga coreografia era stata proposta per l'inaugurazione

dell'acquedotto di Napoli. A Padova, per l'occasione, fu costruita in piazza Unità d'Italia (ora Piazza dei Signori), una fontana formata da travi e tavole di legno. L'uscita di un alto getto d'acqua fu salutata dalla banda che suonava la marcia Reale (inno nazionale del Regno d'Italia).
La Commissione di Collaudo, composta dai professori Pio Chicchi e Enrico Bernardi e dall'ingegner Francesco Turola rassegnò il certificato di collaudo l'1 settembre 1889, certificando che l'opera era perfettamente collaudata, in ottimo stato di manutenzione e completamente funzionante.

1.1.9 Il riscatto da parte del Comune

Fin dall'inizio del funzionamento dell'acquedotto si andavano manifestando in città dissensi sulla gestione e richieste di riscatto da parte del Municipio. Una richiesta significativa fu quella fatta dalla Società di igiene per la città e Provincia di Padova il 29 aprile 1890.
Nella relazione *L'acquedotto ed i bisogni di Padova in linea igienica*, letta all'assemblea generale della Società il 29 aprile 1890, Napoleone D'Ancona ricordava che la canaletta poteva convogliare quotidianamente fino a $36.000\,m^3$, che i pozzi Norton fino allora infissi erano capaci di una portata giornaliera di $21.600\,m^3$ ossia di 450 litri per abitante al giorno «come *New York*», contro una quantità erogata di soli $520\,m^3$, equivalenti a poco più di 10 litri per abitante, contro i 150 fissati dagli igienisti. Citava ancora che il numero delle case a Padova oltrepassava le 5.000, con 11.000 famiglie che vi abitavano: a fronte di questo gli allacciamenti all'acqua potabile erano solamente 411. Inoltre, lamentava, nessuna fontana abbelliva la città: solo 12 fontanini con portata di 150 litri all'ora erano a disposizione dei poveri. Indicava, quali cause del mancato sviluppo, la circostanza che la rete di distribuzione aveva uno sviluppo inferiore ai $24\,km$ contro uno sviluppo delle strade cittadine di circa $50\,km$: ovviamente le strade escluse erano quelle dei quartieri più poveri, debitamente elencate. Un'altra ragione era individuata nell'elevato costo dell'allacciamento a carico dell'utente, specie quando la casa era sita dal lato opposto della strada rispetto alla collocazione della tubatura. Al termine dell'adunanza fu approvato all'unanimità un ordine del giorno che chiedeva il riscatto dell'acquedotto da parte del Comune o nuovi accordi con la Società concessionaria.
Di fatto, ad acquedotto ultimato, lo sviluppo degli allacciamenti, e quindi dei consumi, fu assai più contenuto del previsto. Si rese quindi indispensabile l'intervento del Comune che, con una spesa di 2.150.000 lire (meno quindi di quanto

speso dalla Veneta per la costruzione che risultò di 2.750.000 lire) riscattò l'acquedotto.

Dopo il riscatto da parte del Comune, divenuto operativo il 17 maggio 1893, la tariffa fu ridotta da 25 a 16 centesimi il m^3; ed iniziarono i lavori di ampliamento della rete che terminarono il 30 ottobre dello stesso anno: la rete risultante fu di $40.084\,m$, con un incremento di 77 saracinesche di scarico, 53 bocche da incendio e 85 fontanine pubbliche allora chiamate, alla francese, *born fontaine*. La spesa fu di lire 117.040,39. Per adeguare l'impianto di sollevamento alle nuove richieste furono installati, in aggiunta ai due ordinari, altri due gruppi di pompe Giraud ed una caldaia a vapore identica alla prima.

1.2 Ampliamenti della rete d'adduzione

La *canaletta* fu, fino al 1959, l'unica fonte di approvvigionamento della città ed è tuttora in funzione unitamente al *Nuovo acquedotto*. Quest'ultimo, dotato di una condotta di cemento amianto di $900\,mm$ di diametro, preleva ancora dalla zona di Dueville, poco a valle della zona di presa della canaletta, e, con percorso all'incirca parallelo a quello della canaletta, porta l'acqua alla centrale Codalunga (Figura 1.9). L'opera fu ultimata il 15 dicembre 1958 e messa in funzione l'anno successivo. Dal 2000 è in funzione una nuova linea di adduzione in acciaio del diametro di $1.300\,mm$, che, partendo dall'impianto di Saviabona, a nord di Vicenza, raggiunge il territorio comunale di Padova seguendo un percorso parallelo alla condotta da $900\,mm$.

Attualmente la rete acquifera padovana è composta da:

- una linea di adduzione, una canaletta trapezoidale in calcestruzzo costruita nel 1890, che parte direttamente da Villaverla e raggiunge il territorio di Padova con un percorso che ha luogo lungo la ferrovia Vicenza-Bassano, per poi proseguire lungo la statale Vicenza-Padova;
- una linea di adduzione del diametro di $900\,mm$ in cemento, costruita nel 1958, che convoglia l'acqua raccolta nei pozzi posti a nord del Comune di Vicenza e recapita a Padova con un percorso che si snoda lungo la ferrovia Padova-Vicenza;
- una nuova linea di adduzione in acciaio del diametro di $1.300\,mm$, entrata in esercizio nel 2000, che, partendo dall'impianto di Saviabona, raggiunge il territorio comunale di Padova seguendo un percorso parallelo alla condotta da $900\,mm$;

- cinque stazioni di sollevamento (Montà, viale Codalunga, Brentelle, Stanga, Volta Brusegana);
- sei serbatoi di stoccaggio per un volume totale di 150.000 m^3;
- cinque serbatoi pensili di stoccaggio e compensazione della pressione della rete di distribuzione;
- una rete globale di condotte per uno sviluppo di circa 1.246 km.

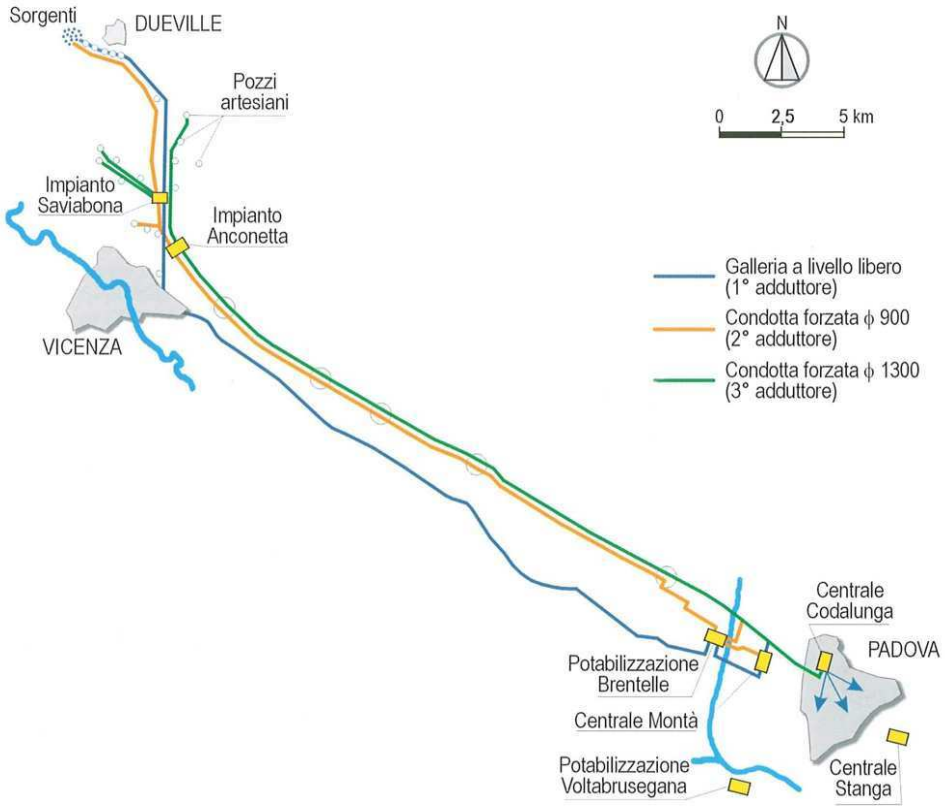


Figura 1.9 - *L'attuale sistema di adduzione dal vicentino: la canaletta, la condotta da 900 mm e la condotta da 1300 mm.*

Capitolo 2
-
Studi precedenti sulla previsione dei consumi idrici

Lo studio dei modelli per la previsione dei consumi idrici urbani ha di recente assunto notevole interesse per la moderna gestione delle reti di distribuzione. Attraverso tali modelli è possibile effettuare, con ridotti margini di incertezza, la stima dei consumi idrici futuri alle scale temporali di interesse, consentendo di adeguare con il dovuto anticipo il funzionamento degli impianti alle situazioni attese, allo scopo di ridurre disservizi, pianificare le operazioni di manutenzione e minimizzare il consumo energetico per l'erogazione del servizio.

2.1 Modello di previsione di Acegas-Aps (1982)

Alla fiera di Padova dedicata ai temi ambientali e al risparmio energetico del 1982 (SEP[10]) l'ing. Francesco Davanzo di Acegas-Aps[11] ha presentato un modello di previsione dei consumi idrici applicato alla città di Padova. Lo scopo era quello di minimizzare le risorse energetiche necessarie per l'erogazione dell'acqua alle utenze cittadine. Il modello realizzato ha permesso un uso migliore delle risorse adduttive (condotte) e capacitive (serbatoi) consentendo un significativo risparmio energetico relativamente ai sistemi di pompaggio dell'acqua nelle condotte. In particolare, la bontà del modello realizzato dall'ing. Davanzo ha portato nell'anno 1982 ad una diminuzione dei costi di energia elettrica impiegata del 5%, nonostante l'aumento nella produzione di acqua del 2,5%.

---

[10] *SEP*: Salone internazionale delle Ecotecnologie organizzato periodicamente da PadovaFiere con il patrocinio del Ministero dell'Ambiente. Vengono trattati argomenti sui servizi per la gestione di energia, aria, acqua, rifiuti per la salvaguardia delle risorse ambientali, la promozione della sostenibilità sociale, ambientale ed economica.

[11] *Acegas-Aps S.p.A.* è un'azienda italiana che opera nel settore dei servizi di pubblica utilità in particolare in Veneto e in Friuli-Venezia Giulia. Nasce il 19 dicembre 2003 dalla fusione di due imprese che hanno operato per più di cent'anni nel campo della fornitura di servizi pubblici nelle province di Padova e Trieste, rispettivamente APS e Acegas. Gestisce a Padova la raccolta dei rifiuti, la rete fognaria, i servizi di distribuzione dell'acqua e del gas e la manutenzione degli impianti di illuminazione pubblica.

2.1.1 Il sistema di adduzione e la previsione dei consumi settimanali

Il primo ambito di interesse analizzato dal team coordinato dall'ing. Davanzo è il sistema di produzione e adduzione dell'acqua. Il sistema padovano attingeva circa il 90% dell'acqua prodotta dalle falde pedemontane vicentine. Nel tratto Vicenza - Padova era presente un sistema di adduzione in pressione del diametro di $900\,cm$ che consentiva l'erogazione per gravità di 600 litri di acqua al secondo. Questa portata era aumentabile fino a 900 litri al secondo mediante l'impiego di due pompe a giri variabili della potenza di oltre 300 kW. A valle del sistema di adduzione venivano alimentati dei serbatoi che avevano una capienza di $45.000\,m^3$, dai quali attingevano gli impianti per la distribuzione in rete. Fino a prima dell'applicazione del modello risultante dal lavoro svolto dall'ing. Davanzo, il servizio di approvvigionamento di acqua dalle sorgenti del vicentino si basava sul rifornimento giornaliero pari al consumo del giorno precedente: nel corso della notte i serbatoi venivano completamente riempiti per compensare il consumo realizzato durante il giorno.

La messa in produzione del nuovo modello di esercizio ha comportato una programmazione dell'adduzione che seguisse un ciclo settimanale: stimando il consumo previsto per la settimana successiva venivano regolate le pompe in modo da erogare acqua a portata costante per l'intero periodo. Infatti, solo in tal modo l'impiego di energia elettrica può essere minimizzato, in forza dell'andamento cubico che legava il consumo di energia elettrica in funzione della portata come riportato in Figura 2.1.

In linea di principio sarebbe stato conveniente considerare un periodo più lungo di una settimana per ridurre ulteriormente i consumi di energia elettrica, ma furono considerati due fattori: da una parte l'attendibilità della stima dei consumi idrici è inversamente proporzionale alla durata del periodo da stimare, dall'altra va tenuto conto che i serbatoi hanno una capacità finita.

Ottimizzare l'adduzione consiste nel ridurre i costi di gestione garantendo l'erogazione del servizio. Questo obiettivo si raggiunge riempiendo i serbatoi con portata costante in modo tale da rispondere comunque alle richieste dei consumi che sono invece variabili. I vincoli ai quali si è soggetti sono: la quantità di acqua nei serbatoi deve essere superiore ad una soglia minima predefinita e inferiore alla capacità massima degli stessi.

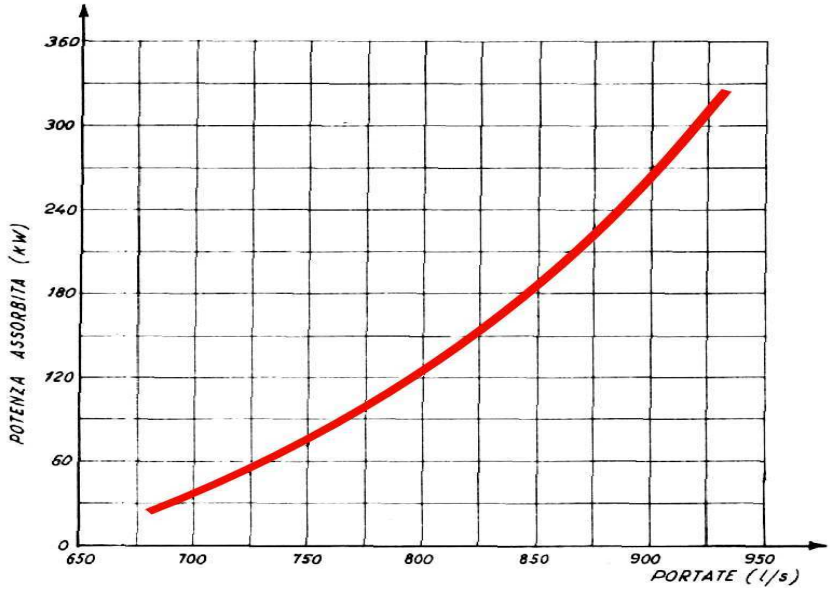


Figura 2.1 *Potenza assorbita dall'impianto di pompaggio in funzione della portata (funzione cubica).*

2.1.1.1 Costruzione del modello

L'analisi del 1982 ha considerato i dati relativi agli anni 1969-1980 (12 anni) e ha preso in esame le seguenti variabili:

- giorno di riferimento ($gg/mm/aa$);
- volume di acqua distribuito ($m^3/giorno$);
- temperatura ambientale massima del giorno ($°C$).

La prima osservazione ha riguardato la stagionalità annua dei consumi idrici: i volumi di acqua erogata sono stati inferiori nel periodo invernale e maggiori in quello estivo, con valori intermedi nelle altre stagioni, sono state rilevate inoltre evidenti diminuzioni in occasione dei periodi di ferie.

Un secondo punto considerato è stato l'influenza della temperatura ambientale rispetto al consumo idrico. Se da un lato temperatura e consumi sono correlati nella stagionalità annua, è stato evidenziato che un aumento della temperatura, rispetto alla temperatura media di riferimento per quel periodo, porta ad una quota aggiuntiva di consumo di acqua.

A fronte di queste osservazioni è stata effettuata un'analisi per realizzare un modello di previsione le cui fasi sono:

- l'individuazione dei consumi medi settimanali e del coefficiente che indica il "peso" di ciascuna settimana rispetto al volume annuo,
- la considerazione dell'influenza delle temperature di ciascuna settimana per spiegare la variabilità residua.

2.1.1.2 Modellazione della componente stagionale

Per individuare i consumi medi settimanali è stato necessario dividere i 365 giorni dell'anno in settimane. Sono state ipotizzare due soluzioni: l'individuazione delle settimane da lunedì a domenica, a prescindere dal giorno (numerico) di inizio oppure la suddivisione delle settimane sulla base della data di calendario, indipendentemente dal giorno di inizio settimana. Per esemplificare, si supponga che il 1° gennaio sia mercoledì: con il primo metodo la prima settimana dell'anno inizia dal 6 gennaio (lunedì) e finisce il 12 gennaio (domenica); con il secondo metodo la prima settimana inizia dal 1° gennaio (mercoledì) e finisce il 7 gennaio (martedì). Il primo metodo presenta il problema di dover escludere dall'analisi le settimane a cavallo dell'anno e di considerare settimane di anni diversi non sempre sovrapponibili in termini di giorni festivi (es. ferragosto) che potrebbero capitare in settimane diverse tra un anno e l'altro. Si è deciso di adottare il secondo metodo di suddivisione dell'anno in settimane poiché non risente di queste problematiche.

Sono stati considerati i dati dei consumi giornalieri dei 12 anni di interesse e, per ciascuno di essi, i primi 364 giorni (7 giorni x 52 settimane = 364 giorni).

Indichiamo con j l'anno, con i la settimana e con c_{ij} il consumo medio della settimana i nell'anno j (i=1,…,52; j=1,…,12).

Quindi:

Il calcolo del consumo giornaliero medio annuo è:

$$\bar{c}_j = \frac{1}{52}\sum_{i=1}^{52} c_{ij} \qquad j=1,\ldots,12\ . \tag{2.1}$$

Al fine di eliminare l'effetto del volume annuo dei consumi è stata considerata, per ogni anno, la quantità derivante dal rapporto tra il consumo giornaliero medio di ciascuna settimana e il consumo giornaliero medio annuo; detta quantità viene chiamata *indice settimanale*, in seguito r_{ij} , dove i sono le settimane e j gli anni:

$$r_{ij} = \frac{c_{ij}}{\bar{c}_j} \qquad i=1,\ldots,52 \quad j=1,\ldots,12. \tag{2.2}$$

Per costruzione risulta che il valore medio sulle 52 settimane degli indici settimanali è pari a 1:

$$\mathrm{E}_i(r_{ij}) = \frac{1}{52}\sum_{i=1}^{52} r_{ij} = \frac{1}{52}\sum_{i=1}^{52}\frac{c_{ij}}{\bar{c}_j} = \frac{\sum_{i=1}^{52} c_{ij}}{52\cdot\bar{c}_j} = \frac{\bar{c}_j}{\bar{c}_j} = 1 \qquad j=1,\ldots,12\ . \tag{2.3}$$

La media dell'*indice settimanale* effettuata sui 12 anni rappresenta invece il consumo medio di ciascuna settimana appartenente a quell'arco di anni. Viene considerata detta quantità come un valore tipico di ciascuna settimana, in seguito $\hat{r}_i$:

$$\hat{r}_i = \frac{1}{12}\sum_{j=1}^{12} r_{ij} \qquad i=1,\ldots,52\ . \tag{2.4}$$

Gli indici $\hat{r}_i$ possono essere utilizzati per stimare il consumo settimanale di un generico anno. Il consumo medio della settimana *i* può essere stimato dalla quantità $\hat{c}\cdot\hat{r}_i$, dove $\hat{c}$ è il consumo medio dell'anno di interesse, noto o stimato.

2.1.1.3 Introduzione del fattore temperatura.

Nella Figura 2.2 si riporta l'analisi grafica dell'andamento dell'indice $\hat{r}_i$ presente nel lavoro dell'ing. Davanzo, nella quale sono rappresentati anche i valori massimi e minimi di r_{ij} al variare di *j*:

$$r_i^{\max} = \max_j(r_{ij}) \qquad i=1,\ldots,52 \tag{2.5}$$

$$r_i^{\min} = \min_j(r_{ij}) \qquad i=1,\ldots,52\ . \tag{2.6}$$

Nello stesso grafico sono stati rappresentati anche i valori della temperatura ambientale media, calcolati come segue:

$$t_i = \frac{1}{12}\sum_{j=1}^{12} t_{ij} \qquad i=1,\ldots,52\ , \tag{2.7}$$

dove:

t_{ij} = media delle temperature ambientali massime giornaliere della settimana *i* dell'anno *j*.

La semplice osservazione del grafico evidenzia che la stima basata solo sul valore medio della settimana, così come proposta dal modello finora illustrato, non è sufficiente. In particolare risulta eccessivamente imprecisa la stima dei consumi idrici relativa al periodo estivo; questo si evince dall'aumento della fascia di scostamento dal valor medio.

Si è tentato quindi di spiegare la variabilità residua utilizzando come variabile dipendente lo scostamento della temperatura media della settimana rispetto alla media dello stesso valore calcolato sui 12 anni disponibili (temperatura tipo di quella settimana).

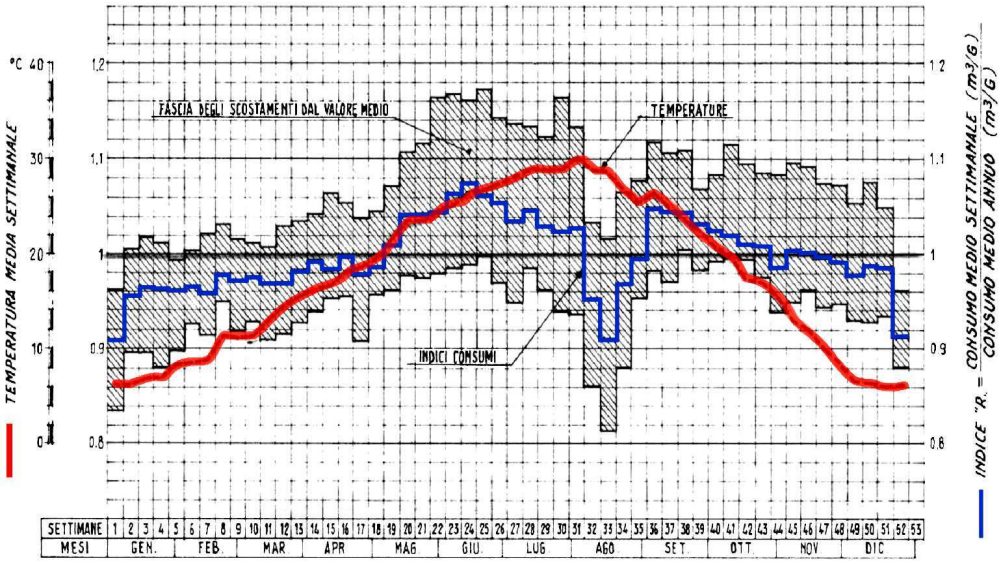


Figura 2.2 *Valori dell'indice settimanale* $\hat{r}_i$ *(linea blu), suo campo di variazione (area tratteggiata) e confronto con la temperatura massima giornaliera (linea rossa).*

Se indichiamo con t_{ij} il valore della temperatura media settimanale registrato nella settimana *i* dell'anno *j*, la temperatura tipo della settimana *i* è:

$$t_i = \frac{1}{12}\sum_{j=1}^{12} t_{ij} \qquad i=1,\ldots,52 \tag{2.8}$$

Per considerare la correlazione tra il consumo di acqua e la temperatura media della settimana *i*-esima è stato costruito lo stimatore $\hat{\hat{r}}$:

$$\hat{\hat{r}}_{ij} = \hat{r}_i(1+\Delta t_{ij}\alpha_i), \tag{2.9}$$

dove $\Delta t_{ij} = t_{ij} - t_i$ è lo scarto della temperatura media della settimana *i* dell'anno *j* rispetto alla temperatura tipo della stessa settimana e α_i è il coefficiente di relazione tra consumo idrico e temperatura relativo alla settimana *i*.

L'errore dello stimatore $\hat{\hat{r}}$ è stato espresso come differenza tra l'indice effettivo calcolato per la settimana *i* dell'anno *j* e il valore stimato sulla base del modello:

$$E_{ij} = r_{ij} - \hat{\hat{r}}_{ij} = r_{ij} - r_i \cdot (1+\Delta t_{ij} \cdot \alpha_i) \tag{2.10}$$

La determinazione dei coefficienti α_i è stata ottenuta cercando di minimizzare la somma del valore assoluto degli scarti per i 12 anni osservati.

Attraverso un metodo iterativo si sono cercati i valori $\hat{\alpha}_i$ che minimizzano le quantità seguenti:

$$\hat{\alpha}_i : \min_{\alpha_i} \sum_{j=1}^{12} \left|E_{ij}\right| = \min_{\alpha_i} \sum_{j=1}^{12} \left|r_{ij} - r_i \cdot (1+\Delta t_{ij} \cdot \alpha_i)\right| \tag{2.11}$$

Sono stati ottenuti i coefficienti riportati nel grafico di in Figura 2.3**.**

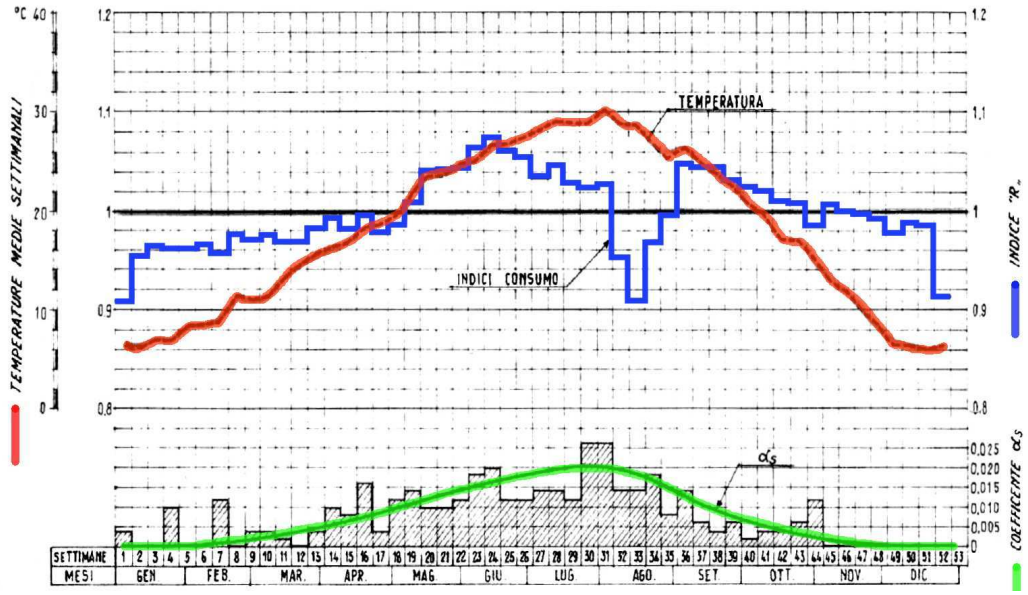


Figura 2.3 - *Temperatura media settimanale (linea rossa), indice settimanale* $\hat{r}_i$ *(linea blu) e coefficienti* α_i *che minimizzano l'errore assoluto (linea verde).*

2.1.2 La rete di distribuzione urbana e la previsione dei consumi giornalieri.

Abbiamo visto al paragrafo 2.1.1 come realizzare un modello di previsione dei consumi di una determinata settimana rispetto al consumo annuale. Questo ha agevolato l'esercizio dell'adduzione idrica, ovvero del trasporto dell'acqua dal sistema di produzione di Vicenza ai serbatoi di Padova.

Un elemento altrettanto importante è l'esercizio della distribuzione idrica all'interno della settimana, al fine di ottimizzare la distribuzione urbana dell'acqua dai serbatoi di accumulo alle utenze servite. Stimato il volume di consumo settimanale occorre considerare come questo viene ripartito nei vari giorni che compongono la settimana.

Analogamente a quanto fatto per lo studio della periodicità annuale, sono stati considerati i valori dei consumi dei singoli giorni in rapporto alla media dei consumi della settimana di appartenenza. Per ogni anno osservato si sono costruiti 52 rapporti per il giorno di lunedì, 52 per il martedì e così via. Considerando i 12 anni di osservazione si sono ottenuti 624 valori per l'indice di ciascun giorno della settimana.

Il calcolo dei rapporti può essere formalizzato come segue.

Detto *k* l'indice che rappresenta il giorno della settimana (1=lunedì,…,7=domenica) si indica con:

c_{ijk} il consumo relativo a: giorno *k*, settimana *i*, anno *j*
(*k*=1,…,7;*i*=1,…,52;*j*=1,…,12),

$\bar{c}_{ij}$ il consumo medio relativo a: settimana *i*, anno *j*: (*i*=1,…,52; *j*=1,…,12)

$$\bar{c}_{ij} = \frac{1}{7}\sum_{k=1}^{7} c_{ijk} \,, \tag{2.12}$$

s_{ijk} il rapporto del giorno *k* nella settimana *i* dell'anno *j*:

$$s_{ijk} = \frac{c_{ijk}}{\bar{c}_{ij}} \,. \tag{2.13}$$

Fissata la settimana *i* dell'anno *j*, il valore s_{ijk} vale 1 se nel giorno *k* il consumo è stato pari al consumo medio di quella settimana, sarà maggiore di 1 in caso di consumo superiore e minore di 1 in caso di consumo inferiore.
Sia $\bar{s}_k$ il rapporto medio nel giorno *k*.

$$\bar{s}_k = \frac{1}{12 \cdot 52}\sum_{j=1}^{12}\sum_{i=1}^{52} s_{ijk} \,. \tag{2.14}$$

La quantità $\bar{s}_k$ rappresenta il "peso" medio del giorno *k* nella ripartizione del consumo settimanale sui 7 giorni e viene definita *indice giornaliero*.
Nella Figura 2.4 si riportano gli indici giornalieri e i loro campi di variazione.
Dalla rappresentazione grafica si ricava che:

- i consumi dei giorni da lunedì al venerdì sono superiori alla media settimanale di una percentuale che va dall'1,5% al 2%;
- i consumi del sabato si attestano sulla media settimanale;
- i consumi della domenica sono inferiori dell'8% ÷ 10% rispetto alla media settimanale.

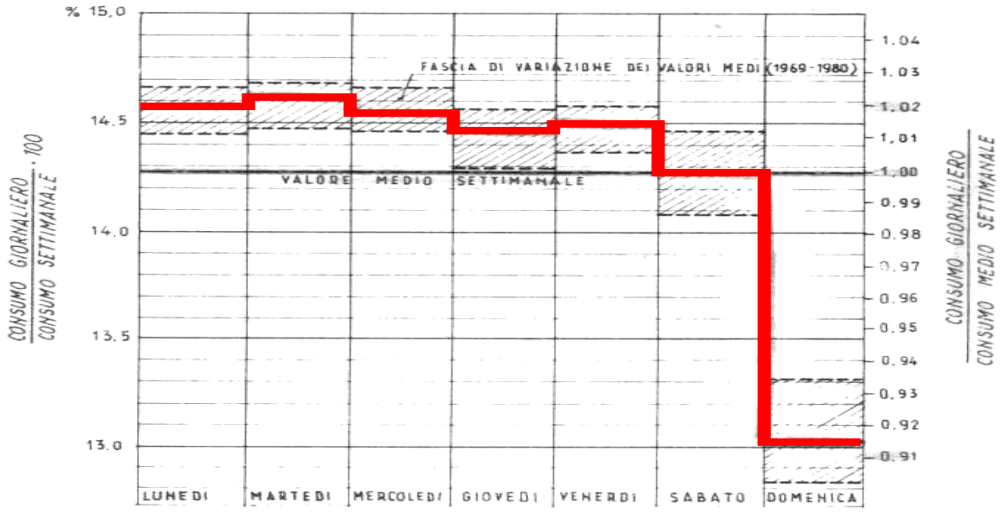


Figura 2.4 - *Indici giornalieri (linea rossa) e loro campo di variazione (area tratteggiata).*

2.2 Rielaborazione del modello Acegas-Aps (2003)

Una riproposizione del modello di Davanzo è stata realizzata nel 1993 dall'ing. Danillo Calaon, sempre di Acegas-Aps, analizzando i dati relativi al consumo idrico di Padova relativo al periodo 1995-2002. Seguendo le linee guida tracciate da Davanzo è stato ricalcolato il valore degli indici $\hat{r}_i$ per quel periodo, i cui valori sono rappresentati nel grafico superiore di Figura 2.5. Si conferma la presenza di valori elevati nel periodo estivo, bassi nel periodo invernale e intermedi in primavera e autunno. Sono presenti in modo marcato i cali dei valori dell'indice nei periodi di ferragosto (33^a settimana) e di Natale - capodanno (1^a e 52^a settimana). Nella parte inferiore del grafico è invece presente l'andamento medio della temperatura nel periodo osservato. I valori sono calcolati facendo riferimento alla temperatura massima giornaliera; di questi sono poi calcolati i valori medi nelle varie settimane e per ciascuna di queste la media negli anni a disposizione.

Confrontando congiuntamente i due andamenti al variare della settimana, appare evidente la correlazione tra le due variabili.

Raffrontando i due grafici appenda descritti con quanto rilevato da Davanzo nel 1982, rappresentato in Figura 2.2**,** si nota come questi siano quasi perfettamente sovrapponibili, segno che le abitudini di consumo idrico, a livello settimanale, siano rimaste sostanzialmente immutate nel tempo, così come l'andamento "sinusoidale" della temperatura massima.

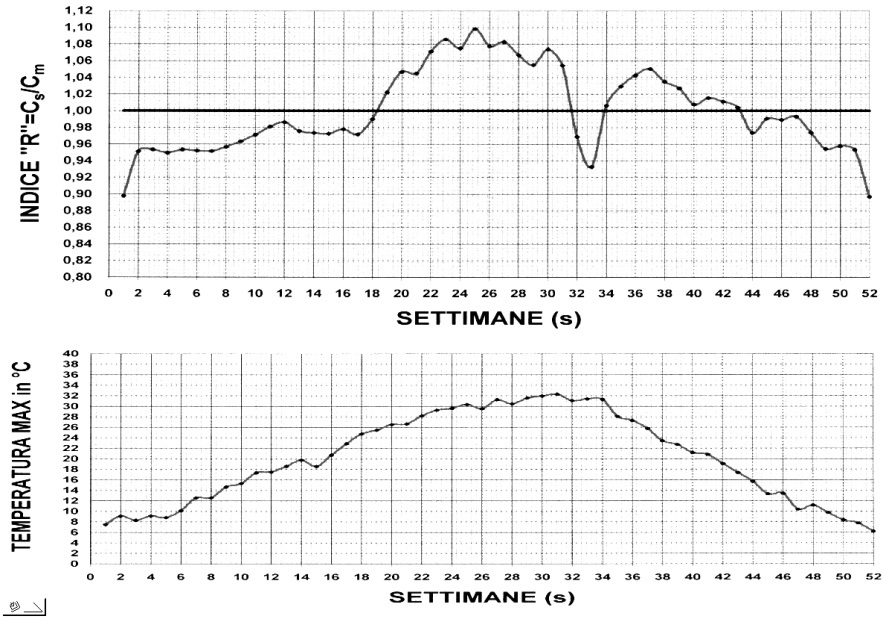


Figura 2.5 - *Confronto tra indice* $\hat{r}_i$ *, in alto, e temperatura massima giornaliera (media settimanale), in basso al variare della settimana.*

Il grafico in Figura 2.6 rappresenta il campo di variazione del coefficiente $\hat{r}_i$ al variare della settimana; la linea rossa indica il valore massimo (di ciascuna settimana) registrato negli 8 anni del periodo di osservazione, la linea verde il valore minimo e la linea nera il valore medio. Così come nel lavoro di Davanzo (vedi Figura 2.2), anche Calaon evidenzia la maggior variabilità dell'indice per le settimane del periodo centrale dell'anno, quando i valori del consumo idrico sono più elevati.

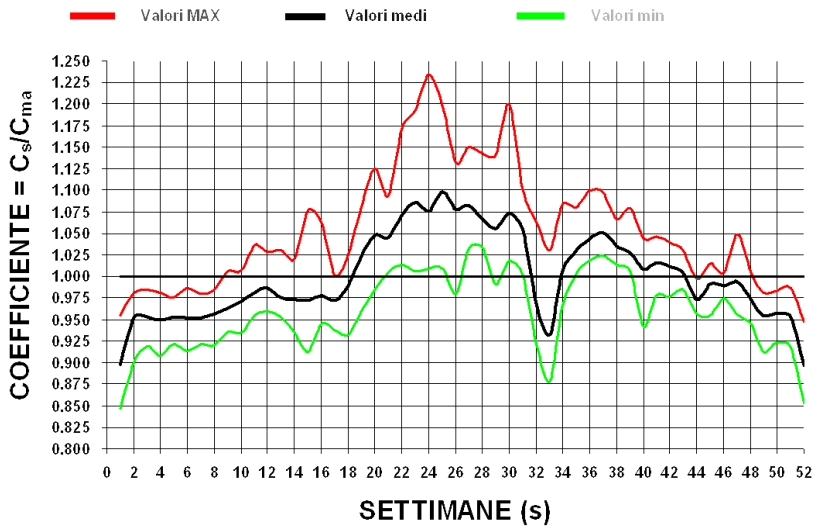


Figura 2.6 - *Calcolo dell'indice settimanale* $\hat{r}_i$ *al variare della settimana e suo campo di variazione per il periodo 1995-2002.*

Calaon propone una versione dello stimatore basato sul valore dell'erogato dell'anno precedente, ampliato o diminuito in base al valore di un *coefficiente termico* e alla differenza di temperatura. Si tratta di un caso particolare del modello di Davanzo, nel quale il coefficiente $\hat{r}_i$ è calcolato soltanto con i dati dell'anno precedente e la stima del consumo totale annuo si ricava per trasposizione di quanto realizzato l'anno precedente. In luogo del coefficiente $\hat{\alpha}_i$ viene utilizzato il *coefficiente termico* settimanale rappresentato nel grafico di Figura 2.7 che ne è sostanzialmente una versione semplificata. A titolo di esempio, per la 28ª settimana il consumo può essere stimato sulla base di quanto erogato la medesima settimana dell'anno precedente, aumentato (o ridotto) del 2% per ogni grado in più (o in meno) rispetto alla temperatura dell'anno precedente.

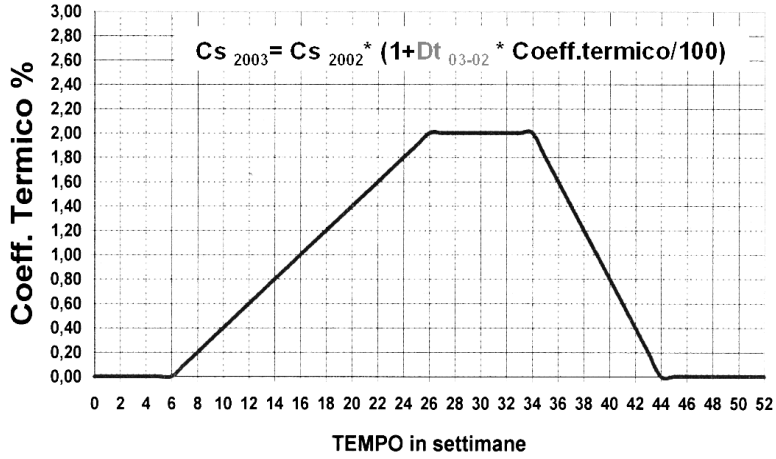


Figura 2.7 - *Coefficiente termico al variare della settimana.*

2.3 La previsione dei consumi idrici urbani attraverso reti neurali (2007)

Il 28 e 29 giugno 2007 si è tenuto a Ferrara un convegno sul tema "Approvvigionamento e distribuzione idrica: esperienze, ricerca ed innovazione" nel quale è stato presentato un modello di previsione dei consumi idrici che si basa sull'utilizzo di reti neuronali artificiali. È uno studio che pone evidenza sulla metodologia seguita per il calcolo dei pesi della rete neurale, per i quali viene utilizzato un algoritmo di tipo Bayesiano "Shuffled Complex Evolution Metropolis – Uncertainty Assessment" (SCEM-UA) (Campisano *et al.,* 2007).

Sono state sviluppate due famiglie di modelli: una per la previsione dei consumi idrici a scala oraria con orizzonte temporale di 1 ora, 2 ore, …, 24 ore e una per la previsione giornaliera con orizzonte di 1 giorno, 2 giorni,…, 7 giorni.

Le reti neuronali artificiali utilizzate sono del tipo *feedforward* con un singolo strato nascosto. Le funzioni di attivazione dello strato di input e intermedio sono di tipo logistico, quelle dello strato di output di tipo lineare. Per gli aspetti metodologi e teorici sulle reti neurali si rimanda al Capitolo 4.

Il numero di neuroni presenti nello strato nascosto varia con gli orizzonti temporali in funzione di un migliore adattamento della rete, mantenendo un valore non superiore al doppio del numero di neuroni dello strato di input, al fine di ridurre la possibilità di *overfitting*.

I modelli sono stati realizzati su una parte della rete di distribuzione della città di Catania la cui utenza è rappresentativa dell'intera rete di distribuzione cittadina, e hanno riguardato il periodo di osservazione 1/1/2003 – 31/12/2004. Sono stati utilizzati i dati dell'anno 2003 per la calibrazione dei modelli e quelli dell'anno 2004 per la validazione.

Le prestazioni dei modelli costruiti sono comparate con altri 3 metodi di stima: il primo basato sempre su reti neuronali con algoritmo di calibrazione dei parametri di tipo Bayesiano, il secondo su un modello regressivo lineare e il terzo su sistemi inferenziali di *neuro-fuzzy* (ANFIS). A tale scopo è stato calcolato, per i modelli realizzati, l'indice di efficienza EI di Nash e Sutcliffe (1970):

$$EI = 1 - \frac{\sum \left(f_{obs} - f_{pred} \right)^2}{\sum \left(f_{obs} - f_{mean} \right)^2} \ , \tag{2.15}$$

Dove:

f_{obs} = valore reale osservato,

f_{pred} = valore previsto dal modello,

f_{mean} = valore medio delle osservazioni.

L'indice assume valore 1 in caso di previsione coincidente con i valori reali, valore 0 nel caso la bontà della stima sia equivalente alla semplice media dei dati reali e valori negativi nei casi in cui la media sia migliore delle stime prodotte dal modello.

La bontà dell'adattamento della rete calibrata mediante l'algoritmo SCEM-UA è paragonabile a quella degli altri modelli basati sulle reti neurali e superiore a quanto realizzato mediante modelli di regressione lineare, sia considerando le previsioni a scala giornaliera, sia considerando quelle a scala oraria. Il vantaggio della soluzione proposta risiede nella possibilità di fornire un intervallo di confidenza per i valori stimati.

2.3.1 La previsione a scala giornaliera

Nei modelli di previsione a scala giornaliera, indicando con t l'ultimo giorno del quale si conosce il consumo idrico realizzato, l'obiettivo è la stima del valore del consumo del giorno $t+i$, dove i rappresenta l'orizzonte temporale di previsione e assume valori da 1 a 7. Per ogni valore di i viene costruito uno specifico modello basato su reti neurali.

Ciascun modello presenta 3 neuroni di input corrispondenti alle seguenti variabili:

Q_t consumo idrico giornaliero al tempo t, cioè l'ultimo valore disponibile;

ξ_{t+i} assume valore 0 o 1 e caratterizza il giorno $t+i$ rispettivamente come feriale o festivo;

d_{t+i} assume valori da 1 a 7 e specifica il giorno di previsione $t+i$ all'interno della settimana.

Il neurone dello strato di output, corrispondente al consumo idrico stimato al tempo $t+i$, viene indicato con Q_{t+i}.

I modelli realizzati, al variare dell'orizzonte temporale di previsione, hanno mostrato buone capacità predittive come si può verificare dal grafico di Figura 2.8. I valori dell'indice di efficienza di Nash-Sutcliffe ottenuti nel caso di reti neurali SCEM-UA sono comparabili con i valori ottenuti mediante reti neurali Bayesiane e lievemente migliori dei valori relativi ai modelli basati su leggi di regressione e i modelli ANFIS. In aggiunta, si è osservato un decadimento della capacità predittiva dei diversi modelli al crescere dell'orizzonte temporale di previsione.

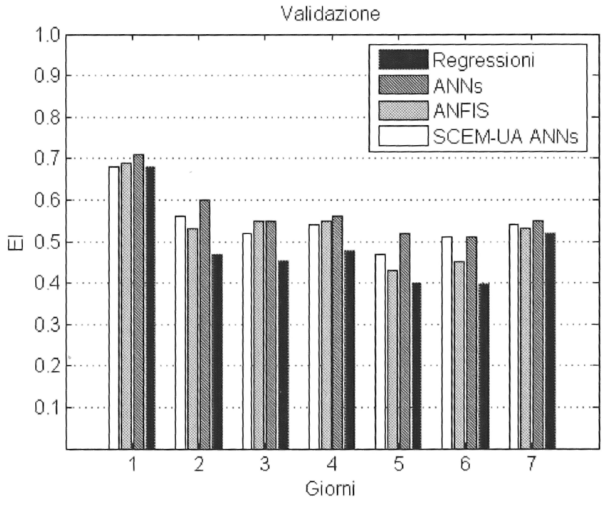


Figura 2.8 - *Indice EI per i modelli giornalieri al variare dell'orizzonte temporale (Validazione)*

2.3.2 La previsione a scala oraria

Nei modelli di previsione a scala oraria, indicando con t l'ultima ora per la quale si conosce una registrazione del consumo idrico realizzato, l'obiettivo è la stima del valore del consumo all'ora $t+h$, dove h rappresenta l'orizzonte temporale di previsione e assume valori da 1 a 24. Per ogni valore di h viene costruito uno specifico modello basato su reti neurali.

Ciascun modello presenta 4 neuroni di input corrispondenti alle seguenti variabili:

Q_t consumo idrico giornaliero all'ora t, cioè l'ultimo valore orario disponibile;

ξ_{t+h} assume valore 0 o 1 e caratterizza il giorno relativo all'ora $t+h$ rispettivamente come feriale o festivo;

d_{t+h} assume valori da 1 a 7 e specifica il giorno relativo all'ora di previsione $t+h$ all'interno della settimana;

ϑ_{t+h} assume valori da 1 a 24 e specifica l'ora di previsione all'interno del giorno.

Il neurone dello strato di output, corrispondente al consumo idrico stimato per l'ora $t+h$, viene indicato con Q_{t+h}.

Anche per questa classe di modelli è stata condotta una comparazione con i modelli equivalenti già descritti in precedenza. Nel grafico di Figura 2.9 si possono osservare i valori dell'indice di efficienza di Nash-Sutcliffe per valutare la bontà delle diverse soluzioni: i modelli basati su reti neurali SCEM-UA sono comparabili con i valori ottenuti mediante reti neurali Bayesiane e lievemente migliori dei valori ricavati mediante le leggi di regressione e i modelli ANFIS.

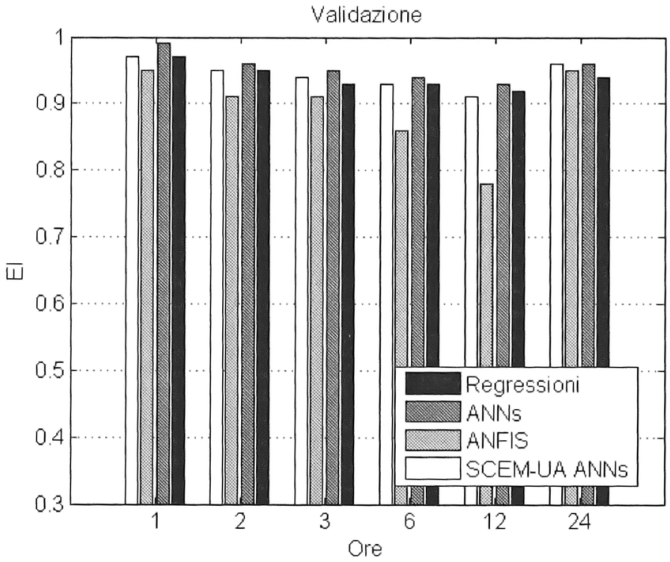


Figura 2.9 - *Indice EI per i modelli orari al variare dell'orizzonte temporale di previsione (validazione)*

Capitolo 3
-
Aggiornamenti e nuovi modelli di previsione per i consumi idrici urbani

In questo capitolo viene dapprima verificata l'attualità della problematica della previsione dei consumi idrici a Padova e la sussistenza delle condizioni necessarie a renderla tutt'oggi di interesse. Fatto ciò, dopo aver presentato il *dataset* attuale a disposizione, sono state condotte analisi per la stima del consumo idrico applicando diversi modelli confrontandoli poi sulla base di indicatori.

Per spiegare il *dataset* utilizzato si è proceduto illustrando l'origine dei dati raccolti, descrivendo tutte le variabili che lo compongono ed infine elaborando statistiche descrittive su queste ultime.

Le analisi per la stima del consumo idrico vengono condotte dapprima applicando i modelli Acegas-Aps al *dataset* attuale e poi costruendo diverse soluzioni mediante l'applicazione di alcuni modelli statistici, in particolare regressione lineare, reti neurali e *Random Forests*.

Per agevolare i riferimenti ai modelli proposti, ciascuno di essi è identificato da un codice che viene riportato tra parentesi nel titolo del paragrafo nel quale viene presentato.

Per le ultime due tipologie di modelli (reti neurali e *Random Forests*) è possibile trovare un approfondimento teorico nel Capitolo 4. Per migliorare la leggibilità, i comandi per la costruzione dei modelli realizzati mediante il software R, nonché gli output che ne riportano le caratteristiche, sono riportati in Appendice.

Nella logica costruttiva dei modelli di Acegas-Aps il riferimento temporale è la settimana. In essi viene stimato il consumo per l'intera settimana di interesse ed in seguito questo viene ripartito nei giorni che la compongono. Nei nuovi modelli elaborati la logica è inversa: vengono dapprima stimati i consumi giornalieri per i singoli giorni e in seguito questi vengono combinati al fine di produrre modelli settimanali. Tale aggregazione è stata fatta solo per i modelli basati su reti neurali e

su *Random Forests* che meglio si adattavano alla rappresentazione della problematica.
Nell'ultimo paragrafo (§3.7) si presenta un'analisi comparativa dei modelli elaborati sulla base delle prestazioni e delle caratteristiche che possano rendere conveniente l'adozione di una soluzione rispetto ad un'altra, viene redatto anche un quadro d'insieme dei modelli elaborati e si espongono le considerazioni conclusive del presente lavoro.

3.1 Il consumo idrico oggi a Padova

La prima questione che è stata posta era la verifica dell'attualità della problematica relativa alla previsione dei consumi idrici. A tale scopo sono stati intervistati Danillo Calaon e Pietro Brazzarola di Acegas-Aps, dai quali si è avuta conferma della persistenza dell'interesse alla previsione dei consumi idrici. Esistono ancora le pompe utilizzate per forzare l'adduzione dai pozzi del vicentino spingendo l'acqua all'interno delle condotte, ancora caratterizzate dalla funzione cubica che lega l'aumento della portata con l'impiego di energia elettrica necessaria. La messa in produzione della nuova condotta in acciaio da $1300\,mm$ di diametro, in affiancamento a quella in cemento-amianto da $900\,mm$ di diametro, non rende superfluo l'uso delle pompe per aumentare la portata poiché permangono dei tratti di percorso serviti solo dalla vecchia condotta. In particolare non sono ancora realizzati gli attraversamenti dei fiumi Tesina e Brentelle, e manca il collegamento tra i serbatoi di Montà e di Brentelle. Quindi le due condotte, pur lavorando per larga parte del tracciato in parallelo, consentono una portata per caduta (senza cioè l'ausilio di pompe) stimata in $600\,l/s$. La previsione dei consumi idrici consente la programmazione dell'adduzione a portata costante e conseguentemente la minimizzazione dell'energia elettrica necessaria al funzionamento delle pompe.
Un ulteriore vantaggio sottolineato da Acegas-Aps derivante dall'esercizio degli impianti con adduzione a portata costante è la minimizzazione degli interventi di manovra sul sistema idrico. Questi, quando vengono eseguiti, comportano degli scompensi all'equilibrio del sistema idrico: ad esempio, a fronte di avvio/arresto delle pompe idrauliche si registrano dei colpi d'ariete[12] che possono alla lunga danneggiare gli elementi del sistema di distribuzione.

[12] Il *colpo d'ariete* è un fenomeno idraulico che si presenta in una condotta quando un flusso di liquido in movimento al suo interno viene bruscamente fermato dalla repentina chiusura di una valvola. O viceversa, quando una condotta chiusa e in pressione viene aperta repentinamente. Consiste in un'onda di pressione che si

Relativamente alla dimensione totale dei serbatoi di accumulo, considerando le tre centrali esistenti di Montà, Brentelle e Stanga, si arriva ad una capienza complessiva di 150.000 m^3, rispetto ai 45.000 m^3 censiti nel 1982.
Il quantitativo erogato medio si aggira intorno a 100.000 m^3/g (metri cubi al giorno), con punte massime che possono arrivare a 150.000 m^3/g.
Relativamente al ciclo giornaliero, questo viene gestito tramite l'accumulo presente nei vari serbatoi pensili che, oltre ad avere funzione di garanzia della pressione nella rete di distribuzione, hanno una capacità di immagazzinamento di acqua che può garantire un'autonomia di circa 24 ore.

3.2 I dati a disposizione

3.2.1 Le informazioni

I dati forniti da Acegas-Aps sono relativi al periodo di tempo 1/1/1995 – 31/12/2008 e riguardano il quantitativo di acqua erogata dall'acquedotto per la distribuzione alle utenze della città di Padova. Tali informazioni, tra le altre, sono registrate e memorizzate da sistemi informatici moderni e verificate dal personale dell'azienda presente in una sala controllo che gestisce, 24 ore su 24, l'esercizio degli impianti idrici cittadini. Il sistema di telecontrollo e telecomando consente di visualizzare la misura di numerose grandezze relative al sistema acquedottistico, ad esempio i livelli dei serbatoi e le portate delle condotte, di registrarne l'andamento storico in opportuni *database* e di impartire ordini alle postazioni periferiche, come l'avvio o l'arresto di pompe idrauliche per variare le portate o l'apertura o chiusura di saracinesche per modificare il funzionamento della rete idraulica.
L'erogato giornaliero di Padova deriva dagli impianti di adduzione che veicolano l'acqua da:

- risorgive e pozzi di Dueville, la cui portata è misurata all'arrivo della canaletta nella centrale di Brentelle;
- condotta in cemento-amianto di diametro 900 *mm* e condotta in acciaio di diametro 1300 *mm* le cui portate sono misurate alla partenza nella centrale di Anconetta (Vicenza);

origina in prossimità della valvola a causa dell'inerzia della colonna di fluido in movimento che impatta contro la parete della valvola chiusa. L'intensità del colpo e il valore della pressione massima dell'onda possono raggiungere livelli tali da far esplodere le condotte.

- impianti di potabilizzazione di Brentelle, che ricavano acqua dal fiume Brentella e dalla sua golena, la portata è misurata all'uscita degli impianti di potabilizzazione.

Mediante l'utilizzo di misuratori venturimetrici o tubi di Venturi[13] è possibile conoscere il valore della portata per ciascuna di queste fonti che sommati consentono di calcolare l'erogato giornaliero. Come tutti gli strumenti di misura, anch'essi presentano un margine di errore che in questo caso può essere stimato intorno al 3%-5%, variabile in ragione del volume della portata: a volumi ridotti l'errore è maggiore in quanto la condotta tende a "svuotarsi" e l'effetto Venturi non avviene. Inoltre l'errore si presenta prevalentemente come distorsione rispetto al vero valore piuttosto che come varianza di misura.

3.2.2 Predisposizione del dataset

Per la realizzazione delle analisi e la costruzione dei modelli è stato utilizzato il pacchetto software R versione 2.9.0. I dati originari forniti da Acegas-Aps in formato Excel sono stati importati nell'ambiente R mediante il package Rcmdr. A seguire, viene proposta una descrizione delle variabili utilizzate, i metodi di predisposizione dei *dataset* ed alcune verifiche preliminari eseguite sui dati a disposizione.

3.2.2.1 Variabili e trasformazione dati

I dati forniti da Acegas-Aps relativi al periodo 1995-2008 si compongono delle seguenti 3 variabili:

`data` data di riferimento (dal 1/1/1995 al 31/12/2008);

`erogato0` consumo realizzato il giorno di riferimento espresso in m^3;

`temperatura` valore della temperatura massima in $°C$, registrata nel giorno di riferimento.

[13] Il venturimetro o tubo di Venturi è uno strumento che serve a misurare la portata di una condotta. Questo strumento sfrutta l'effetto Venturi (dal fisico Giovanni Battista Venturi) che studia la relazione tra la velocità media del fluido e la sua pressione. È poi possibile calcolare la portata volumetrica come prodotto della velocità per l'area della sezione di condotta considerata. Il tubo di Venturi si costruisce mediante la realizzazione nella condotta di una strozzatura graduale, a causa della quale si avrà un incremento di velocità dell'acqua (essendo la portata costante, ad una diminuzione della sezione deve corrispondere un aumento della velocità). Secondo l'equazione di Bernoulli, nelle sezioni di una condotta deve rimanere costante la somma di tre componenti: la quota della condotta (valore costante nello nostro caso), quadrato della velocità (come detto, più elevata nel tratto di condotta a sezione inferiore) e pressione. A causa di questo vincolo, la pressione nel tratto a sezione inferiore è più bassa. La differenza di pressione nei due tratti di condotta è proporzionale, ancora con legge quadratica, alla velocità nella strozzatura, da cui si calcola la portata. Il tubo Venturi fornisce misure di ottima precisione, ed alcuni tipi sono ammessi come misuratori fiscali; per le sue caratteristiche si adatta a un vasto campo di portate, da pochi metri cubi ora a molte migliaia.

A partire da queste tre semplici variabili ne sono state costruite o ricavate altre, da utilizzare come predittori nei modelli realizzati.

Un insieme di variabili sono state ottenute dalla scomposizione della variabile `data` nelle sue componenti (anno, mese, giorno, settimana, ecc.):

`giorno_settimana` rappresenta il giorno della settimana con valori: 1=lunedì, 2=martedì, …, 7=domenica;

`giorno_anno` rappresenta il giorno progressivo all'interno dell'anno, con valori che vanno da 1 per il 1° gennaio di ciascun anno fino a 365 (o 366 per gli anni bisestili) per il 31 dicembre; la variabile è considerata qualitativa;

`giorno_anno_n` è l'equivalente della variabile `giorno_anno`, ma è considerata quantitativa;

`anno` rappresenta l'anno di riferimento e assume valori che vanno da 1995 a 2008;

`mese` rappresenta il mese di riferimento e assume valori che vanno da 1 (gennaio) a 12 (dicembre);

`giorno` rappresenta il giorno del mese e assume valori che vanno da 1 a 31;

`settimana` rappresenta il progressivo della settimana all'interno dell'anno. Seguendo quanto indicato da Davanzo (cfr. §2.1.1.2) la variabile assume valore 1 per giorni dall'1 al 7 gennaio, valore 2 per i giorni dall'8 al 14 gennaio, …, 52 per l'ultima settimana completa dell'anno e 53 per l'ultimo giorno dell'anno (o ultimi 2 nel caso di anno bisestile);

`festivo` variabile dicotomica che assume valore 1 nei giorni festivi sotto elencati, 0 in tutti gli altri casi:

- 1 gennaio;
- 6 gennaio, epifania;
- 25 aprile, festa della Liberazione;
- 1 maggio, festa dei lavoratori;
- 2 giugno, festa della Repubblica (solo dal 2000 in poi);
- 13 giugno (S. Antonio, patrono di Padova);
- 15 agosto, ferragosto;
- 1 novembre, tutti i Santi;
- 8 dicembre, Immacolata Concezione;
- 25 e 26 dicembre, Natale e S. Stefano;
- Pasqua e Pasquetta.

Un altro insieme di variabili è quello generato a partire dalla variabile `erogato0` per traslazione di data. Sulla stessa riga sono stati riportati i volumi di acqua erogati il giorno precedente, 2 giorni precedenti, …, 7 giorni precedenti, rispettivamente memorizzati nelle variabili:

`erogato1` volume di acqua erogata il giorno precedente la data di riferimento;
`erogato2` volume di acqua erogata 2 giorni antecedenti la data di riferimento;
`erogato3` volume di acqua erogata 3 giorni antecedenti la data di riferimento;
`erogato4` volume di acqua erogata 4 giorni antecedenti la data di riferimento;
`erogato5` volume di acqua erogata 5 giorni antecedenti la data di riferimento;
`erogato6` volume di acqua erogata 6 giorni antecedenti la data di riferimento;
`erogato7` volume di acqua erogata 7 giorni antecedenti la data di riferimento.

È stato necessario definire un altro gruppo di variabili che rappresentano la versione normalizzata a valori nell'intervallo $[0,1]$ dei volumi di acqua erogata. La normalizzazione è stata ottenuta sottraendo alla variabile originale il valore minimo di questa e dividendo per la differenza tra i valori massimi e minimi. Indicando con x la variabile originaria, la versione normalizzata $\tilde{x}$ viene espressa:

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.1}$$

Le variabili normalizzate sono:

`erogato0_n` versione normalizzata della variabile `erogato0`;
`erogato1_n` versione normalizzata della variabile `erogato1`;
`erogato2_n` versione normalizzata della variabile `erogato2`;
`erogato3_n` versione normalizzata della variabile `erogato3`;
`erogato4_n` versione normalizzata della variabile `erogato4`;
`erogato5_n` versione normalizzata della variabile `erogato5`;
`erogato6_n` versione normalizzata della variabile `erogato6`;
`erogato7_n` versione normalizzata della variabile `erogato7`.

Un'altra variabile è stata ottenuta dal valore dell'*indice settimanale*, come definito nel §2.1.1.1:

`rapp_settimana` valore dell'indice r_{ij} relativo alla stessa settimana (i=`settimana`) della data di riferimento nell'anno precedente (j=`anno`-1); per costruzione la variabile presenta dati mancanti per tutto il primo anno di osservazione (1995). Tale variabile fornisce un'indicazione del "peso", in termini di consumo idrico, della settimana all'interno dell'anno, rilevato l'anno precedente.

L'ultima variabile presente nel *dataset* viene introdotta per suddividere il campione di osservazioni in due parti, una utilizzata per calcolare i parametri dei modelli realizzati, l'altra per verificarne le prestazioni su un insieme di dati differente:

`training_validation` assume valore "T" quando la data di riferimento è inferiore o uguale al 31/12/2006, valore "V" per valori della data di riferimenti superiori od uguali a 1/1/2007. I primi 12 anni del campione vengono usati per la costruzione dei modelli, gli ultimi 2 anni per la loro validazione.

3.2.2.2 Suddivisioni del dataset

Il *dataset* complessivo, denominato `pd.all`, risultante dall'importazione dei dati è composto da 5.114 osservazioni. A partire da questo, sono stati ricavati altri due *dataset*, sulla base della variabile `training_validation`. Il primo *dataset*, denominato `df.train`, contiene le 4.383 osservazioni per le quali la variabile assume la modalità "T" e viene utilizzato per la costruzione dei modelli. Il secondo *dataset*, denominato `df.valid`, contiene le 731 osservazioni per le quali la variabile assume la modalità "V" e viene utilizzato per la validazione dei modelli elaborati e per il confronto tra di essi.

Poiché la variabile `rapp_settimana` non è definita nel primo anno di osservazione (1995), per la costruzione dei modelli che utilizzano come predittore tale variabile, è stato necessario considerare un sottoinsieme del *dataset* `df.train`, chiamato `df.train_r`, contenente solo i dati dal 1/1/1996 al 31/12/2006.

3.2.2.3 Dati mancanti e controlli del dataset

Nel campione in esami sono presenti solo due valori mancanti della variabile `erogato0` relativi ai giorni 30/9/2000 e 5/6/2005. In entrambi i casi il dato mancante è stato sostituito con la media delle osservazioni dei due giorni adiacenti.

Sono state realizzate delle semplici tabelle di frequenze per verificare la "congruenza" dei dati a disposizione, ad esempio per controllare che tutte le settimane contenessero 7 osservazioni, che tutti gli anni ne contenessero 365, o 366 se bisestili, e così via. Da queste non sono emersi ulteriori problemi con i dati a disposizione.

3.2.3 Statistiche descrittive

Di seguito vengono presentate alcune statistiche descrittive delle variabili componenti il *dataset* a disposizione. Nell'Appendice A è possibile visualizzare i comandi dell'ambiente R necessari per la realizzazione dei grafici.

Nella Figura 3.1 si può vedere la distribuzione delle frequenze della variabile `erogato0`, di forma campanulare, con valore medio pari a $113.720\,m^3$ e scarto quadratico medio di $9.692\,m^3$.

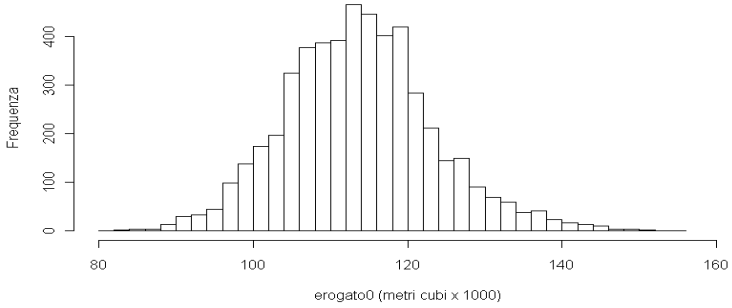


Figura 3.1 - *Distribuzione di frequenze del volume di acqua erogato (1995-2008)*

La distribuzione è caratterizzata da una forma che si adatta abbastanza bene a quella gaussiana come evidenziato anche dal *normal probability plot* di Figura 3.2.

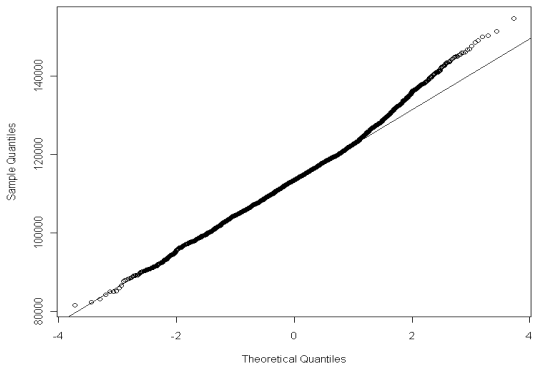


Figura 3.2 - *Normal probability plot della variabile* `erogato0`

Il grafico di Figura 3.3 è realizzato visualizzando i valori della temperatura massima giornaliera e del volume di acqua erogato per i giorni nel periodo 1995-2008. Appare evidente la relazione diretta che lega le due grandezze osservate.

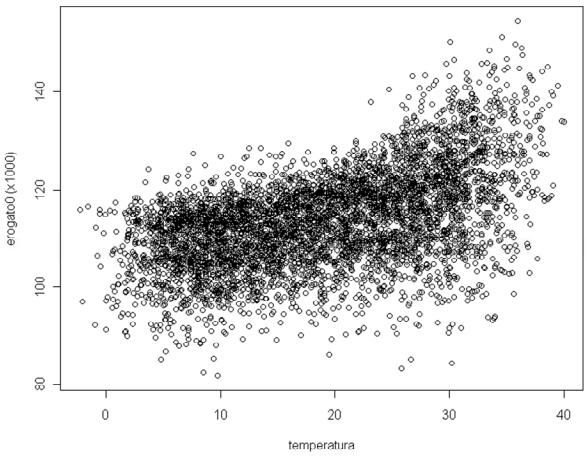


Figura 3.3 - *Valori della temperatura massima e volume di acqua erogato, valori giornalieri 1995-2008.*

Analizzando l'erogato sulla base del giorno della settimana (Figura 3.4**)** è evidente come questo sia pressoché costante nei giorni dal lunedì al venerdì, inferiori il sabato e soprattutto la domenica. Questi risultati sono in linea con quanto riscontrato da Davanzo (cfr. §2.1.2).

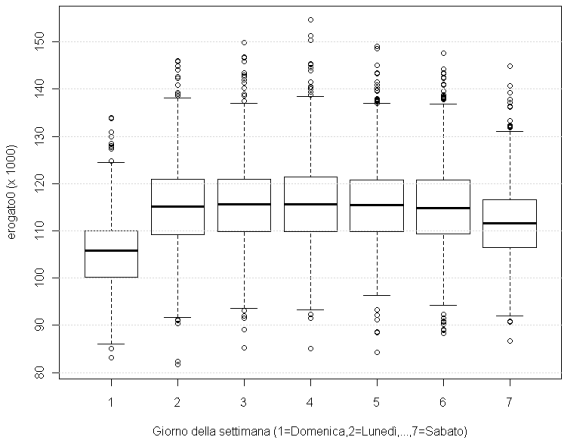


Figura 3.4 - *Boxplot del volume di acqua erogato per giorno della settimana. Dati giornalieri dal 1995 al 2008.*

Nel grafico di Figura 3.5 viene evidenziato l'andamento del valore dell'erogato giornaliero rispetto al giorno dell'anno. Anche in questo caso è possibile effettuare un parallelo con quanto rilevato da Davanzo (vedi Figura 2.2), si osserva, infatti, un aumento dell'erogato nel periodo centrale dell'anno e una diminuzione marcata in corrispondenza dei periodi festivi (Natale e capodanno) e di ferie estive, nei quali si registra un numero inferiore di presenze in città.

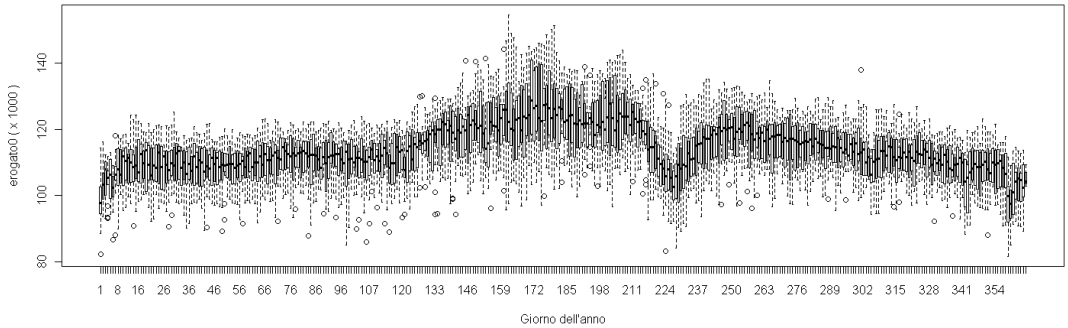


Figura 3.5 - *Boxplot del volume di acqua erogato per giorno dell'anno (1…365). Dati giornalieri dal 1995 al 2008.*

Analoghe considerazioni si possono trarre dalla visualizzazione dei grafici di Figura 3.6 e Figura 3.7, nei quali il medesimo andamento è visualizzato rispettivamente a livello di settimana e di mese.

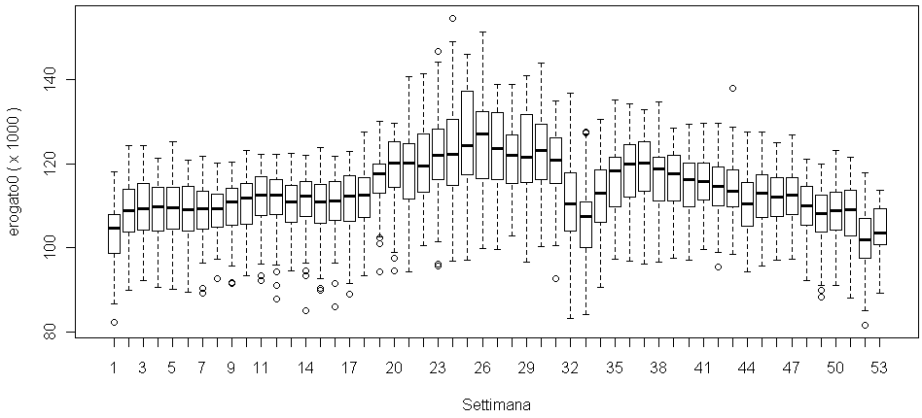


Figura 3.6 - *Boxplot del volume di acqua erogato per settimana. Dati giornalieri dal 1995 al 2008.*

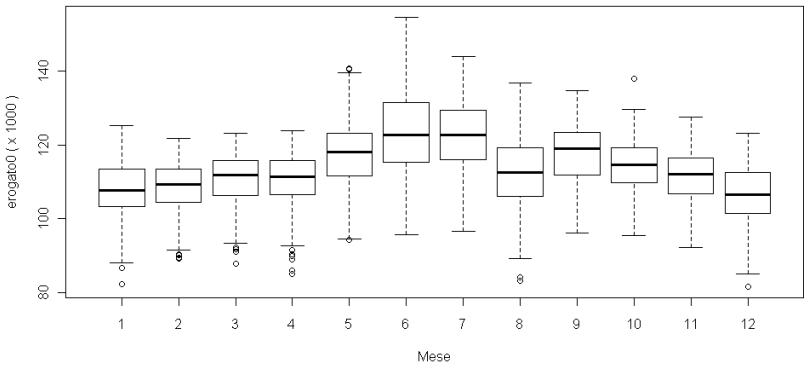


Figura 3.7 - *Boxplot del volume di acqua erogato per mese. Dati giornalieri dal 1995 al 2008.*

L'andamento annuale della variabile `erogato0` è rappresentato in Figura 3.8. Si nota un aumento dei consumi idrici dal 1995 al 2003 seguita da un calo dal 2003 al 2008. Per spiegare tali variazioni si possono ipotizzare fenomeni quali l'ottimizzazione

della rete di distribuzione in termini di diminuzione delle perdite, un aumento dell'uso di elettrodomestici quali lavatrici e lavastoviglie che necessitano di quantità ridotte di acqua, un'ottimizzazione dei consumi sia domestici (minor spreco) che industriali (aggiornamenti tecnologici) oppure una variazione nei regimi climatici.

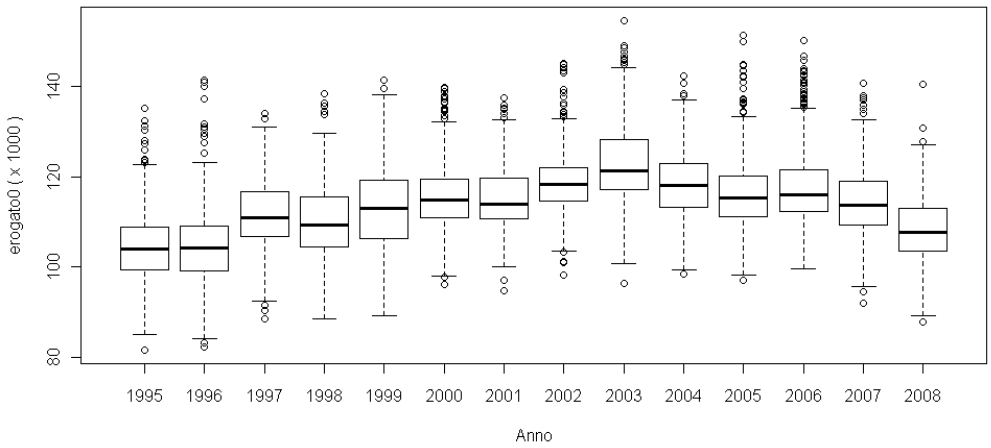


Figura 3.8 - *Boxplot del volume di acqua erogato per anno. Dati giornalieri dal 1995 al 2008.*

L'andamento della temperatura nelle settimane che compongono l'anno è rappresentato in Figura 3.9, nella quale è evidente l'andamento sinusoidale di periodicità annua.

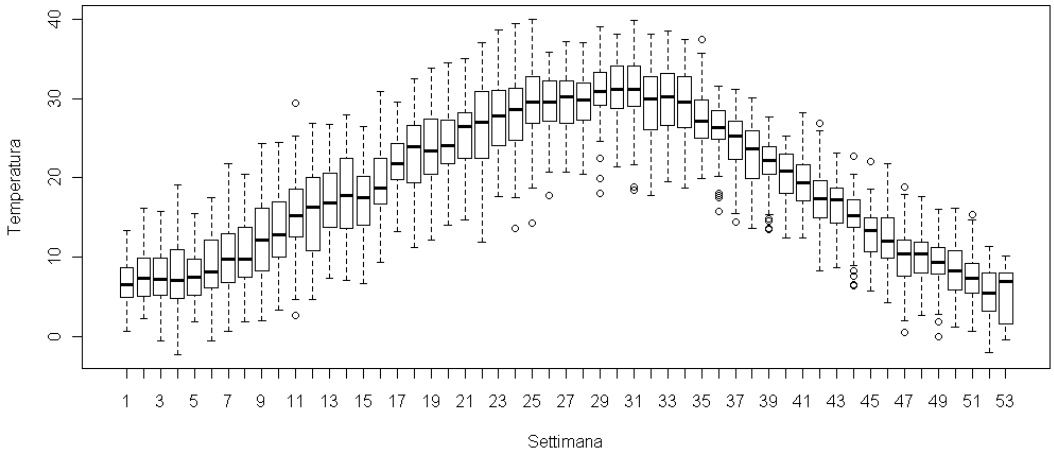


Figura 3.9 - *Boxplot della temperatura massima (°C) per settimana. Dati giornalieri dal 1995 al 2008.*

Viene considerato infine l'effetto sul volume di acqua erogato operato dalla variabile `festivo` descritta al §3.2.2.1. Appare evidente la diminuzione dell'erogato in corrispondenza dei giorni festivi.

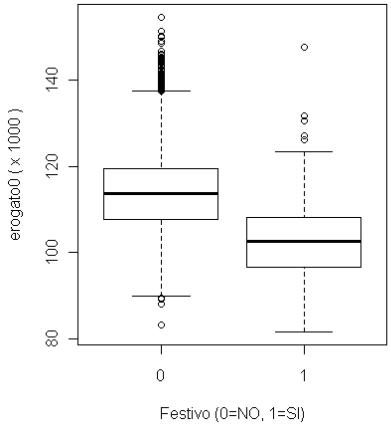


Figura 3.10 - *Boxplot del volume di acqua erogato per giorno festivo o feriale. Dati giornalieri dal 1995 al 2008.*

3.3 Modello Acegas-Aps con dati attuali

I primi modelli elaborati sui dati a disposizione sono quelli di Acegas-Aps proposti da Davanzo.

Secondo quanto previsto dalle linee guida del lavoro di Davanzo, l'analisi della previsione dei consumi a 7 giorni si basa sulla stagionalità annuale dei consumi settimanali, valutata per le settimane intere, con esclusione della settimana 53ª; è stata quindi operata una riduzione del *dataset* considerando solo i primi 364 dati per ogni anno.

Il *dataset* a disposizione è stato suddiviso in due parti: il *training set*, contenente i dati dal 1995 al 2006, per la costruzione dei modelli e il *validation set* per la loro validazione contenente i dati del 2007 e 2008.

3.3.1 Previsione dei consumi settimanali

3.3.1.1 Modellazione della componente stagionale

La prima elaborazione è stata la determinazione del consumo giornaliero medio per ogni anno osservato calcolando le quantità:

$$\bar{c}_j = \frac{1}{52 \cdot 7} \sum_{i=1}^{52} \sum_{k=1}^{7} c_{ijk} \qquad j=1,\ldots,12 \text{ (anni dal 1995 al 2006)} \qquad (3.2)$$

dove c_{ijk} è il consumo (variabile `erogato0`) del giorno k (variabile `giorno_settimana`, k=1,..,7) della settimana i (variabile `settimana`, i=1,…,52) dell'anno j (variabile `anno`, j=1,…,12).

I valori calcolati sono riportati nel grafico di Figura 3.11, dove per confronto si riporta anche la temperatura massima giornaliera (media annua).

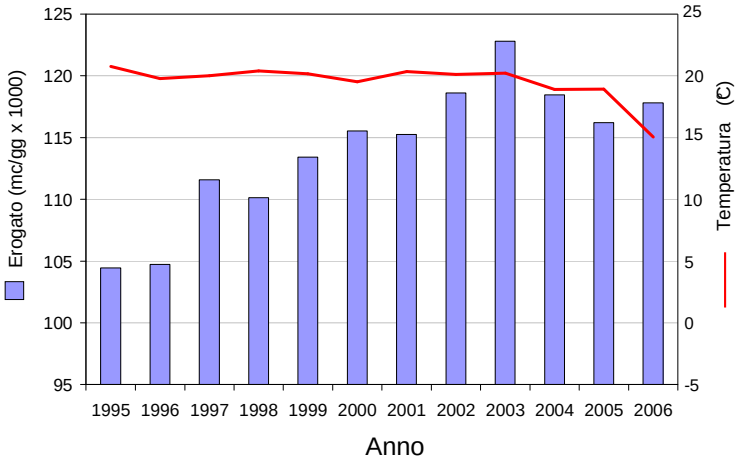


Figura 3.11 - *Consumo giornaliero e temperatura massima giornaliera per anno (medie annue).*

Sono stati calcolati i valori di consumo medio settimanale:

$$\bar{c}_{ij} = \frac{1}{7}\sum_{k=1}^{7} c_{ijk} \qquad i=1,\ldots,52 \;\; j=1,\ldots,12\,, \tag{3.3}$$

gli indici settimanali nel modo seguente:

$$r_{ij} = \frac{\bar{c}_{ij}}{\bar{c}_j} \qquad i=1,\ldots,52 \;\; j=1,\ldots,12 \tag{3.4}$$

e la loro media calcolata sui 12 anni del *training set*:

$$\hat{r}_i = \frac{1}{12}\sum_{j=1}^{12} r_{ij} \qquad i=1,\ldots,52. \tag{3.5}$$

Per ciascuna settimana sono stati calcolati i valori massimi e minimi rilevati nei 12 anni osservati:

$$r_i^{\max} = \max_j (r_{ij}) \qquad i=1,\ldots,52 \tag{3.6}$$

$$r_i^{\min} = \min_j (r_{ij}) \qquad i=1,\ldots,52. \tag{3.7}$$

Nel grafico di Figura 3.12 si riporta la rappresentazione degli indici $\hat{r}_i$ ottenuti, assieme ai valori minimi $r_i^{\min}$ e massimi $r_i^{\max}$.

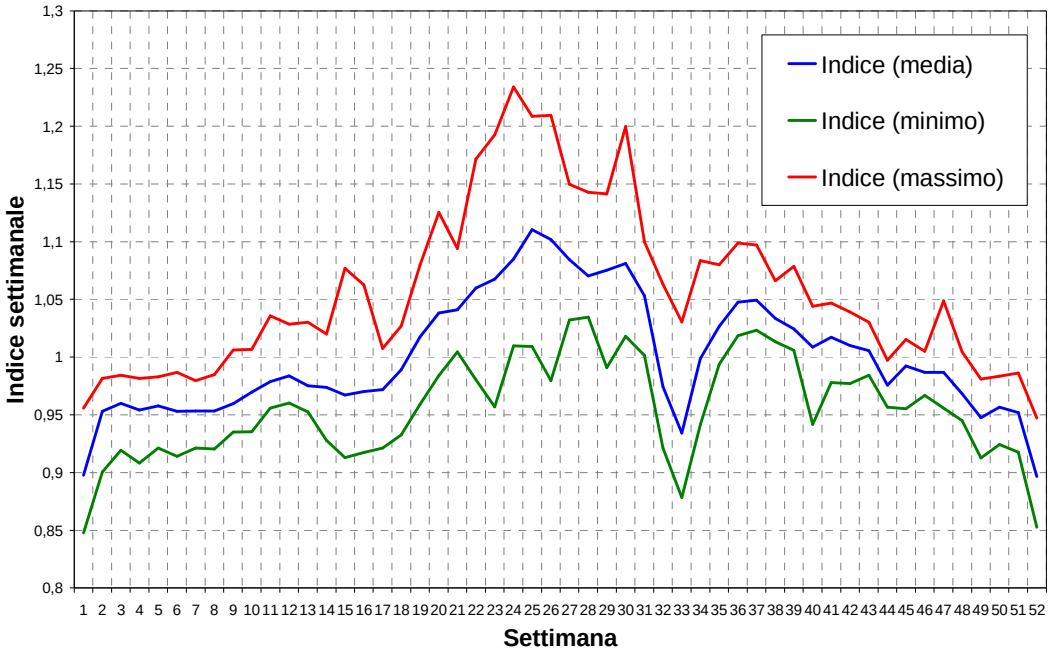


Figura 3.12 - *Indici settimanali al variare della settimana: valore medio (in blu), valore minimo (in verde) e valore massimo (in rosso).*

Sono state calcolate le temperature medie settimanali:

$$\bar{t}_{ij} = \sum_{k=1}^{7} t_{ijk} \qquad i=1,\ldots,52 \;\; j=1,\ldots,12 \,, \tag{3.8}$$

dove t_{ijk} è la temperatura massima registrata il giorno k della settimana i dell'anno j. Sono stati calcolati i valori della temperatura media settimanale dell'anno tipo, come media dei valori ottenuti nei 12 anni:

$$\bar{t}_{i} = \frac{1}{12} \sum_{j=1}^{12} \bar{t}_{ij} \qquad i=1,\ldots,52 \,. \tag{3.9}$$

Rappresentando in un unico grafico i valori degli indici settimanali e della temperatura media settimanale dell'anno tipo si ottiene quanto illustrato in Figura 3.13.

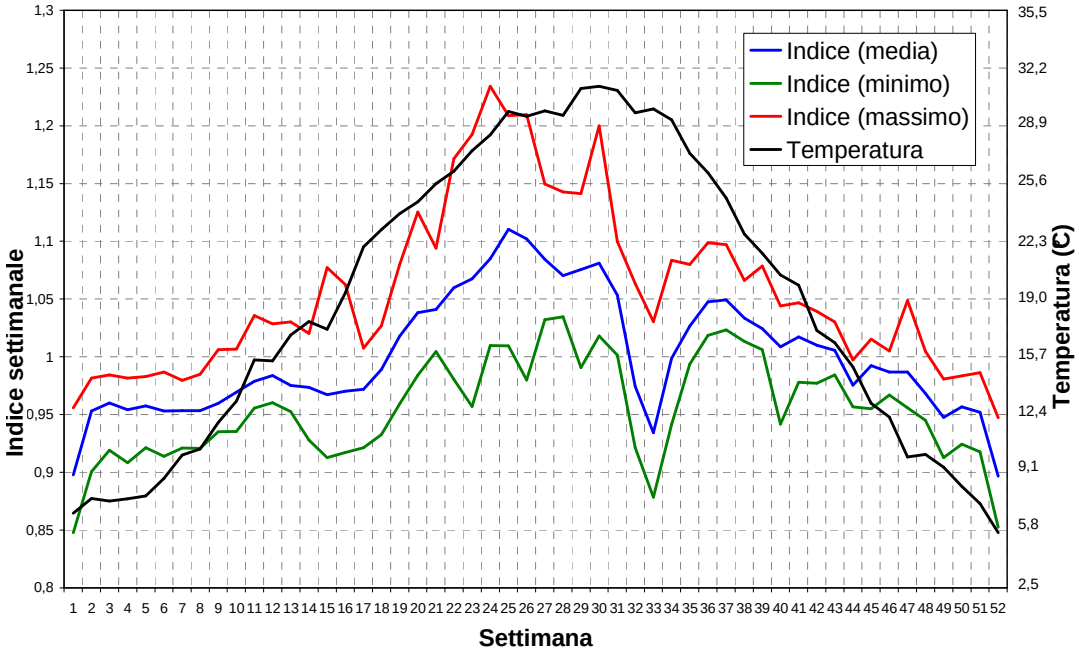


Figura 3.13 - *Indici settimanali medi e confronto con temperatura settimanale media.*

Si nota come la temperatura media sia correlata con il consumo idrico, specialmente nelle settimane centrali (16 – 44) corrispondenti al periodo che va da fine aprile a fine ottobre, ad eccezione del periodo a cavallo di ferragosto (settimana 33).
Il legame esistente tra temperatura giornaliera e consumo idrico è evidenziato dal grafico in Figura 3.14.

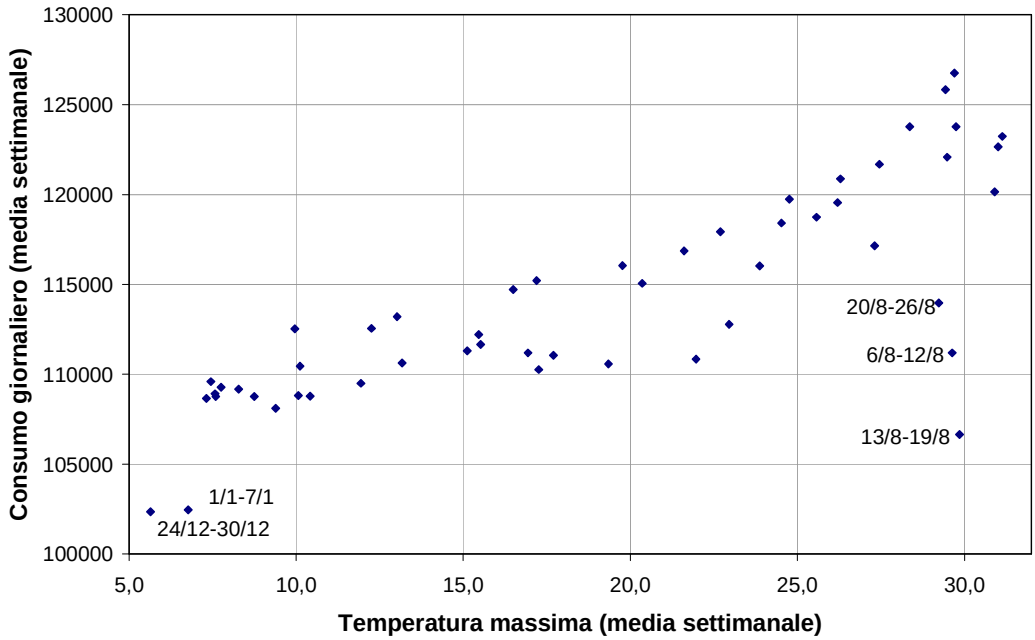


Figura 3.14 - *Temperatura massima giornaliera e consumo idrico (valori medi settimanali).*

3.3.1.2 Primo modello

Sulla base della media degli indici di consumo settimanali ricavati $\hat{r}_i$, è possibile realizzare un primo modello di stima dei consumi. Si può cercare di ricostruire il consumo della settimana *i* di un qualsiasi anno *j*, indicato con $\hat{c}_{ij}$, come prodotto dell'indice $\hat{r}_i$ per il consumo medio dell'anno *j*, indicato con $\hat{c}_j$. A rigore di logica quest'ultimo valore non è noto, ma è possibile dedurne una stima, ad esempio mediante estrapolazione dal consumo storico (vedi grafico in Figura 3.11). In formule:

$$\hat{c}_{ij} = \hat{r}_i \hat{c}_j \qquad i=1,\ldots,52 \;\; j=1,\ldots,12 . \tag{3.10}$$

Per valutare l'errore di stima compiuto si è proceduto calcolando le quantità $\hat{c}_{ij}$ prima sui dati del *training set* e poi su quelli del *validation set*, usando come valore $\hat{c}_j$ il vero valore $\bar{c}_j$ osservato, ipotizzando quindi di non commettere errori nella stima del consumo medio annuo. I valori $\hat{c}_{ij}$ ottenuti sono stati confrontati con i valori osservati c_{ij}. Per ogni valore è stato calcolato l'errore relativo:

$$\varepsilon_{ij} = \frac{\hat{c}_{ij} - c_{ij}}{c_{ij}} \qquad i=1,\ldots,52 \;\; j=1,\ldots,12 . \tag{3.11}$$

Nella Tabella 3.1 viene riportata la distribuzione di frequenze dell'errore relativo per le stime prodotte sul *training set*.

Tabella 3.1 - *Distribuzione dell'errore relativo nel primo modello di stima (training set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	3	3	0%		
-10% -9%	3	6	1%		
-9 % -8%	5	11	2%		
-8% -7%	5	16	3%	6%	12%
-7% -6%	6	22	4%		
-6% -5%	13	35	6%		
-5% -4%	15	50	8%		
-4% -3%	27	77	12%		
-3% -2%	49	126	20%		
-2% -1%	60	186	30%		
-1% 0%	113	299	48%	89%	75%
0% 1%	99	398	64%		
1% 2%	89	487	78%		
2% 3%	58	545	87%		
3% 4%	28	573	92%		
4% 5%	19	592	95%		
5% 6&	15	607	97%		
6% 7%	8	615	99%		
7% 8%	4	619	99%	5%	13%
8% 9%	2	621	100%		
9% 10%	0	621	100%		
>10%	3	624	100%		
				100%	100%

L'errore commesso secondo questo metodo di stima è, per il 75% dei casi, inferiore al 3% e, per l'89% dei casi, inferiore al 5%.

Nella Figura 3.15 è rappresentata la distribuzione di frequenze degli errori del primo modello di stima applicato al *training set*. La media di tale distribuzione è pari a 0,09 e lo scarto quadratico medio a 3,06.

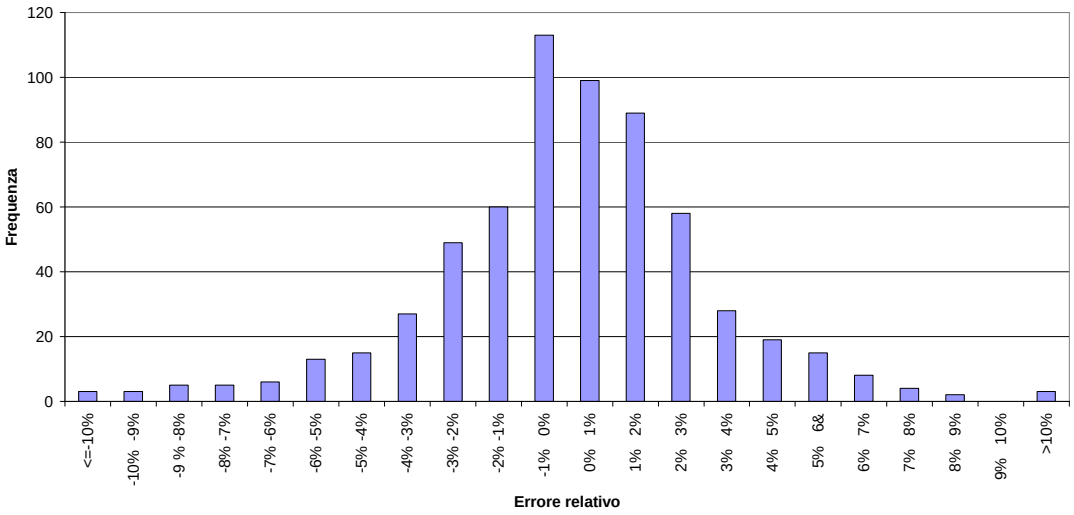


Figura 3.15 - *Distribuzione di frequenze dell'errore relativo nel primo modello di stima (training set).*

Il modello è stato applicato anche ai dati del *validation set* per verificare la bontà dell'adattamento a dati non utilizzati per la sua costruzione. In Tabella 3.3 è presente la distribuzione delle frequenze dell'errore relativo: il 91% dei casi registra un errore di stima inferiore al 5% e il 72% inferiore al 3%.

Tabella 3.2 - *Distribuzione dell'errore relativo nel primo modello di stima (validation set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	0	0	0%		
-10% -9%	0	0	0%		
-9 % -8%	1	1	1%		
-8% -7%	0	1	1%	5%	13%
-7% -6%	2	3	3%		
-6% -5%	2	5	5%		
-5% -4%	1	6	6%		
-4% -3%	8	14	13%		
-3% -2%	8	22	21%		
-2% -1%	16	38	37%		
-1% 0%	13	51	49%	91%	72%
0% 1%	12	63	61%		
1% 2%	17	80	77%		
2% 3%	9	89	86%		
3% 4%	8	97	93%		
4% 5%	3	100	96%		
5% 6&	2	102	98%		
6% 7%	1	103	99%		
7% 8%	0	103	99%	4%	14%
8% 9%	1	104	100%		
9% 10%	0	104	100%		
>10%	0	104	100%		
				100%	100%

Nella Figura 3.15 è rappresentata la distribuzione di frequenze degli errori del primo modello di stima applicato al *validation set*. La media di tale distribuzione è pari a 0,06 è lo scarto quadratico medio a 2,90.

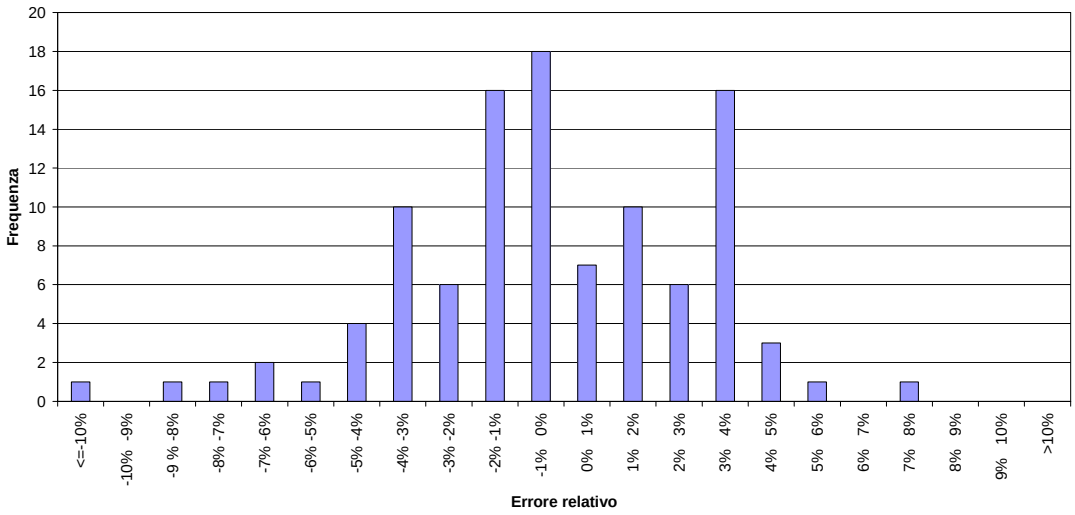


Figura 3.16 - *Distribuzione di frequenze dell'errore relativo nel primo modello di stima (validation set).*

Nel grafico di Figura 3.17 sono riportati gli errori relativi, riportando sulle ascisse la settimana sulle ordinate l'errore relativo per i due anni del *validation set.*

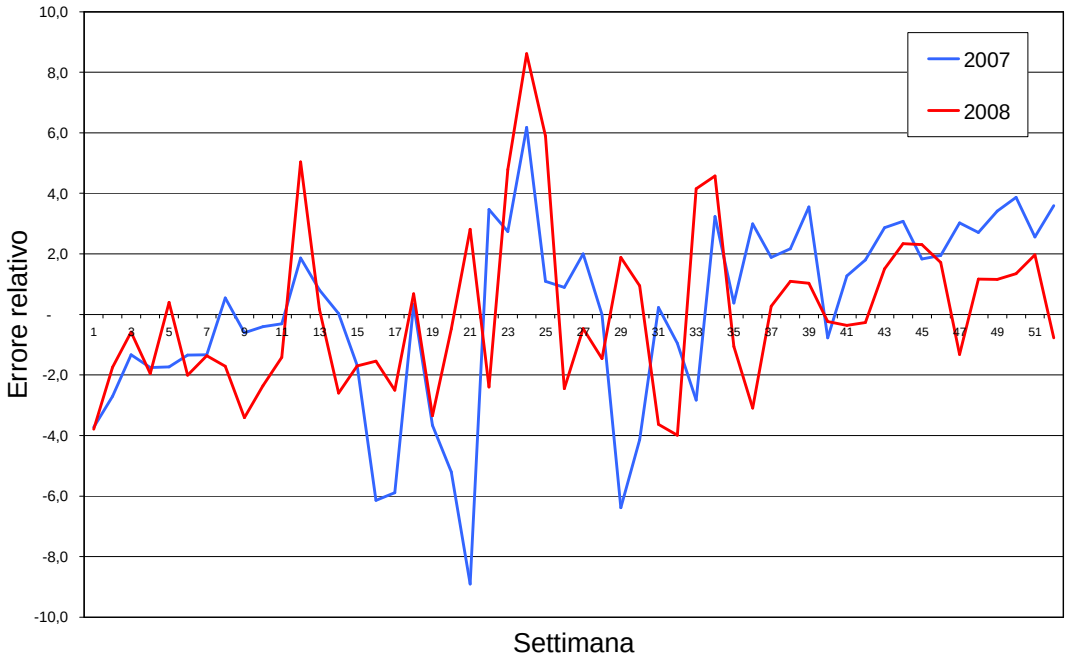


Figura 3.17 - *Errore relativo nel primo modello di stima, per settimana e anno (validation set).*

Dal grafico possiamo trarre alcune considerazioni:

- nelle settimane centrali di tutti gli anni osservati si nota un aumento dell'errore relativo, indicativamente tra la 14ª e la 39ª settimana: sono i mesi in cui i consumi sono più elevati e pertanto l'errore è ancora più rilevante;
- si evidenzia un limite del modello stimato: è presente una correlazione tra l'errore di stima in una certa settimana e l'errore di stima della settimana seguente;
- se consideriamo la media annua dell'errore relativo troviamo un valore prossimo allo zero: tale valore nasconde però una situazione con errori di stima elevati in alcuni mesi compensati da errori di segno opposto nei mesi seguenti.

3.3.1.3 Secondo modello

Come previsto dal modello di Davanzo, si è proceduto a realizzare un secondo modello di stima introducendo la variabile temperatura allo scopo di ridurre la variabilità residua.

Dai dati di partenza sono state reperite le temperature medie settimanali t_{ij}, come media aritmetica delle temperature massime dei 7 giorni componenti la settimana:

$$t_{ij} = \frac{1}{7}\sum_{k=1}^{7} t_{ijk} \qquad i=1,\ldots,52 \;\; j=1,\ldots,12\,, \tag{3.12}$$

dove t_{ijk} è la temperatura massima del giorno *k* della settimana *i* dell'anno *j*.

Sono state calcolate quindi le temperature tipo per ogni settimana, indicate con t_i, come media di quelle rilevate nei 12 anni osservati:

$$t_i = \frac{1}{12}\sum_{j=1}^{12} t_{ij} \qquad i=1,\ldots,52 \tag{3.13}$$

e, per ciascuna settimana di ogni anno osservato, sono state calcolate le differenze tra la temperatura media osservata e la temperatura tipo:

$$\Delta t_{ij} = t_{ij} - t_i \qquad i=1,\ldots,52 \;\; j=1,\ldots,12\,. \tag{3.14}$$

Le differenze Δt_{ij} sono utilizzate per il calcolo di un nuovo modello di stima, nel quale il nuovo coefficiente di consumo viene espresso come segue:

$$\hat{\hat{r}}_{ij} = \hat{r}_i(1+\Delta t_{ij}\alpha_i) \qquad i=1,\ldots,52 \;\; j=1,\ldots,12\,, \tag{3.15}$$

dove:

$\hat{r}_i$ sono i coefficienti di consumo ottenuti in precedenza mediante il primo modello;

α_i sono parametri che rappresentano la relazione tra consumo e temperatura presente nella settimana *i*.

La stima dei parametri α_i viene determinata minimizzando l'errore assoluto:

$$\hat{\alpha}_i : \min_{\alpha_i} \sum_{j=1}^{12} |E_{ij}| = \min_{\alpha_i} \sum_{j=1}^{12} |r_{ij} - r_i \cdot (1 + \Delta t_{ij} \cdot \alpha_i)| \qquad i=1,\ldots,52 \tag{3.16}$$

Per il calcolo di $\hat{\alpha}_i$ si è proceduto definendo la funzione:

$$f_i(\alpha_i) = \sum_{j=1}^{12} |r_{ij} - r_i \cdot (1 + \Delta t_{ij} \cdot \alpha_i)| \tag{3.17}$$

e calcolando il suo minimo rispetto ad α_i. Essendo presenti punti angolosi nei quali la funzione non è derivabile, si è proceduto utilizzando un metodo grafico. A titolo esemplificativo si riporta quanto realizzato per la settimana *i*=1. Sostituendo nell'espressione il valore α_1 con *x*, la funziona da minimizzare risulta la seguente:

$$\begin{aligned} f_1(x) = {} & |0{,}8978 - 0{,}89777(1 + 0{,}2 \cdot x)| + \\ & + |0{,}86164 - 0{,}89777(1 - 0{,}71429 \cdot x)| + \\ & + |0{,}95575 - 0{,}89777(1 + 1{,}04286 \cdot x)| + \\ & + |0{,}87192 - 0{,}89777(1 + 2{,}17143 \cdot x)| + \\ & + |0{,}84779 - 0{,}89777(1 + 1{,}34286 \cdot x)| + \\ & + |0{,}90723 - 0{,}89777(1 + 0{,}62857 \cdot x)| + \\ & + |0{,}90345 - 0{,}89777(1 + 0{,}3 \cdot x)| + \\ & + |0{,}93761 - 0{,}89777(1 + 0{,}72857 \cdot x)| + \\ & + |0{,}87516 - 0{,}89777(1 + 0{,}05714 \cdot x)| + \\ & + |0{,}90648 - 0{,}89777(1 - 1{,}92857 \cdot x)| + \\ & + |0{,}89071 - 0{,}89777(1 + 2{,}75714 \cdot x)| + \\ & + |0{,}91768 - 0{,}89777(1 - 1{,}78571 \cdot x)| \end{aligned} \tag{3.18}$$

Il valore *x* che minimizza $f_1(x)$ è stato ricavato dal grafico di Figura 3.18**,** nel punto di ascissa $x = -0{,}0050 = \hat{\alpha}_1$.

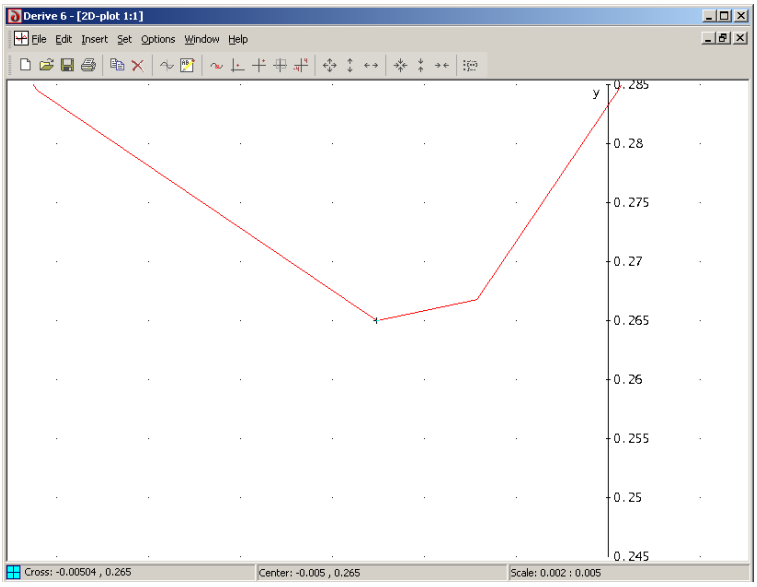


Figura 3.18 - *Determinazione grafica del minimo del parametro* α_1.

I valori di $\hat{\alpha}_i$ ricavati per i=1,…,52 sono rappresentati nel grafico di Figura 3.19.

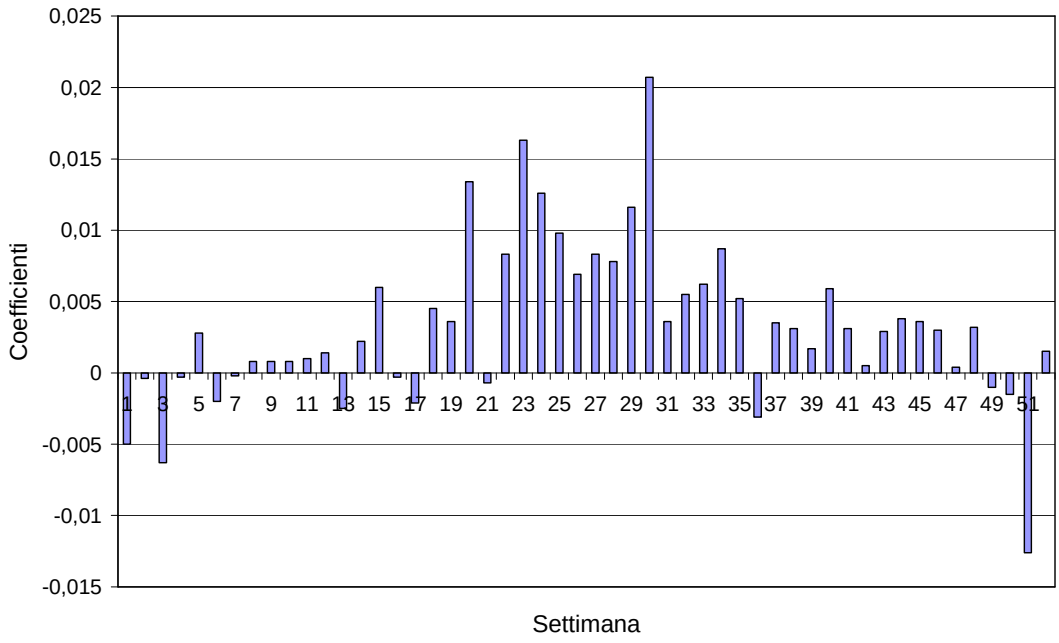


Figura 3.19 - *Valore ottimale dei coefficienti* α_i *al variare della settimana.*

Fatta eccezione per alcune settimane a cavallo dell'anno, il coefficiente assume generalmente valori rilevanti positivi a partire dalla 18ª settimana (inizio maggio) fino alla 46ª (metà novembre). In questo periodo, il fattore temperatura influenza i consumi al di là di quanto previsto dalla settimana tipo. Ad esempio, nella settimana 25ª il valore ottimale del coefficiente è circa 0,01; questo indica che per ogni grado di temperatura in più (o in meno) rispetto alla settimana tipo ci si può attendere un aumento (o una diminuzione) del consumo dell'1%.

Confrontando quanto ora ottenuto (Figura 3.19) con quanto ottenuto da Davanzo nell'analisi originale del 1982 (Figura 2.3) si nota che il periodo in cui il coefficiente è significativamente positivo si è spostato in avanti, passando dal periodo Aprile-Settembre al periodo Maggio-Novembre.

Sulla base dei valori $\hat{\alpha}_i$ ricavati si è proceduto al calcolo degli indici di consumo secondo il nuovo modello:

$$\hat{\hat{r}}_{ij} = \hat{r}_i(1+\Delta t_{ij}\hat{\alpha}_i) \qquad i=1,\ldots,52 \;\; j=1,\ldots,12 \tag{3.19}$$

e al conseguente modello di stima dei consumi:

$$\hat{\hat{c}}_{ij} = \hat{\hat{r}}_{ij} \cdot \hat{c}_i \qquad i=1,\ldots,52 \;\; j=1,\ldots,12\,. \tag{3.20}$$

Ancora una volta si è valutato l'errore di stima compiuto calcolando le quantità $\hat{\hat{c}}_{ij}$ utilizzando dapprima i dati del *training set* e quindi quelli del *validation set*, usando come valore $\hat{c}_i$ il vero valore $\overline{c}_i$ osservato, nell'ipotesi quindi di non commettere errori nella stima del consumo medio annuo.

I valori $\hat{\hat{c}}_{ij}$ ottenuti sono stati confrontati con i valori osservati c_{ij} ed è stato calcolato l'errore relativo:

$$\varepsilon_{ij} = \frac{\hat{\hat{c}}_{ij} - c_{ij}}{c_{ij}} \qquad i=1,\ldots,52 \;\; j=1,\ldots,12\,. \tag{3.21}$$

Nella Tabella 3.3 viene riportata la distribuzione di frequenze dell'errore relativo per le stime prodotte sul *training set*.

Tabella 3.3 - *Distribuzione dell'errore relativo nel secondo modello di stima (training set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	0	0	0%		
-10% -9%	1	1	0%		
-9 % -8%	4	5	1%		
-8% -7%	1	6	1%	3%	7%
-7% -6%	5	11	2%		
-6% -5%	8	19	3%		
-5% -4%	10	29	5%		
-4% -3%	16	45	7%		
-3% -2%	45	90	14%		
-2% -1%	69	159	25%		
-1% 0%	121	280	45%	92%	80%
0% 1%	135	415	67%		
1% 2%	80	495	79%		
2% 3%	50	545	87%		
3% 4%	32	577	92%		
4% 5%	19	596	96%		
5% 6%	17	613	98%		
6% 7%	3	616	99%		
7% 8%	3	619	99%	4%	13%
8% 9%	1	620	99%		
9% 10%	1	621	100%		
>10%	3	624	100%		
				100%	100%

Le stime prodotte utilizzando questo secondo modello sui dati dal *training set* registrano un errore che nel 79% dei casi è inferiore al 3% e nel 93% dei casi inferiore al 5%.
Nella Figura 3.20 è rappresentata la distribuzione di frequenze degli errori la cui media è pari a 0,31 e il cui scarto quadratico medio è pari a 2,63.

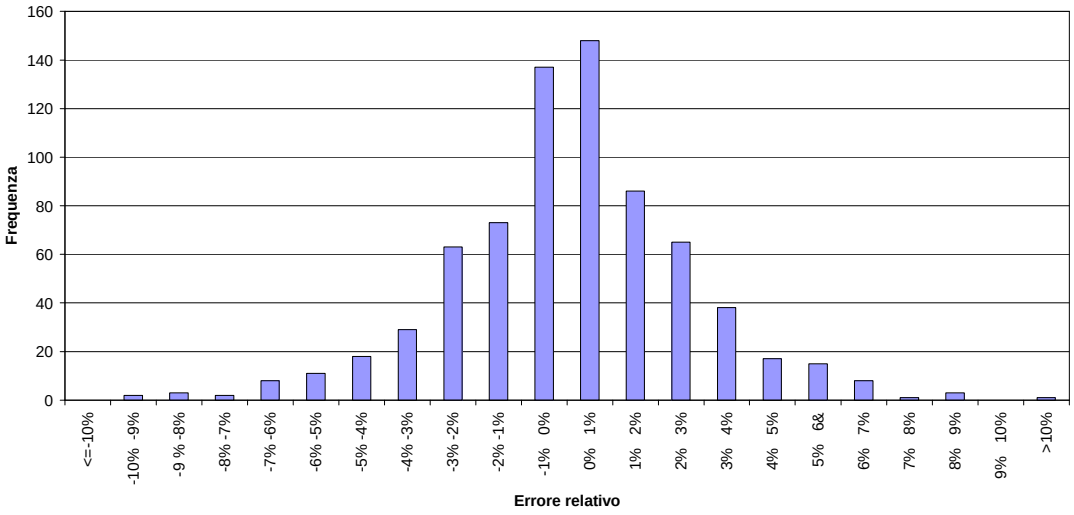


Figura 3.20 - *Distribuzione di frequenze dell'errore relativo nel secondo modello di stima (training set).*

Il secondo modello è stato applicato anche ai dati del *validation set* per verificare la bontà l'adattamento a dati non utilizzati per la sua costruzione. In Tabella 3.4 è presente la distribuzione delle frequenze dell'errore relativo: il 92% dei casi regista un errore di stima inferiore al 5% e il 61% inferiore al 3%.

Tabella 3.4 - *Distribuzione dell'errore relativo nel primo modello di stima (validation set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	1	1	1%		
-10% -9%	0	1	1%		
-9 % -8%	1	2	2%		
-8% -7%	1	3	3%	6%	19%
-7% -6%	2	5	5%		
-6% -5%	1	6	6%		
-5% -4%	4	10	10%		
-4% -3%	10	20	19%		
-3% -2%	6	26	25%		
-2% -1%	16	42	40%		
-1% 0%	18	60	58%	92%	61%
0% 1%	7	67	64%		
1% 2%	10	77	74%		
2% 3%	6	83	80%		
3% 4%	16	99	95%		
4% 5%	3	102	98%		
5% 6%	1	103	99%		
6% 7%	0	103	99%		
7% 8%	1	104	100%	2%	20%
8% 9%	0	104	100%		
9% 10%	0	104	100%		
>10%	0	104	100%		
				100%	100%

Nella Figura 3.21 è rappresentata la distribuzione di frequenze degli errori relativi al secondo modello di stima applicato al *validation set*. La media di tale distribuzione è pari a -0,27 è lo scarto quadratico medio a 3,21.

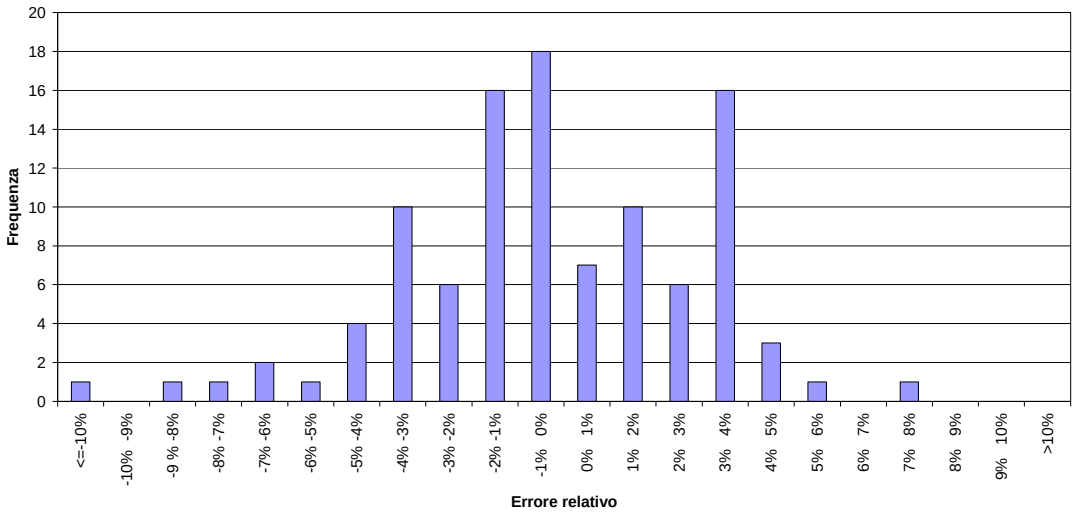


Figura 3.21 - *Distribuzione di frequenze dell'errore relativo nel secondo modello di stima (training set).*

Nel grafico di Figura 3.22 sono riportati gli errori relativi, riportando sulle ascisse la settimana e sulle ordinate l'errore relativo per i due anni del *validation set.*

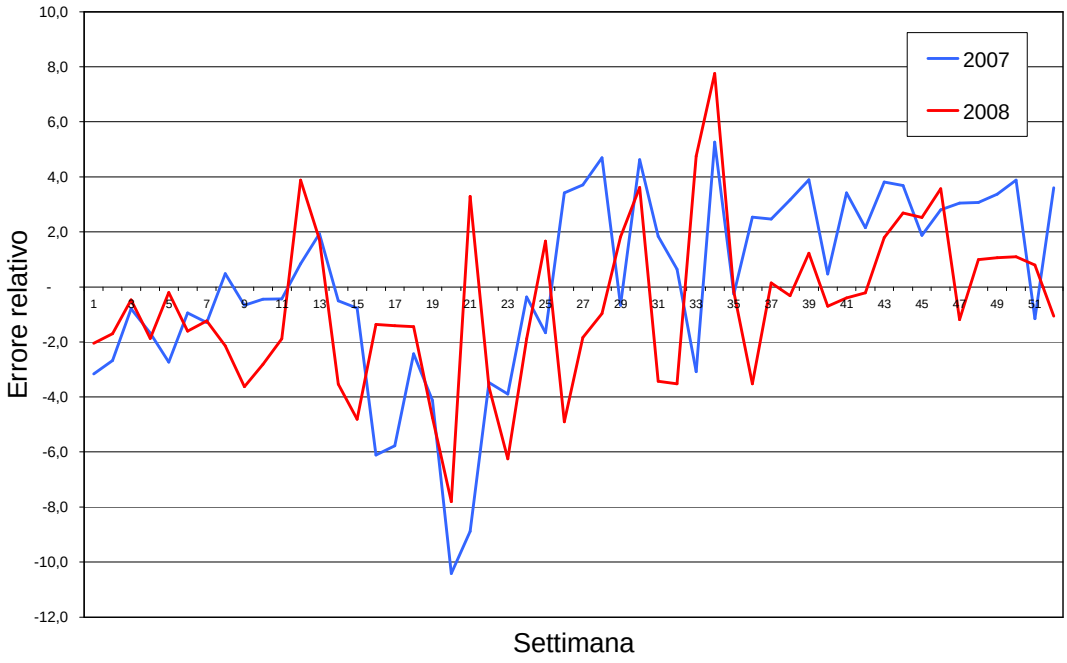


Figura 3.22 - *Errore relativo nel primo modello di stima, per settimana e anno (validation set).*

Anche per il secondo modello valgono le stesse considerazioni espresse per il primo: nei mesi centrali dell'anno gli errori sono più elevati, è presente correlazione tra

l'errore di una settimana e quello della settimana successiva, la media dell'errore su base annua è bassa ma si compone di errori settimanali anche elevati.

3.3.2 Previsione dei consumi giornalieri

Ottenuta una stima del volume dei consumi di una determinata settimana è necessario capire come questo si distribuisca nei sette giorni che la compongono.
Partendo dai dati del *training set* si considerano i consumi idrici c_{ijk}, dove k è l'indice che rappresenta il giorno della settimana (1=lunedì,…, 7=domenica), i indica la settimana all'interno dell'anno (i=1,…,52) e j l'anno (j=1,…,12).
Vengono calcolate le medie dei consumi giornalieri per tutte le settimane nel periodo di osservazione:

$$\bar{c}_{ij} = \frac{1}{7}\sum_{k=1}^{7} c_{ijk} \qquad i=1,\ldots,52 \quad j=1,\ldots,12\ . \tag{3.22}$$

Si calcolano i rapporti:

$$s_{ijk} = \frac{c_{ijk}}{\bar{c}_{ij}} \qquad k=1,\ldots,7 \quad i=1,\ldots,52 \quad j=1,\ldots,12\ . \tag{3.23}$$

Fissata la settimana i dell'anno j, il valore s_{ijk} vale 1 se nel giorno k il consumo è stato pari al consumo medio di quella settimana, sarà maggiore di 1 in caso di consumo superiore e minore di 1 in caso di consumo inferiore.
Si calcolano quindi i pesi medi dei 7 giorni settimanali:

$$\bar{s}_k = \frac{1}{12 \cdot 52}\sum_{j=1}^{12}\sum_{i=1}^{52} s_{ijk} \qquad k=1,\ldots,7. \tag{3.24}$$

La quantità $\bar{s}_k$ rappresenta il "peso" medio del giorno k nella determinazione del consumo settimanale.
Vengono quindi calcolati i campi di variazione dei pesi giornalieri:

$$s^{\min}{}_k = \min_{j=1}^{12}\left(\min_{i=1}^{52} s_{ijk}\right) \tag{3.25}$$

$$s^{\max}{}_k = \max_{j=1}^{12}\left(\max_{i=1}^{52} s_{ijk}\right) \tag{3.26}$$

Nel grafico di Figura 3.23 sono riportati i valori medi ottenuti e i loro campi di variazione.

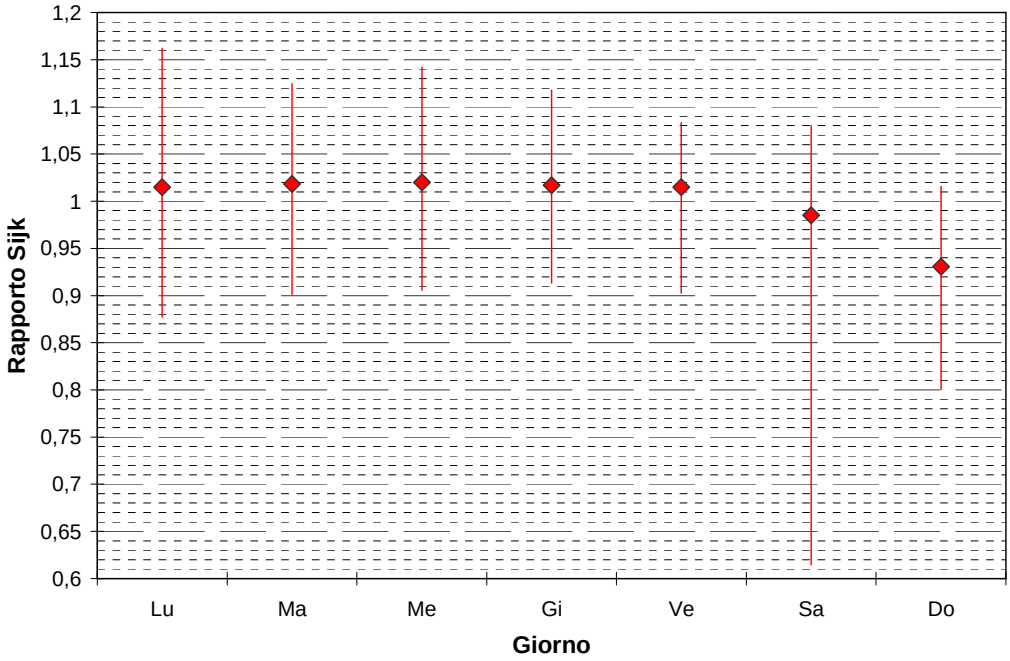


Figura 3.23 - *Pesi giornalieri e loro campi di variazione nei giorni della settimana (training set).*

Dall'osservazione del grafico emerge quanto segue:

- i consumi dei giorni da lunedì al venerdì sono superiori alla media settimanale di una percentuale che va dall'1% al 2%;
- i consumi del sabato si attestano sotto la media settimanale dell'1,5%;
- i consumi della domenica sono inferiori del 7% rispetto alla media settimanale;

Tali risultati confermano sostanzialmente quanto ottenuto da Davanzo nel 1982 (vedi Figura 2.4), evidenziando nel confronto con questo una certa diminuzione dei consumi nella giornata di sabato a favore di un corrispondente aumento di domenica.

3.3.2.1 Primo modello giornaliero

È stata effettuata la previsione giornaliera partendo dalle stime realizzate dal primo modello settimanale, ripartendo i consumi settimanali previsti sulla base dei pesi giornalieri. Le previsioni giornaliere ottenute sono state confrontate con il consumo reale, considerando prima le osservazioni del *training set* e poi quelle del *validation set*.

Nella Tabella 3.5 viene riportata la distribuzione di frequenze dell'errore relativo per le stime prodotte sul *training set*. Le previsioni realizzare presentano un errore che nel 62% dei casi è inferiore al 3% e nel 82% dei casi inferiore al 5%.

Tabella 3.5 - *Distribuzione dell'errore relativo nel primo modello giornaliero. (training set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	54	54	1%		
-10% -9%	30	84	2%		
-9 % -8%	45	129	3%		
-8% -7%	63	192	4%	9%	18%
-7% -6%	70	262	6%		
-6% -5%	120	382	9%		
-5% -4%	167	549	13%		
-4% -3%	248	797	18%		
-3% -2%	329	1126	26%		
-2% -1%	473	1599	37%		
-1% 0%	571	2170	50%	82%	62%
0% 1%	502	2672	61%		
1% 2%	456	3128	72%		
2% 3%	382	3510	80%		
3% 4%	251	3761	86%		
4% 5%	182	3943	90%		
5% 6%	116	4059	93%		
6% 7%	89	4148	95%		
7% 8%	77	4225	97%	10%	20%
8% 9%	40	4265	98%		
9% 10%	28	4293	98%		
>10%	75	4368	100%		
				100%	100%

Nella Figura 3.24 è rappresentata la distribuzione di frequenze degli errori la cui media è pari a 0,13 e il cui scarto quadratico medio è pari a 4,09.

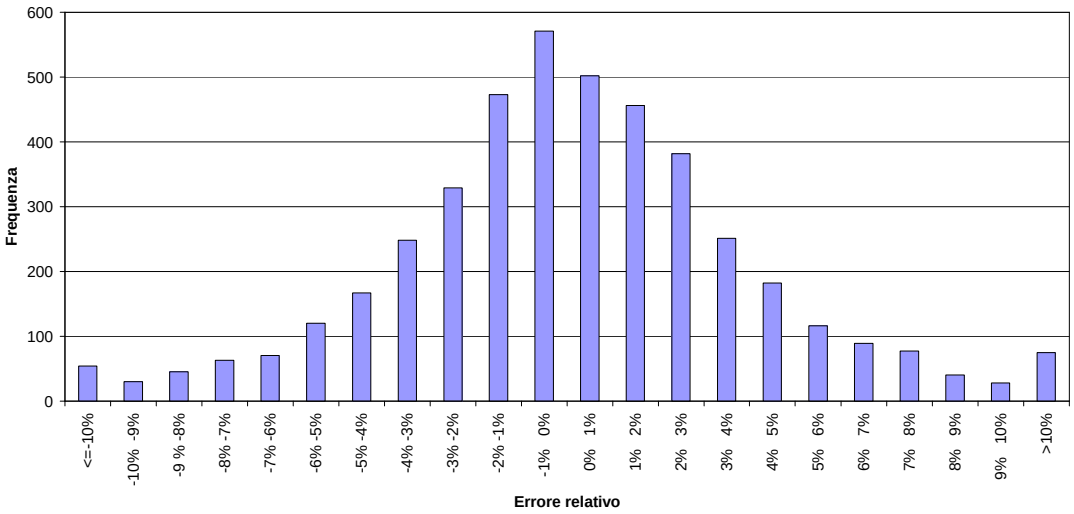


Figura 3.24 - *Distribuzione di frequenze dell'errore relativo nel primo modello giornaliero (training set).*

Nella Tabella 3.6 viene riportata la distribuzione di frequenze dell'errore relativo per le stime prodotte sul *validation set*. Le previsioni realizzare presentano un errore che nel 61% dei casi è inferiore al 3% e nel 82% dei casi inferiore al 5%.

Tabella 3.6 - *Distribuzione dell'errore relativo nel primo modello giornaliero. (validation set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	5	5	1%		
-10% -9%	5	10	1%		
-9 % -8%	4	14	2%		
-8% -7%	14	28	4%	9%	20%
-7% -6%	18	46	6%		
-6% -5%	18	64	9%		
-5% -4%	21	85	12%		
-4% -3%	58	143	20%		
-3% -2%	60	203	28%		
-2% -1%	75	278	38%		
-1% 0%	78	356	49%	82%	61%
0% 1%	82	438	60%		
1% 2%	75	513	70%		
2% 3%	71	584	80%		
3% 4%	45	629	86%		
4% 5%	34	663	91%		
5% 6%	22	685	94%		
6% 7%	12	697	96%		
7% 8%	7	704	97%	9%	20%
8% 9%	5	709	97%		
9% 10%	5	714	98%		
>10%	14	728	100%		
				100%	100%

Nella Figura 3.25 è rappresentata la distribuzione di frequenze degli errori la cui media è pari a 0,15 e il cui scarto quadratico medio è pari a 4,06.

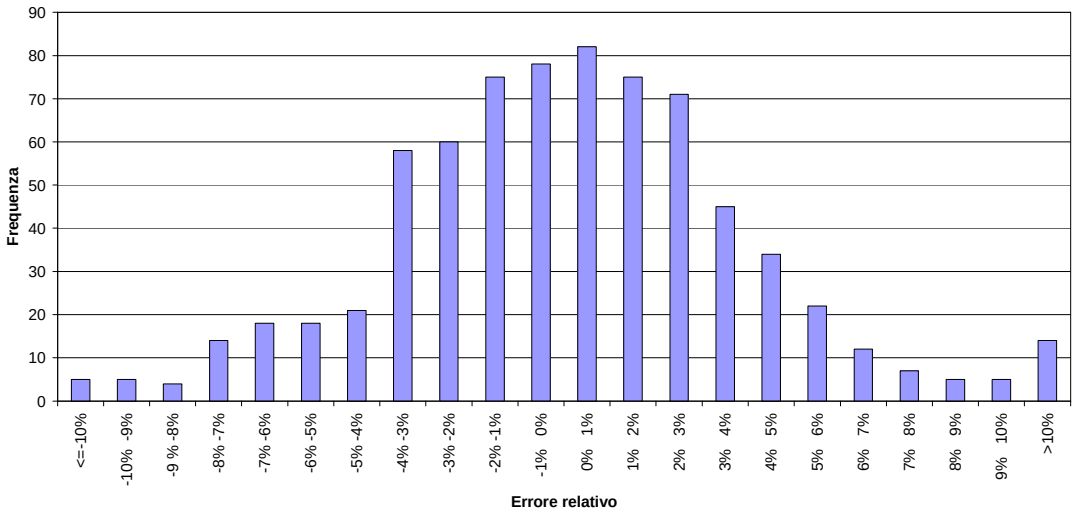


Figura 3.25 - *Distribuzione di frequenze dell'errore relativo nel primo modello giornaliero (validation set).*

3.3.2.2 Secondo modello giornaliero

Analogamente a quanto fatto per il primo modello giornaliero, si è proceduto ripartendo i consumi settimanali previsti dal secondo modello settimanale sulla base dei pesi giornalieri. Le previsioni giornaliere ottenute sono state confrontate con il

consumo reale, considerando prima le osservazioni del *training set* e poi quelle del *validation set*.

Nella Tabella 3.7 viene riportata la distribuzione di frequenze dell'errore relativo per le stime prodotte sul *training set*. Le previsioni realizzare presentano un errore che nel 65% dei casi è inferiore al 3% e nel 84% dei casi inferiore al 5%.

Tabella 3.7 - *Distribuzione dell'errore relativo nel secondo modello giornaliero. (training set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	26	26	1%		
-10% -9%	20	46	1%		
-9 % -8%	35	81	2%		
-8% -7%	37	118	3%	7%	15%
-7% -6%	49	167	4%		
-6% -5%	126	293	7%		
-5% -4%	146	439	10%		
-4% -3%	234	673	15%		
-3% -2%	333	1006	23%		
-2% -1%	512	1518	35%		
-1% 0%	566	2084	48%	84%	65%
0% 1%	558	2642	60%		
1% 2%	508	3150	72%		
2% 3%	368	3518	81%		
3% 4%	254	3772	86%		
4% 5%	185	3957	91%		
5% 6%	114	4071	93%		
6% 7%	81	4152	95%		
7% 8%	64	4216	97%	9%	19%
8% 9%	46	4262	98%		
9% 10%	29	4291	98%		
>10%	77	4368	100%		
				100%	100%

Nella Figura 3.26 è rappresentata la distribuzione di frequenze degli errori la cui media è pari a 0,36 e il cui scarto quadratico medio è pari a 3,82.

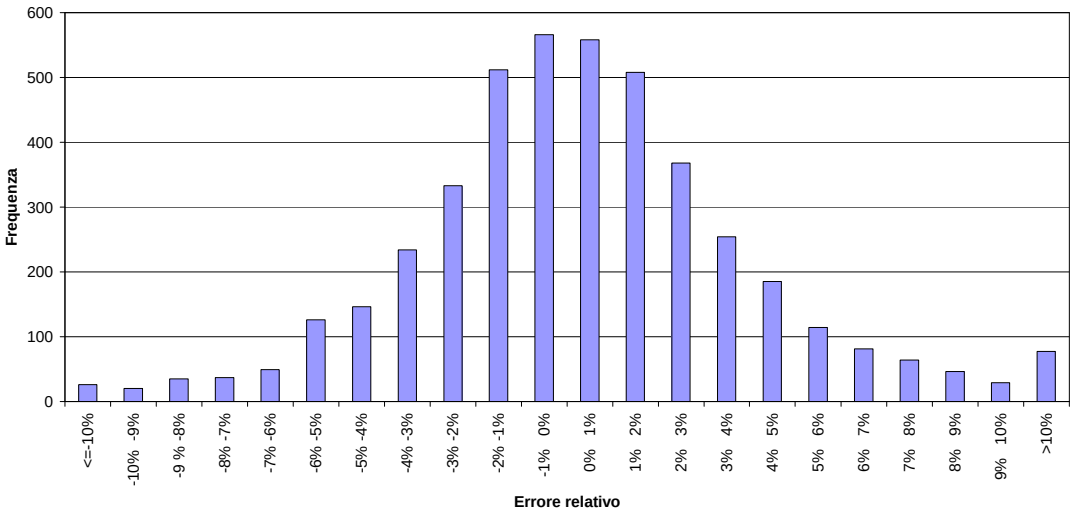


Figura 3.26 - *Distribuzione di frequenze dell'errore relativo nel secondo modello giornaliero (training set).*

Nella Tabella 3.8 viene riportata la distribuzione di frequenze dell'errore relativo per le stime prodotte sul *validation set*. Le previsioni realizzare presentano un errore che nel 56% dei casi è inferiore al 3% e nel 80% dei casi inferiore al 5%.

Tabella 3.8 - *Distribuzione dell'errore relativo nel secondo modello giornaliero. (validation set).*

Errore	Frequenza	Freq. Cum	Freq.Cum.Rel	Errore 5%	Errore 3%
<=-10%	12	12	2%		
-10% -9%	9	21	3%		
-9 % -8%	6	27	4%		
-8% -7%	14	41	6%	12%	23%
-7% -6%	17	58	8%		
-6% -5%	26	84	12%		
-5% -4%	31	115	16%		
-4% -3%	50	165	23%		
-3% -2%	74	239	33%		
-2% -1%	60	299	41%		
-1% 0%	74	373	51%	80%	56%
0% 1%	74	447	61%		
1% 2%	74	521	72%		
2% 3%	50	571	78%		
3% 4%	53	624	86%		
4% 5%	40	664	91%		
5% 6%	22	686	94%		
6% 7%	12	698	96%		
7% 8%	9	707	97%	9%	22%
8% 9%	8	715	98%		
9% 10%	3	718	99%		
>10%	10	728	100%		
				100%	100%

Nella Figura 3.27 è rappresentata la distribuzione di frequenze degli errori la cui media è pari a -0,18 e il cui scarto quadratico medio è pari a 4,26.

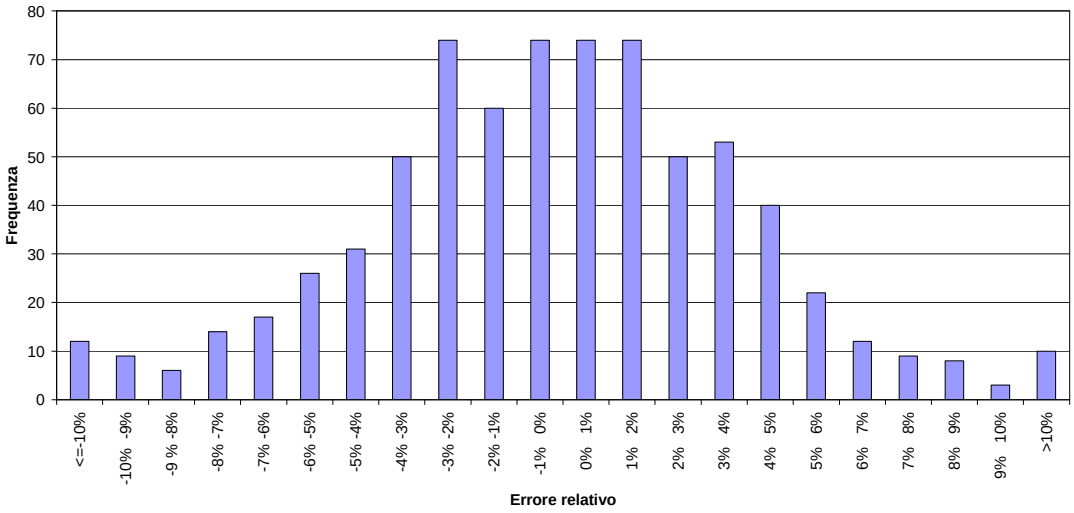


Figura 3.27 - *Distribuzione di frequenze dell'errore relativo nel secondo modello giornaliero (validation set).*

3.4 Modelli basati su regressione lineare

La prima famiglia di nuovi modelli realizzati si basa sulla regressione lineare. La loro costruzione, oltre a rappresentare l'approccio classico alla modellazione, realizza una sorta di *benchmark* per valutare la bontà dei modelli che utilizzano altre metodologie. Dopo aver brevemente esposto la teoria dei modelli di regressione lineare, si presenteranno alcuni modelli calcolati sui dati a disposizione. In ogni modello la variabile indipendente sarà il volume di acqua erogato nel giorno di riferimento (`erogato0`) mentre i predittori presi in considerazione saranno inizialmente i volumi erogati nei 7 giorni precedenti, poi verrà inserita come predittore la variabile temperatura del giorno di riferimento e infine anche il coefficiente settimanale.

3.4.1 Richiami teorici

Il modello di regressione lineare multipla è esprimibile come:

$$y = b_0 + b_1 x_1 + ... + b_j x_j + ... + b_k x_k + \varepsilon \tag{3.27}$$

dove y (vettore di n elementi) è la variabile dipendente, $x_1, x_2, ..., x_k$ (tutti vettori di n elementi) sono le variabili esplicative (linearmente indipendenti), b_j sono i parametri o coefficienti di regressione (uno per ogni variabile indipendente) e b_0 è l'intercetta. Il valore ε è la componente accidentale del modello; di solito si assume che abbia distribuzione normale di media nulla e varianza costante.

Il modello viene chiamato di regressione lineare perché è lineare nei parametri, cioè i coefficienti b_j sono semplici moltiplicatori delle variabili indipendenti, non è invece necessario che il modello di regressione lineare sia lineare nelle variabili indipendenti.

I coefficienti b_j (o coefficienti di regressione parziale) misurano il cambiamento nel valore (medio) di y al variare di un'unità nella x_j, mantenendo costanti tutte le altre variabili.

I coefficienti di regressione sono legati all'unità di misura delle variabili. Ciò significa che la grandezza di un particolare coefficiente non è un buon indicatore della sua importanza. I confronti tra i coefficienti di diverse variabili possono essere fatti attraverso i coefficienti standardizzati:

$$\beta_j = b_j \frac{s_{x_j}}{s_y}, \tag{3.28}$$

dove s_{x_j} e s_y indicano le deviazioni standard di x_j e di y. L'intercetta è nulla per definizione.

Per la stima dei parametri tramite minimi quadrati, l'unica assunzione necessaria riguarda l'indipendenza lineare tra le k variabili esplicative. La matrice X delle variabili esplicative deve avere rango pieno, e quindi $X'X$ deve avere determinante non nullo, altrimenti non si ottengono le stime. Se una variabile esplicativa dovesse essere linearmente dipendente dalle altre (collinearità perfetta), significherebbe che le informazioni in essa contenute sono in realtà già presenti nel *dataset* attraverso le altre variabili. Pertanto, l'eliminazione di tale variabile non comporterebbe perdita di informazione e porterebbe alla definizione di una nuova matrice X, questa volta di rango pieno pari a $k-1$ e quindi alla stima di $k-1$ parametri.

Le stime ai minimi quadrati dei parametri del modello di regressione lineare multipla non richiedono ipotesi distributive ma solo l'assenza di collinearità perfetta. Tali stime sono non distorte (cioè il valore atteso delle stime coincide con il valore vero dei parametri) ed hanno varianza minima nella classe degli stimatori lineari e non distorti (teorema di Gauss-Markov).

Una volta stimato un modello è anche possibile calcolarne la bontà dell'adattamento tramite l'indice R^2, compreso tra 0 e 1:

$$R^2 = \frac{somma\ dei\ quadrati\ di\ regressione}{somma\ dei\ quadrati\ totale} = \frac{SSR}{SST}. \tag{3.29}$$

Poiché R^2 cresce all'aumentare del numero di variabili inserite nel modello, se si vuole utilizzare questo criterio per la scelta tra modelli, risulta più adatto l'indice corretto $\overline{R}^2$:

$$\overline{R}^2 = 1 - \frac{n-1}{n-k-1}\left(1-R^2\right) \tag{3.30}$$

Per fare inferenza statistica sul modello di regressione è necessario fare ipotesi sulla distribuzione della componente d'errore ε:

$$\varepsilon_i \sim N(0,\sigma^2), \qquad i = 1,2,...,n. \tag{3.31}$$

In tal caso le stime dei minimi quadrati coincidono con quelle di massima verosimiglianza e quindi godranno delle stesse proprietà che consentono di condurre test d'ipotesi sui parametri (ad esempio per verificare la significatività delle variabili) o sulla significatività globale del modello.

Per verificare se una particolare variabile x_j abbia influenza sulla variabile dipendente y si considera l'ipotesi $H_0 : b_j = 0$ (variabile non significativa) e la statistica test su cui ci si basa è:

$$\hat{t}_j = \frac{\hat{b}_j}{\sqrt{\Sigma_{jj}}} \tag{3.32}$$

con $\Sigma = \sigma^2 (X'X)^{-1}$ e $\hat{b} = (X'X)^{-1} X'y$, che sotto H_0 ha la distribuzione t di Student con $n - p$ gradi di libertà.

Per valutare la bontà complessiva del modello si considera l'ipotesi nulla $H_0 : R^2 = 0$, che si basa sulla statistica test:

$$F_c = \frac{SSR}{SSE} \frac{n-k}{k-1}, \tag{3.33}$$

che sotto H_0 si distribuisce come una F di Snedecor con $k-1$ e $n-k$ gradi di libertà.

3.4.2 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti (lm.1)

In questo modello lineare vengono considerate come variabili predittive solo i 7 giorni precedenti. Il modello è statisticamente significativo con α osservato inferiore a $2{,}2 \cdot 10^{-16}$, così come le variabili `erogato1`, `erogato7` e in misura minore `erogato3` che rappresentano rispettivamente l'erogato del giorno precedente, quello di 7 giorni prima, cioè lo stesso giorno della settimana precedente, e quello di 3 giorni prima.

Il valore di R^2 è pari a 0,77, lo scarto quadratico medio della distribuzione degli errori relativi calcolati sul *training set* e sul *validation set* è pari rispettivamente a 4,11 e 4,23.

Nella Figura 3.28 è stato realizzato il plot dei valori stimati verso i valori reali, per il *dataset* di *training*, a sinistra, e per il *dataset* di *validation*, a destra.

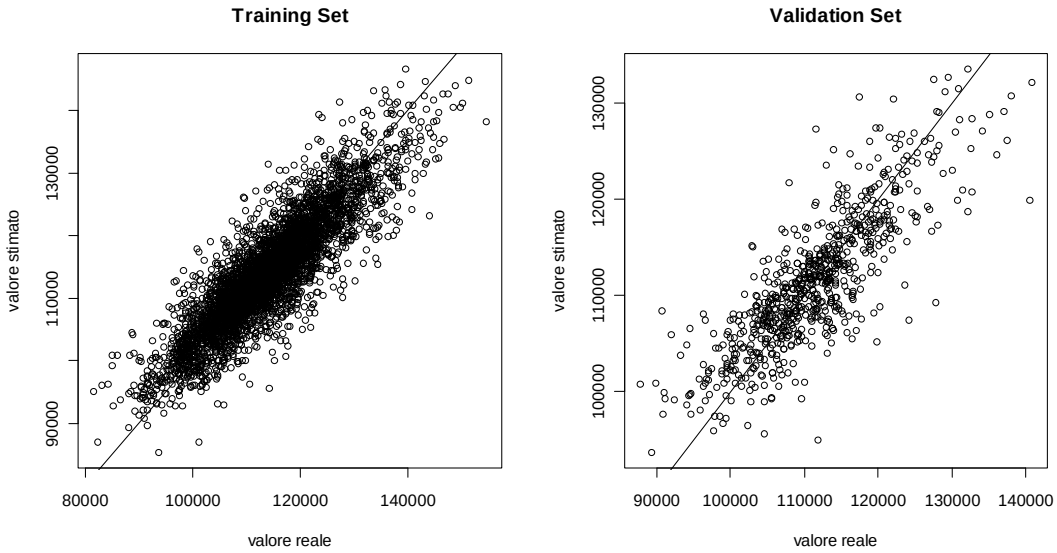


Figura 3.28 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello lm.1.*

La retta tracciata, bisettrice del primo e terzo quadrante, rappresenta la linea ideale nella quale valori stimati e valori reali coincidono.
Nella parte sinistra di Figura 3.29 è presente il plot dei residui verso i valori stimati calcolati sul *training set*, dal quale non si evidenziano particolari tendenze nella distribuzione degli errori. Nella parte destra è rappresentata la distribuzione di frequenze degli errori relativi calcolati sul *validation set* che presenta l'atteso andamento campanulare; il 58% degli errori è inferiore al 3% mentre il 78% è inferiore al 5%.

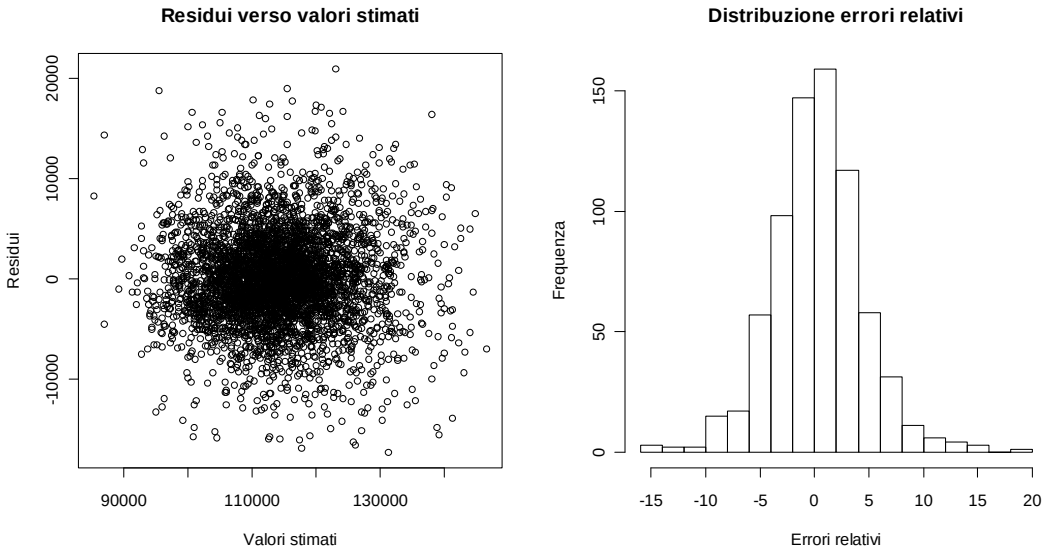


Figura 3.29 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello lm.1.*

3.4.3 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e la temperatura (lm.2)

In aggiunta al precedente modello lm.1 è stata inserita come predittore la variabile `temperatura` che registra appunto temperatura massima del giorno di stima. In un'ottica di previsione a ben vedere tale valore non è noto, ma è abbastanza agevole realizzarne una previsione. Sono statisticamente significativi gli stessi predittori che lo erano nel precedente modello lm.1, così come lo è il nuovo predittore `temperatura`. Il modello nel suo complesso è significativo con α osservato inferiore a $2{,}2 \cdot 10^{-16}$.
Il valore di R^2 è pari a 0,78, lo scarto quadratico medio della distribuzione degli errori relativi calcolati sul *training set* e sul *validation set* è pari rispettivamente a 4,07 e 4,16, leggermente inferiori rispetto agli analoghi del modello lm.1.

In Figura 3.30 è presente il plot dei valori reali verso i valori stimati dal modello, a sinistra per i dati del *training set*, a destra per il *validation set.*

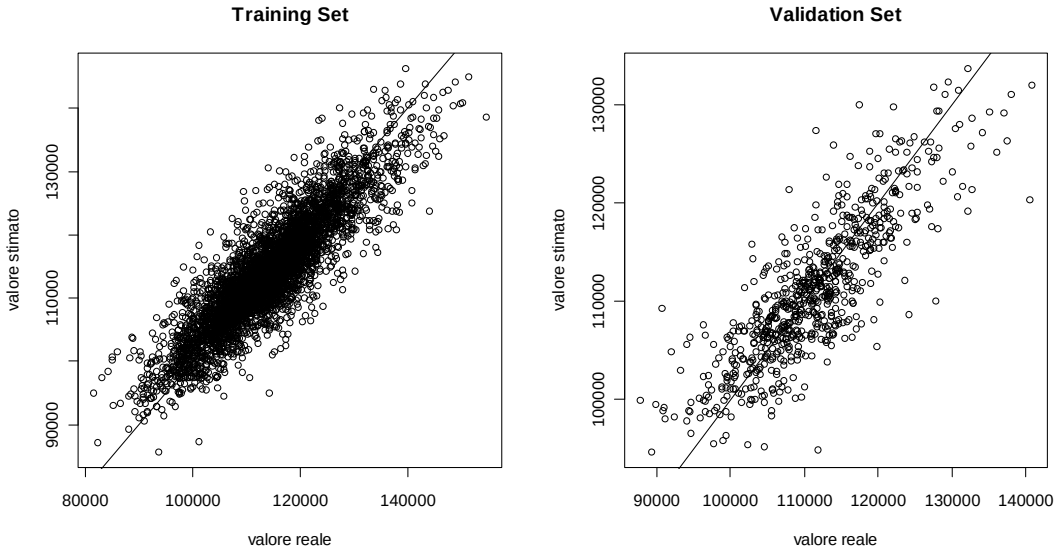


Figura 3.30 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello lm.2.*

Nella Figura 3.31 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata.

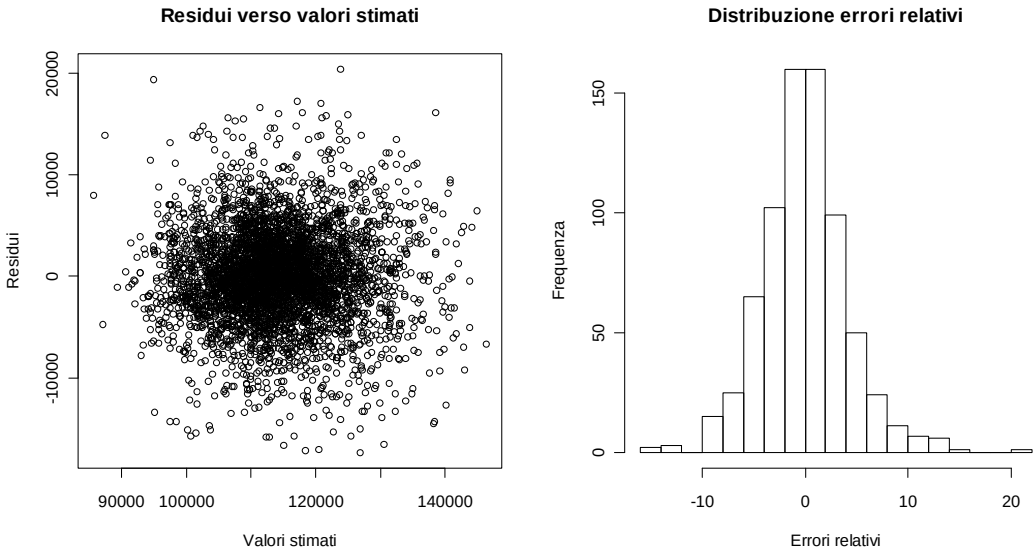


Figura 3.31 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello lm.2.*

Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set* che ne mostra l'andamento campanulare; il 60% degli errori è inferiore

al 3% mentre il 79% è inferiore al 5%, con un sensibile aumento rispetto al modello lm.1.

3.4.4 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale (lm.3)

In questo modello, oltre alle variabili considerate in precedenza, viene inserito come predittore la variabile `rapp_settimana`, che rappresenta il coefficiente settimanale. È stato necessario utilizzare per il calcolo dei parametri il *training set* `df.train_r`, che esclude le osservazioni del primo anno per le quali la variabile non è definita.
Il modello risulta statisticamente significativo con α osservato inferiore a $2{,}2 \cdot 10^{-16}$; lo sono anche le stesse variabili che lo erano nel precedente modello lm.2, mentre è statisticamente non significativo il nuovo predittore `rapp_settimana`.
Il valore di R^2 è pari a 0,77, di poco inferiore al precedente modello lm.2. Lo scarto quadratico medio della distribuzione degli errori relativi calcolati sul *training set* e sul *validation set* è pari rispettivamente a 4,02 e 4,16, in linea rispetto agli analoghi del modello lm.2.
In Figura 3.32 è presente il plot dei valori reali verso i valori stimati dal modello, a sinistra per i dati del *training set*, a destra per il *validation set.*

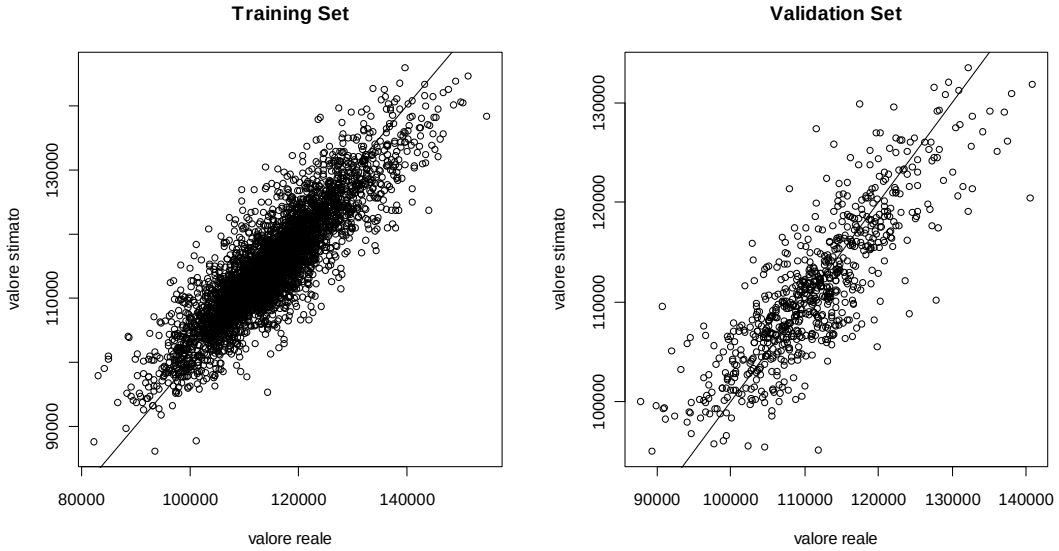


Figura 3.32 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello lm.3.*

Nella Figura 3.33 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il

validation set che ne mostra l'andamento campanulare; il 62% degli errori è inferiore al 3% mentre il 79% è inferiore al 5%, in linea con quanto rilevato per il modello lm.2.

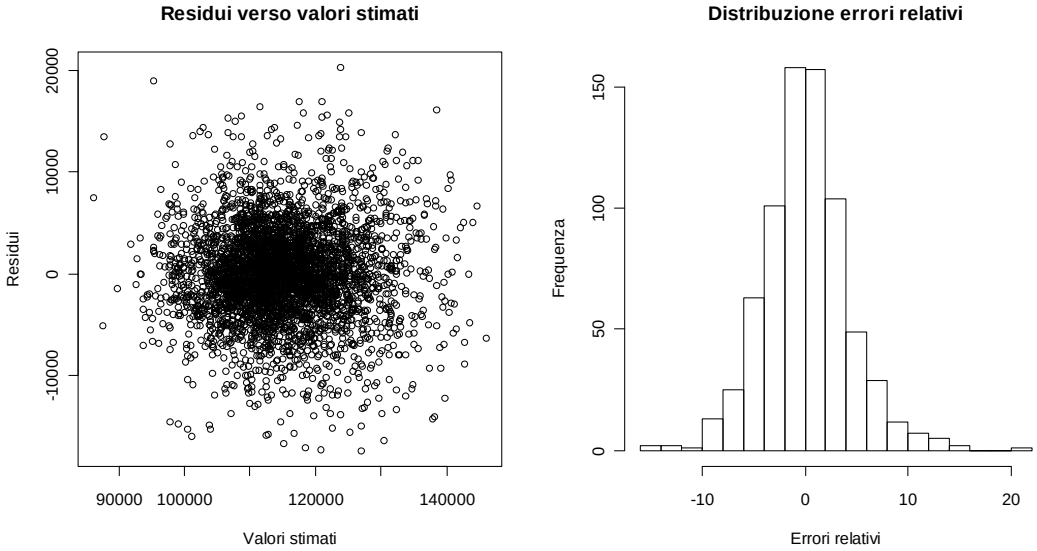


Figura 3.33 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello lm.3.*

3.5 Modelli basati su reti neurali

La seconda famiglia di modelli realizzati si basa sulle reti neurali, per i quali è stato necessario utilizzare per le variabili che rappresentano l'erogato le rispettive versioni scalate. La variabile risposta è quindi `erogato0_n`.

Vengono calcolati tre modelli (nn.1, nn.2 e nn.3) con gli stessi predittori utilizzati per i modelli basati su regressione lineare, allo scopo di verificare la bontà dell'adattamento rispetto al *benchmark*. Nel modello nn.4 viene valutato l'effetto della sola variabile `rapp_settimana` e nel modello nn.5 viene aggiunta la variabile `temperatura`, escludendo per entrambi il valore dell'erogato nei giorni precedenti. Il modello nn.6 è equivalente al modello nn.5 ma considera la versione standardizzata della variabile `temperatura`. Il modello nn.7 è equivalente al modello nn.1, avente come predittori l'erogato dei 7 giorni precedenti, ma con valori normalizzati anziché scalati nell'intervallo [0,1]. Il modello nn.8 utilizza come predittori l'erogato dei 7 giorni precedenti e la temperatura. I modelli nn.9 e nn.10 presentano un numero di neuroni nello strato nascosto pari a 18, a differenza di tutti gli altri che ne hanno solo 4. L'idea è di verificare se, al crescere del numero di neuroni dello strato nascosto e quindi dei parametri del modello, si realizzi il fenomeno dell'*overfitting*, ovvero un miglioramento dell'adattamento nel *training set* e un peggioramento nel *validation*

set. Nel modello nn.11 viene inserita la variabile `festivo` e nel modello nn.12 anche la variabile `giorno_settimana`; quest'ultimo presenta il miglior adattamento, valutato sul *validation set*, tra quelli considerati.
Tutti i modelli descritti finora hanno l'obiettivo di prevedere il consumo di un singolo giorno a partire dai dati disponibili fino a quel momento. Al fine di realizzare il modello settimanale nn.S che fornisca una previsione del consumo per l'intera settimana, sono stati realizzati dapprima i modelli nn.S0, nn.S1,…,nn.S6 che, utilizzando gli stessi predittori del modello nn.12, consentono la stima del consumo dei 7 giorni e successivamente sono stati combinati opportunamente i loro risultati.

3.5.1 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti (nn.1)

In questo primo modello basato su reti neurali vengono considerate come variabili predittive solo i 7 giorni precedenti. Questo modello è preferibile rispetto all'analogo basato su regressione lineare (lm.1): la deviazione standard sul *validation set* passa da 4,23 a 3,69.
Nella Figura 3.34 è stato realizzato il plot dei valori stimati verso i valori reali, per il *dataset* di *training*, a sinistra, e per il *dataset* di *validation*, a destra. La retta tracciata, bisettrice del primo e terzo quadrante, rappresenta la linea ideale nella quale valori stimati e valori reali coincidono.

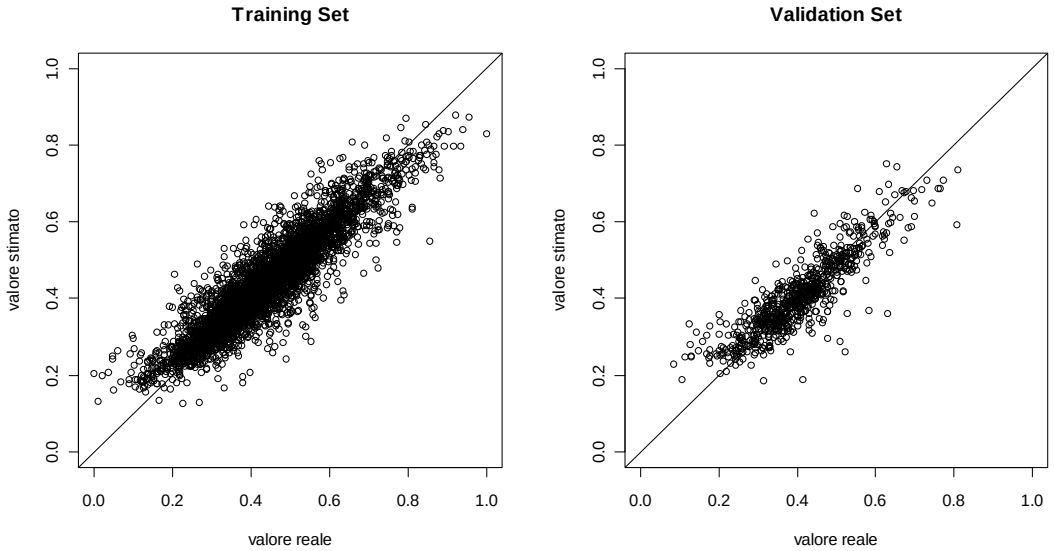


Figura 3.34 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.1.*

Nella parte sinistra di Figura 3.35 è presente il plot dei residui verso i valori stimati calcolati sul *training set*, dal quale non si evidenziano particolari tendenze nella

distribuzione degli errori. Nella parte destra è rappresentata la distribuzione di frequenze degli errori relativi calcolati sul *validation set* che presenta l'atteso andamento campanulare; il 68% (58% nel modello lm.1) degli errori è inferiore al 3% mentre l'87% (78% nel modello lm.1) è inferiore al 5%.

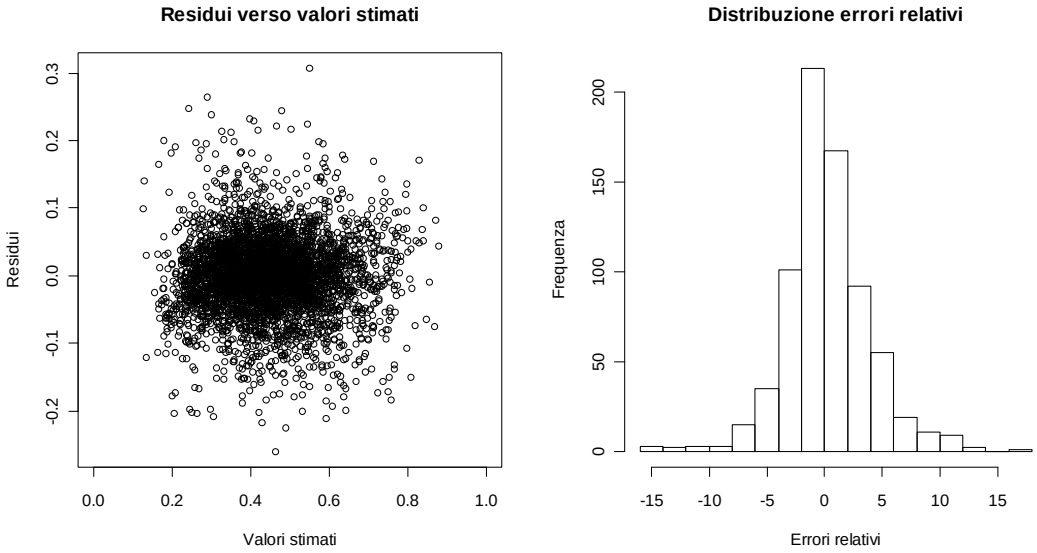


Figura 3.35 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.1.*

3.5.2 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e la temperatura (nn.2)

In aggiunta al modello precedente è stata inserita tra i predittori la variabile che registra la temperatura massima del giorno. Anche questo modello, rispetto all'equivalente basato su regressione lineare (lm.2) mostra una diminuzione della deviazione standard degli errori relativi da 4,16 a 3,67. Rispetto al modello precedente (nn.1), il cui valore era invece pari a 3,69, l'introduzione della temperatura non sembra portare ad un significativo miglioramento.

In Figura 3.36 è presente il plot dei valori reali verso i valori stimati dal modello, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

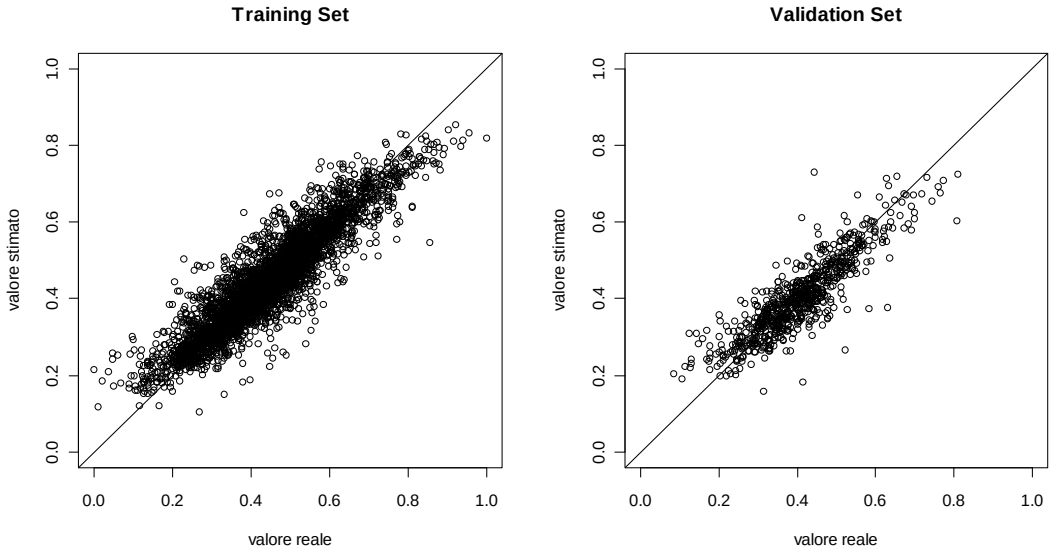


Figura 3.36 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.2.*

Nella Figura 3.37 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set* che ne mostra l'andamento campanulare; il 67% degli errori è inferiore al 3% mentre l'87% valori prossimi a quanto ottenuto nel precedente modello nn.1.

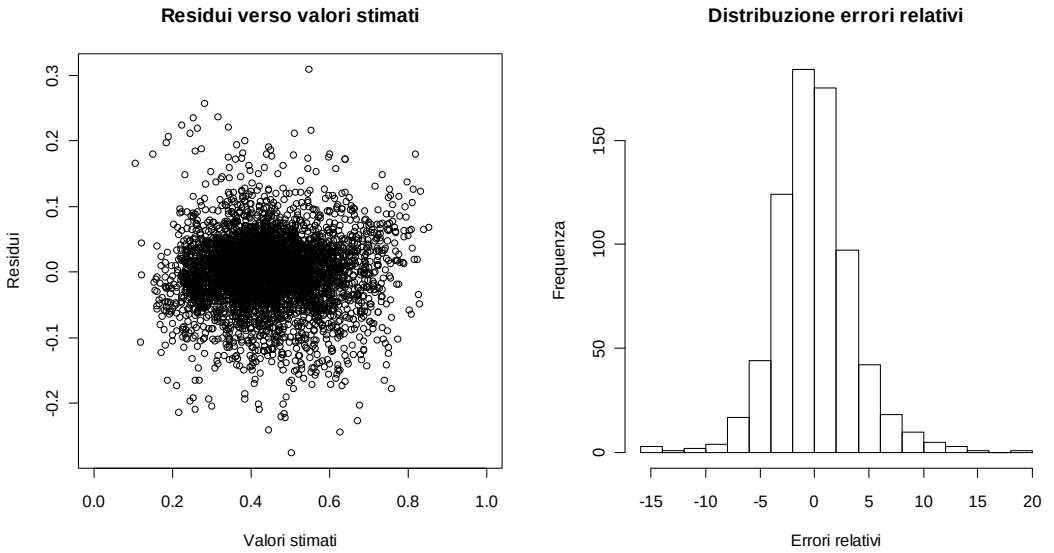


Figura 3.37 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.2.*

3.5.3 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale (nn.3)

Rispetto al precedente modello nn.2 viene inserito come predittore anche il coefficiente settimanale, rappresentato dalla variabile `rapp_settimana`. Lo scarto quadratico medio nella distribuzione degli errori relativi, valutato sul *validation set*, è pari a 3,72 contro 3,67 del precedente modello nn.2, ad indicare la sostanziale equivalenza dei due modelli.

Nel confronto con il modello corrispondente basato sulla regressione lineare (lm.3) la riduzione è invece sensibile: da 4,16 a 3,72; ciò conferma la superiorità dei modelli basati su reti neuronali.

In Figura 3.38 è presente il plot dei valori reali verso i valori stimati dal modello, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

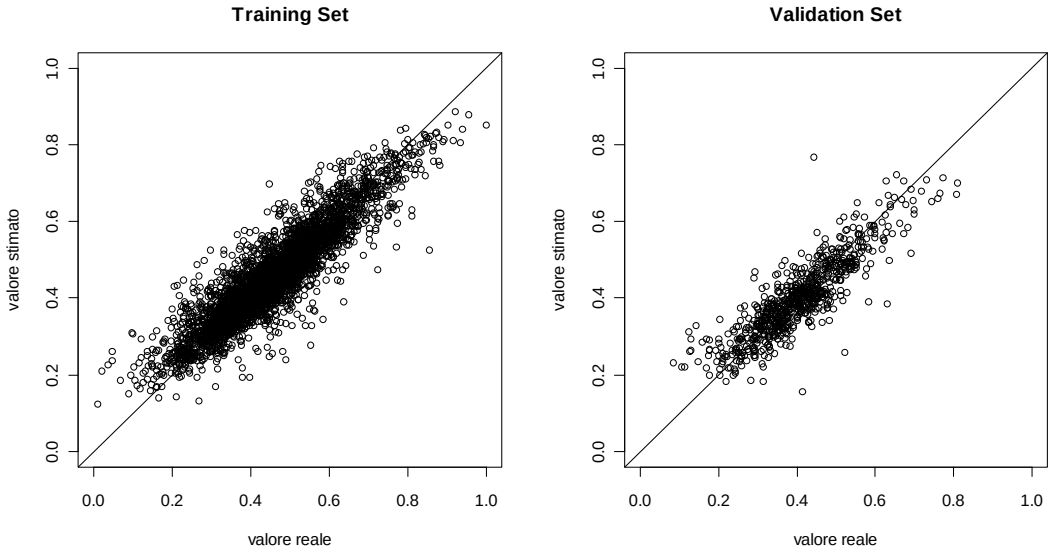


Figura 3.38 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.3.*

Nella Figura 3.37 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set* che ne mostra l'andamento campanulare; il 65% degli errori è inferiore al 3% mentre l'86% degli errori è inferiore al 5%, in linea con quanto ottenuto dal precedente modello nn.3. Si evince quindi che il predittore `rapp_settimana` non contribuisce in modo originale alla spiegazione della variabile risposta.

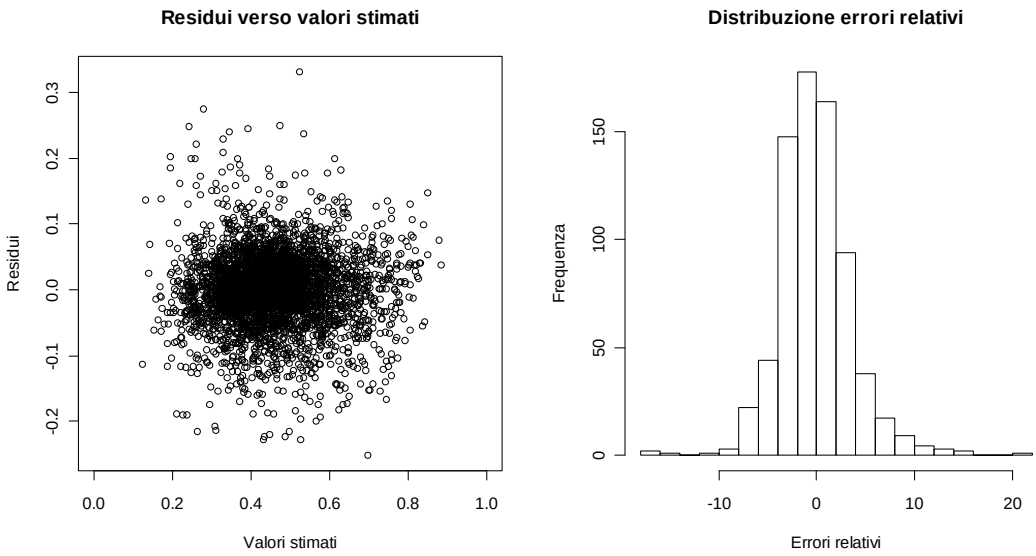


Figura 3.39 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.3.*

3.5.4 Modello avente come predittore il solo coefficiente settimanale (nn.4)

Visto che dal precedente modello nn.3 la variabile `rapp_settimana` non ha fornito un contributo all'aumento della bontà delle stime del consumo, si è voluto valutarne la capacità predittiva ponendola come unico predittore del modello.

L'idea è di fare una sorta di parallelo con il primo modello giornaliero di Acegas-Aps (§3.3.2.1): questi aveva prodotto uno scarto quadratico medio della distribuzione degli errori relativi, calcolata sui dati del *validation set*, pari a 4,06, molto migliore di quanto ottenuto dal presente modello pari a 6,25.

Anche nel confronto con il precedente modello nn.3 appare evidente il peggioramento delle stime: lo scarto quadratico medio della distribuzione degli errori relativi, calcolata sui dati del *validation set*, passa da 3,72 a 6,25.

Il peggioramento della capacità predittiva, rispetto ai modelli visti in precedenza, è evidente dai grafici di Figura 3.40 che riportano il plot dei valori reali verso i valori stimati dal modello, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale: è evidente la dispersione dei valori attorno alla linea retta.

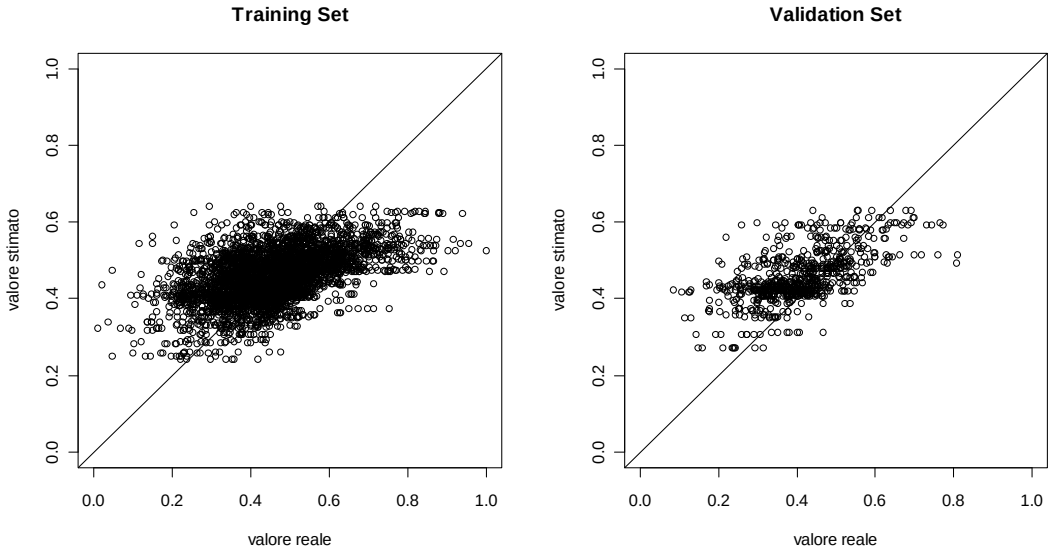


Figura 3.40 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.4.*

Nella Figura 3.41 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata.

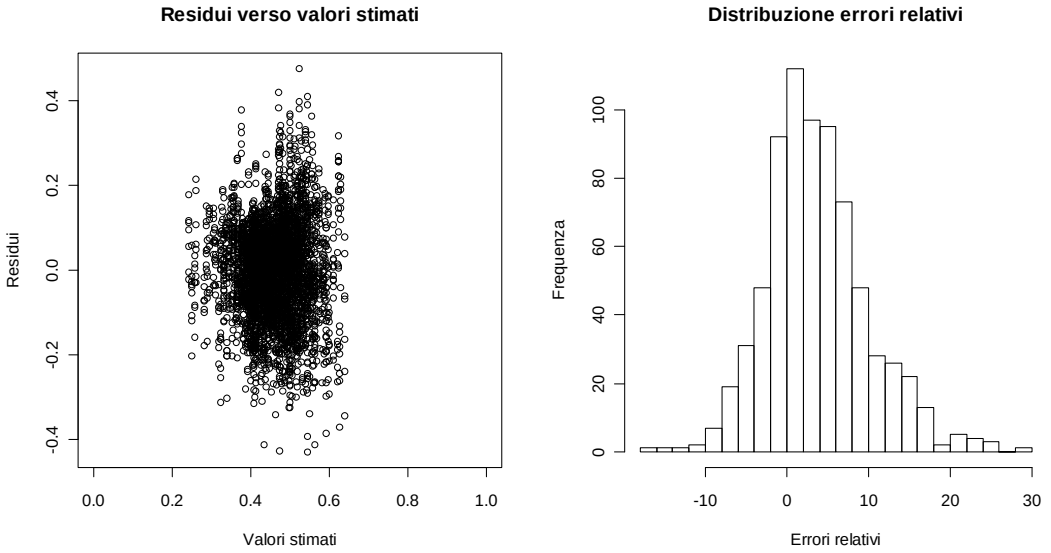


Figura 3.41 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.4.*

Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 40% degli errori è inferiore al 3% mentre il 57% degli errori è inferiore al 5%. Appare evidente la distorsione delle stime prodotte, testimoniata dal valor medio della distribuzione pari a 3,7: mediamente il modello fornisce una stima superiore del 3,7% rispetto al valore reale.

3.5.5 Modello avente come predittori la temperatura e il coefficiente settimanale (nn.5)

Rispetto al precedente modello nn.4 viene aggiunto il predittore `temperatura`. Il parallelo è, in questo caso, con il secondo modello giornaliero di Acegas-Aps (§3.3.2.2). Nel confronto si nota che lo scarto quadratico medio della distribuzione degli errori relativi valutati sul *validation set* passa da 4,26 per Acegas-Aps a 6,02 per il presente modello dimostrandone l'inferiorità.
Invece, nel confronto con il precedente modello nn.4, la variazione dello scarto quadratico medio della distribuzione degli errori relativi valutati sul *validation set* è modesta: l'introduzione della temperatura lo fa scendere da un valore 6,25 a 6,02.
I grafici di Figura 3.42 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale e confermano la scarsa capacità predittiva del modello.

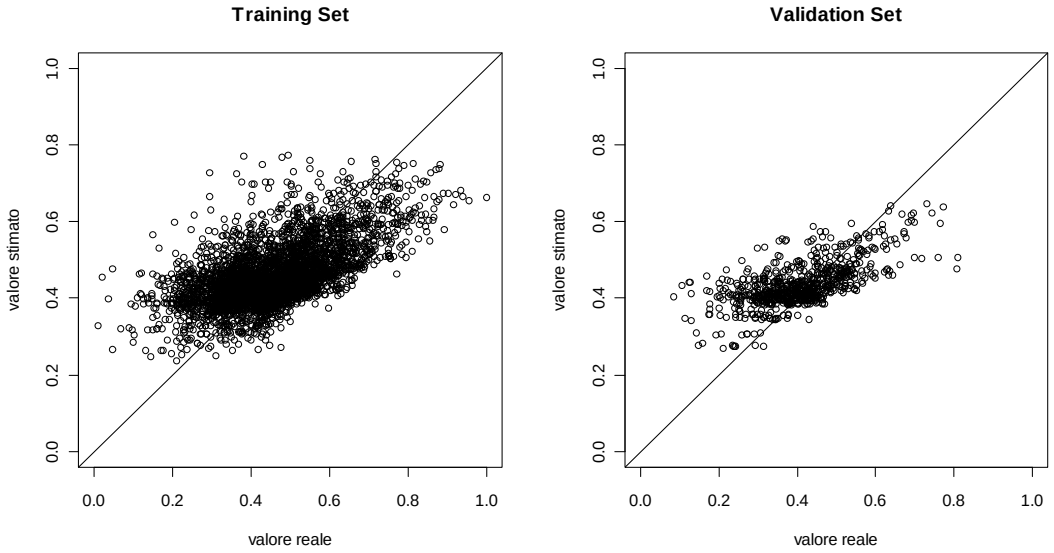


Figura 3.42 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.5.*

Nella Figura 3.43 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 44% degli errori è inferiore al 3% mentre il 64% degli errori è inferiore al 5%. Anche in questo caso è presente una distorsione delle stime prodotte pari a 2,3: mediamente il modello fornisce una stima superiore del 2,3% rispetto al valore reale. Il modello è leggermente migliore del precedente nn.4, ma risulta nettamente peggiore degli altri.

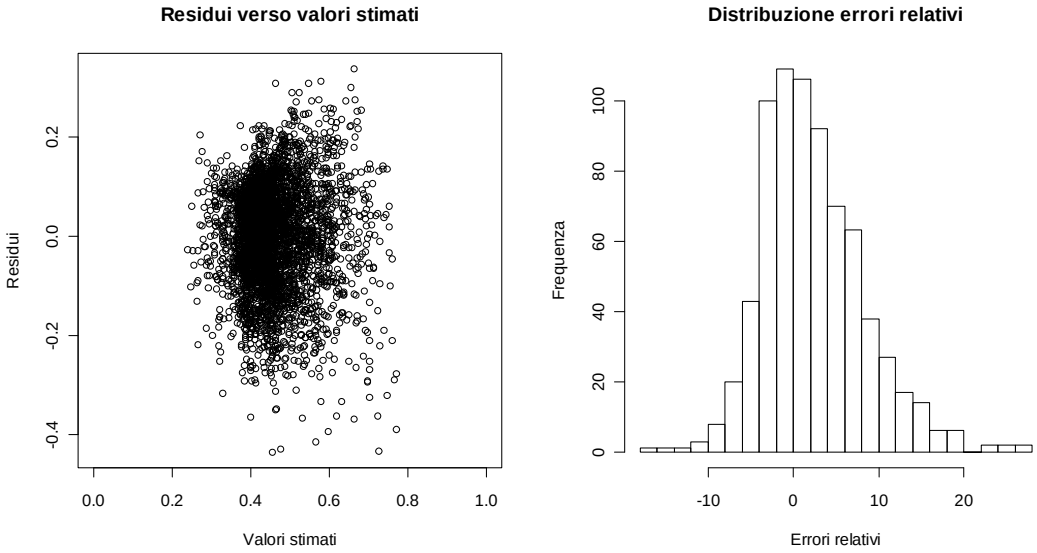


Figura 3.43 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.5.*

3.5.6 Modello avente come predittori la temperatura normalizzata e il coefficiente settimanale (nn.6)

Finora nei modelli basati su reti neurali la variabile `temperatura` è stata considerata senza trasformazioni di scala. Rispetto al precedente modello nn.5 a questa variabile viene applicata la funzione `scale()` che la trasforma sottraendole la media e dividendola per lo scarto quadratico medio. Il confronto tra i due modelli sulla base dello scarto quadratico medio della distribuzione degli errori relativi valutati sul *validation set* evidenzia la loro equivalenza: i valori sono 6,02 per nn.5 e 6,01 per nn.6.

I grafici di Figura 3.44 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale e confermano anche in questo caso la scarsa capacità predittiva del modello.

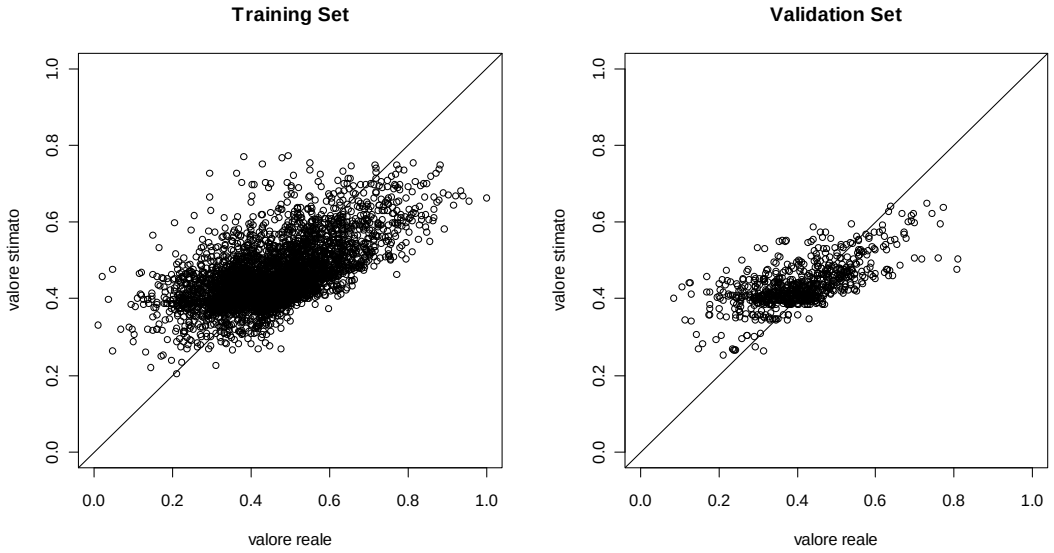


Figura 3.44 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.6.*

Nella Figura 3.44 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 44% degli errori è inferiore al 3% mentre il 63% degli errori è inferiore al 5%. Anche in questo caso è presente una distorsione delle stime prodotte del 2,3%. Tutti questi indicatori sono molto simili ai corrispondenti del modello nn.5.

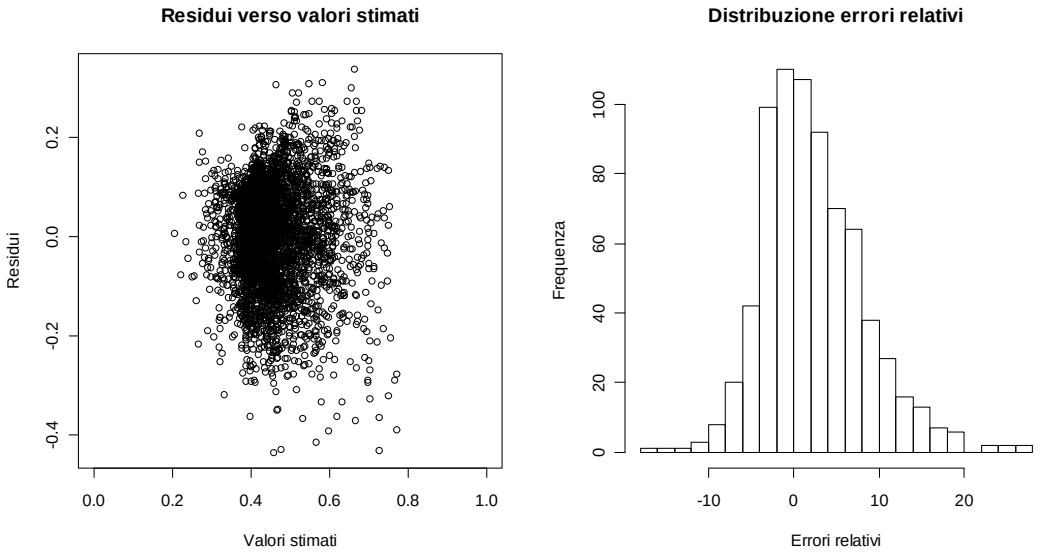


Figura 3.45 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.6.*

3.5.7 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, valori normalizzati (nn.7)

In questo modello vengono nuovamente inseriti tra i predittori i valori dell'erogato nei 7 giorni precedenti, indispensabili per ottenere un buon adattamento. I predittori sono gli stessi visti per il modello nn.1 con l'unica differenza che vengono normalizzati a media nulla e varianza unitaria invece che scalati nell'intervallo [0,1]. La bontà di questo modello è equivalente a quella rilevata per il modello nn.1: considerando le osservazioni del *validation set* lo scarto quadratico medio della distribuzione degli errori relativi è 3,67 contro 3,69 del modello nn.1. Il fatto che i predittori siano scalati sull'intervallo [0,1] o siano normalizzati non varia quindi la bontà dell'adattamento.

I grafici di Figura 3.46 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale e confermano il buon adattamento del modello.

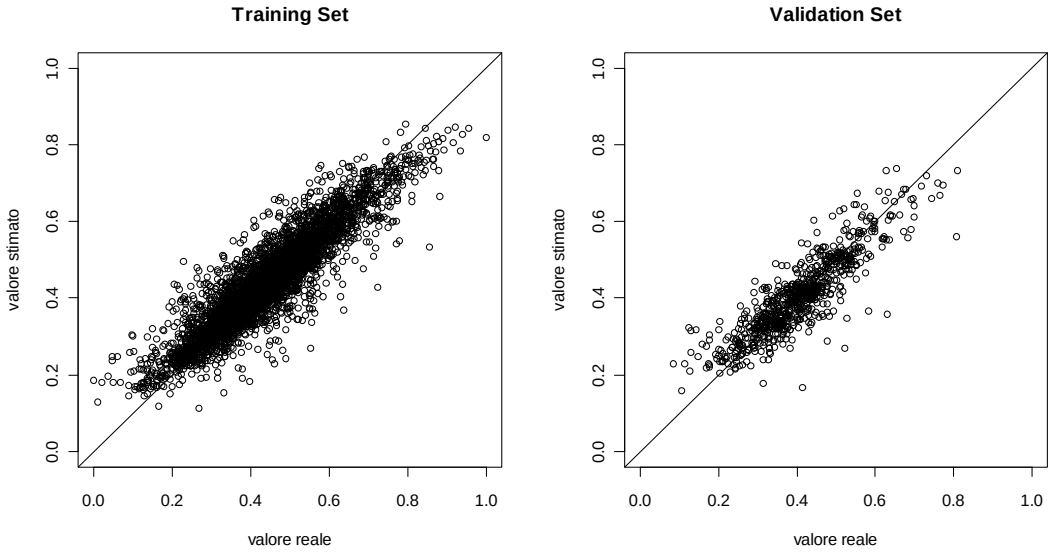


Figura 3.46 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.7.*

Nella Figura 3.47 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 67% degli errori è inferiore al 3% mentre l'87% degli errori è inferiore al 5%. La distorsione è quasi inesistente: la media degli errori relativi è pari a 0,08%. Il modello nn.1 riporta valori molto simili.

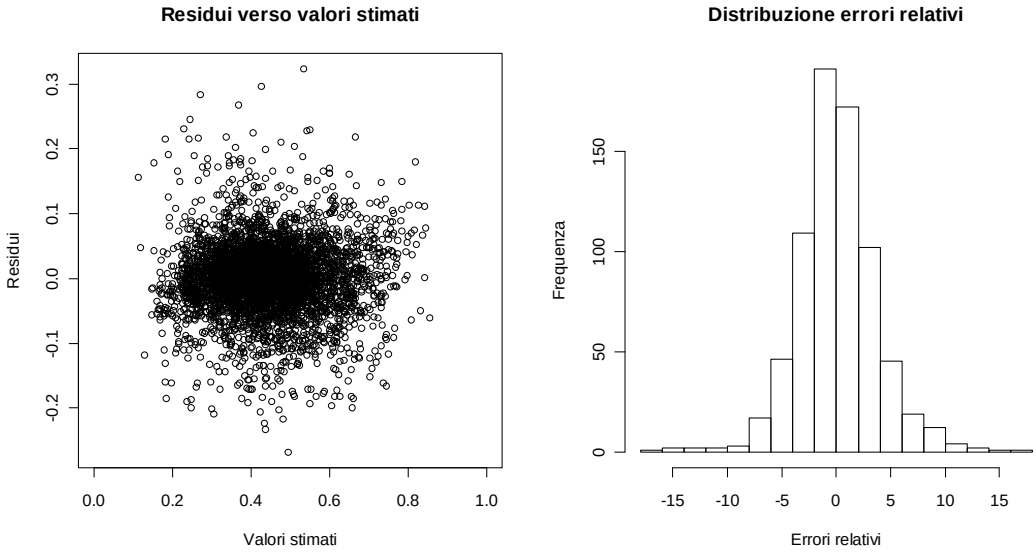


Figura 3.47 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.7.*

3.5.8 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e il coefficiente settimanale (nn.8)

In questo modello vengono inseriti come predittori i valori dell'erogato nei 7 giorni precedenti scalati nell'intervallo [0,1] e la variabile `rapp_settimana`. L'adattamento di questo modello è molto simile a quello rilevato per il modello nn.3 nonostante quest'ultimo considerasse oltre a questi predittori anche la variabile `temperatura`. In particolare, lo scarto quadratico medio degli errori relativi calcolato sul *validation set* è 3,66 per nn.8 e 3,72 per nn.3. Se ne deduce che l'introduzione della temperatura come elemento di previsione dei consumi, in aggiunta ai valori di acqua erogata nei giorni precedenti, non porta grandi benefici.

I grafici di Figura 3.48 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale e confermano il buon adattamento del modello.

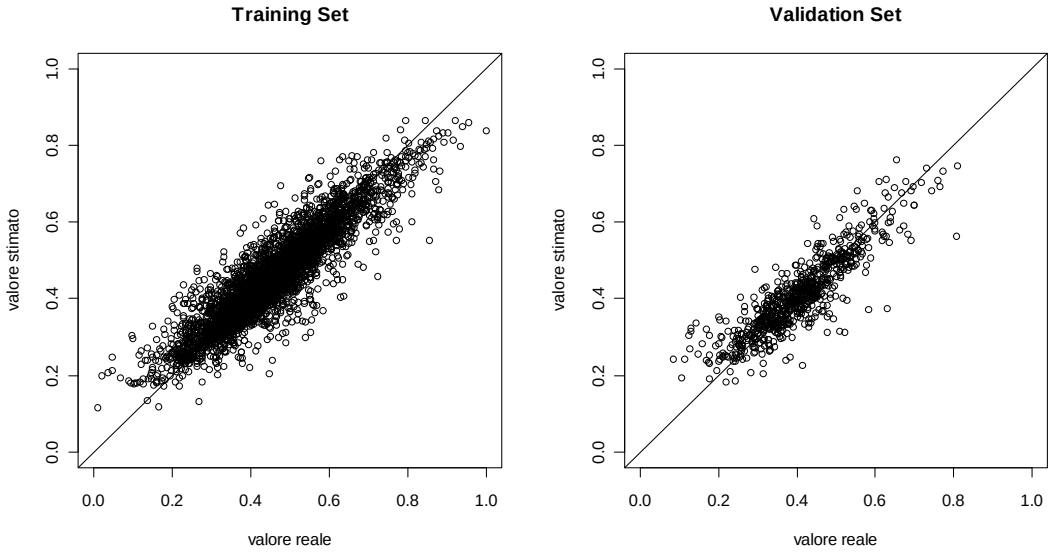


Figura 3.48 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.8.*

Nella Figura 3.49 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 70% degli errori è inferiore al 3% mentre l'86% degli errori è inferiore al 5%. La distorsione è quasi inesistente: la media degli errori relativi è pari a 0,35%. Il modello nn.3 riporta valori molto simili.

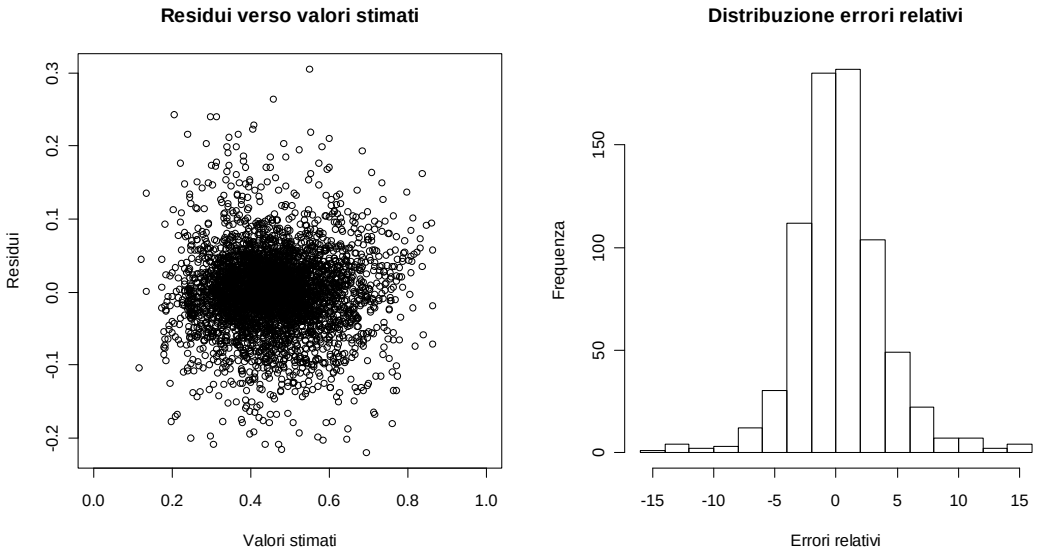


Figura 3.49 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.8.*

3.5.9 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale. Verifica overfitting (nn.9)

Questo modello considera gli stessi predittori del modello nn.3 ma viene aumentato il numero di neuroni presenti nello strato nascosto: invece dei 4 neuroni considerati nei precedenti modelli ne vengono considerati 18, pari al doppio del numero degli ingressi, valore considerato limite massimo per evitare un eccessivo adattamento ai dati. L'obiettivo è appunto quello di verificare il manifestarsi del fenomeno di *overfitting*. Si rileva che a fronte di una leggera diminuzione dell'errore sul *training set* (lo scarto quadratico medio degli errori di stima passa da 3,55 a 3,14) vi è un notevole aumento della medesima quantità nel *validation set* che passa da 3,72 a 5,32 confermando che già con 18 neuroni nello strato nascosto si è in presenza di *overfitting*.

I grafici di Figura 3.50 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale. Si vede la maggior dispersione presente nel plot relativo al *validation set*.

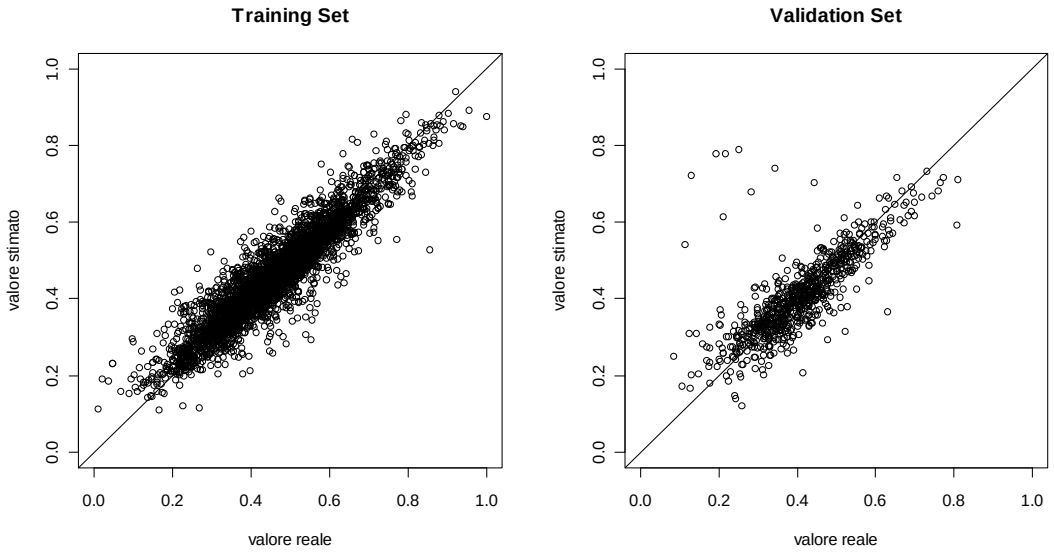


Figura 3.50 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.9.*

Nella Figura 3.51 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 65% degli errori è inferiore al 3% mentre l'86% degli errori è inferiore al 5%.

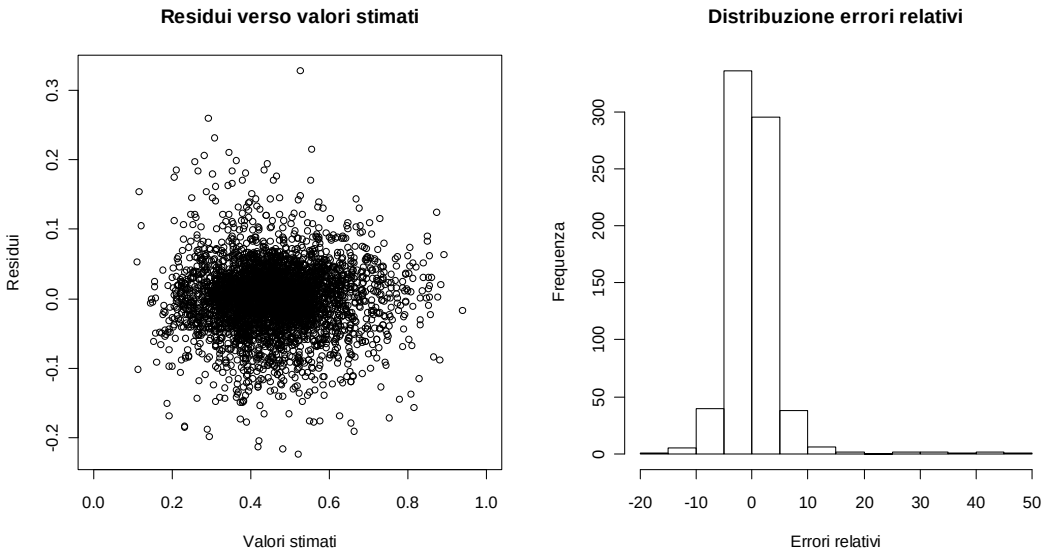


Figura 3.51 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.9.*

3.5.10 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e il coefficiente settimanale. Molti neuroni nello strato nascosto (nn.10)

Anche in questo modello viene considerato un numero elevato di neuroni nello strato nascosto, escludendo però come variabile predittiva la temperatura. Anche in questo caso si verifica un certo livello di *overfitting*. Lo scarto quadratico medio degli errori relativi calcolato sul *training set* è pari a 3,15 mentre sul *validation set* è pari a 4,66.
I grafici di Figura 3.52 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale. Anche in questo caso è possibile notare la maggior dispersione presente nel plot relativo al *validation set*.

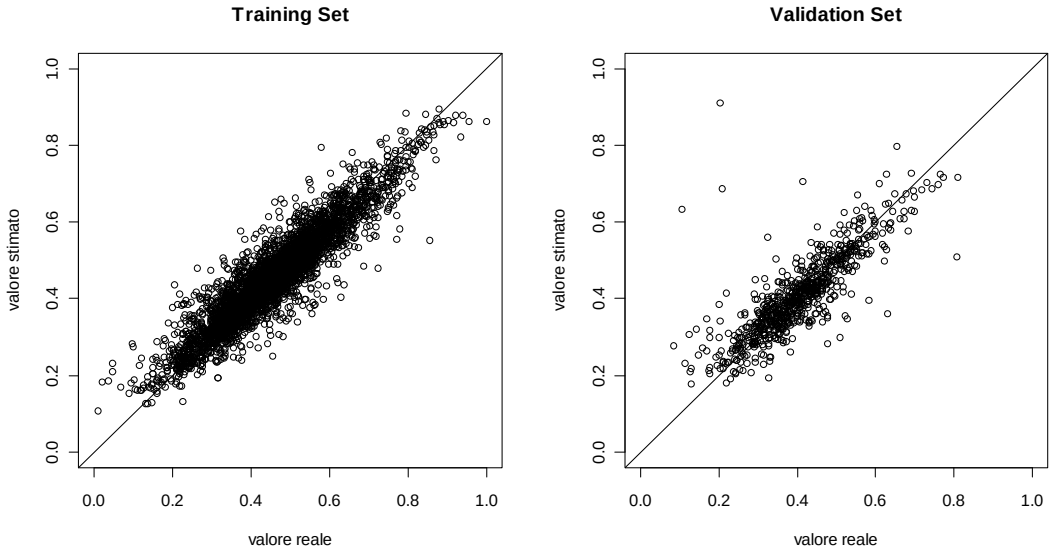


Figura 3.52 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.10.*

Nella Figura 3.53 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 69% degli errori è inferiore al 3% mentre l'87% degli errori è inferiore al 5%.

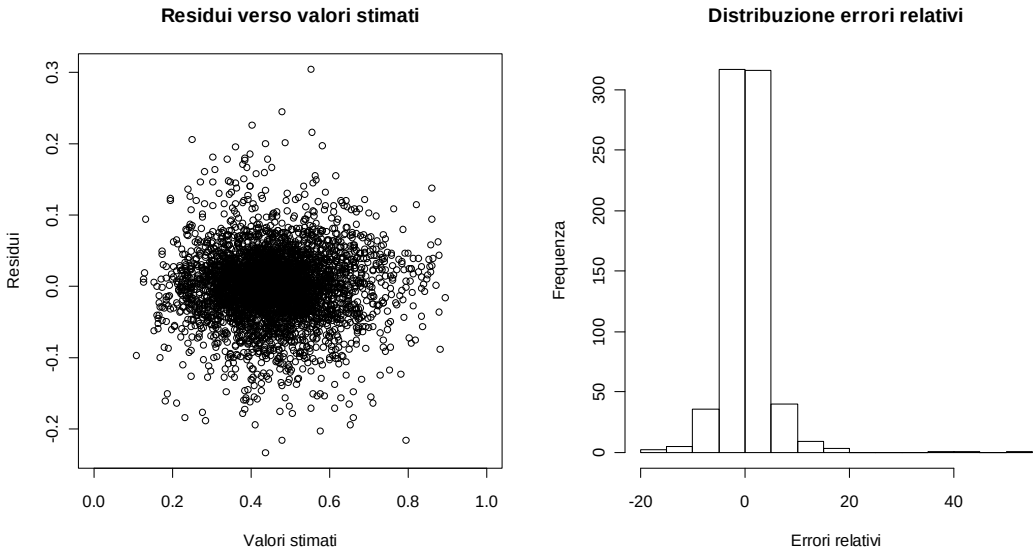


Figura 3.53 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.10.*

3.5.11 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il coefficiente settimanale e il giorno festivo (nn.11)

In questo modello, così come nei successivi, si è tornati a considerare 4 neuroni nello strato nascosto. I predittori sono i medesimi presenti nel modello nn.3 con l'aggiunta del fattore `festivo`, variabile a valore 1 o 0 a seconda che il giorno sia rispettivamente festivo o meno, così come descritto nel §3.2.2.1.
Lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* rispetto modello nn.3 è rimasto sostanzialmente invariato: in entrambi i casi è prossimo al valore 3,72.
I grafici di Figura 3.54 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale, dai quali è possibile notare un buon adattamento del modello.

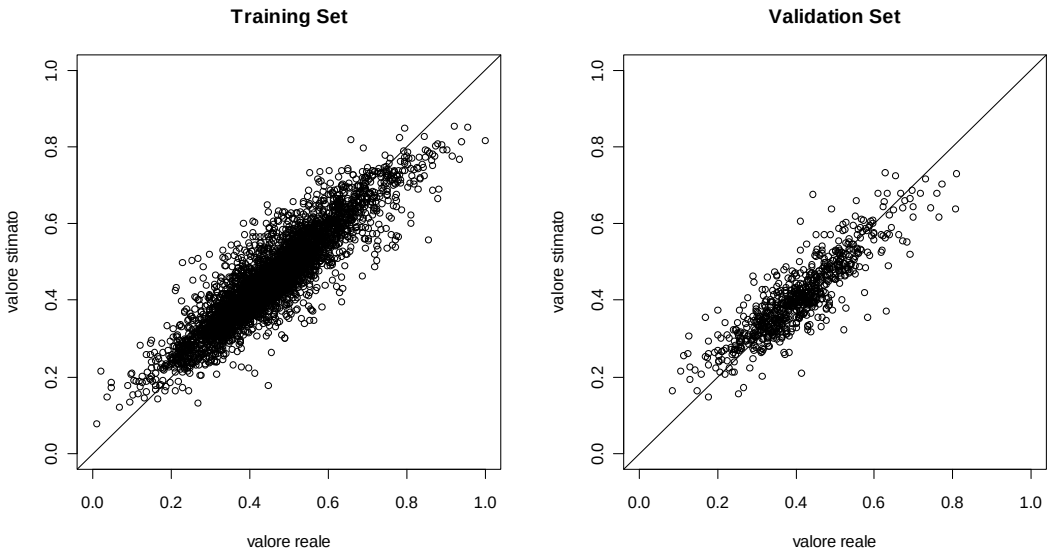


Figura 3.54 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.11.*

Nella Figura 3.55 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 65% degli errori è inferiore al 3% mentre l'86% degli errori è inferiore al 5%.

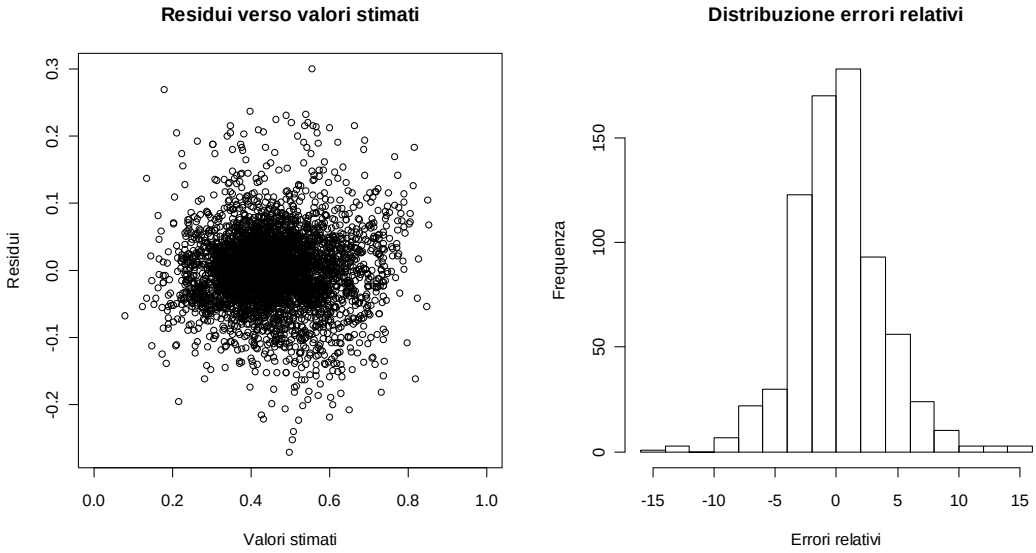


Figura 3.55 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.11.*

3.5.12 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il coefficiente settimanale, il giorno festivo e il giorno della settimana (nn.12)

In questo modello è stata aggiunta rispetto al modello precedente nn.11 la variabile `giorno_settimana`, fattore a 7 livelli che identifica appunto il giorno della settimana. Lo scarto quadratico medio dell'errore relativo, calcolato sul *validation set* scende da 3,72 a 2,78, quelli calcolato sul *training set* scende da 3,60 a 2,56.

I grafici di Figura 3.56 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale. È possibile notare anche visivamente il maggior adattamento del modello, notando come i plot siano più concentrati sulle rette di previsione ideale rispetto ai corrispondenti grafici di Figura 3.54.

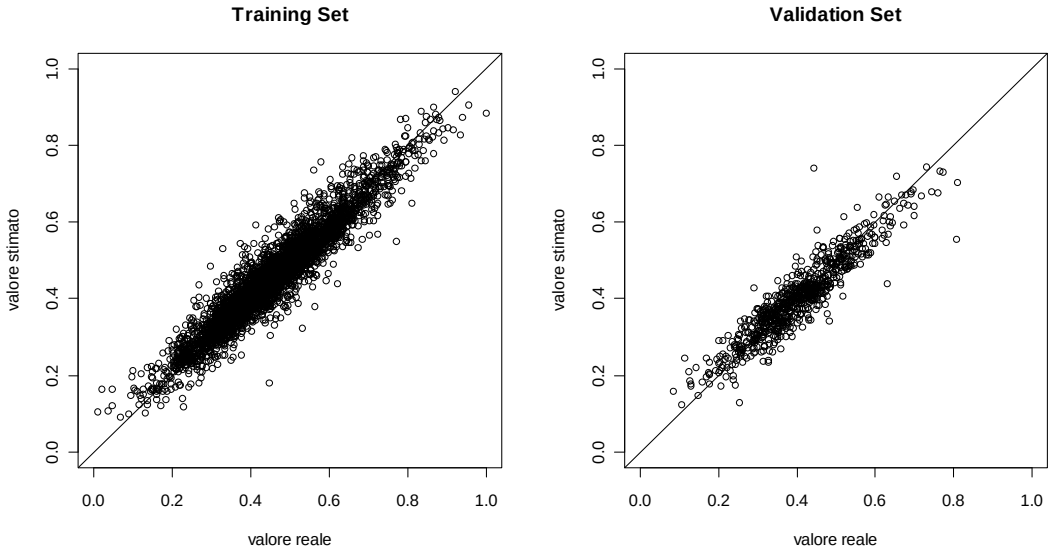


Figura 3.56 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.12.*

Nella Figura 3.57 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 76% degli errori è inferiore al 3% e ben il 94% degli errori è inferiore al 5%.

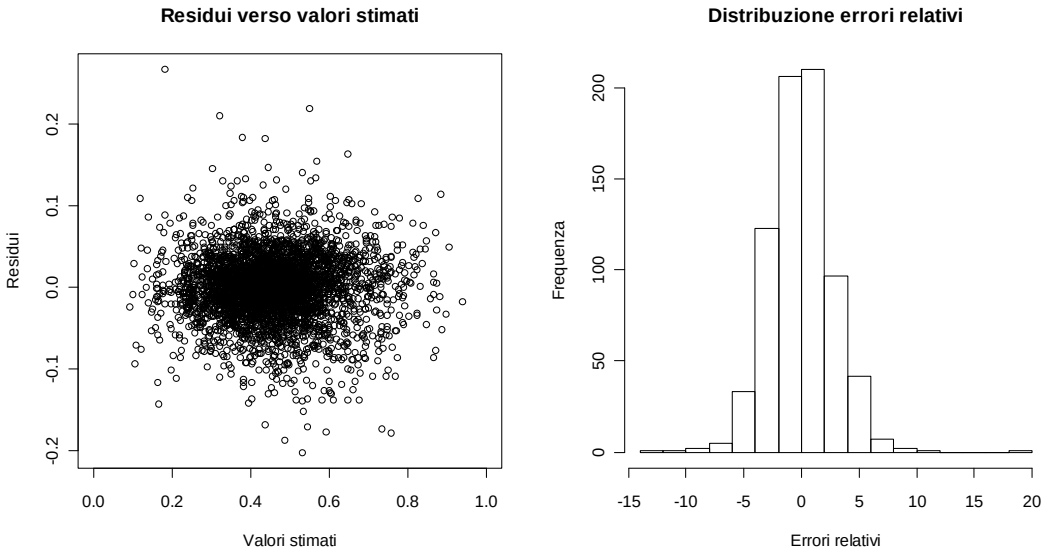


Figura 3.57 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.12.*

3.5.13 Previsione settimanale

Nei modelli considerati finora è stata considerata la stima del volume di acqua erogata il giorno successivo (variabile `erogato0_n`), ponendo quindi l'orizzonte temporale di stima a 24 ore. Nella gestione dell'esercizio degli impianti è tuttavia indispensabile ottenere anche una previsione sulla quantità prevista di consumo nei successivi 7 giorni. Per rispondere a questa necessità è stato predisposto un modello di previsione settimanale dei consumi idrici.

Il primo passaggio è stato la realizzazione di modelli di previsione di stima non solo del giorno seguente, ma anche dei giorni successivi. Mantenendo la variabile risposta `erogato0_n` (riferita al tempo t), per stimare il consumo del giorno successivo si usano come predittori i valori `erogato1_n` (tempo $t-1$), `erogato2_n` (tempo $t-2$),…, `erogato7_n` (tempo $t-7$), esattamente quello che è stato fatto nei modelli precedenti. Per stimare il consumo che si realizzerà tra due giorni è utile continuare a considerare questo rappresentato dalla variabile `erogato0_n`, ponendosi quindi al tempo $t-2$. In questo caso non è possibile considerare come predittore la variabile `erogato1_n` (tempo $t-1$) in quanto non ancora nota. Si possono invece utilizzare i predittori `erogaro2_n`, `erogato3_n`,…, `erogato7_n`. In aggiunta è stata costruita la variabile `erogato8_n` (consumo al tempo $t-8$) in modo da avere a comunque disposizione gli ultimi 7 valori di consumo realizzati in un certo momento. Per costruire tale valore si è dovuto rinunciare alla possibilità di utilizzare la prima riga del *dataset* in quanto per tale unità il valore è non noto.

In modo analogo si possono costruire i modelli di previsione per i consumi a 3,4,…,7 giorni, escludendo di volta in volta anche le variabili `erogato2_n`, `erogato3_n`,…, `erogato6_n` e inserendo al loro posto le variabili `erogato9_n`, `erogato10_n`,…, `erogato13_n` opportunamente costruite, perdendo rispettivamente 2,3,…, 6 unità del *dataset*.

La Tabella 3.9 riporta quali sono le variabili utilizzate per ogni orizzonte di previsione considerato: la variabile risposta è sempre `erogato0_n`; i predittori sono, a seconda dell'orizzonte temporale considerato, quelli indicati con X. L'ultima riga rappresenta il numero di unità che devono essere escluse quando si utilizzano i rispettivi predittori nei modelli di previsione.

Oltre ai predittori ora descritti, vengono considerati anche i predittori utilizzati nel modello nn.12, risultato il migliore di quelli considerati, cioè `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`.

Tabella 3.9 - *Predittori utilizzati sulla base dell'orizzonte di previsione e unità non disponibili per i modelli.*

	Orizzonte di previsione (gg)						
Predittori	**1**	**2**	**3**	**4**	**5**	**6**	**7**
erogato1	X						
erogato2	X	X					
erogato3	X	X	X				
erogato4	X	X	X	X			
erogato5	X	X	X	X	X		
erogato6	X	X	X	X	X	X	
erogato7	X	X	X	X	X	X	X
erogato8		X	X	X	X	X	X
erogato9			X	X	X	X	X
erogato10				X	X	X	X
erogato11					X	X	X
erogato12						X	X
erogato13							X
Unità escluse	0	1	2	3	4	5	6

3.5.13.1 Modello di previsione a 1 giorno (nn.S0)

Il modello di previsione per il giorno seguente è, di fatto, uguale al modello nn.12. La variabile risposta è `erogato0_n` e i predittori sono `erogato1_n`,…,`erogato7_n`, oltre a `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Il *data frame* utilizza tutte le osservazioni.

Lo scarto quadratico medio dell'errore relativo, calcolato sul *validation set* è pari a 2,78, identico valore rilevato per il modello nn.12.

I grafici di Figura 3.58 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

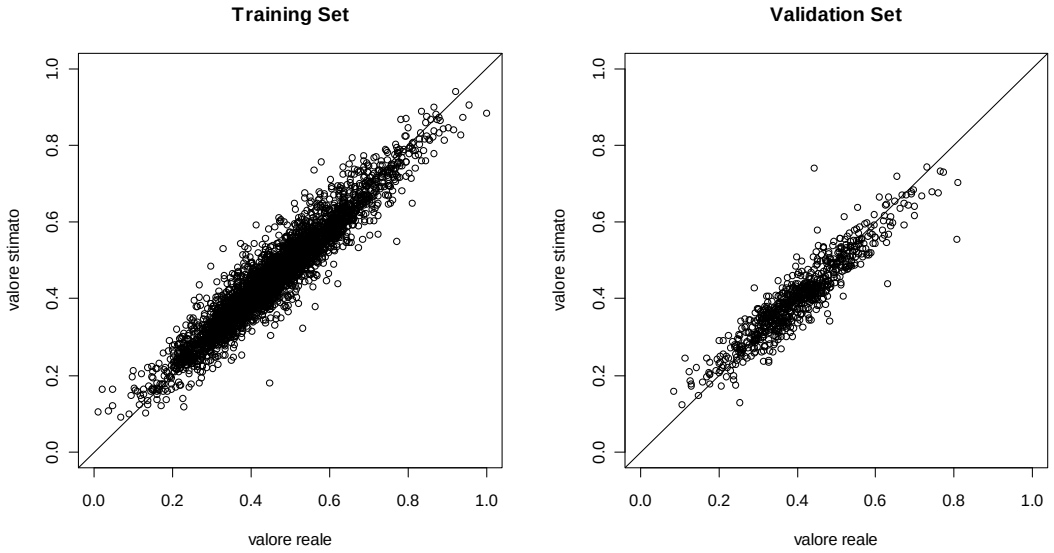


Figura 3.58 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S0.*

Nella Figura 3.59 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 76% degli errori è inferiore al 3% e ben il 94% degli errori è inferiore al 5%.

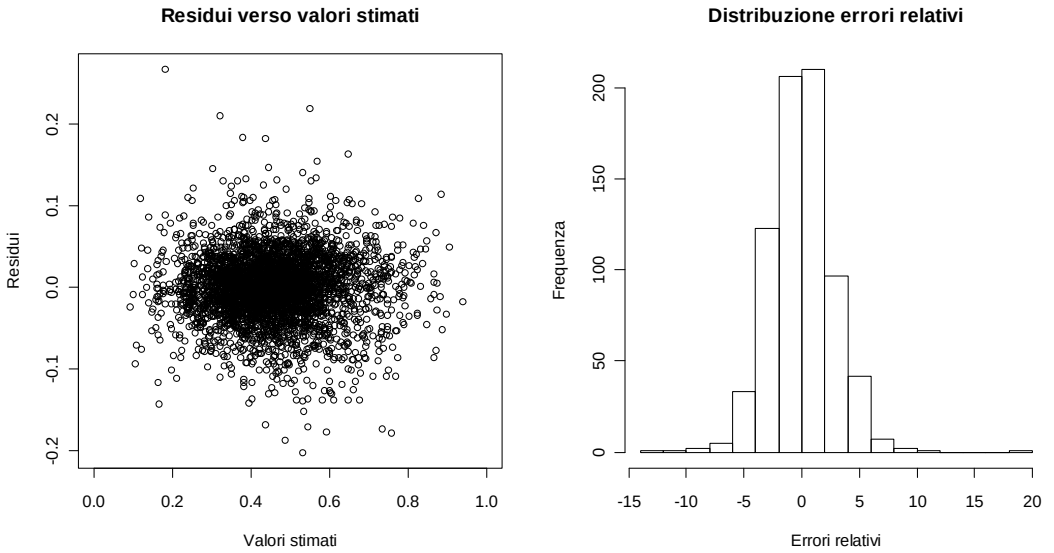


Figura 3.59 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S0.*

3.5.13.2 Modello di previsione a 2 giorni (nn.S1)

Per la stima del volume erogato il giorno $t+2$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0_n`. I predittori sono invece

`erogato2_n`,…, `erogato8_n`, quindi si considerano i valori disponibili fino a due giorni antecedenti. Gli altri predittori sono `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Viene esclusa la prima osservazione del *data frame* in quanto non dispone del valore sulla variabile `erogato8_n`.

Lo scarto quadratico medio degli errori relativi, valutati sul *validation set*, sale da 2,78 del modello nn.S0 a 3,28 del presente modello. È infatti atteso che allungando il periodo di previsione la stima diventi meno precisa.

I grafici di Figura 3.60 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

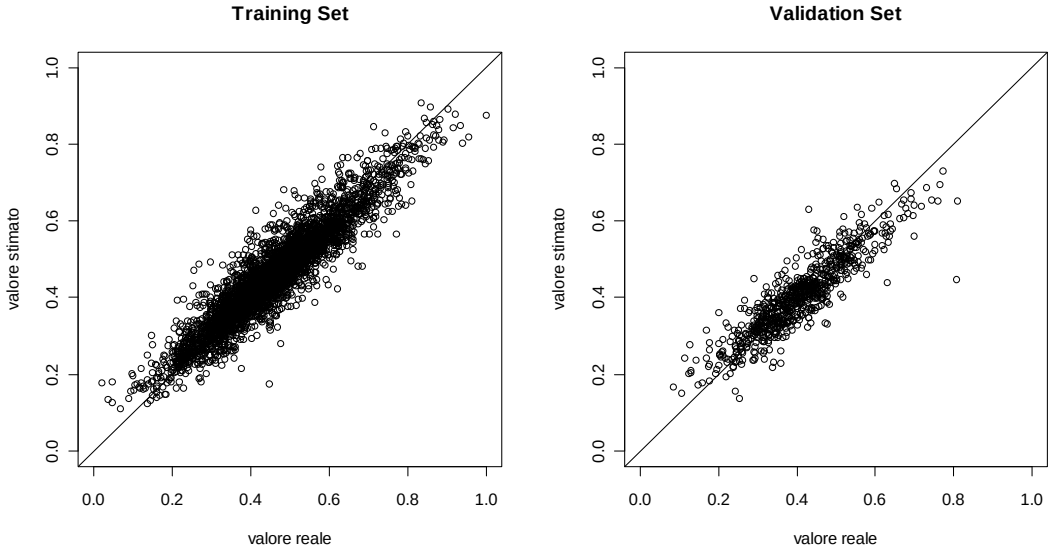


Figura 3.60 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S1.*

Nella Figura 3.61 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 70% degli errori è inferiore al 3% e l'89% degli errori è inferiore al 5%, mentre per il modello nn.S0 i corrispondenti valori erano 76% e 94%.

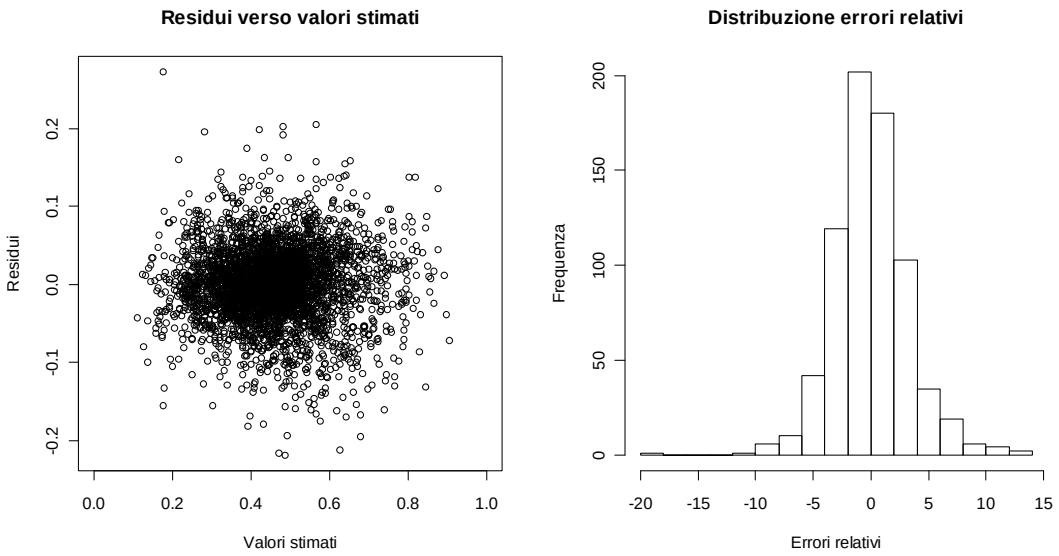


Figura 3.61 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S1.*

3.5.13.3 Modello di previsione a 3 giorni (nn.S2)

Per la stima del volume erogato il giorno $t+3$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0_n`. I predittori sono invece `erogato3_n`,…, `erogato9_n`, quindi si considerano i valori disponibili fino a 3 giorni antecedenti. Gli altri predittori sono `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Vengono escluse le prime 2 osservazioni del *data frame* in quanto non dispongono del valore sulla variabile `erogato9_n`.

Lo scarto quadratico medio degli errori relativi, valutati sul *validation set*, sale ulteriormente da 3,28 del modello nn.S1 a 3,46.

I grafici di Figura 3.62 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

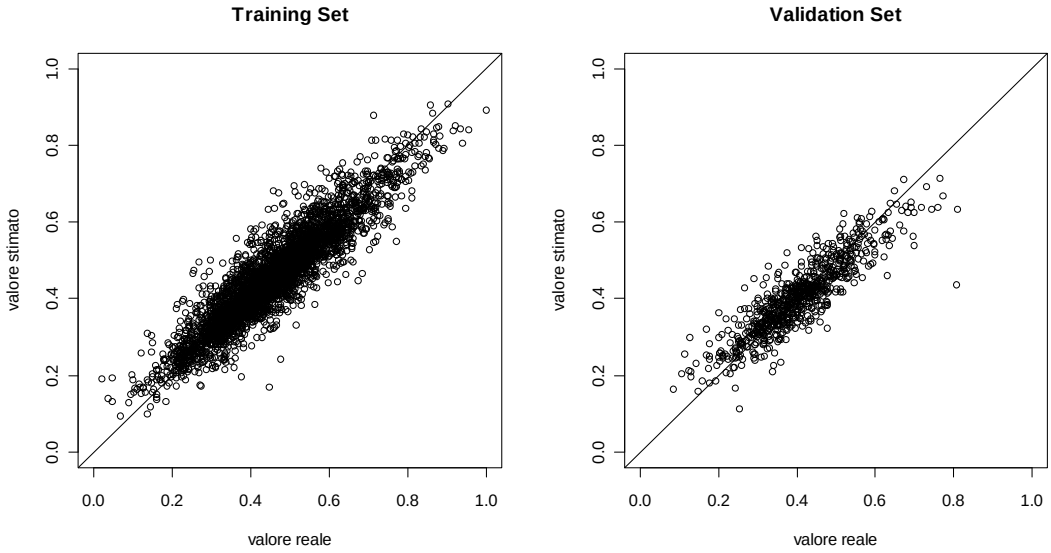


Figura 3.62 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S2.*

Nella Figura 3.63 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 68% degli errori è inferiore al 3% e l'86% degli errori è inferiore al 5%, mentre per il modello nn.S0 i corrispondenti valori erano 70% e 89%.

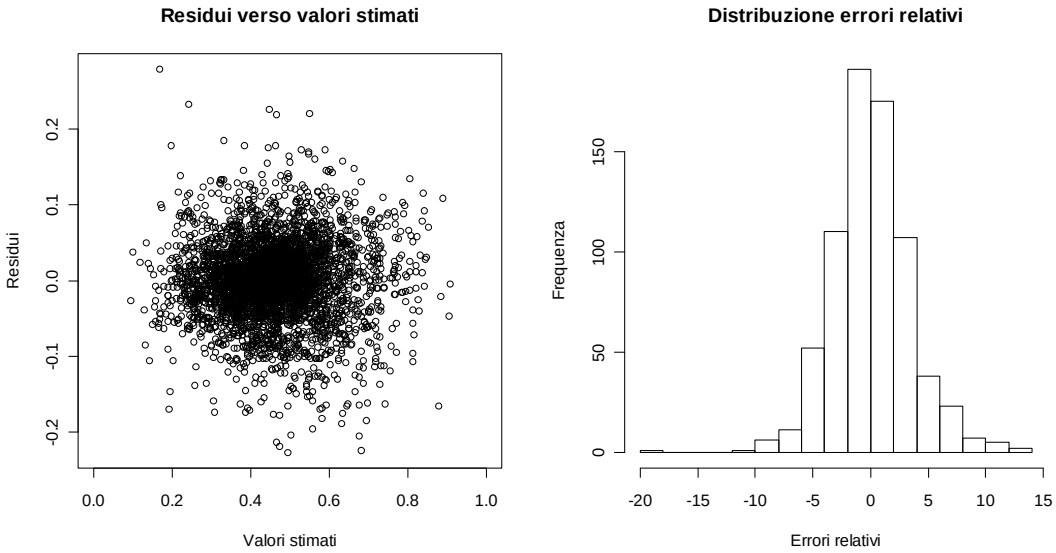


Figura 3.63 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S2.*

3.5.13.4 Modello di previsione a 4 giorni (nn.S3)

Per la stima del volume erogato il giorno $t+4$ è stato predisposto un opportuno modello che ha come variabile risposta sempre il valore `erogato0_n`. I predittori sono invece `erogato4_n`,…, `erogato10_n`, quindi si considerano i valori disponibili fino a 4 giorni antecedenti. Gli altri predittori sono `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Vengono escluse le prime 3 osservazioni del *data frame* in quanto non dispongono del valore sulla variabile `erogato10_n`.

Lo scarto quadratico medio degli errori relativi, valutati sul *validation set*, sale ulteriormente da 3,46 del modello nn.S2 a 3,65.

I grafici di Figura 3.64 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

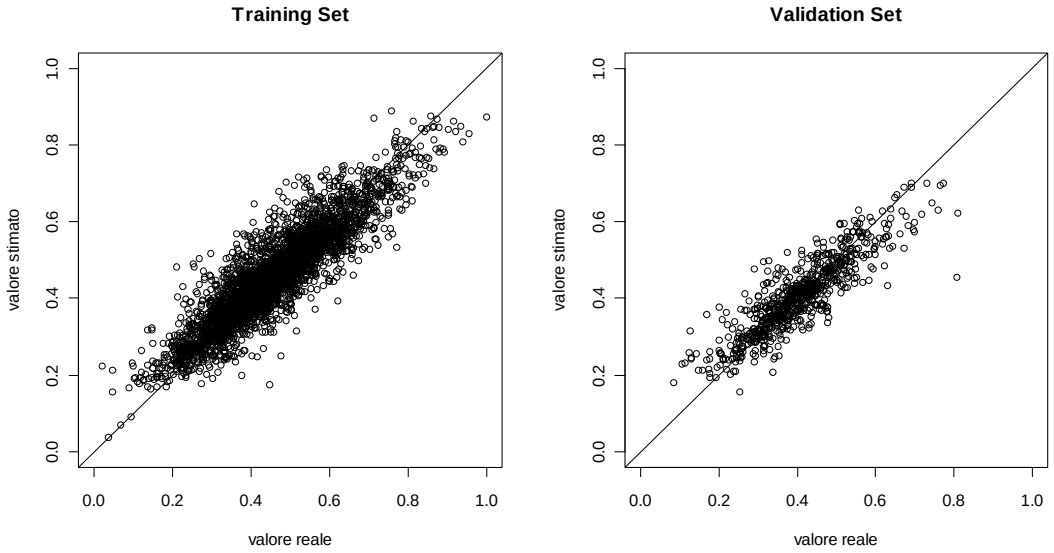


Figura 3.64 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S3.*

Nella Figura 3.65 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 69% degli errori è inferiore al 3% e l'84% degli errori è inferiore al 5%, mentre per il modello nn.S0 i corrispondenti valori erano 68% e 86%.

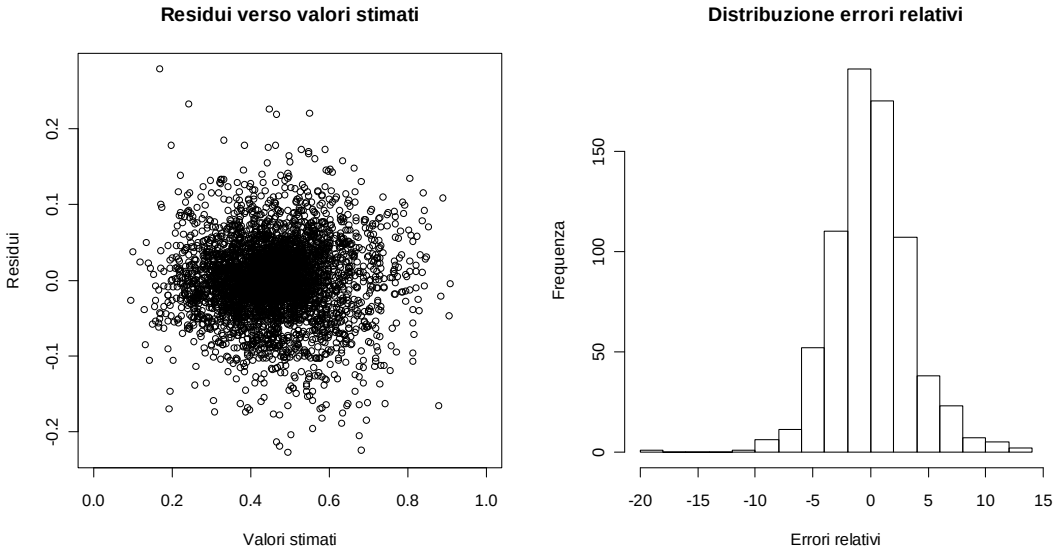


Figura 3.65 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S3.*

3.5.13.5 Modello di previsione a 5 giorni (nn.S4)

Per la stima del volume erogato il giorno $t+5$ è stato predisposto un opportuno modello che ha come variabile risposta sempre il valore `erogato0_n`. I predittori sono invece `erogato5_n`,…, `erogato11_n`, quindi si considerano i valori disponibili fino a 5 giorni antecedenti. Gli altri predittori sono `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Vengono escluse le prime 4 osservazioni del *data frame* in quanto non dispongono del valore sulla variabile `erogato11_n`.

Lo scarto quadratico medio degli errori relativi, valutati sul *validation set*, sale ulteriormente da 3,65 del modello nn.S3 a 3,87.

I grafici di Figura 3.66 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

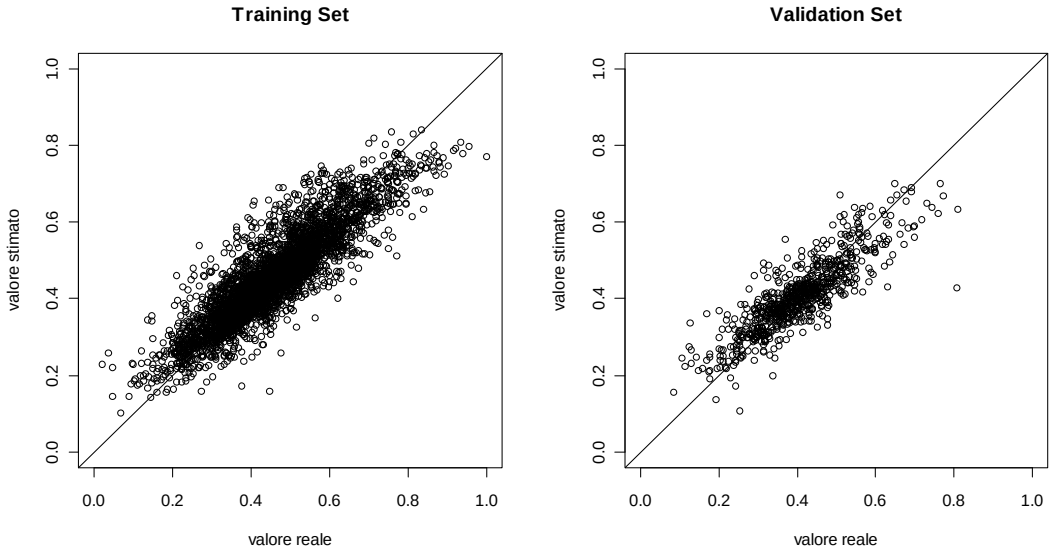


Figura 3.66 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S4.*

Nella Figura 3.67 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 64% degli errori è inferiore al 3% e l'82% degli errori è inferiore al 5%, mentre per il modello nn.S0 i corrispondenti valori erano 69% e 84%.

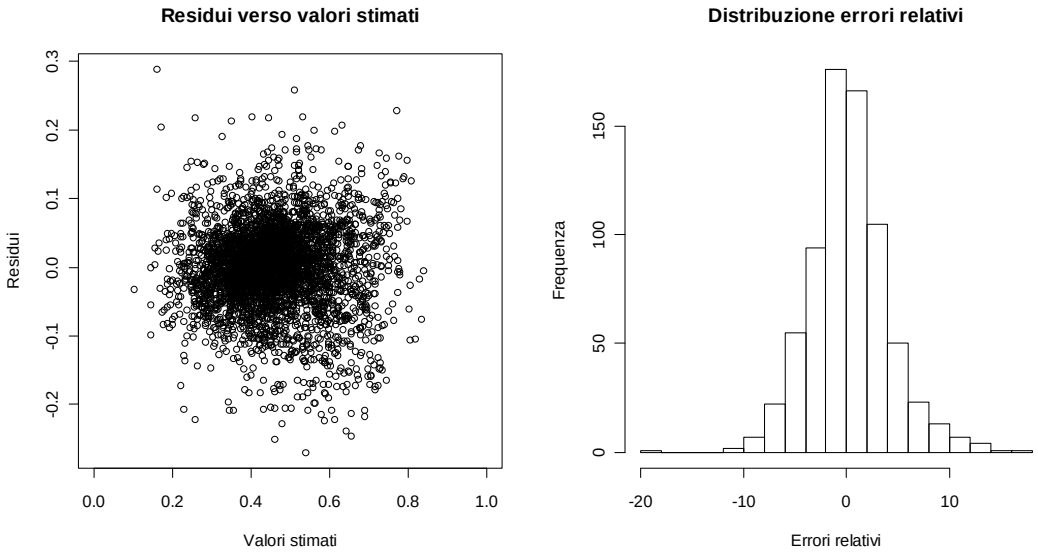


Figura 3.67 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S4.*

3.5.13.6 Modello di previsione a 6 giorni (nn.S5)

Per la stima del volume erogato il giorno $t+6$ è stato predisposto un opportuno modello che ha come variabile risposta sempre il valore `erogato0_n`. I predittori sono

invece `erogato6_n`,…, `erogato12_n`, quindi si considerano i valori disponibili fino a 6 giorni antecedenti. Gli altri predittori sono `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Vengono escluse le prime 5 osservazioni del *data frame* in quanto non dispongono del valore sulla variabile `erogato12_n`.
Lo scarto quadratico medio degli errori relativi, valutati sul *validation set*, sale ulteriormente da 3,87 del modello nn.S4 a 3,95.
I grafici di Figura 3.68 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

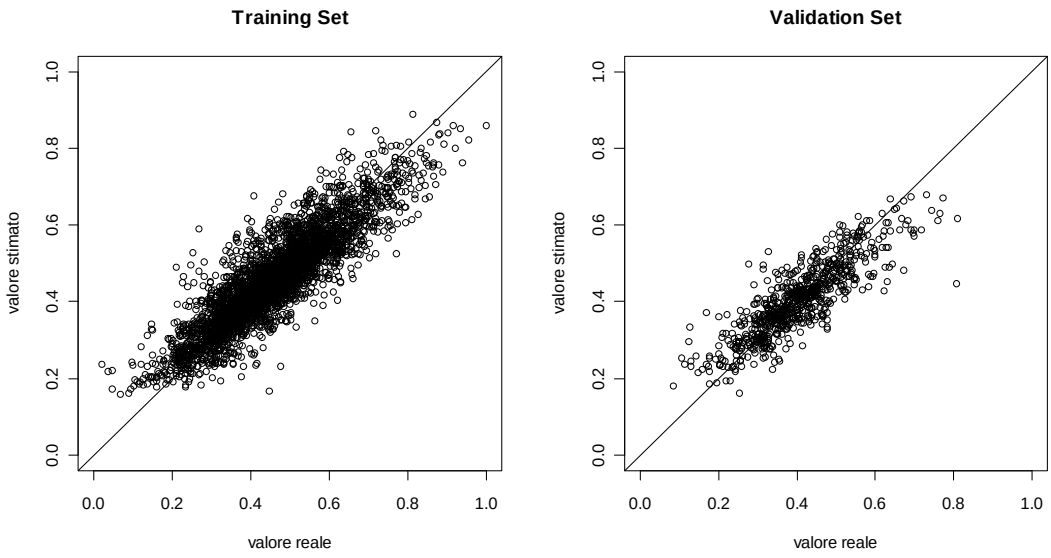


Figura 3.68 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S5.*

Nella Figura 3.69 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 64% degli errori è inferiore al 3% e l'81% degli errori è inferiore al 5%, mentre per il modello nn.S0 i corrispondenti valori erano 64% e 82%.

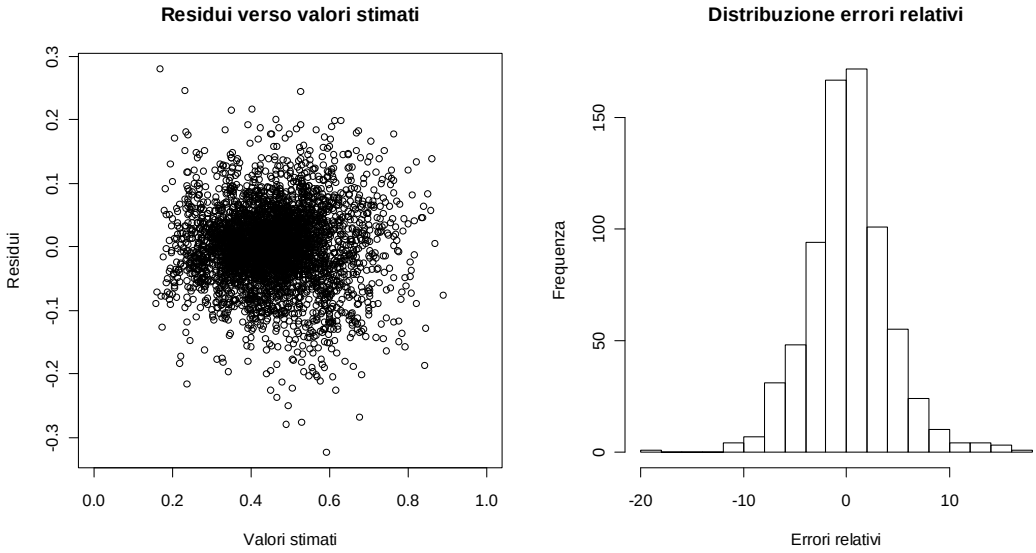


Figura 3.69 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S5.*

3.5.13.7 Modello di previsione a 7 giorni (nn.S6)

Per la stima del volume erogato il giorno $t+7$ è stato predisposto un opportuno modello che ha come variabile risposta sempre il valore `erogato0_n`. I predittori sono invece `erogato7_n`,…, `erogato13_n`, quindi si considerano i valori disponibili fino a 7 giorni antecedenti. Gli altri predittori sono `temperatura`, `rapp_settimana`, `festivo` e `giorno_settimana`. Vengono escluse le prime 6 osservazioni del *data frame* in quanto non dispongono del valore sulla variabile `erogato13_n`.

Lo scarto quadratico medio degli errori relativi, valutati sul *validation set*, sale ulteriormente da 3,95 del modello nn.S5 a 4,02.

I grafici di Figura 3.70 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

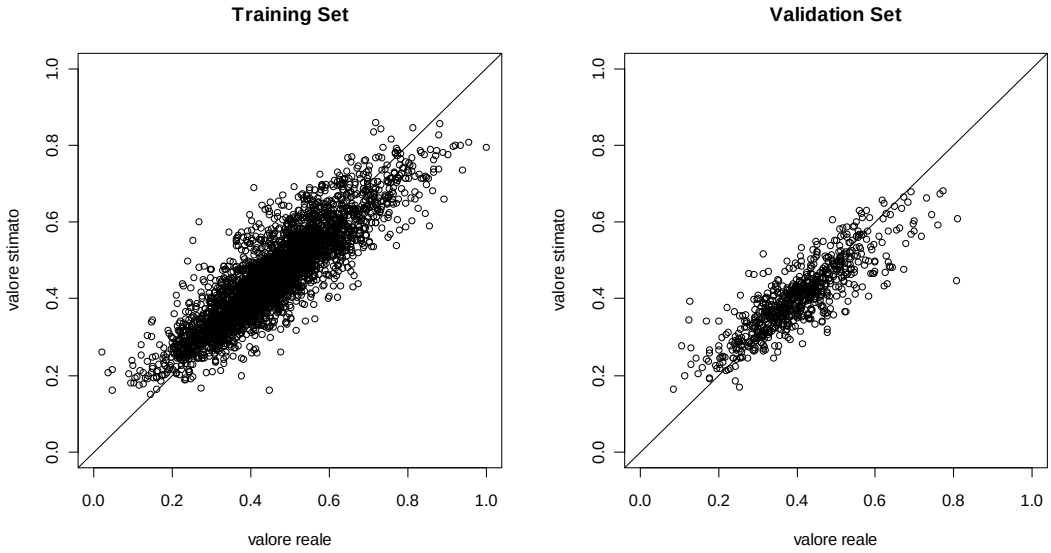


Figura 3.70 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S6.*

Nella Figura 3.71 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 64% degli errori è inferiore al 3% e l'82% degli errori è inferiore al 5%, mentre per il modello nn.S0 i corrispondenti valori erano 64% e 81%.

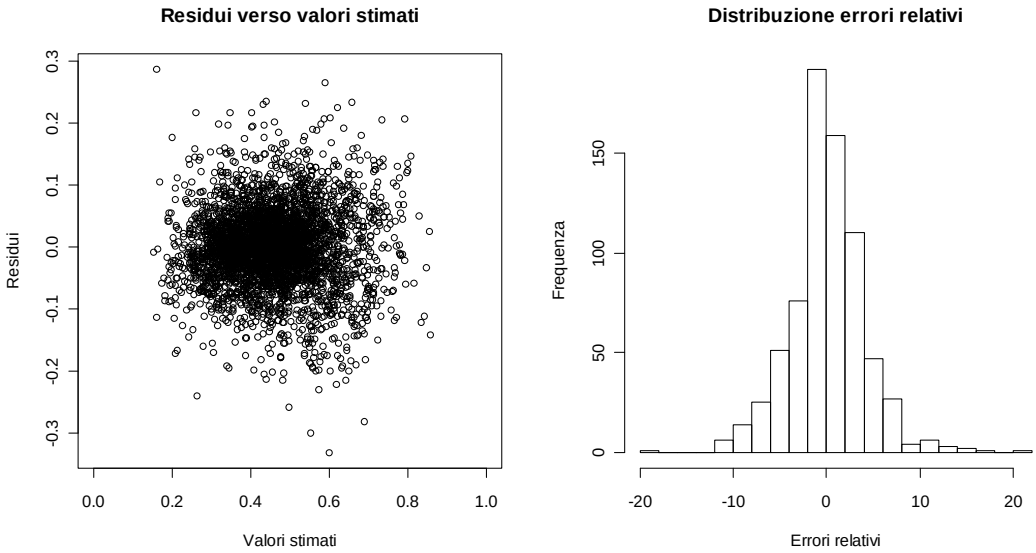


Figura 3.71 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S6.*

3.5.13.8 Modello settimanale (nn.S)

Una volta realizzati i modelli di previsione per i singoli giorni successivi, i valori predetti sono sommati per ottenere una stima complessiva del consumo nella

settimana successiva. È necessario costruire per ciascuna settimana il consumo reale e il suo valore stimato sia per il *dataset training* che per il *dataset validation*.
Il consumo reale per il *dataset training* (`df.train_r`) è stato ottenuto ordinando le 4018 osservazioni per data e suddividendole in 574 gruppi di 7 elementi consecutivi. Indicando con i la posizione della singola osservazione ($i = 1,...,4018$) il primo gruppo è composto dalle osservazioni caratterizzate da valori di i che vanno da 1 al 7, il secondo da 8 al 13, e così via fino al 574° gruppo con i da 4012 a 4018. Per ciascun gruppo n ($n = 1,...,574$) è stato sommato il valore `erogato0` degli elementi che lo compongono creando la variabile `cs.nn_train`.
Formalizzando:

$$\text{cs.nn\_train}_n = \sum_{i=7\cdot n-6}^{7\cdot n} \text{erogato0}_i \qquad n = 1,...,574 \tag{3.34}$$

La variabile `cs.nn_train` rappresenta il consumo reale settimanale per il *dataset training*. Si procede quindi al calcolo del consumo settimanale stimato per il medesimo *dataset*. Per ciascun gruppo sono state calcolate le stime dei consumi dei 7 giorni, utilizzando per ciascun giorno il modello di previsione adeguato al giorno stesso: indicando con t il primo giorno del gruppo ($t = 1,8,15,\ldots,4012$), si tratta di stimare i consumi dei giorni $t, t+1,...,t+6$ avendo a disposizione i dati fino al giorno $t-1$. Per stimare il consumo del giorno t viene utilizzato il modello nn.S0 che usa oltre agli altri predittori le variabili `erogato1`, `erogato2`,…, `erogato7`. Per il secondo giorno ($t+1$) viene utilizzato il modello nn.S1 che usa tra gli altri i predittori `erogato2`, `erogato3`,…, `erogato8` riferiti al giorno ($t+1$) e così via fino al giorno ($t+6$) nel quale si utilizza il modello nn.S6 con predittori `erogato7`, `erogato8`,…, `erogato13`. In questo modo sono stati introdotti nei vari modelli sempre gli stessi valori, cioè i consumi realizzati nei 7 giorni antecedenti il giorno t, ovverosia quelli dell'ultima settimana nota per realizzare la stima del consumo settimanale. Il valore ottenuto come somma delle stime dei singoli giorni è rappresentato dalla variabile `ss.nn_train`:

$$\text{ss.nn\_train}_n = \sum_{k=1}^{7} \text{pred}_{[k]}(7n + k - 7) \tag{3.35}$$

dove $\text{pred}_{[k]}(i)$ è la previsione effettuata utilizzando il modello che considera i predittori `erogato(k)`, `erogato(k+1)`,…, `erogato(k+7)` utilizzando in input i dati della riga i del *training set*.
In modo del tutto analogo sono state costruite le variabili `cs.nn_valid` e `ss.nn_valid` che contengono i consumi reali e stimati per le 104 settimane presenti nel *validation set*.

Le quantità appena calcolate sono rappresentate nei grafici di Figura 3.126**:** a sinistra il plot dei valori stimati verso i valori reali per il *training set* e a destra l'analogo per il *validation set*.

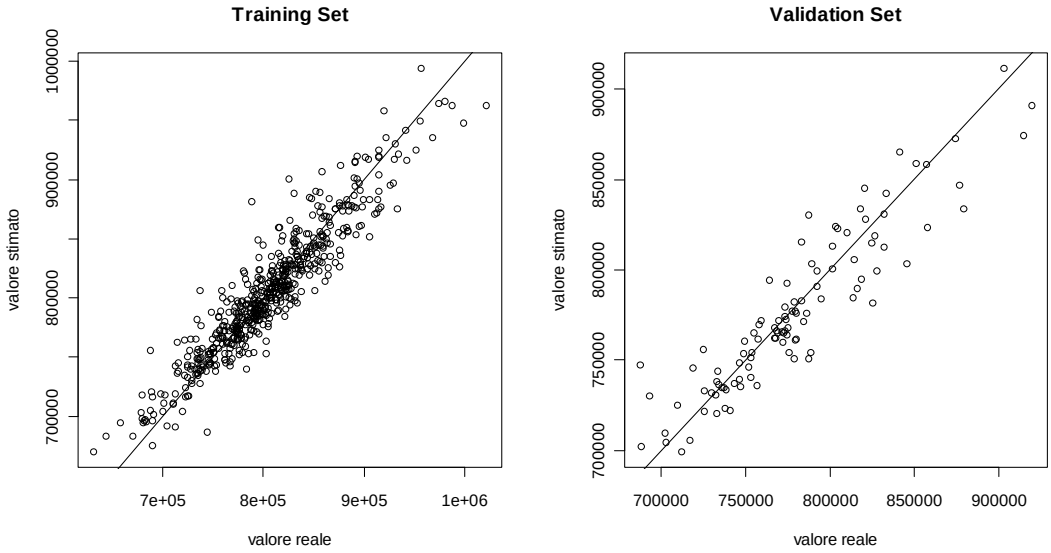


Figura 3.72 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello nn.S.*

È stato prodotto il grafico dei residui verso i valori stimati e la distribuzione degli errori relativi calcolati sul *validation set* (Figura 3.73).

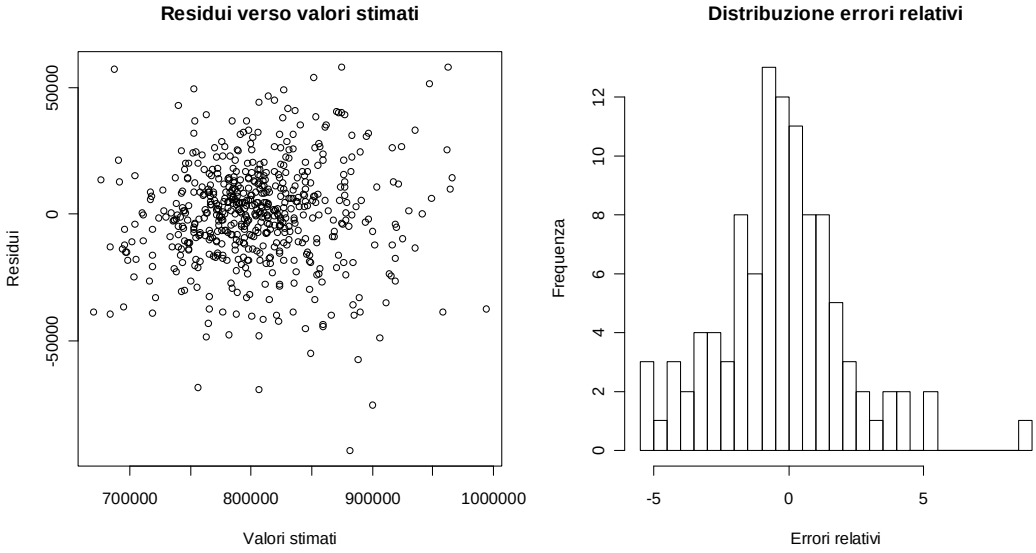


Figura 3.73 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello nn.S.*

Come indicatore di bontà dell'adattamento è stato calcolato lo scarto quadratico medio degli errori di stima relativi, sia per il *training set* che per il *validation set*, pari rispettivamente a 2,51 e 2,37.
Gli errori commessi nella stima delle osservazioni del *validation set* sono nell'80% dei casi inferiori al 3% e nel 94% inferiori al 3%.

3.6 Modelli basati su Random Forests

I modelli basati sull'algoritmo *Random Forests* non necessitano che le variabili predittive o la variabile risposta sia riscalata o normalizzata. Essi, infatti, sono invarianti per trasformazioni monotone delle variabili, proprietà ereditata dagli alberi di classificazione e regressione sui quali l'algoritmo si basa. I modelli che seguono quindi utilizzano le variabili nella scala originale.
Nelle reti neurali il "meccanismo" di previsione del modello risultante rimane nascosto all'interno della rete che viene quindi vista come una "scatola nera" cui fornire i valori di ingresso e osservarne le uscite. Un modello basato su *Random Forests*, invece, ha la possibilità di evidenziare l'importanza dei singoli predittori nella spiegazione della variabile risposta. Nella realizzazione software dell'algoritmo *Random Forests* presente in R, sono calcolati due indicatori di importanza (Kuhn *et al*, 2008):

- %IncMSE: è costruito permutando una alla volta i valori di ciascuna variabile del *test set*, composto dai dati *out-of-bag*, registrandone la previsione e confrontandola con quella del *test set* non permutato. Trattandosi di alberi di regressione, viene calcolato come aumento medio dello scarto quadratico medio nei residui quando la variabile viene permutata. I dettagli dell'algoritmo di calcolo sono riportati al §4.2.2.3. Valori elevati dell'indicatore rappresentano un'elevata importanza della variabile.
- IncNodePurity: misura la qualità di ciascuno *split* per ogni variabile (e quindi ogni nodo) di un albero utilizzando l'indice di Gini. Ogni volta che viene effettuato uno *split* di un nodo sulla base dei valori di una variabile, la misura dell'impurità dei due nodi generati è nel complesso inferiore del nodo padre. Questa riduzione viene attribuita alla variabile che realizza lo *split*; la somma di tutti questi valori, effettuata su tutti i nodi di tutti gli alberi, fornisce un'indicazione dell'importanza dei predittori che spesso è congruente con quanto realizzato dall'indicatore %IncMSE. Un valore

elevato dell'indice ancora una volta indica un'elevata importanza del predittore.

Nei modelli realizzati mediante *Random Forests* che vengono presentati nei paragrafi seguenti è presente come nei precedenti modelli la valutazione dei valori stimati nel *training set*. A differenza degli algoritmi basati su regressione lineare o su reti neurali, nei quali i valori stimati sul *training set* si ottengono a partire dalle stesse osservazioni che hanno realizzato la stima dei parametri del modello e che pertanto possono presentare fenomeni di *overfitting*, nei modelli basati su *Random Forests* questo rischio non sussiste in quanto i valori sul training set vengono valutati quando i singoli valori ricadono negli insiemi o*ut-of-bag*. I valori di adattamento sul *training set* possono pertanto risultare inferiori rispetto alle altre tipologie di modelli.

3.6.1 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti (rf.1)

Il primo modello proposto presenta gli analoghi predittori del modello nn.1, cioè i valori dell'erogato nei 7 giorni precedenti, ottenendo una varianza spiegata pari all'82,15%. L'adattamento è comparabile con l'analogo modello basato sulle reti neurali: lo scarto quadratico medio della distribuzione degli errori relativi calcolati sul *validation set* è pari a 3,74 contro il valore di 3,69 realizzato nel modello nn.1.

Il limite di 500 alberi generati è sufficientemente elevato. Dal grafico di Figura 3.74 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 200 alberi.

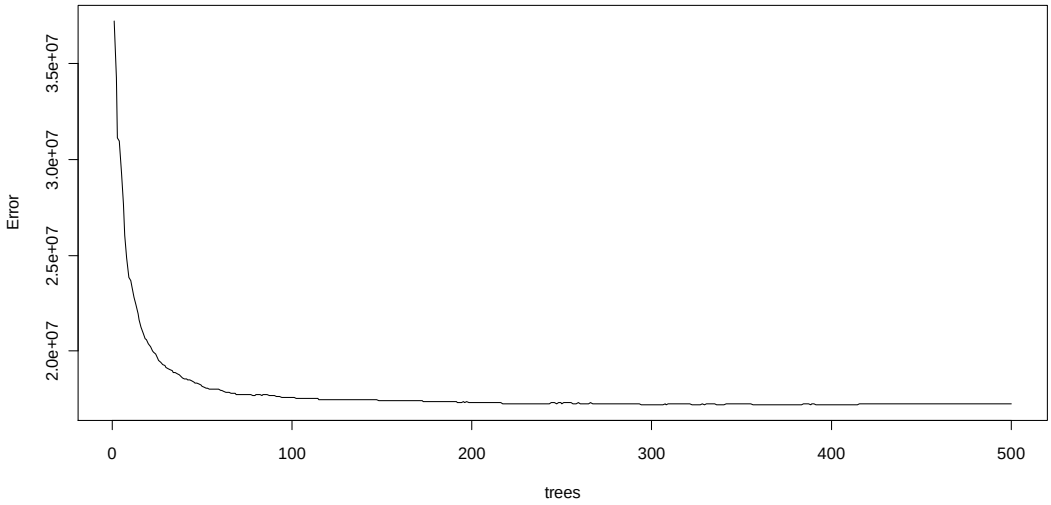


Figura 3.74 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.1.*

Dai grafici di Figura 3.75 si evince che i predittori più importanti sono la quantità di acqua erogata 7 giorni prima (`erogato7`) e quella del giorno precedente (`erogato1`)ed in misura minora anche le variabili `erogato2` ed `erogato6`.

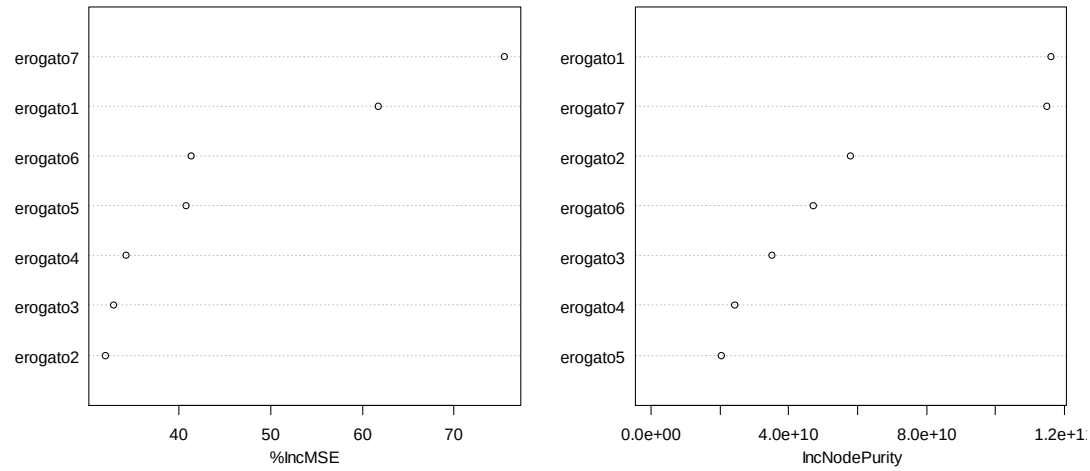


Figura 3.75 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.1.*

I grafici di Figura 3.76 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

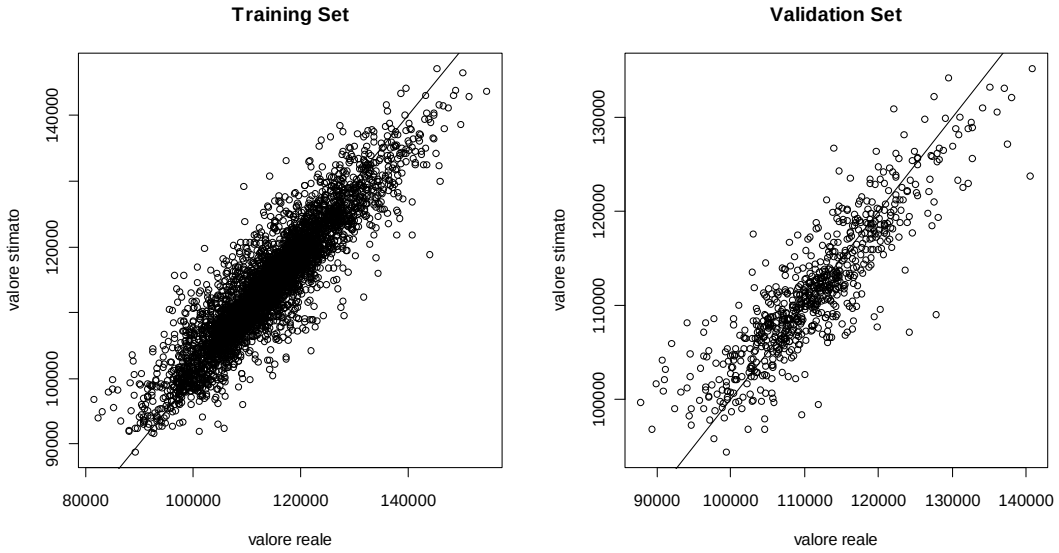


Figura 3.76 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.1.*

Nella Figura 3.77 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il

validation set: il 67% degli errori è inferiore al 3% e l'85% degli errori è inferiore al 5%, mentre per il modello nn.1 i corrispondenti valori erano 68% e 87%.

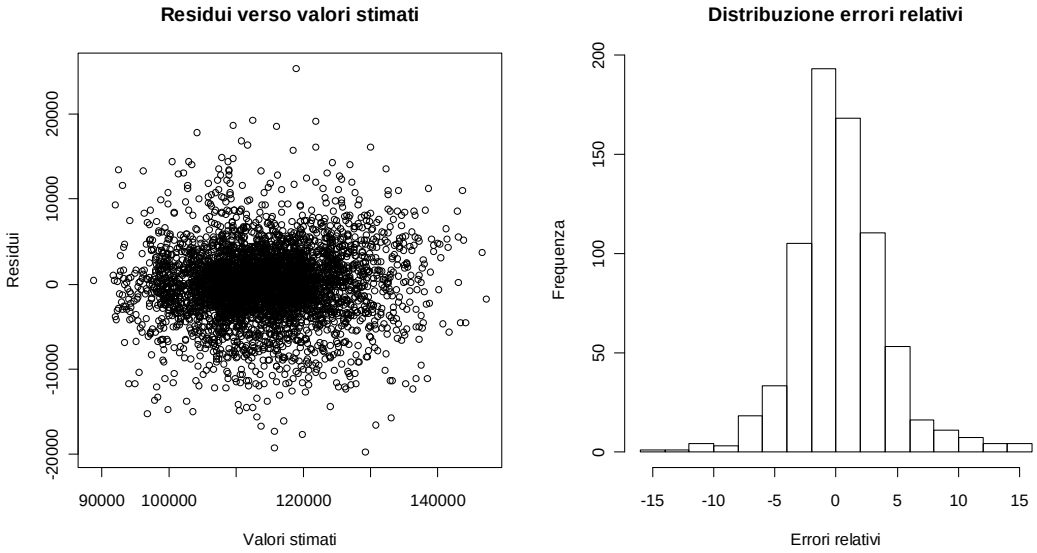


Figura 3.77 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.1.*

3.6.2 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e la temperatura (rf.2)

In aggiunta al modello rf.1 viene introdotta la variabile `temperatura`; i predittori sono corrispondenti a quelli del modello nn.2. La percentuale di varianza spiegata sale da 82,15% a 83,19%, mentre lo scarto quadratico medio degli errori relativi calcolati sul *validation set* rimane sostanzialmente invariato: 3,67 contro 3,74 del modello rf.1.
Anche in questo caso il numero di alberi generati è sufficiente: in Figura 3.78 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

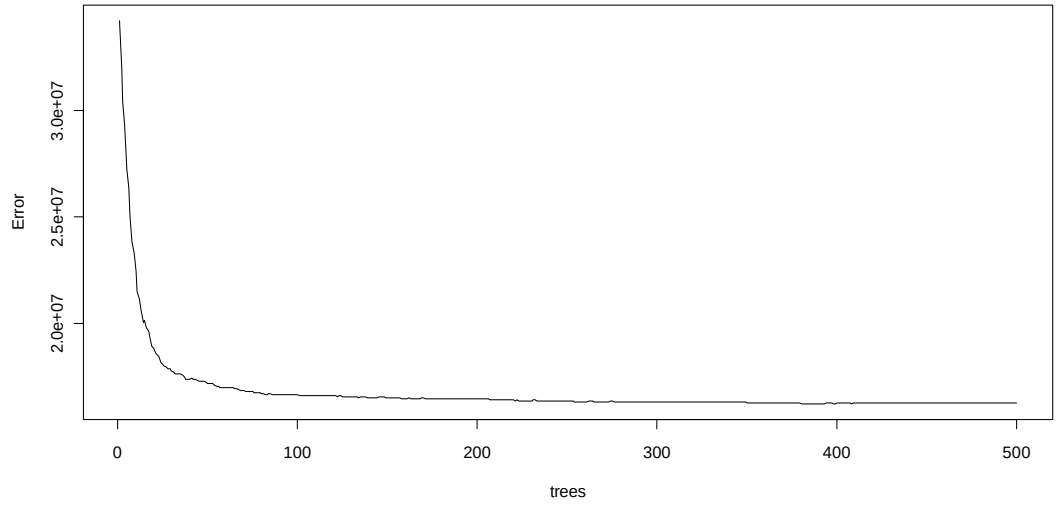


Figura 3.78 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.2.*

Relativamente all'importanza dei predittori, i grafici di Figura 3.83 confermano come variabili più significative `erogato7`, `erogato1` ed in misura minora anche le variabili `erogato2`, `erogato6` e `temperatura`.

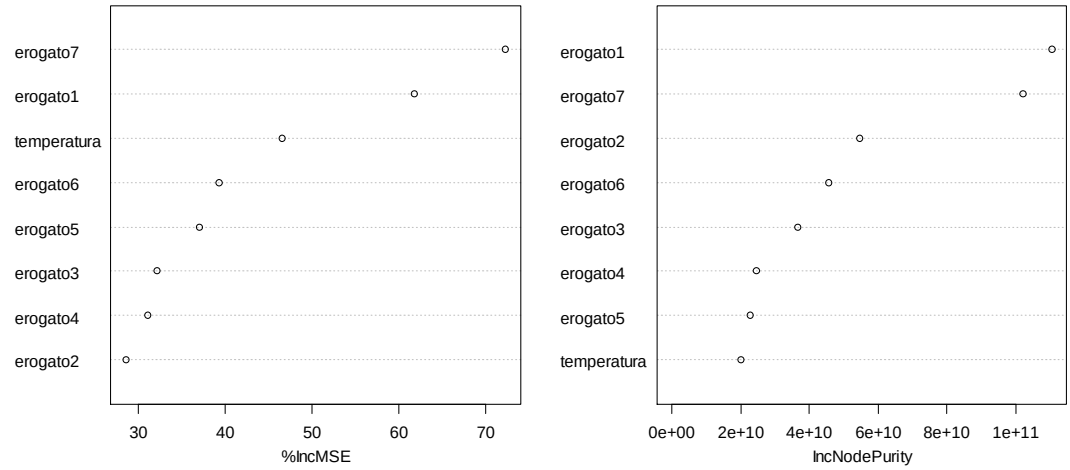


Figura 3.79 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.2.*

I grafici di Figura 3.80 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

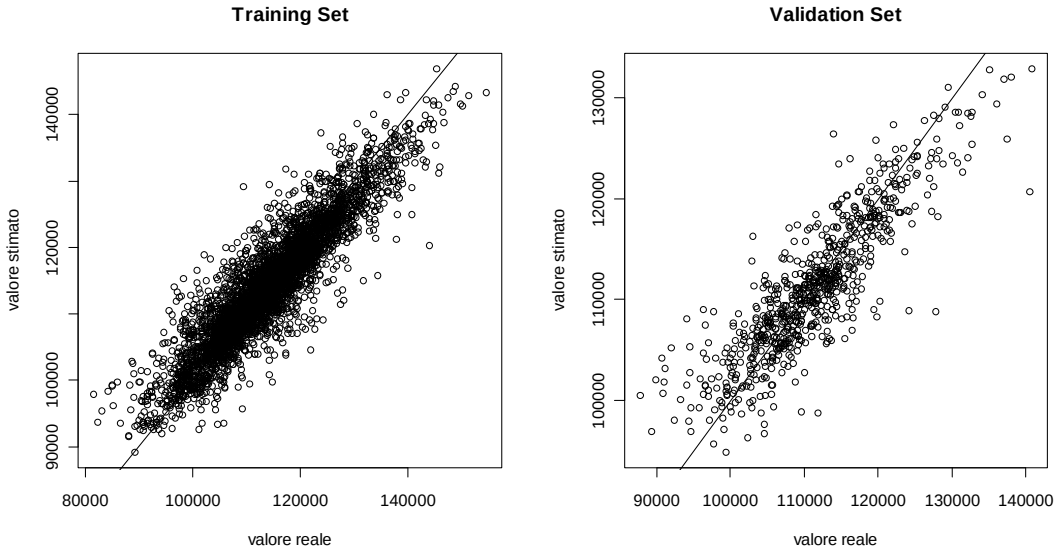


Figura 3.80 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.2.*

Nella Figura 3.81 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 65% degli errori è inferiore al 3% e il 86% degli errori è inferiore al 5%, mentre per il modello nn.2 i corrispondenti valori erano 67% e 87%.

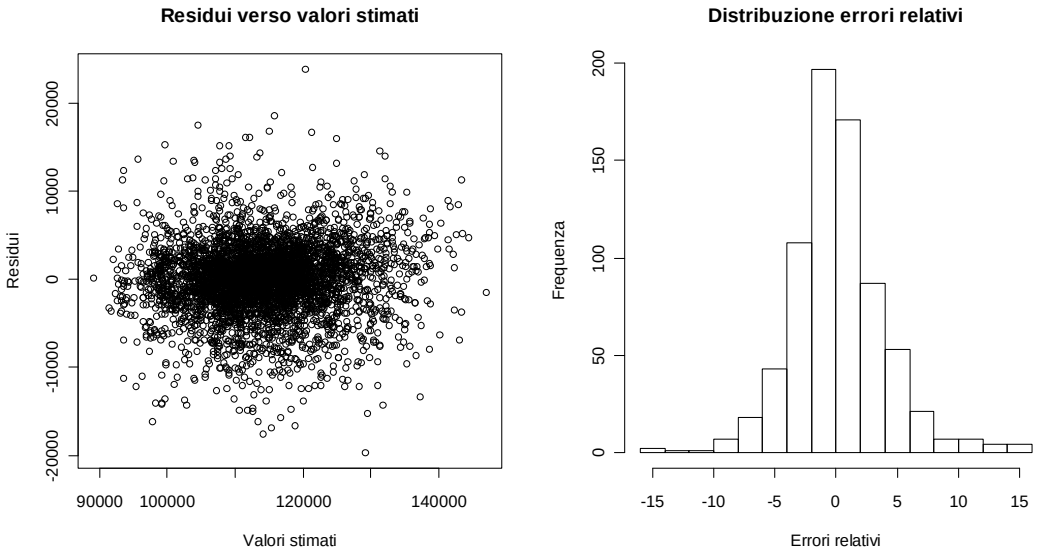


Figura 3.81 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.2.*

3.6.3 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale (rf.3)

Rispetto al precedente modello rf.2 viene inserito come predittore anche il coefficiente settimanale, rappresentato dalla variabile `rapp_settimana`; i predittori sono corrispondenti a quelli del modello nn.3. Non è evidente nessun miglioramento rispetto al modello precedente: la percentuale di varianza spiegata risulta l'82,6% contro 83,19% del modello rf.2; lo scarto quadratico medio degli errori relativi calcolati sul *validation set* rimane sostanzialmente invariato: 3,72 per il modello rf.3 e 3,67 per il modello rf.2.
Anche in questo caso il numero di alberi generati è sufficiente: in Figura 3.82 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

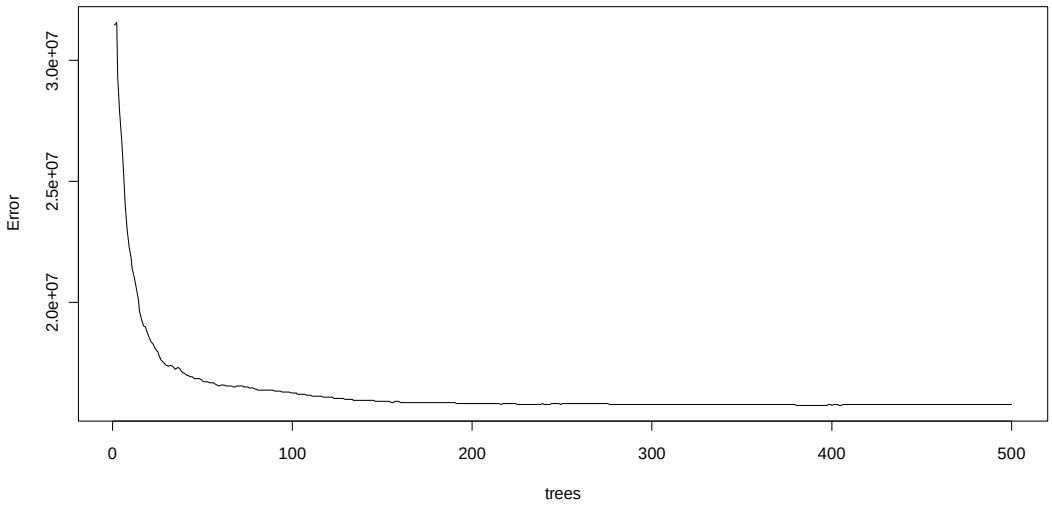


Figura 3.82 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.3.*

Relativamente all'importanza dei predittori, i grafici di Figura 3.83 confermano come variabili più significative `erogato7`, `erogato1` ed in misura minore anche le variabili `erogato2`, `erogato6` e `temperatura`.

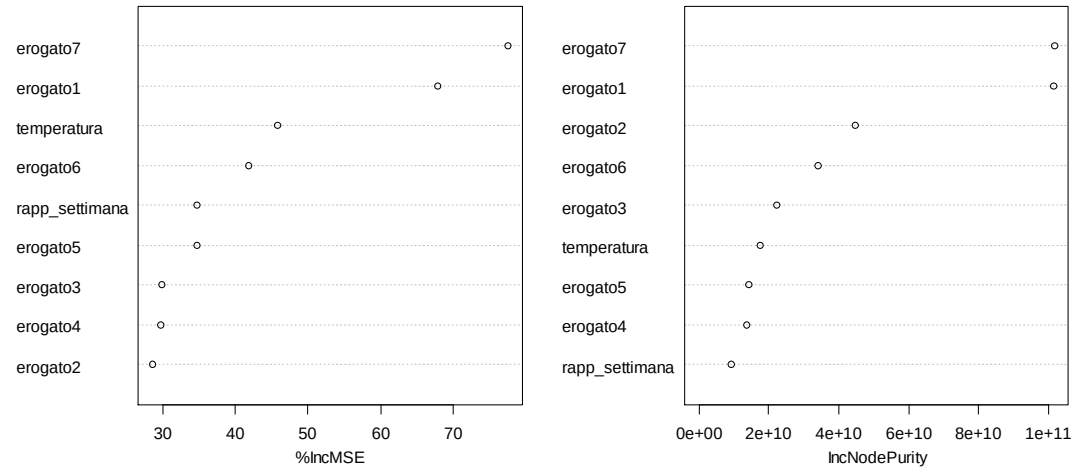


Figura 3.83 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.3.*

I grafici di Figura 3.84 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

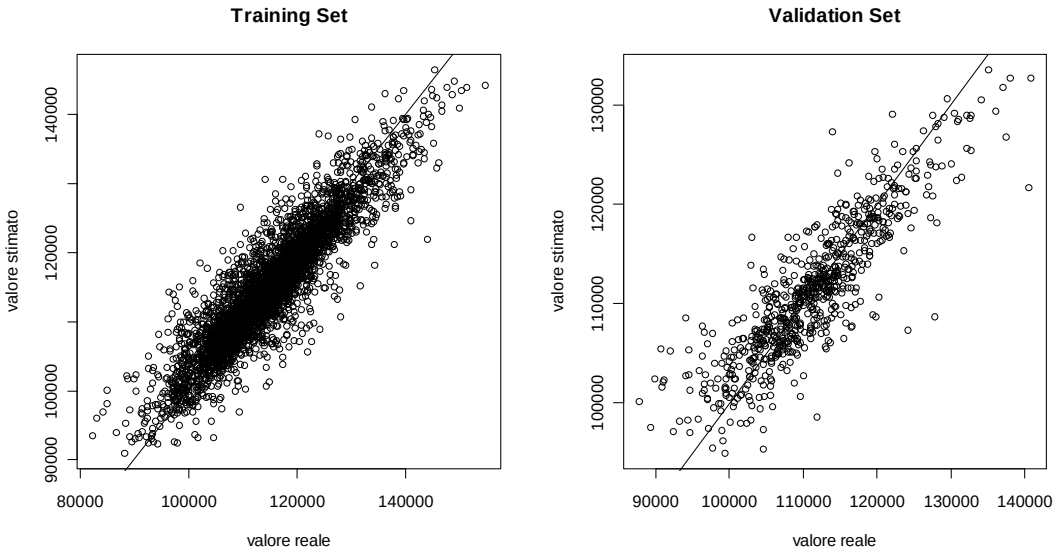


Figura 3.84 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.3.*

Nella Figura 3.85 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 66% degli errori è inferiore al 3% e l'86% degli errori è inferiore al 5%, mentre per il modello nn.3 i corrispondenti valori erano 65% e 86%.

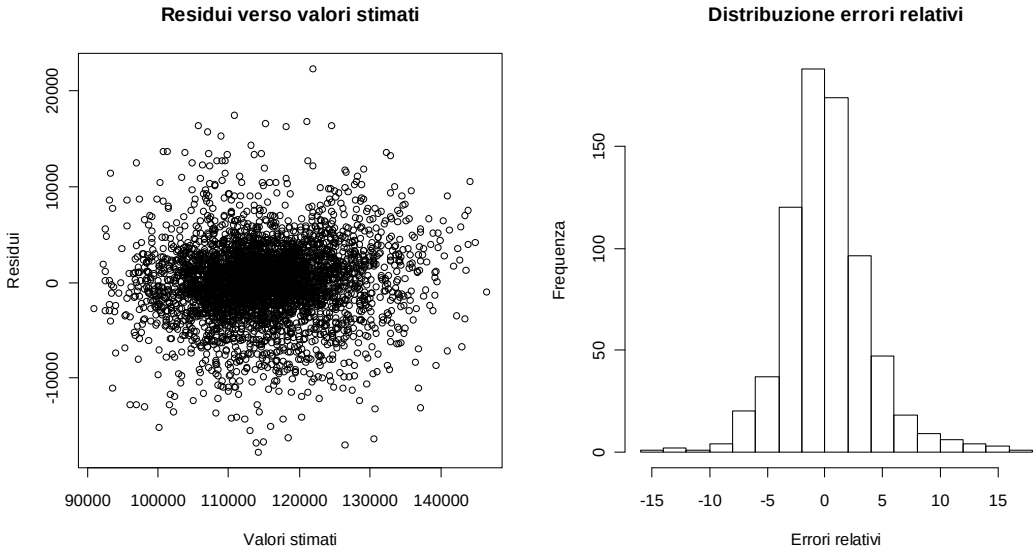


Figura 3.85 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.3.*

3.6.4 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il giorno dell'anno (rf.4)

Rispetto al precedente modello rf.3 viene aggiunto come predittore il giorno dell'anno, con valori da 1 a 366 (anni bisestili), identificato dalla variabile `giorno_anno_n`, ed escluso il coefficiente settimanale `rapp_settimana` risultato scarsamente informativo: l'idea è che l'andamento stagionale annuo che si è tentato di modellare con il coefficiente settimanale possa venire sostituito dalle opportune suddivisioni della variabile `giorno_anno_n` operate automaticamente dall'algoritmo *Random Forests*. I risultati ottenuti dal modello rf.4 così costruito sono leggermente migliori di quelli del modello rf.3: la percentuale di varianza spiegata è pari rispettivamente a 83,98% e 82,60%, lo scarto quadratico medio dei residui degli errori relativi calcolati sul *validation set* è pari rispettivamente a 3,64 e 3,72.

Anche per questo modello il numero di alberi generati è adeguato: in Figura 3.86 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

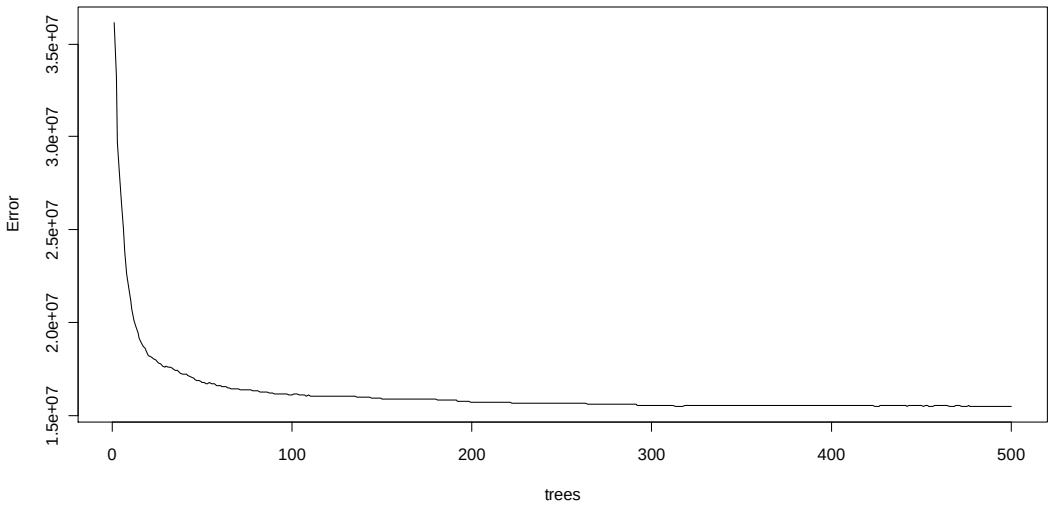


Figura 3.86 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.4.*

Relativamente all'importanza dei predittori, i grafici di Figura 3.87 confermano come variabili più significative `erogato7`, `erogato1` ed in misura minora anche le variabili `erogato2`, `erogato6` e `temperatura`. Il nuovo predittore `giorno_anno_n` risulta scarsamente esplicativo.

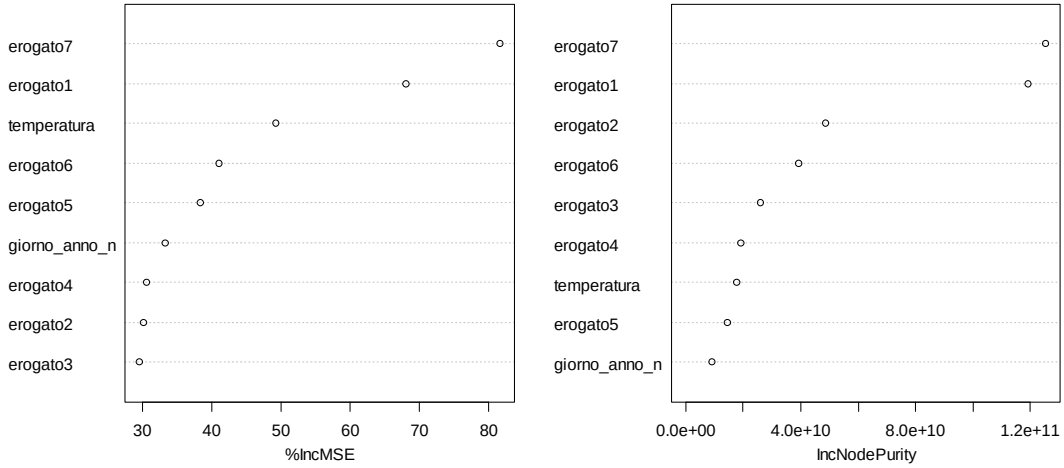


Figura 3.87 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.4.*

I grafici di Figura 3.87 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

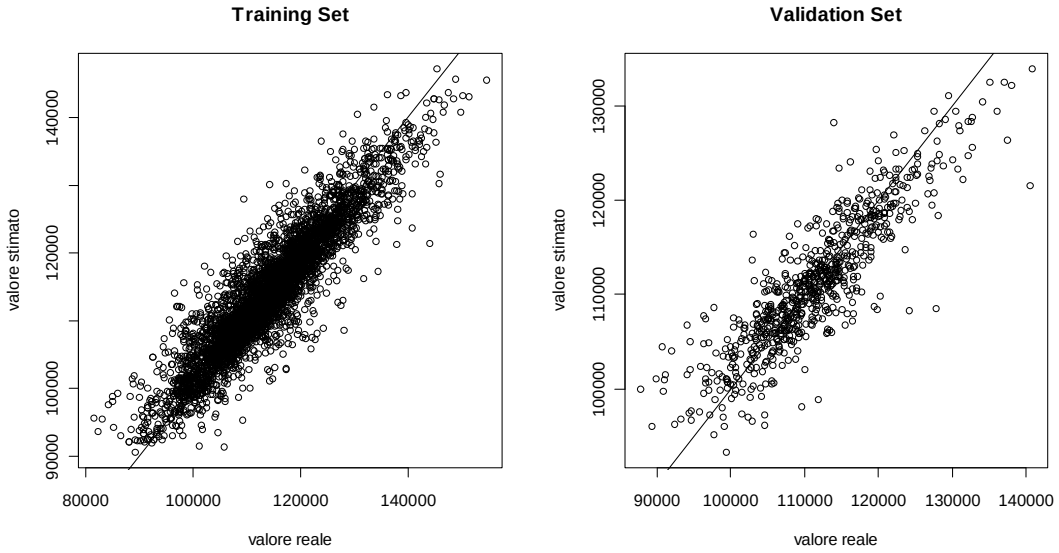


Figura 3.88 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.4.*

Nella Figura 3.89 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 68% degli errori è inferiore al 3% e l'87% degli errori è inferiore al 5%, mentre per il modello rf.3 i corrispondenti valori erano 66% e 86% che confermano un certo miglioramento nell'adattamento.

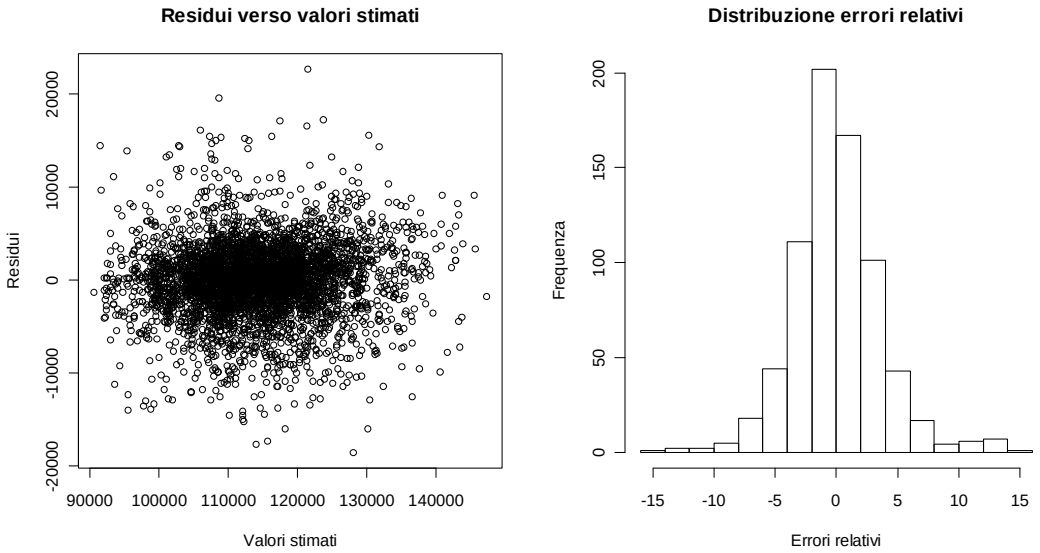


Figura 3.89 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.4.*

3.6.5 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il giorno dell'anno e il giorno festivo (rf.5)

Rispetto al precedente modello rf.4 viene aggiunto come predittore la variabile dicotomica `festivo` che identifica appunto il giorno di previsione come festivo o meno (vedi §3.2.2.1). Il modello risulta migliore del precedente: l'introduzione del nuovo predittore ha fatto salire la varianza spiegata a 85,44% dal precedente 83,98%. Anche lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* è sensibilmente diminuito: da 3,64 del modello rf.4 a 3,44 di quello attuale.
Anche per questo modello il numero di alberi generati è adeguato: in Figura 3.90 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

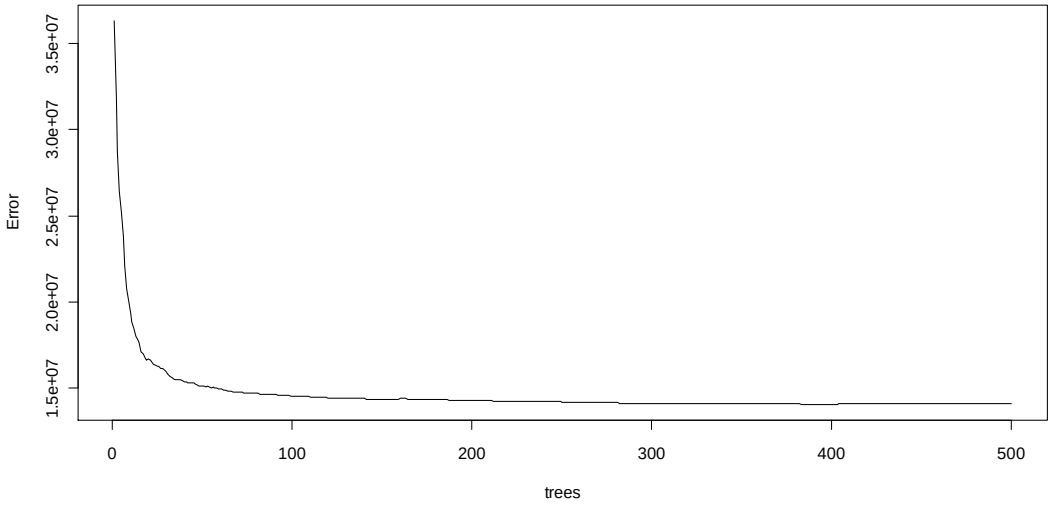


Figura 3.90 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.5.*

Relativamente all'importanza dei predittori, i grafici di Figura 3.91 confermano come variabili più significative `erogato7`, `erogato1` e del nuovo predittore `festivo`, in misura minora anche le variabili `erogato2`, `erogato6` e `temperatura`.

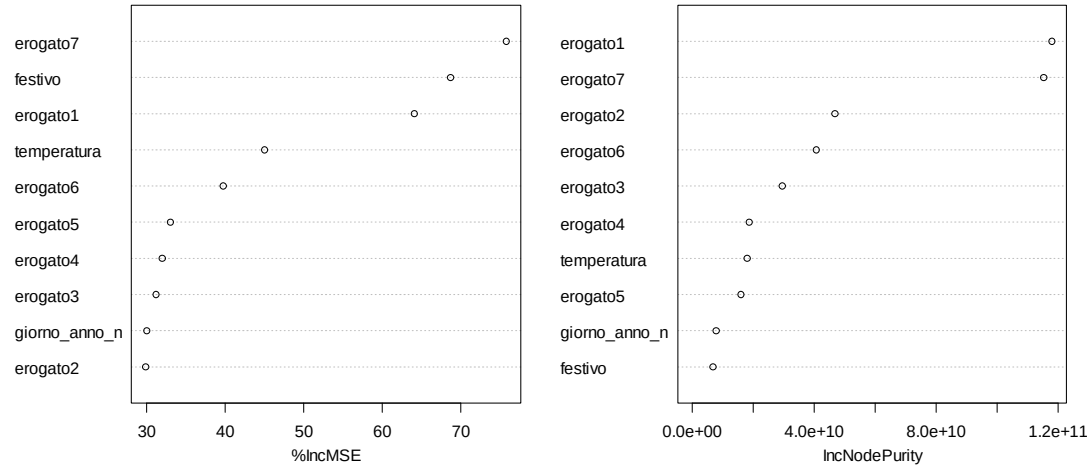


Figura 3.91 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.5.*

I grafici di Figura 3.92 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

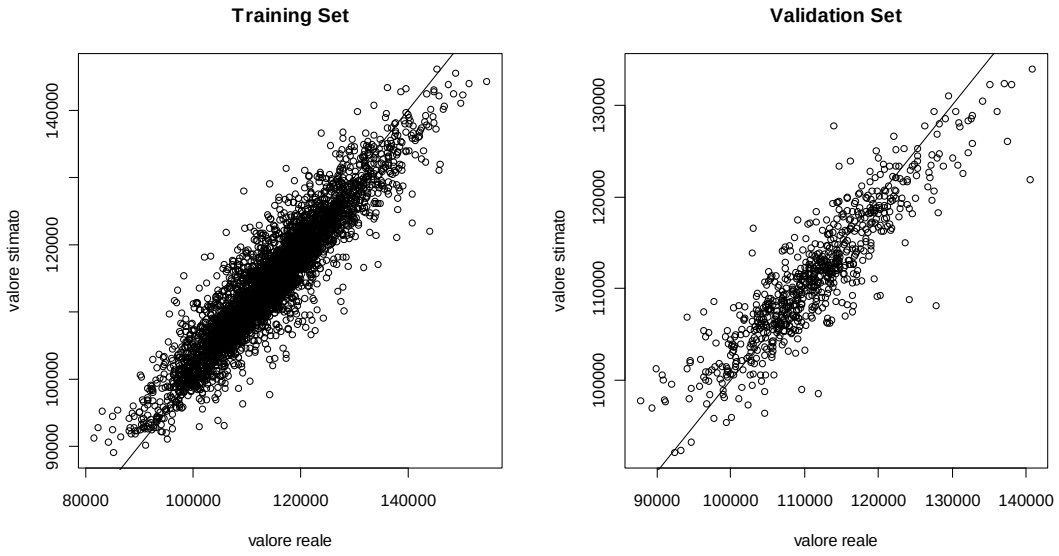


Figura 3.92 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.5.*

Nella Figura 3.93 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 70% degli errori è inferiore al 3% e l'88% degli errori è inferiore al 5%, mentre per il modello rf.4 i corrispondenti valori erano 68% e 87% che confermano un certo miglioramento nell'adattamento.

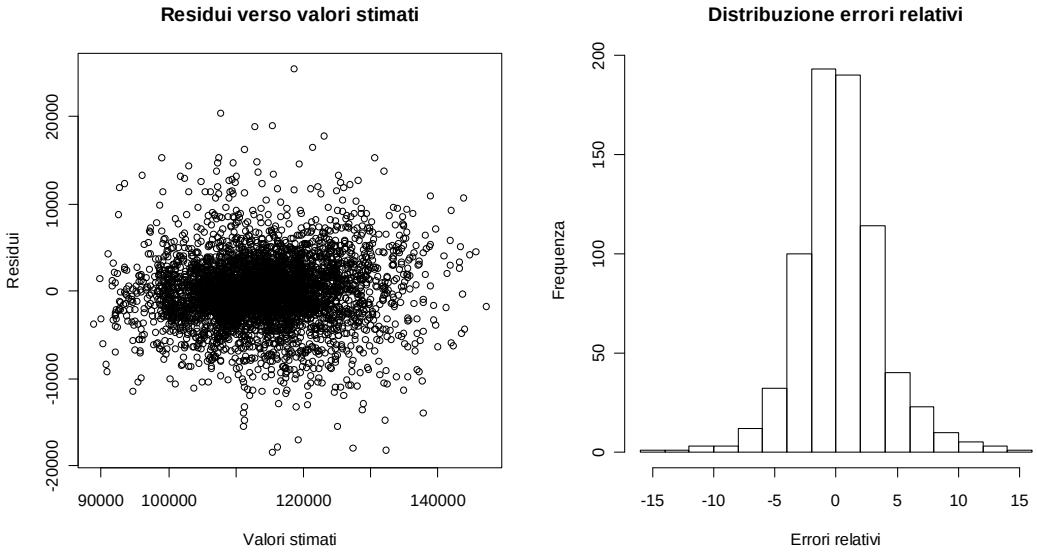


Figura 3.93 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.5.*

3.6.6 Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il giorno dell'anno, il giorno festivo e il giorno della settimana (rf.6)

Rispetto al precedente modello rf.5 viene aggiunto come predittore la variabile qualitativa `giorno_settimana`, fattore a 7 livelli che identifica appunto il giorno della settimana. Il modello risulta migliore del precedente: l'introduzione del nuovo predittore ha fatto salire la varianza spiegata a 89,19% dal precedente 85,44%.

Anche lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* è sensibilmente diminuito: da 3,43 del modello rf.5 a 3,00 di quello attuale.

Anche per questo modello il numero di alberi generati è adeguato: in Figura 3.94 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 250 alberi.

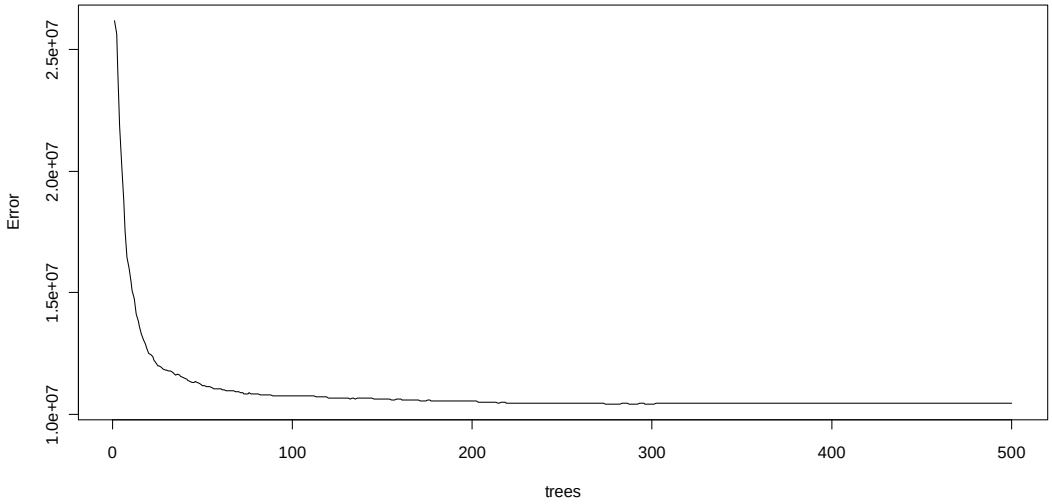


Figura 3.94 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.6.*

Relativamente all'importanza dei predittori, i grafici di Figura 3.95 evidenziano il nuovo predittore `giorno_settimana` tra le variabili più significative, assieme a `festivo`, `erogato7` ed `erogato1`. In misura minora lo sono le variabili `erogato2`, `erogato6` e `temperatura`.

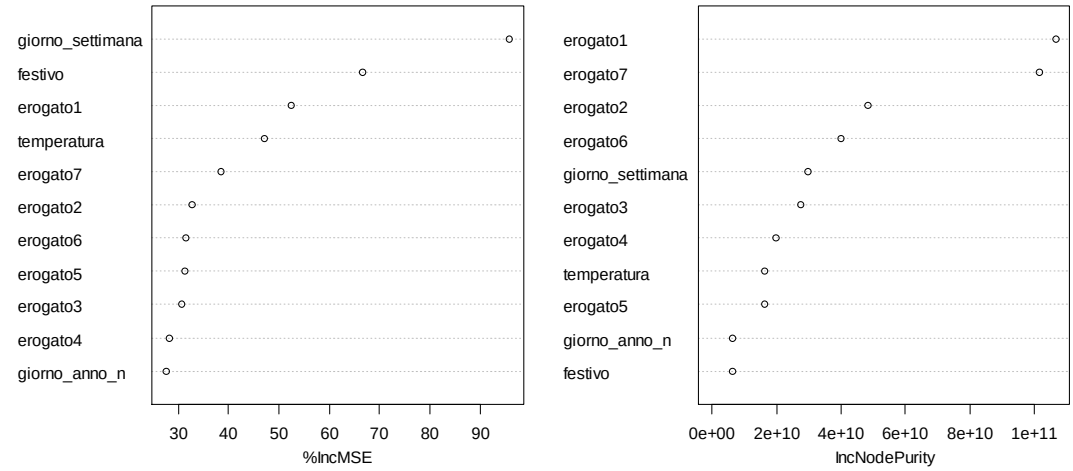


Figura 3.95 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.6.*

I grafici di Figura 3.96 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

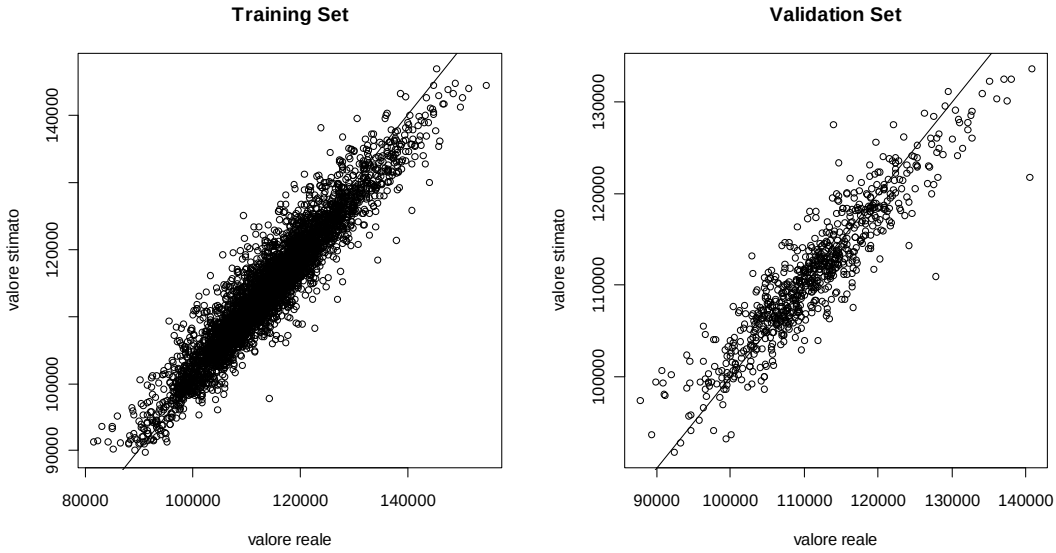


Figura 3.96 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.6.*

Nella Figura 3.97 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 75% degli errori è inferiore al 3% e il 91% degli errori è inferiore al 5%, mentre per il modello rf.5 i corrispondenti valori erano 70% e 88% che confermano il miglioramento nell'adattamento.

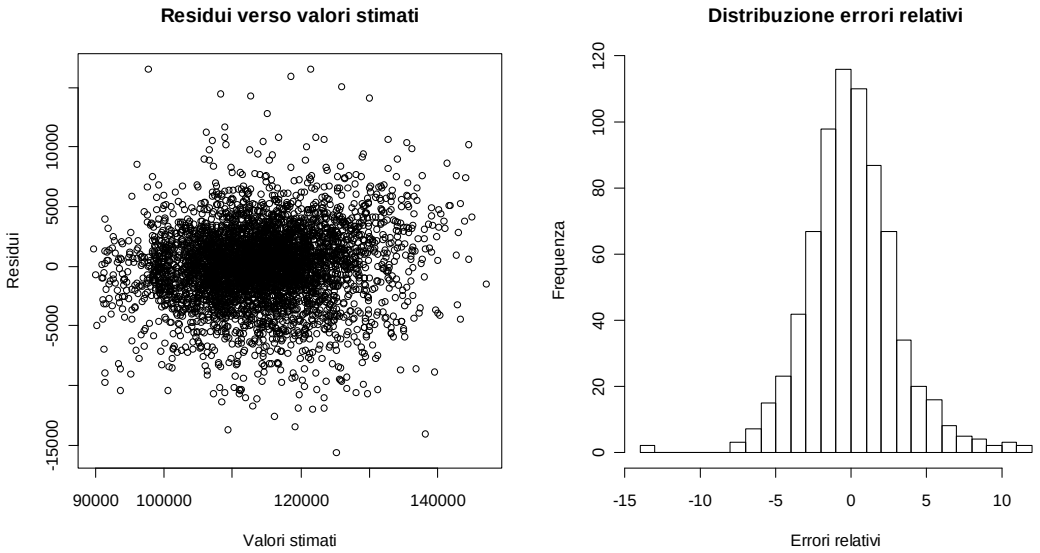


Figura 3.97 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.6.*

3.6.7 Previsione settimanale

Come nel caso dei modelli basati su reti neurali, anche per modelli basati su *Random Forests* viene costruito un modello di previsione settimanale del quantitativo di acqua erogato.

Il primo passaggio è stato la realizzazione di modelli di previsione di stima non solo del giorno seguente, ma anche dei giorni successivi. Mantenendo la variabile risposta `erogato0` (riferita al tempo t), per stimare il consumo del giorno successivo si usano come predittori i valori `erogato1` (tempo $t-1$), `erogato2` (tempo $t-2$),…, `erogato7` (tempo $t-7$), esattamente quello che è stato fatto nei modelli precedenti. Per stimare il consumo che si realizzerà tra due giorni è utile continuare a considerare questo rappresentato dalla variabile `erogato0`, ponendosi quindi al tempo $t-2$. In questo caso non è possibile considerare come predittore la variabile `erogato1` (tempo $t-1$) in quanto non ancora nota. Si possono invece utilizzare i predittori `erogaro2`, `erogato3`,…, `erogato7`. In aggiunta è stata costruita la variabile `erogato8` (consumo al tempo $t-8$) in modo da avere a comunque disposizione gli ultimi 7 valori di consumo realizzati in un certo momento. Per costruire tale valore si è dovuto rinunciare alla possibilità di utilizzare la prima riga del *dataset* in quanto per tale unità il valore è non noto.

In modo analogo posso costruire i modelli di previsione per i consumi a 3,4,…,7 giorni, escludendo di volta in volta anche le variabili `erogato2`, `erogato3`,…, `erogato6` e inserendo al loro posto le variabili `erogato9`, `erogato10`,…, `erogato13` opportunamente costruite, perdendo rispettivamente 2,3,…, 6 unità del *dataset*.

Oltre ai predittori ora descritti, vengono considerati anche i predittori utilizzati nel modello rf.6, cioè `temperatura`, `giorno_anno_n`, `festivo` e `giorno_settimana`.

3.6.7.1 Modello di previsione a 1 giorno (rf.S0)

Il modello di previsione per il giorno seguente è, di fatto, equivalente al modello rf.6. La variabile risposta è `erogato0` e i predittori sono `erogato1`,…,`erogato7`, oltre a `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Il *data frame* utilizza tutte le osservazioni.

La varianza spiegata è pari al 89,19% e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* è pari a 3,00.

Il numero di alberi generati è adeguato: in Figura 3.98 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 250 alberi.

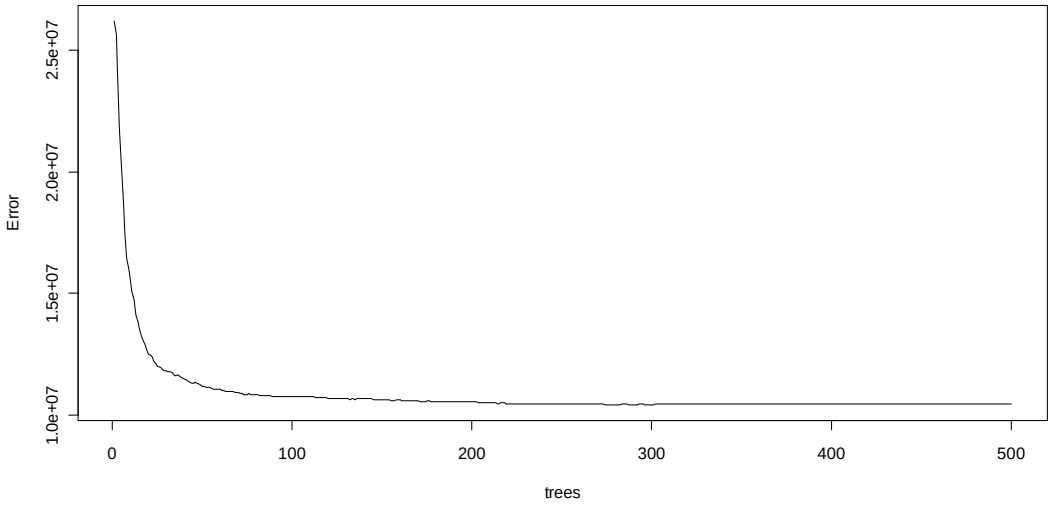


Figura 3.98 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S0.*

Nei grafici di Figura 3.99 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `giorno_settimana`, `festivo`, `erogato1` ed `erogato7`, in misura minore le variabili `erogato2`, `erogato6` e `temperatura`.

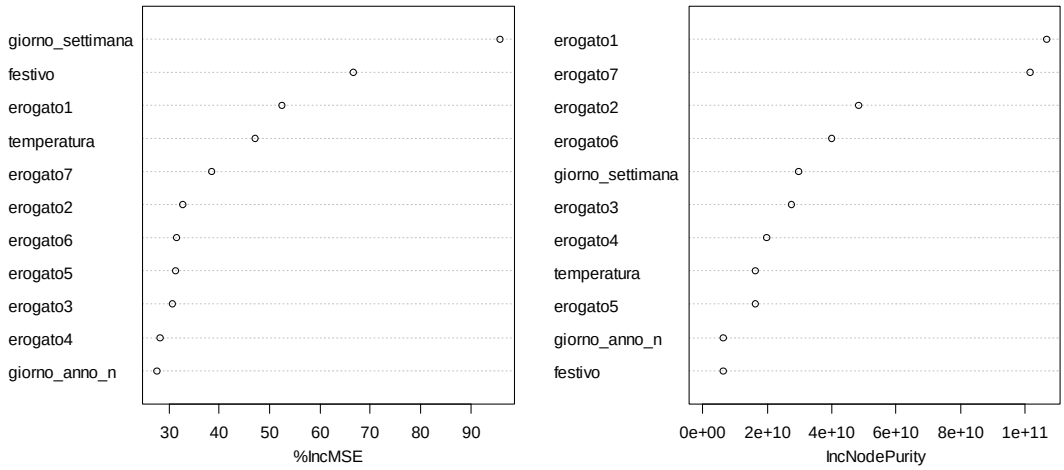


Figura 3.99 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S0.*

I grafici di Figura 3.100 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

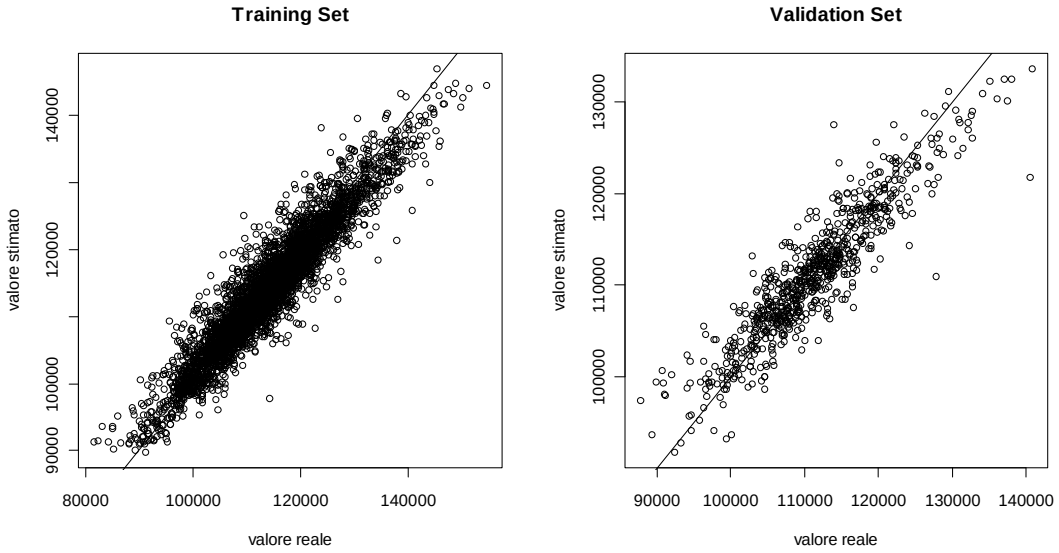


Figura 3.100 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S0.*

Nella Figura 3.101 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 75% degli errori è inferiore al 3% e il 91% degli errori è inferiore al 5%.

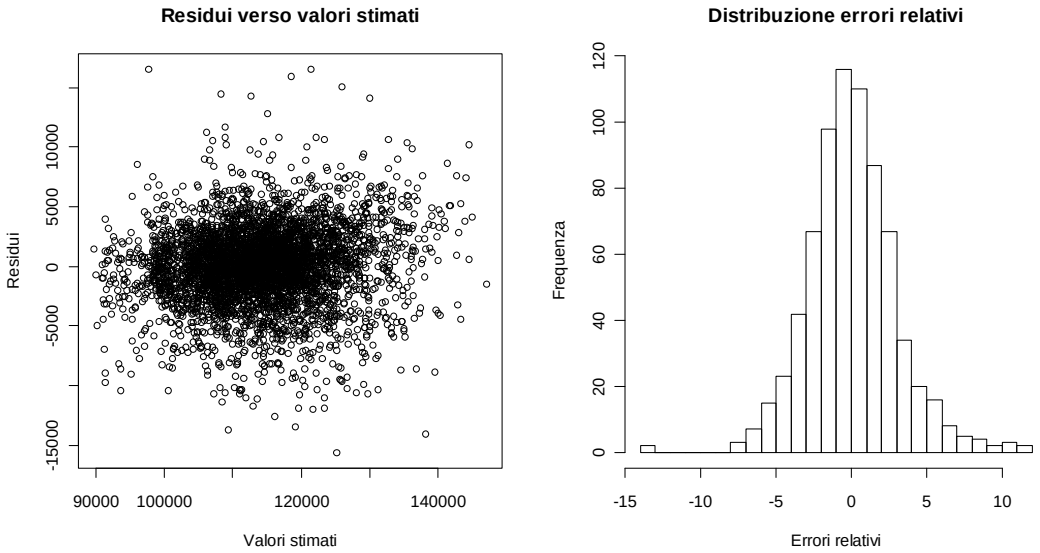


Figura 3.101 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S0.*

3.6.7.2 Modello di previsione a 2 giorni (rf.S1)

Per la stima del volume erogato il giorno $t+2$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0`. I predittori sono invece

`erogato2`,…,`erogato8`, quindi si considerano i valori disponibili fino a due giorni antecedenti. Gli altri predittori sono `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Viene esclusa la prima osservazione del *data frame* in quanto non dispone del valore sulla variabile `erogato8`.

La varianza spiegata scende all'85,47% dall'89,19% del modello rf.S0 e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* sale a 3,46 rispetto al valore 3,00 del modello rf.S0.

Il numero di alberi generati è adeguato: in Figura 3.102 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

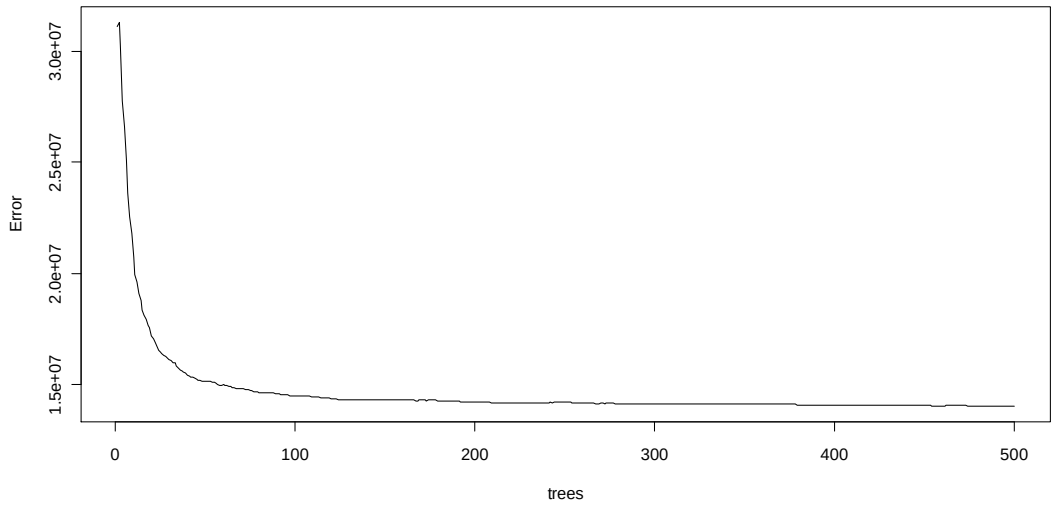


Figura 3.102 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S1.*

Nei grafici di Figura 3.103 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `giorno_settimana`, `festivo`, `erogato7` ed `erogato2`, in misura minore le variabili `erogato6` e `temperatura`.

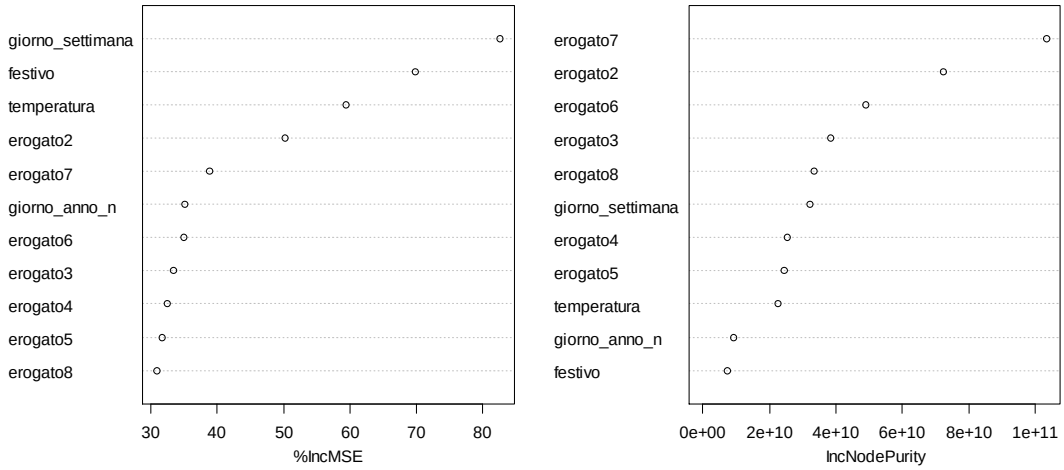


Figura 3.103 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S1.*

I grafici di Figura 3.104 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

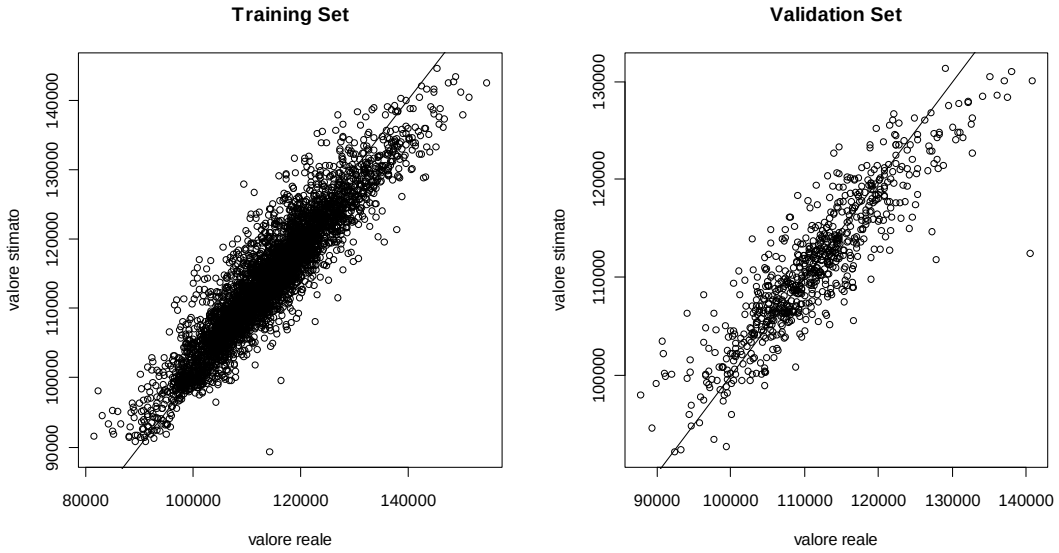


Figura 3.104 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S1.*

Nella Figura 3.105 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 69% degli errori è inferiore al 3% e il 87% degli errori è inferiore al 5%, mentre per il modello rf.S0 i corrispondenti valori erano 75% e 91%.

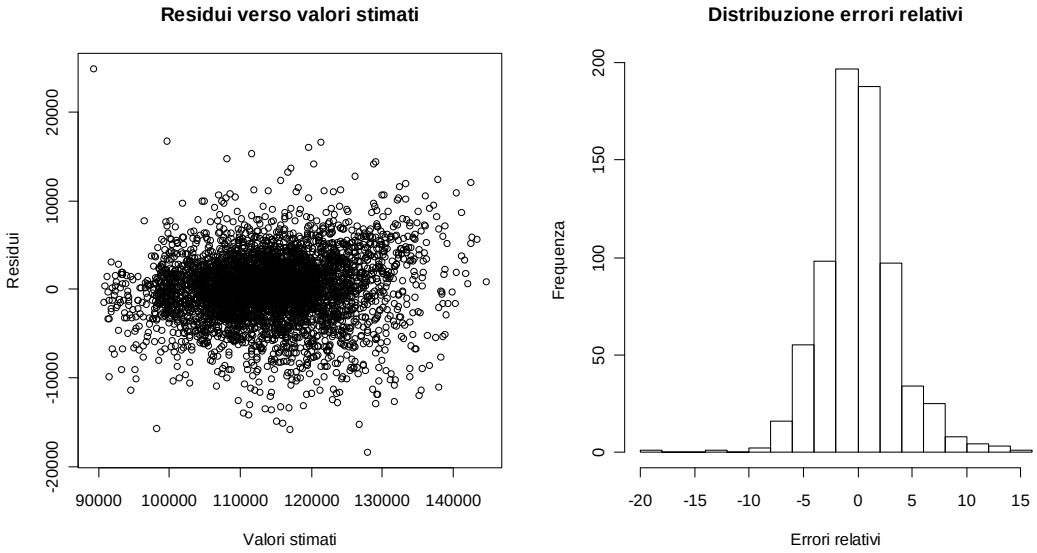


Figura 3.105 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S1.*

3.6.7.3 Modello di previsione a 3 giorni (rf.S2)

Per la stima del volume erogato il giorno $t+3$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0`. I predittori sono invece `erogato3`,…,`erogato9`, quindi si considerano i valori disponibili fino a 3 giorni antecedenti. Gli altri predittori sono `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Vengono escluse le prime 2 osservazione del *data frame* in quanto non dispongono del valore sulla variabile `erogato9`.

La varianza spiegata scende all'83,01% dall'85,47% del modello rf.S1 e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* sale a 3,68 rispetto al valore 3,46 del modello rf.S1.

Il numero di alberi generati è adeguato: in Figura 3.106 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

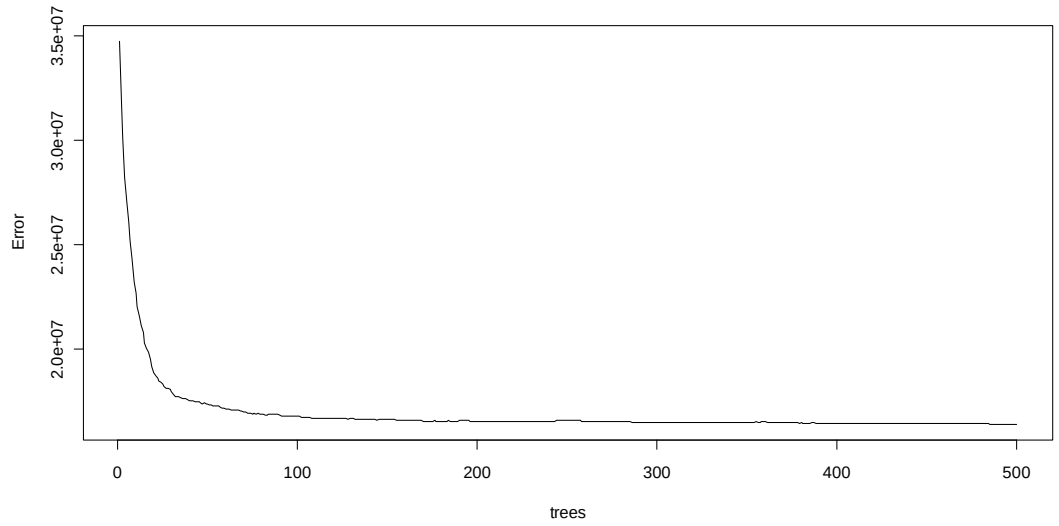


Figura 3.106 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S2.*

Nei grafici di Figura 3.107 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `giorno_settimana`, `festivo`, `erogato7` e `temperatura`, in misura minore le variabili `erogato6` ed `erogato3`.

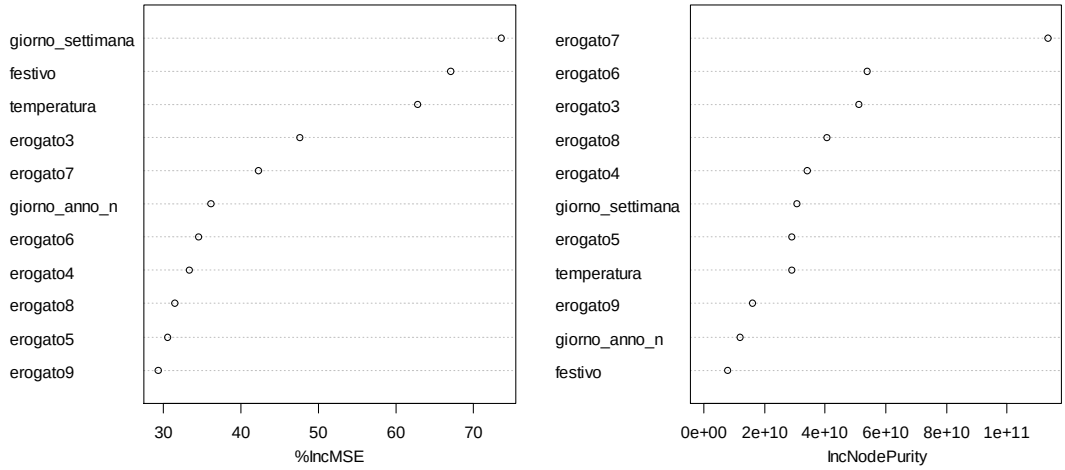


Figura 3.107 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S2.*

I grafici di Figura 3.108 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

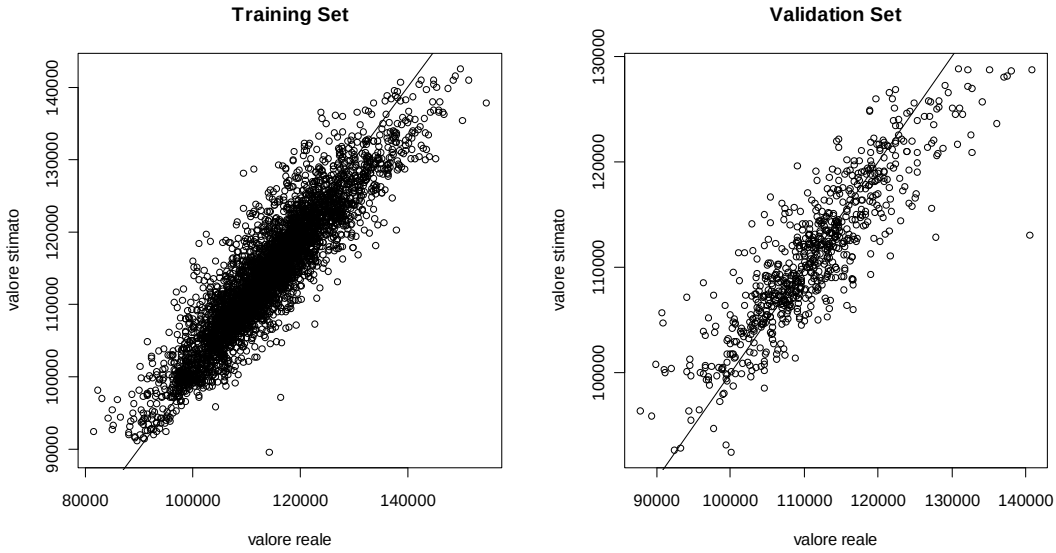


Figura 3.108 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S2.*

Nella Figura 3.109 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 67% degli errori è inferiore al 3% e l'85% degli errori è inferiore al 5%, mentre per il modello rf.S1 i corrispondenti valori erano 69% e 87%.

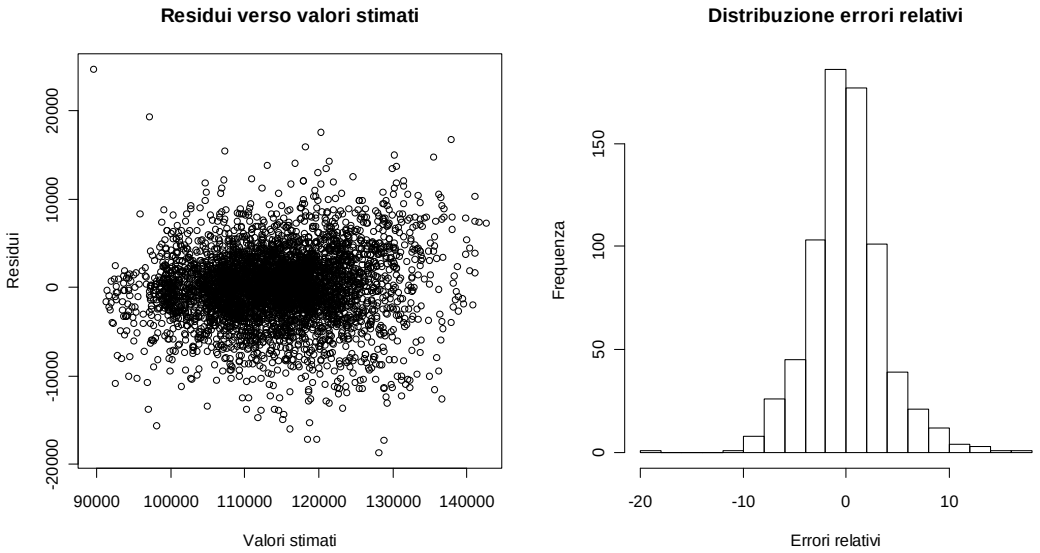


Figura 3.109 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S2.*

3.6.7.4 Modello di previsione a 4 giorni (rf.S3)

Per la stima del volume erogato il giorno $t+4$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0`. I predittori sono invece

`erogato4`,…,`erogato10`, quindi si considerano i valori disponibili fino a 4 giorni antecedenti. Gli altri predittori sono `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Vengono escluse le prime 3 osservazione del *data frame* in quanto non dispongono del valore sulla variabile `erogato10`.
La varianza spiegata scende all'81,19% dall'83,01% del modello rf.S2 e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* sale a 3,80 rispetto al valore 3,68 del modello rf.S2.
Il numero di alberi generati è adeguato: in Figura 3.110 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 350 alberi.

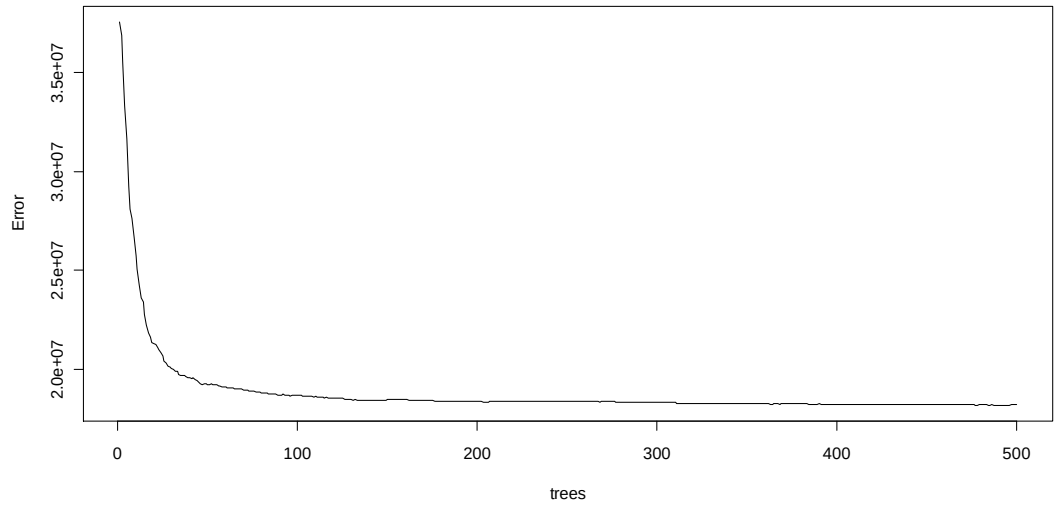


Figura 3.110 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S3.*

Nei grafici di Figura 3.111 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `temperatura`, `festivo`, `giorno_settimana` ed `erogato7`, in misura minore le variabili `erogato6`, `erogato8` ed `erogato4`.

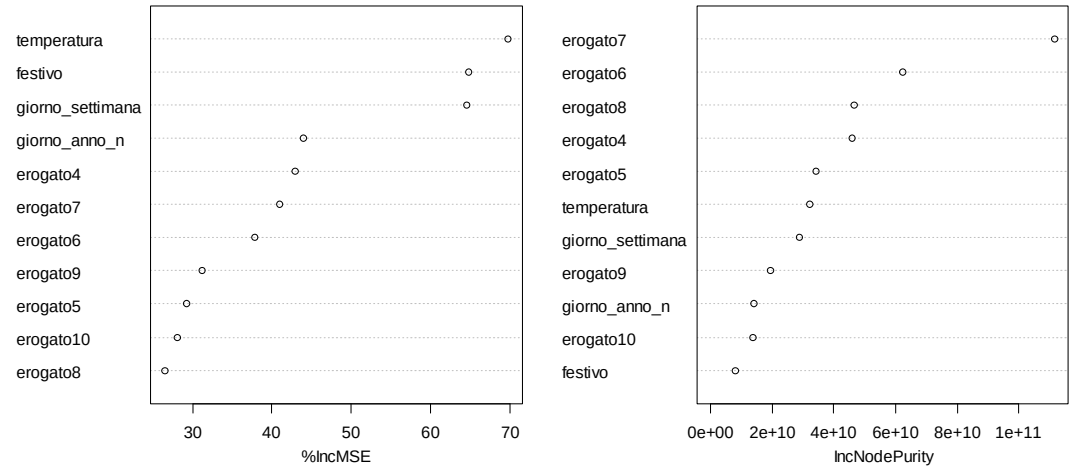


Figura 3.111 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S3.*

I grafici di Figura 3.112 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

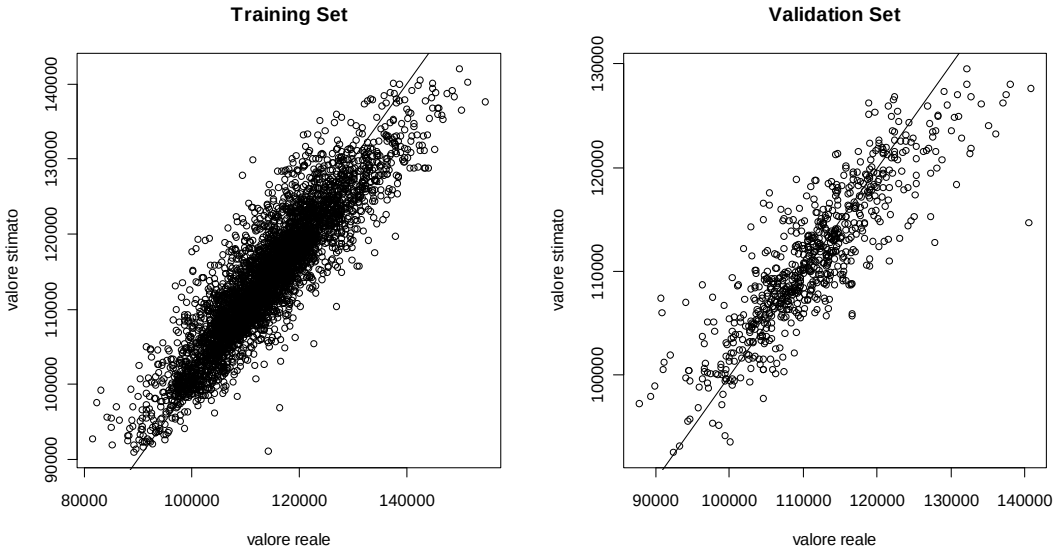


Figura 3.112 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S3.*

Nella Figura 3.113 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 66% degli errori è inferiore al 3% e l'84% degli errori è inferiore al 5%, mentre per il modello rf.S2 i corrispondenti valori erano 67% e 85%.

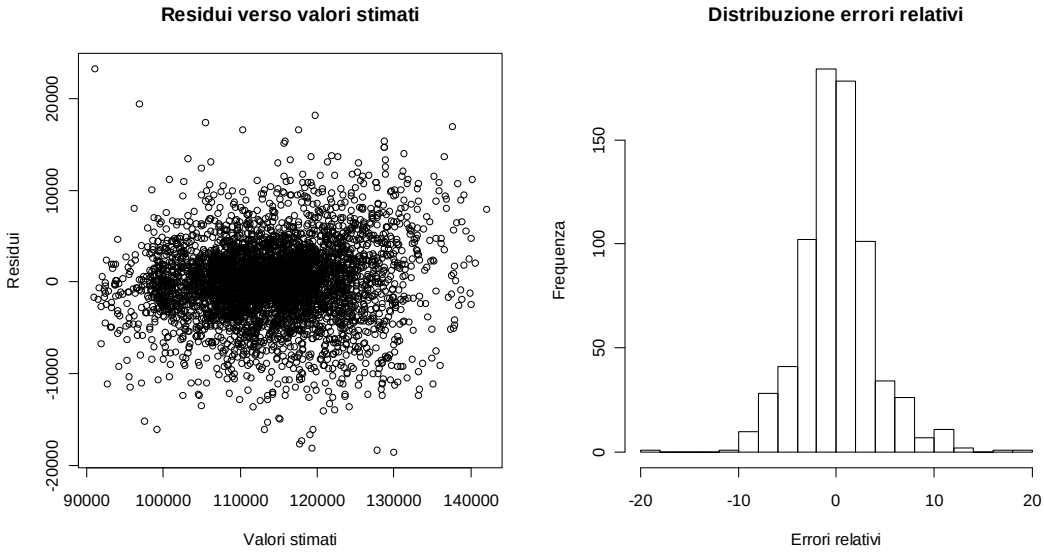


Figura 3.113 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S3.*

3.6.7.5 Modello di previsione a 5 giorni (rf.S4)

Per la stima del volume erogato il giorno $t+5$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0`. I predittori sono invece `erogato5`,…,`erogato11`, quindi si considerano i valori disponibili fino a 5 giorni antecedenti. Gli altri predittori sono `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Vengono escluse le prime 4 osservazione del *data frame* in quanto non dispongono del valore sulla variabile `erogato11`.

La varianza spiegata scende all'80,09% dall'81,19% del modello rf.S3 e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* sale a 3,96 rispetto al valore 3,80 del modello rf.S3.

Il numero di alberi generati è adeguato: in Figura 3.114 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

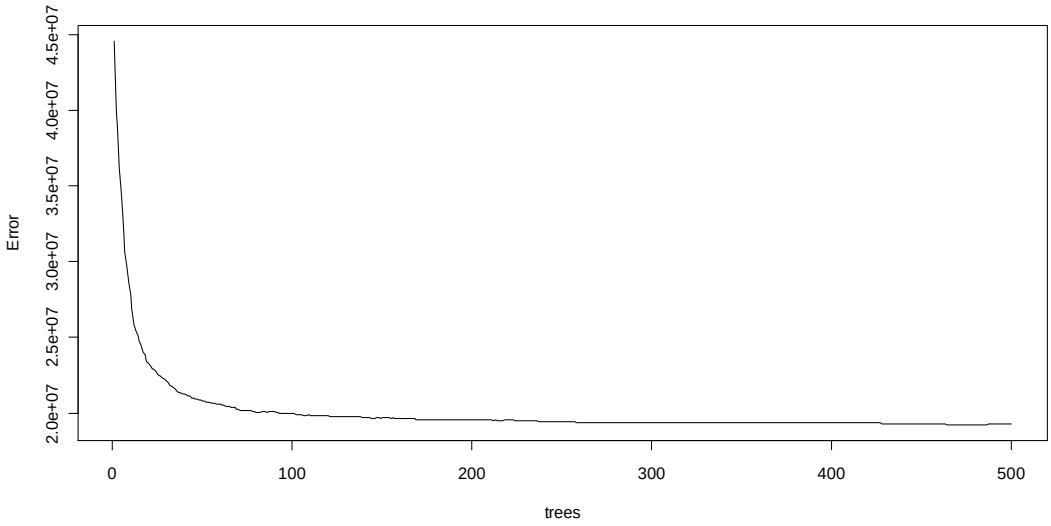


Figura 3.114 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S4.*

Nei grafici di Figura 3.115 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `temperatura`, `festivo`, `giorno_settimana`, `erogato7` ed `erogato6`, in misura minore le variabili `erogato8` ed `erogato5`.

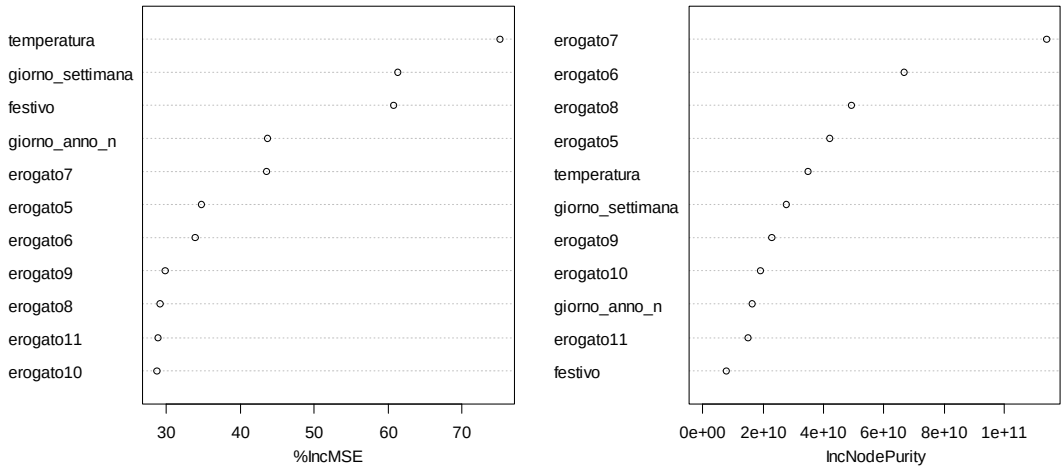


Figura 3.115 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S4.*

I grafici di Figura 3.116 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

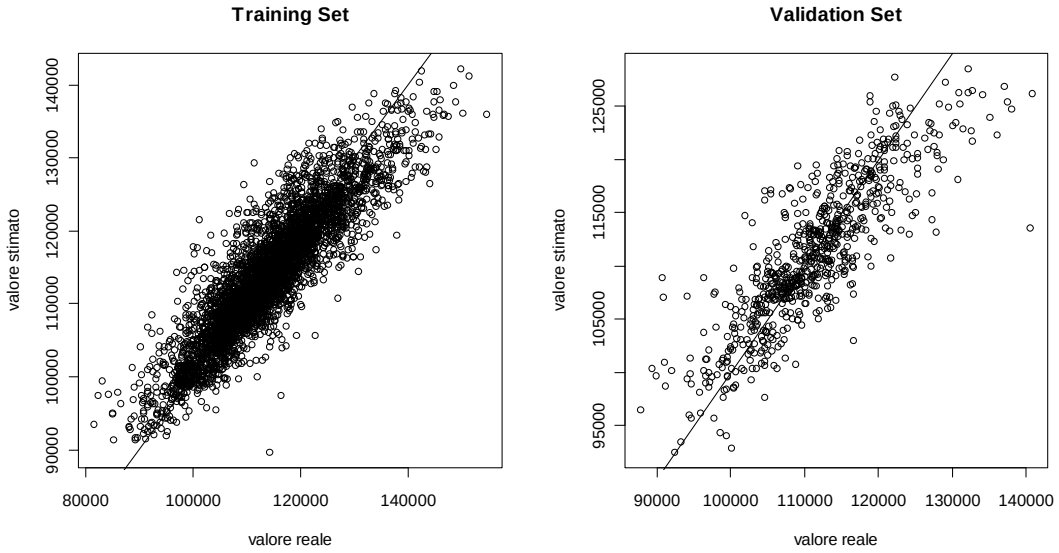


Figura 3.116 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S4.*

Nella Figura 3.117 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 65% degli errori è inferiore al 3% e l'84% degli errori è inferiore al 5%, mentre per il modello rf.S3 i corrispondenti valori erano 66% e 84%.

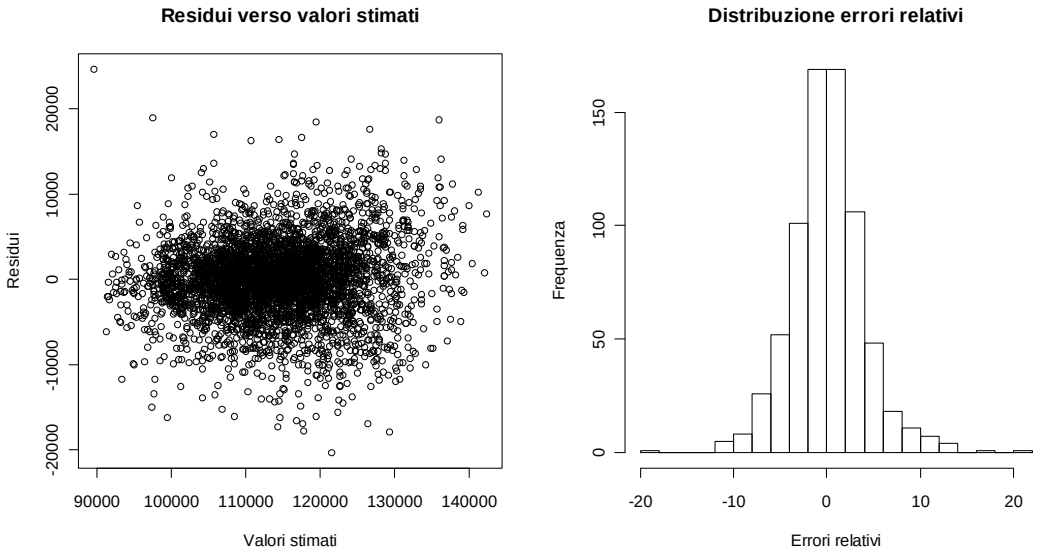


Figura 3.117 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S4.*

3.6.7.6 Modello di previsione a 6 giorni (rf.S5)

Per la stima del volume erogato il giorno $t+6$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0`. I predittori sono invece

`erogato6`,...,`erogato12`, quindi si considerano i valori disponibili fino a 6 giorni antecedenti. Gli altri predittori sono `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Vengono escluse le prime 5 osservazione del *data frame* in quanto non dispongono del valore sulla variabile `erogato12`.
La varianza spiegata scende al 79,29% dall'80,09% del modello rf.S4 e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* sale a 4,05 rispetto al valore 3,96 del modello rf.S4.
Il numero di alberi generati è adeguato: in Figura 3.118 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 300 alberi.

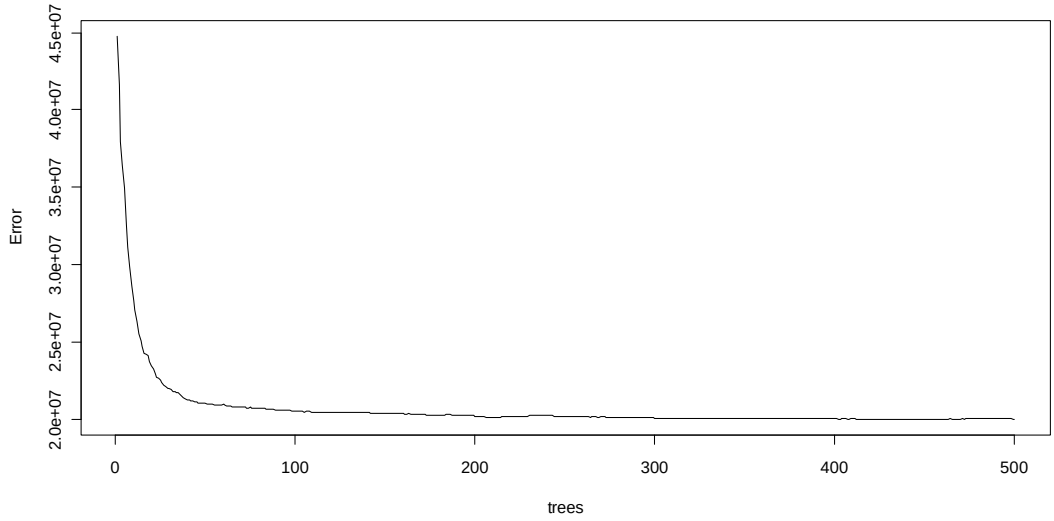


Figura 3.118 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S5.*

Nei grafici di Figura 3.119 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `temperatura`, `festivo`, `giorno_settimana`, `erogato7` ed `erogato6`, in misura minore le variabili `erogato8` ed `giorno_anno_n`.

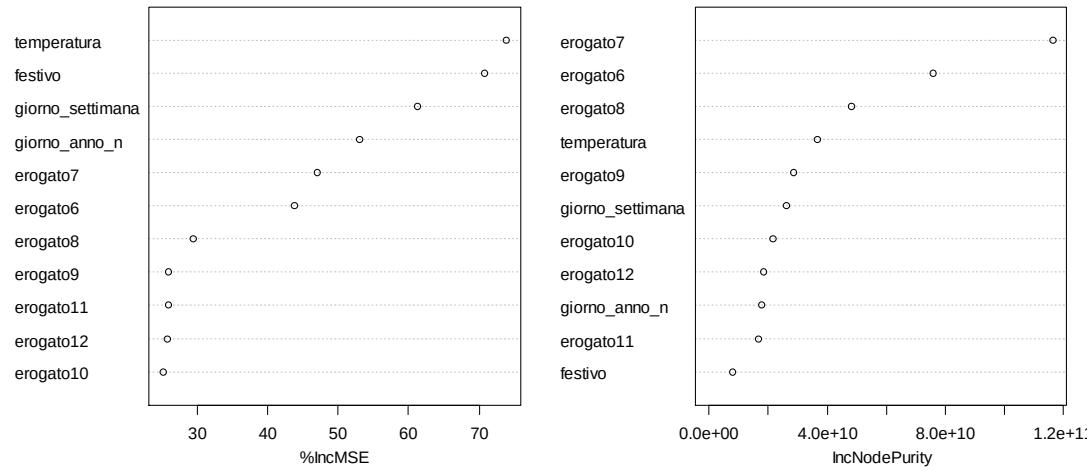


Figura 3.119 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S5.*

I grafici di Figura 3.120 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per il *validation set*, unitamente alla linea di previsione ideale.

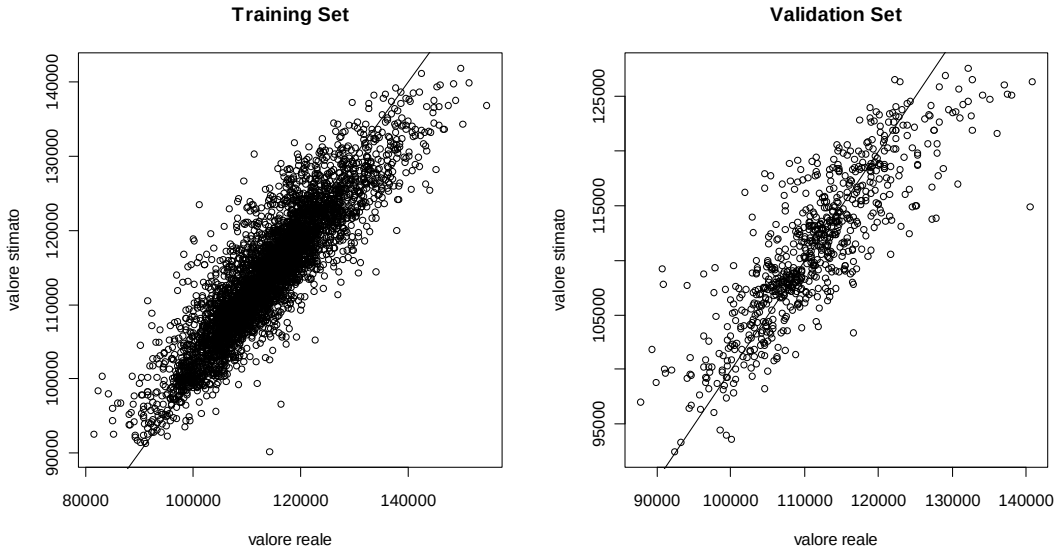


Figura 3.120 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S5.*

Nella Figura 3.121 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 63% degli errori è inferiore al 3% e l'81% degli errori è inferiore al 5%, mentre per il modello rf.S4 i corrispondenti valori erano 65% e 83%.

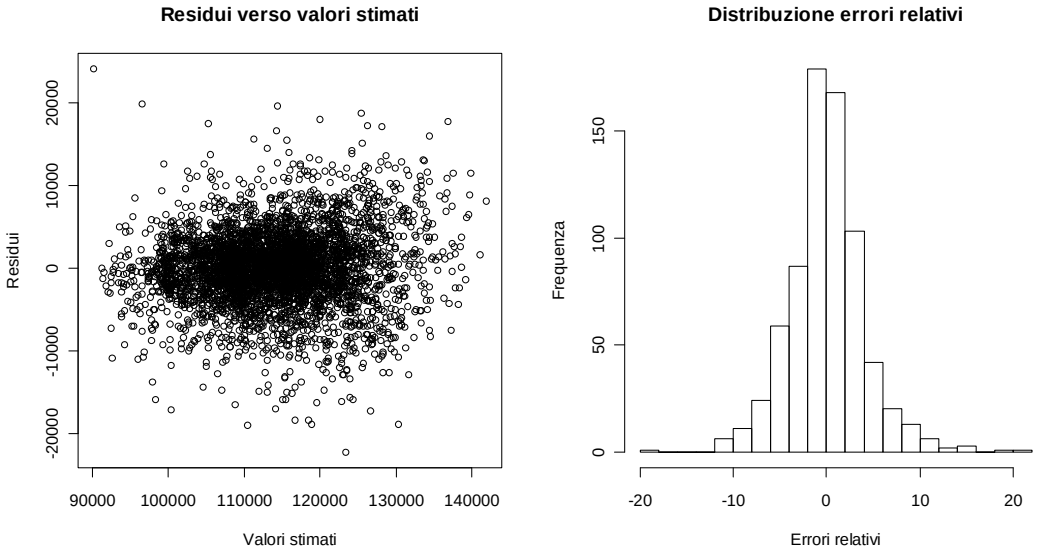


Figura 3.121 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S5.*

3.6.7.7 Modello di previsione a 7 giorni (rf.S6)

Per la stima del volume erogato il giorno $t+7$ è stato predisposto un opportuno modello che ha come variabile risposta il valore `erogato0`. I predittori sono invece `erogato7`,...,`erogato13`, quindi si considerano i valori disponibili fino a 7 giorni antecedenti. Gli altri predittori sono `temperatura`, `festivo`, `giorno_anno_n` e `giorno_settimana`. Vengono escluse le prime 6 osservazione del *data frame* in quanto non dispongono del valore sulla variabile `erogato13`.

La varianza spiegata scende al 78,60% dal 79,29% del modello rf.S5 e lo scarto quadratico medio dell'errore relativo calcolato sul *validation set* sale a 4,14 rispetto al valore 4,05 del modello rf.S5.

Il numero di alberi generati è adeguato: in Figura 3.122 si vede come l'errore quadratico medio rimanga sostanzialmente stazionario già a partire da 250 alberi.

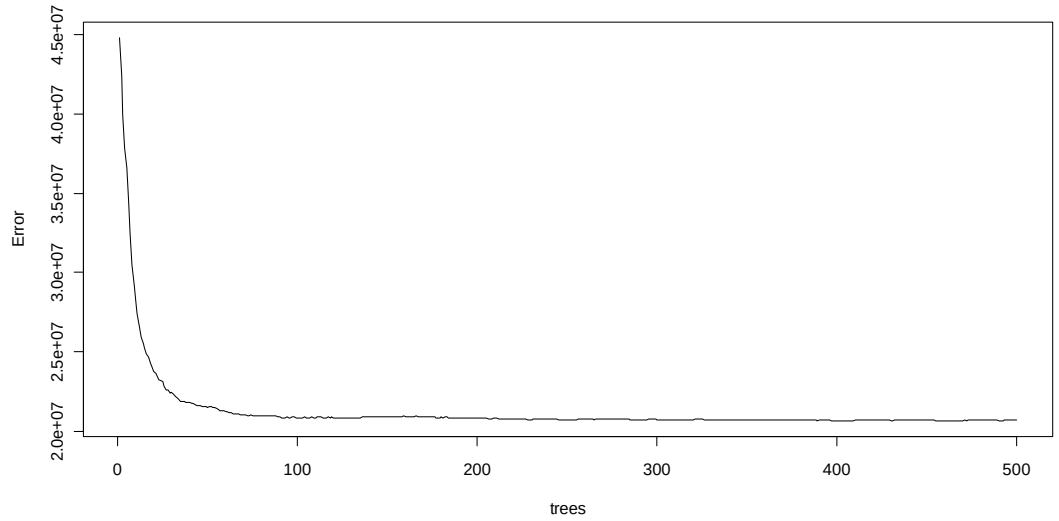


Figura 3.122 - *Errore quadratico medio all'aumentare del numero degli alberi nella foresta, modello rf.S6.*

Nei grafici di Figura 3.123 viene rappresentata l'importanza dei predittori: i più significativi risultano le variabili `temperatura`, `festivo`, `giorno_settimana` ed `erogato7`, in misura minore le variabili `erogato8` ed `giorno_anno_n`.

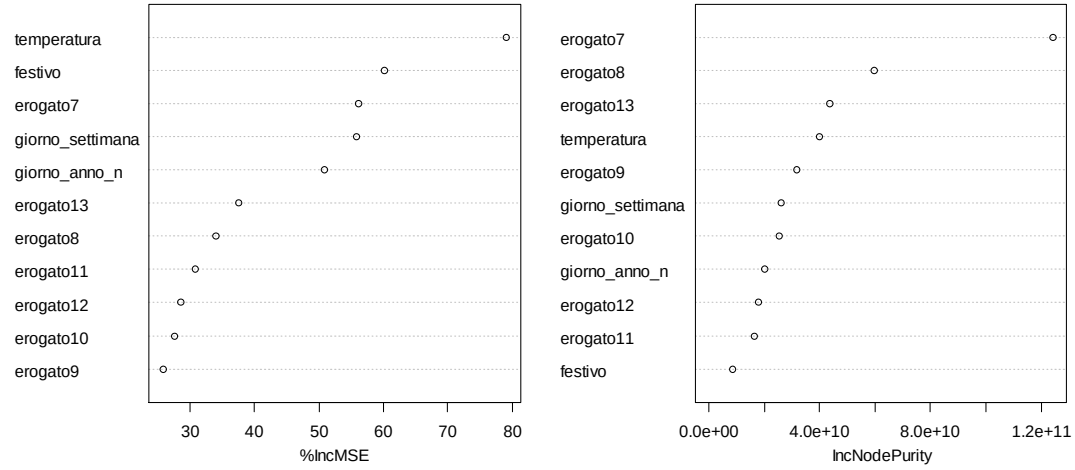


Figura 3.123 - *Importanza dei predittori secondo l'indicatore %IncMSE (a sinistra) e IncNodePurity (a destra), modello rf.S6.*

I grafici di Figura 3.124 riportano i plot dei valori reali verso i valori stimati, a sinistra per i dati del *training set*, a destra per quelli del *validation set*, unitamente alla linea di previsione ideale.

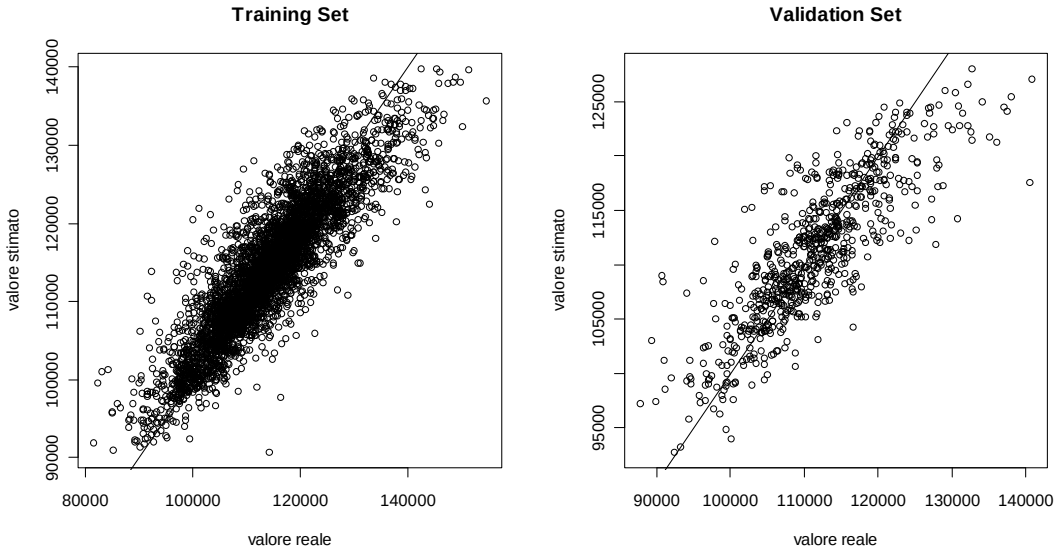


Figura 3.124 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S6.*

Nella Figura 3.125 sono visualizzati a sinistra il plot dei residui verso i valori stimati, dai quali la variabilità dei residui appare costante al variare della grandezza stimata. Nella parte destra della figura si trova la distribuzione degli errori relativi per il *validation set*: il 64% degli errori è inferiore al 3% e l'81% degli errori è inferiore al 5%; per il modello rf.S4 i corrispondenti valori erano 63% e 81%.

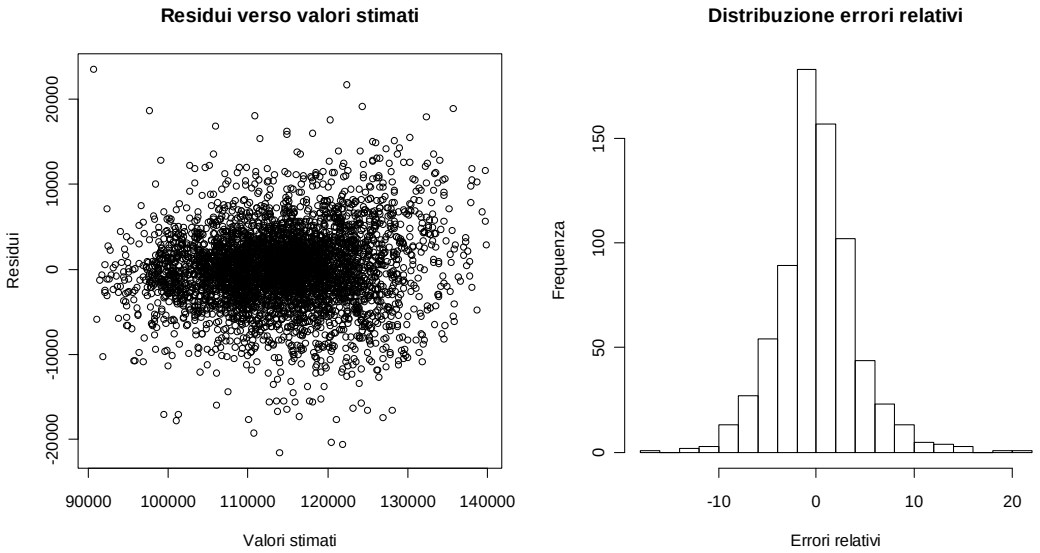


Figura 3.125 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S6.*

3.6.7.8 Modello settimanale (rf.S)

Una volta realizzati i modelli di previsione per i singoli giorni successivi, i valori predetti sono sommati per ottenere una stima complessiva del consumo nella

settimana successiva. È necessario costruire per ciascuna settimana il consumo reale e il suo valore stimato sia per il *dataset training* che per il *dataset validation*.
Il consumo reale per il *dataset training* (`df.train`) è stato ottenuto ordinando le 4376 osservazioni per data e suddividendole in 625 gruppi di 7 elementi consecutivi. Indicando con i la posizione della singola osservazione ($i = 1,...,4376$) il primo gruppo è composto dalle osservazioni caratterizzate da valori di i che vanno da 1 al 7, il secondo da 8 al 13, e così via fino al 625° gruppo con i da 4369 a 4375. L'ultima osservazione non è stata utilizzata. Per ciascun gruppo n ($n = 1,...,625$) è stato sommato il valore `erogato0` degli elementi che lo compongono creando la variabile `cs_train`.
Formalizzando:

$$\text{cs\_train}_n = \sum_{i=7\cdot n-6}^{7\cdot n} \text{erogato0}_i \qquad n = 1,...,625 \tag{3.36}$$

La variabile `cs_train` rappresenta il consumo reale settimanale per il *dataset training*. Si procede quindi al calcolo del consumo settimanale stimato per il medesimo *dataset*. Per ciascun gruppo sono state calcolate le stime dei consumi dei 7 giorni, utilizzando per ciascun giorno il modello di previsione adeguato al giorno stesso: indicando con t il primo giorno del gruppo ($t = 1,8,15,\ldots,4369$), si tratta di stimare i consumi dei giorni $t, t+1,...,t+6$ avendo a disposizione i dati fino al giorno $t-1$. Per stimare il consumo del giorno t viene utilizzato il modello rf.S0 che usa oltre agli altri predittori le variabili `erogato1`, `erogato2`,…, `erogato7`. Per il secondo giorno ($t+1$) viene utilizzato il modello rf.S1 che usa tra gli altri i predittori `erogato2`, `erogato3`,…, `erogato8` riferiti al giorno ($t+1$) e così via fino al giorno ($t+6$) nel quale si utilizza il modello rf.S6 con predittori `erogato7`, `erogato8`,…, `erogato13`. In questo modo sono stati introdotti nei vari modelli sempre gli stessi valori, cioè i consumi realizzati nei 7 giorni antecedenti il giorno t, ovverosia quelli dell'ultima settimana nota, per realizzare la stima del consumo settimanale. Il valore ottenuto come somma delle stime dei singoli giorni è rappresentato dalla variabile `ss_train`:

$$\text{ss\_train}_n = \sum_{k=1}^{7} \text{pred}_{[k]}(7n + k - 7) \qquad n = 1,2,...,624\,, \tag{3.37}$$

dove $\text{pred}_{[k]}(i)$ è la previsione effettuata utilizzando il modello che considera i predittori `erogato(k)`, `erogato(k+1)`,…, `erogato(k+7)` utilizzando in input i dati della riga i del *training set*.
In modo del tutto analogo sono state costruite le variabili `cs_valid` e `ss_valid` che contengono i consumi reali e stimati per le 104 settimane presenti nel *validation set.*

Le quantità appena calcolate sono rappresentate nei grafici di Figura 3.126: a sinistra il plot dei valori stimati verso i valori reali per il *training set* e a destra l'analogo per il *validation set*.

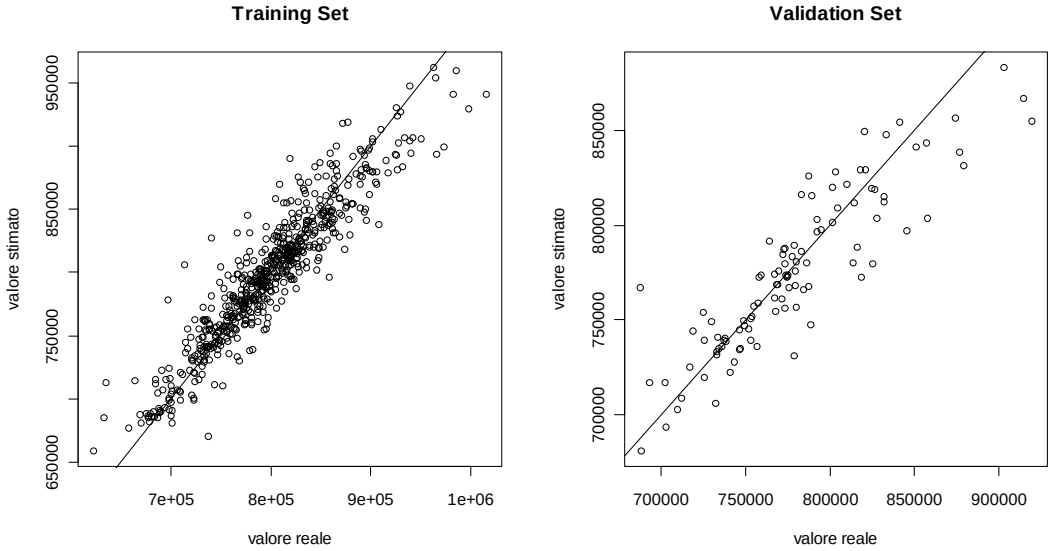


Figura 3.126 - *Plot dei valori stimati verso i valori reali: training set (sinistra) e validation set (destra), modello rf.S.*

In Figura 3.127 si riporta a sinistra il grafico dei valori residui verso i valori stimati e a destra la distribuzione degli errori relativi calcolati sul *validation set*.

Come indicatore di bontà dell'adattamento è stato calcolato lo scarto quadratico medio degli errori di stima relativi, sia per il *training set* che per il *validation set*, pari rispettivamente a 2,73 e 2,74.

Gli errori commessi nella stima delle osservazioni del *validation set* sono nel 77% dei casi inferiori al 3% e nel 90% inferiori al 3%. Il 77% delle stime calcolate sul *validation set* presenta un errore inferiore al 3% e il 90% di esse un errore inferiore al 5%.

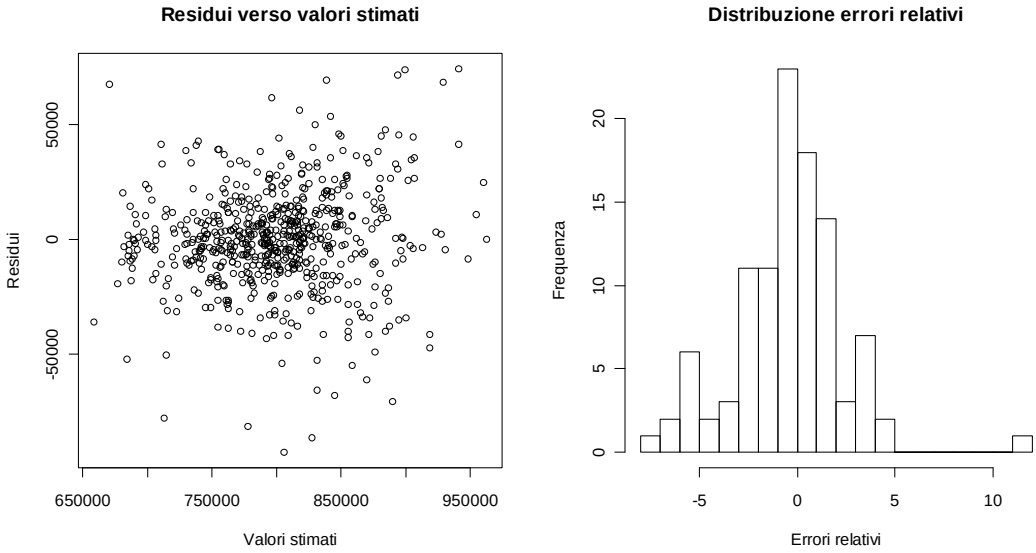


Figura 3.127 - *Plot dei residui verso i valori stimati (training set, a sinistra) e distribuzione degli errori relativi (validation set, a destra), modello rf.S.*

3.7 Confronto tra i modelli elaborati

In questo paragrafo si vogliono sintetizzare i risultati ottenuti dai modelli finora costruiti presentandone le caratteristiche principali e alcuni indicatori relativi alla bontà dell'adattamento. Lo scopo è quello di individuare quale tra questi modelli meglio interpreti la realtà analizzata.

Sono stati elaborati complessivamente n.41 modelli: 4 basati sul modello Acegas-Aps, 3 sulla regressione lineare, 20 sulle reti neurali e 14 sull'algoritmo *Random Forests*. Dei 41 modelli totali, 37 sono giornalieri aventi l'obiettivo di stimare il consumo di acqua del giorno seguente e 4 sono settimanali che mirano a calcolare la stima del consumo dell'intera settimana successiva. I confronti verranno fatti separatamente tra modelli giornalieri e tra modelli settimanali.

3.7.1 Confronto tra modelli giornalieri

Nella Tabella 3.10 viene riportato l'elenco completo di tutti i modelli giornalieri realizzati. La prima colonna contiene il nome del modello: i primi sono quelli di Acegas-Aps, quindi quelli basati sui modelli lineari (lm), poi quelli basati sui reti neurali (nn), infine quelli basati su *Random Forests* (rf). Nella seconda colonna sono riportati i predittori utilizzati per calcolare il modello. Le tre colonne successivi sono peculiari di ogni famiglia di modelli: la colonna "*R-squared*" riporta il valore dell'indice corretto $\overline{R}^2$, come definito nella formula (3.30), calcolato solo per i modelli lineari, la colonna "*N. neuroni strato hidden*" indica il numero di neuroni

presenti nello strato nascosto per i modelli basati su reti neurali, infine la colonna "*% Var spiegata*" riporta la percentuale di varianza spiegata nei modelli basati sull'algoritmo *Random Forests*. Le quattro colonne successive riportano le prestazioni dei modelli valutate sui dati del *training set*: per ogni unità è stato calcolato l'errore relativo di previsione della cui distribuzione si riporta la percentuale dei casi inferiori al 3% e quella inferiore 5%, quindi il valor medio e lo scarto quadratico medio. Le ultime quattro colonne riportano le medesime informazioni calcolate sul *validation set*.

Tabella 3.10 - *Prestazioni dei modelli giornalieri elaborati (continua a pag.136)*

					Training				Validation			
Modello	**Predittori**	**R-squared**	**N. neuroni strato hidden**	**% Var spiegata**	**% err. inferiori 3%**	**% err. inferiori 5%**	**Media**	**Scarto quadratico medio**	**% err. inferiori 3%**	**% err. inferiori 5%**	**Media**	**Scarto quadratico medio**
Acegas-Aps1	Coefficiente settimanale				62%	82%	0,13	4,09	61%	82%	0,15	4,06
Acegas-Aps2	Coefficiente settimanale + temperatura				65%	84%	0,36	3,82	56%	80%	-0,18	4,26
lm.1	erogato1÷erogato7	0,77			60%	81%	0,17	4,11	58%	78%	0,30	4,23
lm.2	erogato1÷erogato7 + temperatura	0,78			61%	82%	0,16	4,07	60%	79%	0,01	4,16
lm.3	erogato1÷erogato7 + temperatura + rapp_settimana	0,77			61%	82%	0,16	4,02	62%	79%	0,11	4,16
nn.1	erogato1_n÷erogato7_n		4		68%	86%	0,14	3,62	68%	87%	0,21	3,69
nn.2	erogato1_n÷erogato7_n + temperatura		4		70%	87%	0,13	3,56	67%	87%	0,02	3,67
nn.3	erogato1_n÷erogato7_n + temperatura + rapp_settimana		4		70%	87%	0,13	3,55	65%	86%	-0,17	3,72
nn.4	rapp_settimana		4		34%	55%	0,50	7,24	40%	57%	3,67	6,25
nn.5	temperatura + rapp_settimana		4		34%	56%	0,46	6,84	44%	64%	2,29	6,02
nn.6	scale(temperatura) + rapp_settimana		4		33%	56%	0,46	6,83	44%	63%	2,29	6,01
nn.7	scale(erogato1_n)÷scale(erogato7_n)		4		70%	87%	0,13	3,61	67%	87%	0,08	3,67
nn.8	erogato1_n÷erogato7_n + rapp_settimana		4		70%	87%	0,12	3,52	70%	86%	0,35	3,66
nn.9	erogato1_n÷erogato7_n + temperatura + rapp_settimana		18		75%	90%	0,10	3,14	65%	86%	0,26	5,32
nn.10	erogato1_n÷erogato7_n + rapp_settimana		18		75%	90%	0,10	3,15	69%	87%	0,44	4,66
nn.11	erogato1_n÷erogato7_n + temperatura + rapp_settimana + festivo		4		68%	87%	0,13	3,60	65%	86%	0,21	3,72
nn.12	erogato1_n÷erogato7_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		81%	94%	0,07	2,56	76%	94%	-0,02	2,78
nn.S0	erogato1÷erogato7 + temperatura + rapp_settimana + festivo + giorno_settimana		4		81%	94%	0,07	2,56	76%	94%	-0,02	2,78
nn.S1	erogato2_n÷erogato8_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		75%	91%	0,09	2,98	70%	89%	0,00	3,28
nn.S2	erogato3_n÷erogato9_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		71%	88%	0,11	3,27	68%	86%	0,08	3,46
nn.S3	erogato4_n÷erogato10_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		67%	85%	0,13	3,61	69%	84%	0,19	3,65
nn.S4	erogato5_n÷erogato11_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		64%	83%	0,16	3,86	64%	82%	0,27	3,87
nn.S5	erogato6_N÷erogato12_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		65%	83%	0,15	3,87	64%	81%	0,17	3,95
nn.S6	erogato7_n÷erogato13_n + temperatura + rapp_settimana + festivo + giorno_settimana		4		63%	82%	0,16	3,94	64%	82%	0,07	4,02

Tabella 3.10 - *Prestazioni dei modelli giornalieri elaborati. (segue da pag.135)*

					Training				Validation			
Modello	**Predittori**	**R-squared**	**N. neuroni strato hidden**	**% Var spiegata**	**% err. inferiori 3%**	**% err. inferiori 5%**	**E(erx100)**	**S(erx100)**	**% err. inferiori 3%**	**% err. inferiori 5%**	**E(erx100)**	**S(erx100)**
rf.1	erogato1÷erogato7			82,15%	69%	86%	0,16	3,69	67%	85%	0,34	3,74
rf.2	erogato1÷erogato7 + temperatura			83,19%	70%	87%	0,16	3,61	65%	86%	0,14	3,67
rf.3	erogato1÷erogato7 + temperatura + rapp_settimana			82,60%	71%	87%	0,16	3,52	66%	86%	0,19	3,72
rf.4	erogato1÷erogato7 + temperatura + giorno_anno_n			83,98%	71%	87%	0,14	3,52	68%	87%	0,04	3,64
rf.5	erogato1÷erogato7 + temperatura + giorno_anno_n + festivo			85,44%	72%	89%	0,17	3,31	70%	88%	0,12	3,44
rf.6	erogato1÷erogato7 + temperatura + giorno_anno_n + festivo + giorno_settimana			89,19%	76%	92%	0,10	2,85	75%	91%	0,08	3,00
rf.S0	erogato1÷erogato7 + giorno_anno_n + festivo + giorno_settimana			89,19%	76%	92%	0,10	2,85	75%	91%	0,08	3,00
rf.S1	erogato2÷erogato8 + giorno_anno_n + festivo + giorno_settimana			85,47%	71%	89%	0,12	3,29	69%	87%	0,07	3,46
rf.S2	erogato3÷erogato9 + giorno_anno_n + festivo + giorno_settimana			83,01%	68%	86%	0,14	3,56	67%	85%	0,05	3,68
rf.S3	erogato4÷erogato10 + giorno_anno_n + festivo + giorno_settimana			81,19%	65%	85%	0,16	3,74	66%	84%	0,08	3,80
rf.S4	erogato5÷erogato11 + giorno_anno_n + festivo + giorno_settimana			80,09%	65%	84%	0,15	3,85	65%	83%	0,07	3,96
rf.S5	erogato6÷erogato12 + giorno_anno_n + festivo + giorno_settimana			79,29%	64%	83%	0,16	3,93	63%	81%	0,06	4,05
rf.S6	erogato7÷erogato13 + giorno_anno_n + festivo + giorno_settimana			78,60%	64%	83%	0,17	4,00	64%	81%	0,09	4,14

I due modelli giornalieri di Acegas-Aps sono stati realizzati seguendo le linee guida del progetto originario: sono stati dapprima costruiti i modelli settimanali, quindi calcolati i pesi giornalieri che sono infine stati utilizzati per ripartire la stima settimanale nei singoli giorni.

I modelli basati sulla regressione lineare hanno realizzato delle performance inferiori rispetto ai corrispondenti modelli basati su reti neurali e *Random Forests*. Per questo motivo si è ritenuto di costruirne solo alcuni da utilizzare come *benchmark* di riferimento.

I primi 12 modelli basati su reti neurali sono stati costruiti per trovare la combinazione migliore dei parametri, sia in termini di selezione delle variabili, sia in relazione alla loro trasformazione di scala, sia in merito al numero di neuroni presenti nello strato nascosto della rete. Sulla base dei predittori e dei parametri del modello risultato migliore (nn.12) sono stati realizzati i 7 modelli giornalieri che forniscono le stime del consumo dei 7 giorni successivi (nn.S0, nn.S1,…, nn.S6), propedeutici alla realizzazione del modello settimanale.

Analogamente a quanto visto per le reti neurali sono stati realizzati i modelli basati sull'algoritmo *Random Forests*: è stato individuato il modello avente parametri e predittori migliori (rf.6) e in base a questo sono stati realizzati i 7 modelli giornalieri (rf.S0, rf.S1,…, rf.S6) necessari alla costruzione del modello settimanale.

3.7.1.1 Confronto a parità di predittori

Un primo tipo di confronto può essere fatto tra i modelli giornalieri basati su metodologie costruttive diverse ma che considerano gli stessi predittori. Si confrontano tra loro i modelli lm.1, nn.1 e rf.1, successivamente lm.2, nn.2 e rf.2 ed infine lm.3, nn.3 e rf.3. I primi tre modelli utilizzano come predittori il consumo dei 7 giorni precedenti, il secondo gruppo di tre anche la temperatura massima giornaliera, l'ultimo gruppo di modelli utilizza oltre ai predittori indicati per i gruppi precedenti anche il coefficiente settimanale. Nel grafico di Figura 3.128 viene rappresentato, per i tre gruppi di tre modelli, il numero dei casi del *validation set* che presentano un errore di stima inferiore al 3% e inferiore al 5%; valori più alti nell'istogramma indicano una migliore capacità predittiva del modello. Si nota che i modelli basati su regressione lineare hanno realizzato un adattamento inferiore, mentre sono pressoché equivalenti fra loro i modelli basati su reti neurali e su *Random Forests*.

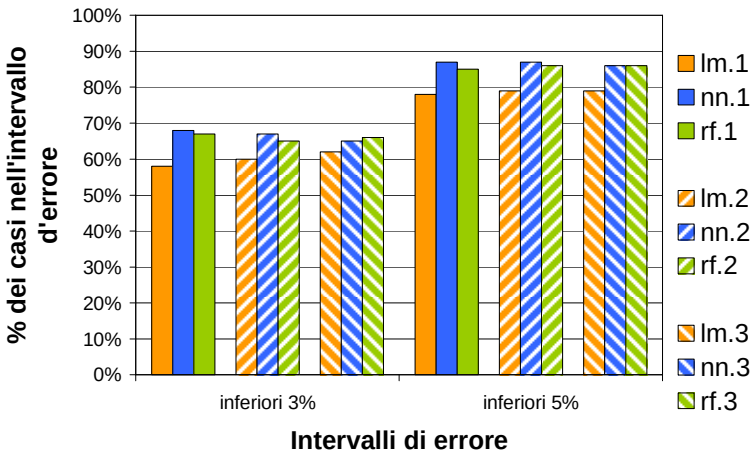


Figura 3.128 - *Confronto tra modelli giornalieri sulla base delle frequenze dell'errore relativo (validation set).*

Confrontando i modelli della stessa famiglia si può notare che l'introduzione tra i predittori della temperatura e del coefficiente settimanale non ha portato ad un miglioramento sostanziale della capacità predittiva dei modelli, anzi in alcuni casi si è verificato una sua diminuzione, in particolare questa si nota nel confronto tra i modelli nn.1, nn.2 e nn.3 (reti neurali) e tra rf.1, rf.2 e rf.3 (*Random Forests*). Un

leggero miglioramento è stato ottenuto invece per i modelli basati su regressione lineare.

In Figura 3.129 si riporta il confronto tra i modelli giornalieri sulla base della media e dello scarto quadratico medio degli errori relativi. La media fornisce indicazioni sulla distorsione delle stime realizzate. Si nota che si attesta per tutti i modelli su valori modesti, inferiori allo 0,5%. Lo scarto quadratico medio fornisce indicazioni sulla dispersione delle stime attorno al valor medio: più questo valore è basso più l'errore di previsione del modello si concentra attorno al valore medio che in questo caso è quasi nullo. Viene confermata la superiorità dei modelli basati su reti neuronali e *Random Forests* rispetto alla regressione lineare.

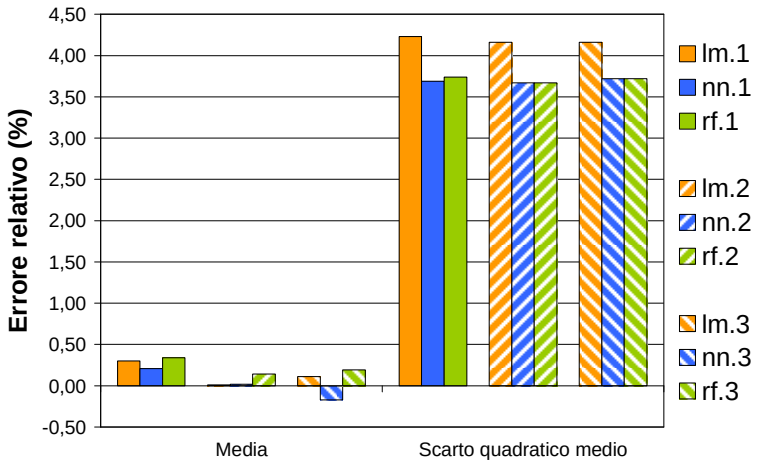


Figura 3.129 - *Confronto tra modelli giornalieri sulla base della media e dello scarto quadratico medio dell'errore relativo (validation set).*

3.7.1.2 Confronto tra i modelli migliori delle famiglie

Un secondo tipo di confronto tra modelli giornalieri viene effettuato comparando il miglior modello basato su reti neurali (nn.12), il migliore basato su *Random Forests* (rf.6) e i due modelli di Acegas-Aps. Nel grafico di Figura 3.130 viene rappresentato, per questi modelli, il numero dei casi del *validation set* che presenta un errore di stima inferiore al 3% e inferiori al 5%.

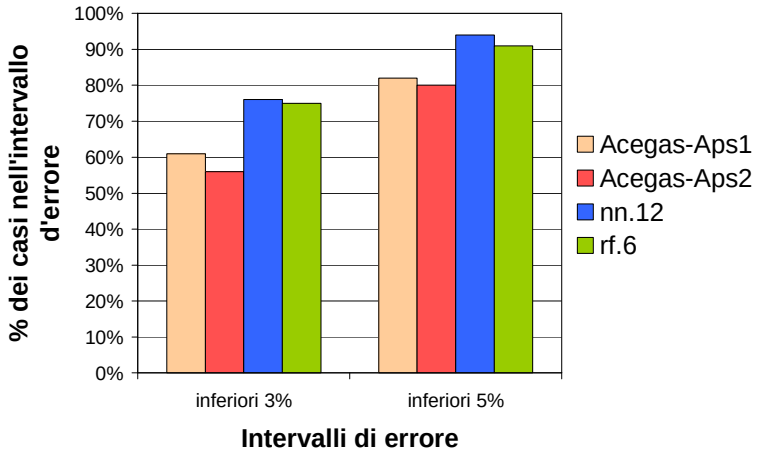


Figura 3.130 - *Confronto tra il miglior modello basato su reti neurali (nn.12), il migliore basato su Random Forests (rf.6) e i modelli Acegas-Aps, sulla base della media e dello scarto quadratico medio dell'errore relativo (validation set).*

I valori più elevati, e quindi le prestazioni migliori, sono stati ottenuti dal modello basato su reti neurali, di poco superiori a quanto realizzato dal modello basato su *Random Forests*. Sensibilmente peggiori sono le stime prodotte dai due modelli Acegas-Aps, in particolare dal secondo.

In Figura 3.131 si riporta la rappresentazione grafica della media e dello scarto quadratico medio degli errori relativi. Per tutti i modelli si osservano per le medie degli errori valori prossimi allo zero indicando la sostanziale assenza di distorsione nelle stime. Valori inferiori per lo scarto quadratico (quindi prestazioni superiori) vengono ottenuti dai modelli basati su reti neurali e *Random Forests* rispetto ai modelli Acegas-Aps.

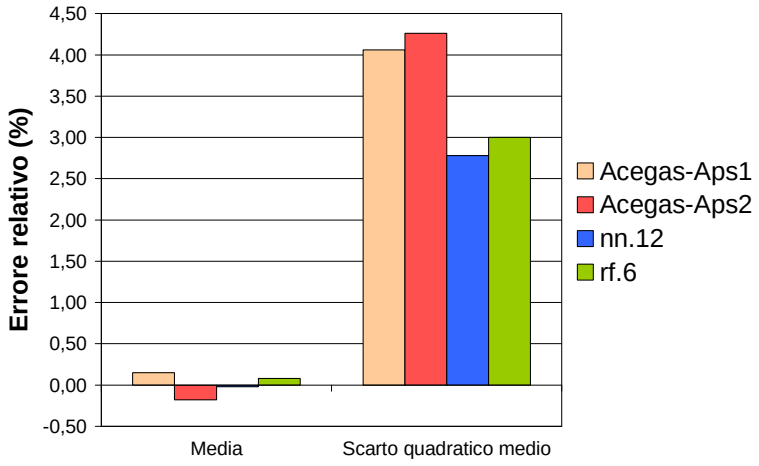


Figura 3.131 - *Confronto tra il miglior modello giornaliero basato su reti neurali (nn.12), il migliore basato su Random Forests (rf.6) e i due di Acegas-Aps sulla base della media e dello scarto quadratico medio dell'errore relativo (validation set).*

3.7.1.3 Influenza dell'orizzonte di previsione giornaliera

Per le famiglie di modelli basati su reti neurali e *Random Forests* si dispone dei modelli di previsione giornaliera non solo per il giorno successivo ma anche per gli altri giorni componenti la settimana a venire. Detti modelli sono stati ricavati nel procedimento costruttivo che ha portato alla realizzazione dei modelli settimanali. Di seguito verranno utilizzati per confrontare le prestazioni delle due famiglie di modelli al variare dell'orizzonte di previsione. Nella Figura 3.132 viene visualizzato lo scarto quadratico medio della distribuzione degli errori relativi di stima, calcolati sulle osservazioni del *validation set*, al variare dell'orizzonte di previsione. Lo scarto quadratico medio, la cui crescita indica una diminuzione della precisione di stima, ha, come atteso, un valore crescente quando si cerca di prevedere il consumo di un giorno più lontano. È possibile inoltre notare come i modelli basati su reti neurali siano migliori rispetto a quelli basati su *Random Forests* a tutti gli orizzonti di previsione.

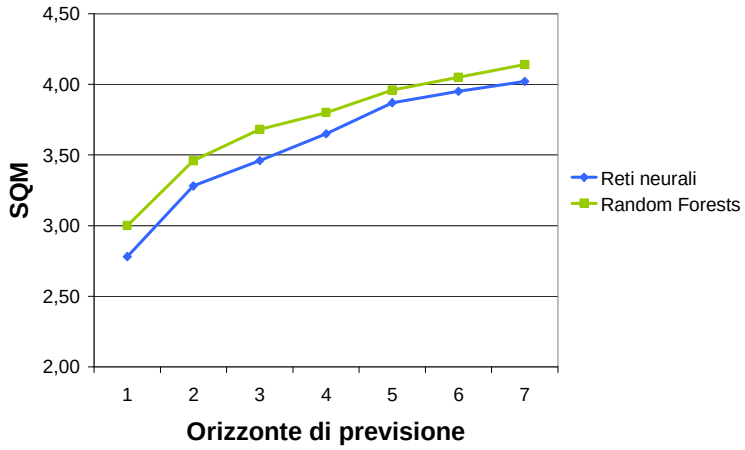


Figura 3.132 – *Andamento dell'errore quadratico medio della distribuzione degli errori relativi per i modelli giornalieri basati su reti neurali e Random Forests al variare dell'orizzonte di previsione (validation set).*

3.7.2 Confronto tra modelli settimanali

Nella Tabella 3.11 viene riportato l'elenco dei modelli settimanali realizzati. I dati presenti sono analoghi a quanto visto per i modelli giornalieri: nome del modello, predittori utilizzati e prestazioni predittive valutate sui dati del *training set* e del *validation set*.

Tabella 3.11 - *Prestazioni dei modelli settimanali elaborati.*

		Training				Validation			
Modello	**Predittori**	**% err. inferiori 3%**	**% err. inferiori 5%**	**Media**	**Scarto quadratico medio**	**% err. inferiori 3%**	**% err. inferiori 5%**	**Media**	**Scarto quadratico medio**
Acegas-Aps-S1	Coefficiente settimanale	75%	89%	0,09	3,06	72%	91%	0,06	2,90
Acegas-Aps-S2	Coefficiente settimanale + temperatura	80%	92%	0,31	2,63	61%	92%	-0,27	3,21
nn.S	erogato1÷erogato13 + temperatura + rapp_settimana + festivo + giorno_settimana	81%	94%	0,07	2,51	80%	94%	-0,25	2,37
rf.S	erogato7÷erogato13 + giorno_anno_n + festivo + giorno_settimana	77%	90%	0,11	2,73	77%	90%	-0,39	2,74

Sono presenti i due modelli settimanali di Acegas-Aps, il modello basato su reti neurali e quello basato su *Random Forests*. Non è stato elaborato il modello basato su regressione lineare in quanto già dalla valutazione dei modelli giornalieri è risultato inferiore.

Nella Figura 3.133 sono riportati per i quattro modelli settimanali la percentuale di errori inferiori al 3% e al 5% calcolati sul *validation set*. Il modello che sembra fornire una migliore capacità predittiva è quello basato su reti neurali, di poco superiore a quello basato su *Random Forests*. I modelli di Acegas-Aps presentano buone prestazioni considerando errori inferiori al 5% ma prestazioni più basse considerando la percentuale di errori inferiori al 3%. In particolare questo accade per il secondo modello, che considera tra i predittori anche la temperatura.

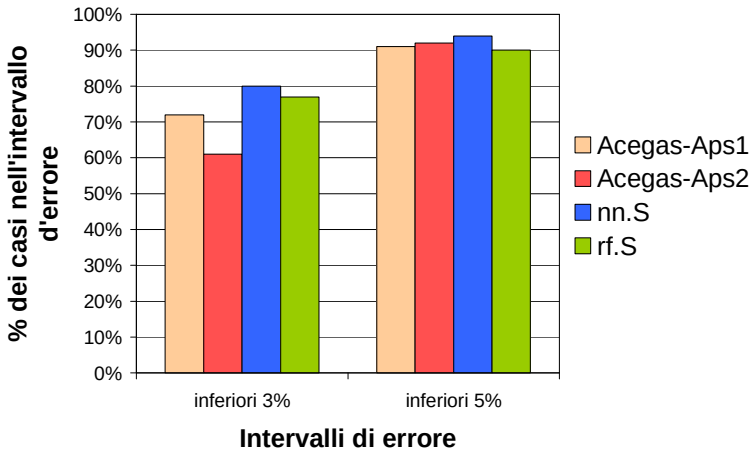


Figura 3.133 - *Confronto tra modelli settimanali sulla base delle frequenze dell'errore relativo.*

In Figura 3.134 si riporta il confronto tra i modelli settimanali sulla base della media e dello scarto quadratico medio degli errori relativi. La media fornisce indicazioni

sulla distorsione delle stime realizzate e si attesta per tutti i modelli su valori relativamente bassi. Lo scarto quadratico medio fornisce indicazioni sulla dispersione delle stime, valori bassi indicano stime più precise. Viene confermata la superiorità del modello basato su reti neurali, a seguire quello basato su *Random Forests* ed il primo modello di Acegas-Aps, infine il secondo modello di Acegas-Aps.

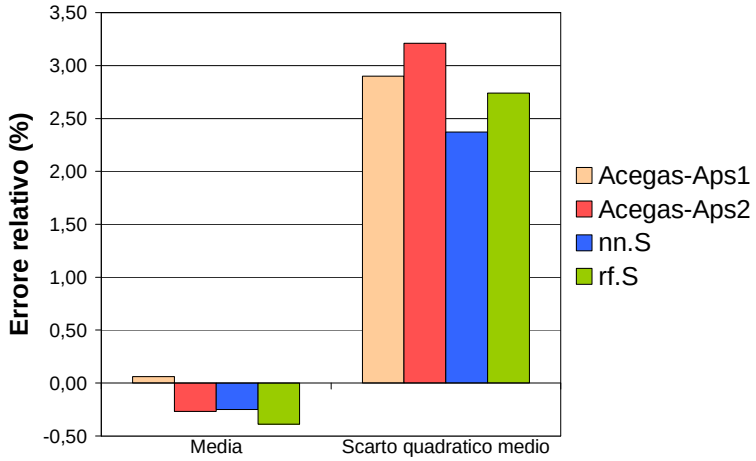


Figura 3.134 - *Confronto tra modelli settimanali sulla base della media e dello scarto quadratico medio dell'errore relativo (validation set).*

3.7.3 Analisi dei modelli Acegas-Aps

È utile focalizzare l'attenzione sui due modelli Acegas-Aps e valutare le prestazioni dei modelli giornalieri e settimanali. Nella Figura 3.135 viene rappresentato lo scarto quadratico medio della distribuzione degli errori di stima per i due modelli giornalieri e per i due settimanali, calcolati per il *training set* e per il *validation set.*

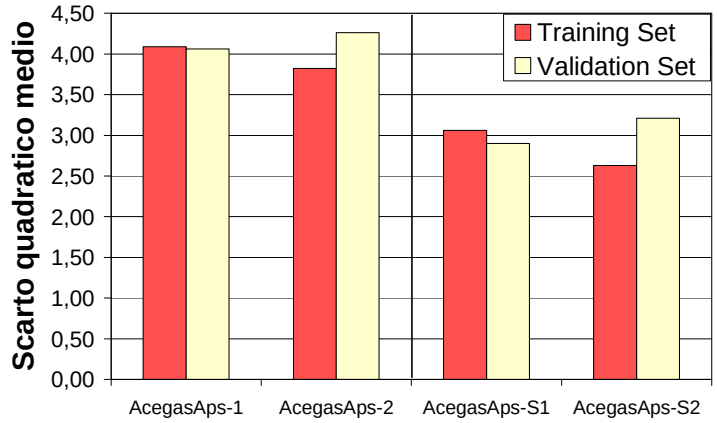


Figura 3.135 - *Confronto tra modelli Acegas-Aps, giornalieri (sinistra) e settimanali (destra), sulla base dello scarto quadratico medio dell'errore relativo (training set e validation set).*

L'introduzione del predittore temperatura per la realizzazione del modello giornaliero AcegasAps-2 e del modello settimanale AcegasAps-S2 porta ad una leggera diminuzione dell'errore nel *training set*, a fronte però di un evidente aumento di questo nel *validation set*. Si ritiene che per questi modelli presentino il fenomeno dell'*overfitting*.

3.7.4 Conclusioni

A seguito del lavoro svolto è possibile trarre alcune considerazioni. Tutti i modelli elaborati si adattano abbastanza bene alla realtà di interesse: non sono evidenti distanze marcate sulla capacità predittive dei modelli. Ciononostante è possibile individuare come modelli con capacità predittiva migliore quelli basati su reti neurali, a seguire quelli basati su *Random Forests*, quindi quelli di Acegas-Aps ed infine quelli basati su regressione lineare.
Le differenze tra le famiglie di modelli sono più evidenti quando si confrontano i modelli giornalieri. In particolare i modelli giornalieri di Acegas-Aps sono penalizzati dalla loro metodologia costruttiva che parte da una previsione di stima settimanale per giungere a quella giornaliera.
È interessante osservare come, contrariamente ad ogni aspettativa, la variabile temperatura introdotta nel secondo modello Acegas-Aps diminuisca la capacità predittiva dello stesso anziché aumentarla.
Con particolare riferimento alla previsione settimanale, era giustificabile pensare che i modelli Acegas-Aps riportassero prestazioni nettamente inferiori rispetto agli altri basati su metodologie costruttive e teorie statistiche consolidate. Ciò non è avvenuto. Si osserva come i modelli settimanali di Acegas-Aps, costruiti su osservazioni empiriche ed esperienza, abbiano dimostrato un livello di adattmento ai dati appena inferiore.
Infine si ritiene importante evidenziare come i modelli di Acegas-Aps presentino il vantaggio di avere una logica costruttiva ed un comportamento noti: l'elaborazione dei dati in ingresso per produrre l'output del modello è trasparente consentendo delle analisi e delle eventuali modifiche ragionate. Tanto nei modelli basati su reti neurali quanto in quelli basati su *Random Forests* si ha per contro la tipica struttura a *black-box*: sebbene gli algoritmi costruttivi siano noti il funzionamento del modello non è trasparente consentendo all'analista solo margini di manovra ridotti per ricercare un'eventuale miglioramento delle prestazioni.

Capitolo 4
-
Fondamenti teorici

Nel presente capitolo vengono presentati, da un punto di vista teorico, i modelli utilizzati per la realizzazione dei modelli esposti al Capitolo 3, ovvero le reti neurali artificiali e le *Random Forests*.

4.1 Reti Neurali Artificiali

4.1.1 Introduzione alle Reti Neurali Artificiali

Le reti neurali artificiali sono nate per riprodurre attività tipiche del cervello umano come la percezione di immagini, il riconoscimento di forme, la comprensione del linguaggio, il coordinamento senso-motorio, ecc.
A tale scopo si sono studiate le caratteristiche del cervello umano. Nel sistema nervoso esistono miliardi di neuroni (cellule nervose). Un neurone è formato da un corpo cellulare e da molti prolungamenti ramificati, detti *dendridi*, attraverso i quali il neurone riceve segnali elettrici da altri neuroni. Ogni neurone ha anche un prolungamento filamentoso chiamato *assone*, la cui lunghezza può variare da circa 1 cm a qualche metro (Lazzerini, 2002). All'estremità l'assone si ramifica formando terminali attraverso i quali i segnali elettrici vengono trasmessi ad altre cellule (ad esempio ai dendridi di altri neuroni). Tra un terminale di un assone e la cellula ricevente esiste uno spazio. I segnali superano questo spazio per mezzo di sostanze chimiche dette neurotrasmettitori. Il punto di connessione tra terminale e dendride è detta sinapsi. Un neurone si 'attiva', cioè trasmette un impulso elettrico lungo il suo assone quando si verifica una differenza di potenziale elettrico tra l'interno e l'esterno della cellula. L'impulso elettrico provoca la liberazione di un neurotrasmettitore dai terminali dell'assone, che a loro volta possono, ad esempio, influenzare altri neuroni. I neuroni biologici sono da 5 a 6 ordini di grandezza più lenti dei componenti elettronici convenzionali: un evento in un chip si verifica in alcuni nanosecondi ($10^{-9} s$) mentre un evento neurale in alcuni millisecondi ($10^{-3} s$). Il cervello umano è un calcolatore complesso, non lineare e parallelo. Pur essendo costituito da elementi di elaborazione molto semplici (i neuroni), è in grado di

eseguire computazioni complesse, come il riconoscimento, la percezione ed il controllo del movimento, molto più velocemente del più veloce degli attuali calcolatori. Il cervello è in grado di modificare le connessioni tra i neuroni in base all'esperienza acquisita, cioé è in grado di imparare.

Nel cervello non esiste un controllo centralizzato, nel senso che le varie zone del cervello funzionano insieme, influenzandosi reciprocamente e contribuendo alla realizzazione di uno specifico compito. Infine, il cervello umano è *fault-tolerant*, cioè se un neurone o una delle sue connessioni sono danneggiate, il cervello continua a funzionare, anche se con prestazioni leggermente degradate. In particolare, le prestazioni del processo cerebrale degradano gradualmente man mano che si distruggono sempre più neuroni (*graceful degradation*).

Quindi, per riprodurre artificialmente il cervello umano, occorre realizzare una rete di elementi molto semplici che sia una struttura distribuita, massicciamente parallela, capace di apprendere e quindi di generalizzare (cioè di produrre uscite in corrispondenza di ingressi non incontrati durante l'addestramento).

La computazione neurale si ispira ai sistemi neurali biologici, dei quali si cerca di modellare la struttura e simulare le funzioni di base. In contrapposizione alla filosofia in cui un singolo processore accentra tutta la capacità computazionale ed esegue le operazioni in sequenza programmata, le reti neurali, ispirandosi ai sistemi biologici, considerano un elevato numero di elementi con capacità computazionali elementari. Questi sono i 'neuroni artificiali', detti anche 'nodi', i quali sono connessi ad altre unità dello stesso tipo. Un neurone artificiale è l'unità computazionale atomica di una rete neurale. Esso valuta l'intensità degli ingressi, ed a ciascuno assegna un peso, cioè un valore numerico che modula l'impatto che tale ingresso ha sulla somma totale. Ogni neurone calcola la somma dei segnali pesati che riceve in ingresso e vi aggiunge la propria 'soglia di attivazione' (*bias*). A tale somma pesata degli ingressi è applicata una funzione, detta funzione di trasferimento o legge di attivazione, di solito non lineare, che determina l'uscita della rete stessa. Tipicamente viene utilizzata la stessa funzione di attivazione per tutti i neuroni che appartengono allo stesso strato di una rete, anche se questo non è indispensabile. Ad ogni connessione è attribuito un peso più o meno debole in modo tale che un neurone possa o meno influenzarne un altro in funzione della 'forza' di connessione fra i due. Ciascun neurone riceve diversi segnali in ingresso, ma ne crea uno solo in uscita. In una rete neurale, generalmente, i neuroni vengono organizzati in strati (*layer*). Il comportamento di un neurone è determinato dalla sua funzione di trasferimento e dalle connessioni pesate lungo le quali invia e riceve segnali. Nella maggior parte delle reti, i neuroni risultano interamente connessi, e quindi se un neurone appartiene

ad uno strato ed è collegato ad un neurone dello strato successivo, allora tutti i neuroni del primo strato sono connessi al secondo neurone. Esistono in letteratura metodologie chiamate di *pruning* (Cammarata, 1997) che provvedono ad eliminare alcune connessioni all'interno della rete, per migliorare le prestazioni ad un costo computazionale minore.

L'architettura di rete maggiormente utilizzata è il 'percettrone multi-strato' (*Multi-Layer Perceptron*, MLP), costituita da uno strato di neuroni di ingresso, da uno o più strati intermedi ('strato nascosto' o *hidden layer*) e da uno strato di uscita (*output layer*).

Uno dei concetti fondamentali della computazione neurale è quello del *training* o addestramento. È, infatti, fondamentale capire come una rete viene addestrata per poter fornire appropriate risposte in relazione a determinati valori degli ingressi. Il metodo più usato per addestrare una rete neurale consiste nel presentare in ingresso alla rete un insieme di esempi (*training set*). La risposta fornita dalla rete per ogni esempio viene confrontata con la risposta desiderata, si valuta la differenza (errore) fra le due, ed in base a tale differenza, si aggiustano i pesi. Questo processo viene ripetuto sull'intero *training set* finché le uscite della rete producono un errore al di sotto di una soglia prestabilita. Gli algoritmi di addestramento specificano in che modo devono essere modificati i pesi per far sì che la rete apprenda. Ci sono due classi di algoritmi: quelli "con supervisore" (*supervised learning*) e quelli "senza supervisore" (*unsupervised learning*). Gli algoritmi con supervisione modificano i pesi delle connessioni in modo che a definiti valori di ingresso (campione di ingresso) corrispondano delle uscite definite a priori (campione d'uscita). Il più famoso tra questi algoritmi è senza dubbio quello denominato di *Back-Propagation*.

Negli algoritmi di addestramento non supervisionato l'aggiustamento della rete neurale non avviene in base ad una risposta desiderata imposta dall'esterno ma la rete si auto-organizza modificando i pesi sinaptici nelle fasi di esposizione ai pattern, sviluppandosi così in maniera autonoma.

Tipicamente, il neurone artificiale ha molti ingressi ed una sola uscita. Ogni ingresso ha associato un peso, che determina la 'conducibilità' del canale di ingresso. L'attivazione del neurone è una funzione della somma pesata degli ingressi.

Anche se di recente introduzione, le reti neurali trovano valida applicazione in settori quali predizione, classificazione, riconoscimento e controllo, portando spesso contributi significativi alla soluzione di problemi difficilmente trattabili con metodologie classiche.

4.1.2 Modello di un Neurone

Un neurone (Figura 4.1) ha n canali di ingresso $x_1,...,x_n$, a ciascuno dei quali è associato un peso. I pesi w_i sono numeri reali che riproducono la sinapsi. Se $w_i > 0$ il canale è detto eccitatorio, se invece $w_i < 0$ il canale è inibitorio. Il valore assoluto di un peso rappresenta la forza della connessione.

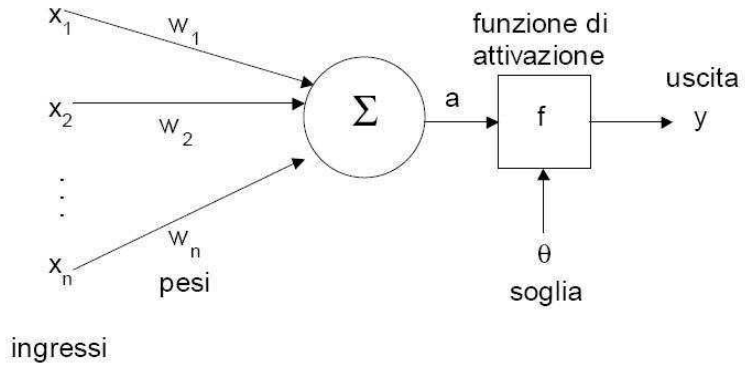


Figura 4.1 - *Modello di neurone*

L'uscita, cioè il segnale attraverso il quale il neurone trasmette la sua attività all'esterno, è calcolata applicando la funzione di attivazione alla somma pesata degli ingressi. Indicando con:

$$a = \sum_{i=1}^{n} w_i x_i \tag{4.1}$$

la somma pesata degli ingressi, l'uscita diviene:

$$y = f(a) = f(\sum_{i=1}^{n} w_i x_i) \tag{4.2}$$

Spesso, nella letteratura, la somma pesata degli ingressi è indicata con la parola *net*. Inoltre, la funzione di attivazione è detta anche funzione di trasferimento.

Nel modello di neurone rappresentato nella precedente Figura 4.1 è stata inclusa anche una soglia (*threshold*), che ha l'effetto di abbassare il valore in ingresso alla funzione di attivazione. Quindi, più correttamente, dobbiamo scrivere:

$$y = f(a) = f(\sum_{i=1}^{n} w_i x_i - \vartheta) \tag{4.3}$$

In questo caso, interpretando la soglia come il peso associato ad un ulteriore canale di ingresso x_0, di valore sempre costante e pari a -1, potremmo anche scrivere:

$$y = f(a) = f(\sum_{i=0}^{n} w_i x_i) \tag{4.4}$$

Il modello di un neurone diventa quindi quello rappresentato nella Figura 4.2.

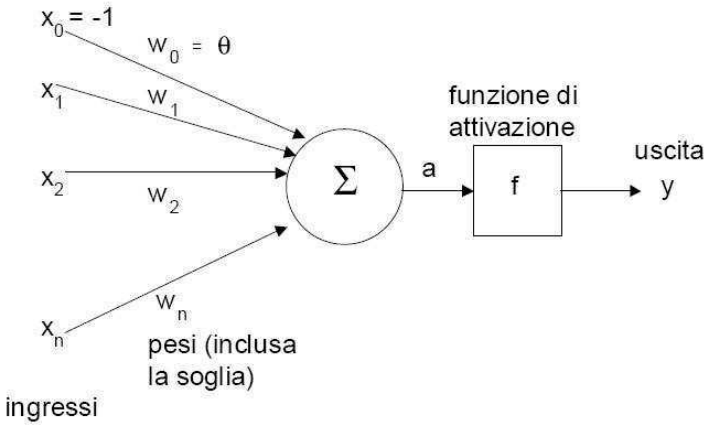


Figura 4.2 - *Modello di neurone*

Osserviamo che in alcuni casi, invece di considerare la soglia, si considera l'opposto della soglia, detto *bias*, che può quindi essere visto come il peso associato ad un ulteriore canale di ingresso di valore costante pari a 1. Avremo ancora:

$$y = f(a) = f(\sum_{i=1}^{n} w_i x_i) \tag{4.5}$$

dove $x_1 = 1$ e $w_1 = b$, in cui b è proprio il *bias*.

4.1.2.1 Funzione di attivazione

La funzione f di attivazione definisce l'uscita di un neurone in funzione del livello di attivazione a.

L'uscita può essere un numero reale, un numero reale appartenente ad un certo intervallo (ad esempio, [0, 1]), oppure un numero appartenente ad un insieme discreto (tipicamente, {0, 1} oppure {−1, +1}).

Vediamo alcuni esempi di funzione di attivazione.

- Funzione a soglia (Figura 4.3)

  L'uscita di un neurone che usa una funzione di attivazione a soglia è:

$$f(a) = \begin{cases} 1 \, se \; a \geq 0 \\ 0 \, se \; a < 0 \end{cases}$$

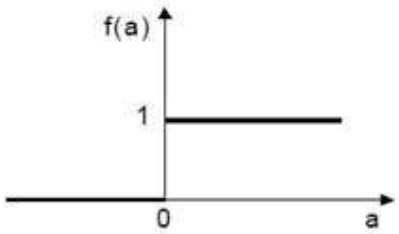


Figura 4.3 - *Funzione a soglia*

- Funzione lineare (Figura 4.4)

$$f(a) = a$$

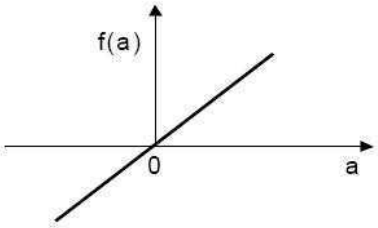


Figura 4.4 - *Funzione lineare*

- Funzione lineare a tratti (Figura 4.5)

$$f(a) = \begin{cases} 0 & se \quad a \le -0{,}5 \\ a + 0{,}5 & se \quad -0{,}5 < a < 0{,}5 \\ 1 & se \quad a \ge 0{,}5 \end{cases}$$

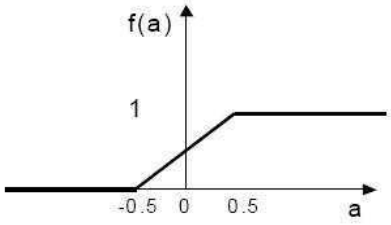


Figura 4.5 - *Funzione lineare a tratti*

- Funzione sigmoide (Figura 4.6)

Assieme alla funzione di soglia, le funzioni sigmoidi sono tra le più usate. Un esempio di funzione sigmoide è la funzione logistica, definita come:

$$f(a) = \frac{1}{1 + e^{-a}} \tag{4.6}$$

Osserviamo che, mentre una funzione a soglia assume solo il valore 0 e 1, una funzione sigmoide assume tutti i valori da 0 a 1. Notiamo, inoltre, che la funzione sigmoide è derivabile, mentre la funzione a soglia non lo è.

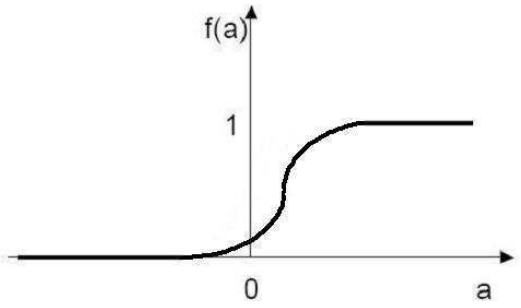


Figura 4.6 - *Funzione sigmoide*

- Funzione segno

Le funzioni di attivazione viste finora assumono valori tra 0 e +1 (esclusa la funzione lineare). A volte è opportuno che la funzione di attivazione assuma valori tra -1 e +1. In particolare, la funzione a soglia è così ridefinita:

$$f(a) = \begin{cases} -1 & se \;\; a < 0 \\ 0 & se \;\; a = 0 \\ 1 & se \;\; a > 0 \end{cases}$$

Tale funzione è nota come “funzione segno”.

4.1.3 Architettura di una rete neurale

Si possono identificare diversi tipi di architettura di rete. In questo lavoro ne presentiamo due.

4.1.3.1 Reti completamente connesse (non stratificate)

In una rete completamente connessa ogni neurone è connesso (in modo bidirezionale) con tutti gli altri (Figura 4.7).

Le connessioni tra i neuroni di una rete completamente connessa sono rappresentate mediante una matrice quadrata W, di dimensione pari al numero di neuroni, il cui generico elemento w_{ij} rappresenta il peso della connessione tra il neurone i ed il neurone j.

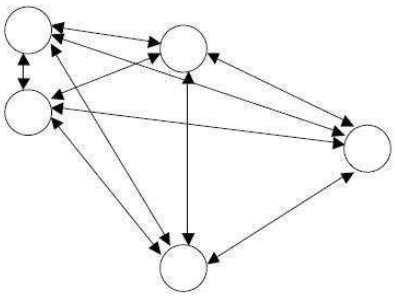


Figura 4.7 - *Esempio di rete completamente connessa.*

4.1.3.2 Reti stratificate

Nelle reti stratificate si individuano degli strati di neuroni tali che ogni neurone è connesso con tutti quelli dello strato successivo, ma non esistono connessioni tra i neuroni all'interno dello stesso strato, né tra neuroni di strati non adiacenti (vedi Figura 4.8).

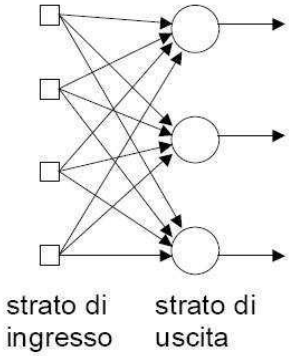


Figura 4.8 - *Esempio di rete stratificata*

Il numero di strati ed il numero di neuroni per strati dipendono dallo specifico problema che si intende risolvere.

Dato che nello strato di ingresso non avviene alcuna computazione (i neuroni di ingresso devono semplicemente passare allo strato successivo i segnali ricevuti dall'ambiente esterno), la rete rappresentata in Figura 2.1 viene di solito considerata come una rete con un solo strato. Inoltre, dato che i segnali viaggiano dallo strato di ingresso verso lo strato di uscita, si parla di rete *feedforward*.

Nella successiva Figura 4.9 viene mostrata una rete stratificata *feedforward* contenente uno strato nascosto, cioè uno strato i cui neuroni non comunicano direttamente con l'esterno. In generale, possono esserci uno o più strati nascosti. I neuroni nascosti permettono alla rete di costruire delle opportune rappresentazioni interne degli stimoli in ingresso in modo da facilitare il compito della rete.

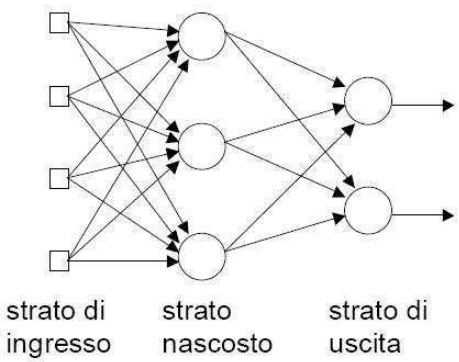


Figura 4.9 - *Esempio di rete stratificata con strato nascosto.*

Le connessioni tra i neuroni di una rete stratificata sono rappresentate mediante tante matrici quante sono le coppie di strati adiacenti. Ogni matrice contiene i pesi delle connessioni tra le coppie di neuroni di due strati adiacenti.

4.1.4 Modalità di attivazione dei neuroni

A seconda dei modelli di rete neurale, un solo neurone per volta può attivarsi oppure tutti i neuroni possono attivarsi contemporaneamente. Nel primo caso si parla di attivazione asincrona, mentre nel secondo di attivazione sincrona o parallela. In particolare, nell'attivazione asincrona il neurone che può attivarsi è scelto in modo casuale.

4.1.4.1 Esempio: la funzione XOR

Costruiamo una rete neurale che calcola l'OR esclusivo (XOR) che è definito dalla Tabella 4.1.

Tabella 4.1 - *Tabella logica XOR*

Ingresso x_1	**Ingresso** x_2	**Uscita**
1	1	0
1	0	1
0	1	1
0	0	0

Consideriamo una rete con uno strato nascosto (Figura 4.10). Per calcolare l'uscita da ogni neurone dello strato nascosto e dello strato di uscita usiamo la funzione a soglia. Ricordiamo che l'uscita di un neurone nello strato di ingresso coincide con il segnale in ingresso a tale neurone.

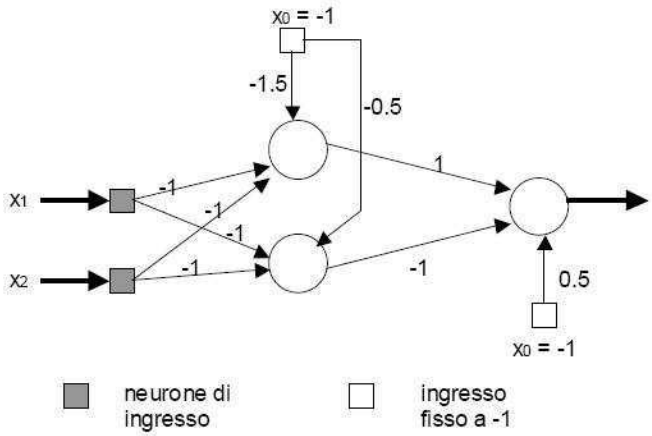


Figura 4.10 - *Rete per il calcolo dello XOR*

Diamo in ingresso alla rete la coppia [1, 1]. Per il primo neurone nascosto con soglia −1,5, abbiamo:

$$\begin{aligned} a &= [x_0 \cdot (-1{,}5)] + [x_1 \cdot (-1)] + [x_2 \cdot (-1)] = \\ &= [-1 \cdot (-1{,}5)] + [1 \cdot (-1)] + [1 \cdot (-1)] = -0{,}5 \end{aligned} \tag{4.7}$$

quindi l'uscita è 0. Per il secondo neurone nascosto con soglia −0,5 abbiamo:

$$\begin{aligned} a &= [x_0 \cdot (-0{,}5)] + [x_1 \cdot (-1)] + [x_2 \cdot (-1)] = \\ &= [-1 \cdot (-0{,}5)] + [1 \cdot (-1)] + [1 \cdot (-1)] = -1{,}5 \end{aligned} \tag{4.8}$$

quindi l'uscita è 0. Per il neurone di uscita con soglia 0.5 abbiamo:

$$\begin{aligned} a &= [x_0 \cdot (+0{,}5)] + [x_1 \cdot (+1)] + [x_2 \cdot (-1)] = \\ &= (-1 \cdot 0{,}5) + [0 \cdot (-1)] + [0 \cdot (-1)] = -0{,}5 \end{aligned} \tag{4.9}$$

quindi l'uscita è 0.

Facendo i calcoli con gli altri ingressi della Tabella 4.1 che definisce il connettivo logico XOR, otteniamo le uscite indicate nella stessa tabella.

4.1.5 Apprendimento supervisionato

Con riferimento all'esempio precedente, possiamo osservare che il corretto funzionamento della rete neurale dipende dall'architettura della rete (cioè dal numero di strati e dal numero di neuroni per strato), dalla funzione di attivazione dei neuroni e dai pesi. I primi due parametri sono fissati prima della fase di addestramento. Il compito dell'addestramento è quindi quello di aggiustare i pesi in modo che la rete produca le risposte desiderate.

Uno dei modi più usati per permettere ad una rete di imparare è l'apprendimento supervisionato, che prevede di presentare alla rete per ogni esempio di addestramento la corrispondente uscita desiderata.

Di solito i pesi vengono inizializzati con valori casuali all'inizio dell'addestramento. Poi si cominciano a presentare, uno alla volta, gli esempi costituenti l'insieme di addestramento (*training set*). Per ogni esempio presentato si calcola l'errore commesso dalla rete, cioè la differenza tra l'uscita desiderata e l'uscita effettiva della rete. L'errore è usato per aggiustare i pesi. Il processo viene di solito ripetuto ripresentando alla rete, in ordine casuale, tutti gli esempi del *training set* finché l'errore commesso su tutto il *training set* (oppure l'errore medio sul *training set*) risulta inferiore ad una soglia prestabilita.

Dopo l'addestramento la rete viene testata controllandone il comportamento su un insieme di dati, detto test set, costituito da esempi non utilizzati durante la fase di *training*. La fase di test ha quindi lo scopo di valutare la capacità di generalizzazione della rete neurale.

Si dice che la rete ha imparato, quando è in grado di fornire risposte anche per ingressi che non le sono mai stati presentati durante la fase di addestramento.

Ovviamente le prestazioni di una rete neurale dipendono fortemente dall'insieme di esempi scelti per l'addestramento. Tali esempi devono quindi essere rappresentativi della realtà che la rete deve apprendere ed in cui verrà utilizzata.

L'addestramento è, in effetti, un processo *ad hoc* dipendente dallo specifico problema trattato.

4.1.5.1 Delta Rule

Riferiamoci al neurone rappresentato in Figura 4.11.

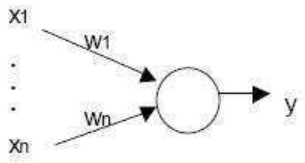


Figura 4.11 - *Neurone delta rule.*

La regola più usata per aggiustare i pesi di un neurone è la *delta rule* o regola di Widrow-Hoff. Sia $x = (x_1,...,x_n)$ l'ingresso fornito al neurone. Se t ed y sono, rispettivamente, l'uscita desiderata e l'uscita neurale, l'errore δ è dato da:

$$\delta = t - y \tag{4.10}$$

La delta rule stabilisce che la variazione del generico peso Δw_i è:

$$\Delta w_i = \eta \delta x_i \tag{4.11}$$

dove η è un numero reale compreso tra 0 ed 1 detto *learning rate*. Il *learning rate* determina la velocità di apprendimento del neurone. La delta rule modifica in maniera proporzionale all'errore solo i pesi delle connessioni che hanno contribuito

all'errore (cioè che hanno $x_i \neq 0$). Al contrario, se $x_i = 0$, non viene modificato poiché non si sa se ha contribuito all'errore.
Il nuovo valore dei pesi è quindi:

$$w_i \leftarrow w_i + \Delta w_i \tag{4.12}$$

4.1.5.2 Esempio di addestramento

Molti problemi, per poter essere risolti, richiedono di adattare una retta o una curva ai dati che si hanno a disposizione. Consideriamo, ad esempio, un insieme di punti nel piano che seguono l'andamento di una linea retta ma non appartengono esattamente ad alcuna linea retta (Figura 4.12).

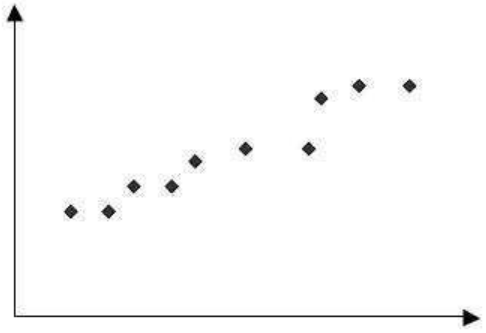


Figura 4.12 - *Punti di addestramento*

Possiamo interpolare i punti mediante una retta, calcolata, ad esempio usando il metodo dei minimi quadrati. Tale metodo ci permette di calcolare la retta che minimizza la somma degli errori quadratici per tutti i punti. Per ogni punto, l'errore è la distanza del punto dalla retta, come rappresentato in Figura 4.13.

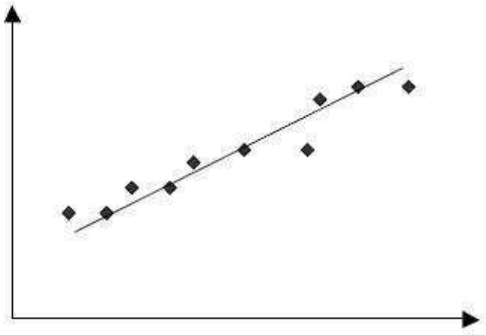


Figura 4.13 - *Retta dei minimi quadrati*

La retta disegnata in Figura 4.13 può essere utile per ricavare o per prevedere valori della variabile dipendente (rappresentata sull'asse delle ordinate) in corrispondenza di valori della variabile indipendente per cui non sono state fatte misurazioni.
Considerando che l'equazione di una retta in forma esplicita è:

$$y = m\,x + c \tag{4.13}$$

dove y ed x sono variabili, m la pendenza e c il punto in cui la retta incontra l'asse y, possiamo calcolare m e c con il metodo dei minimi quadrati risolvendo le equazioni ottenute uguagliando a 0 le derivate parziali (rispetto ad m e c, rispettivamente) della somma degli errori quadratici.

In alternativa al metodo dei minimi quadrati si può usare una rete neurale che approssima una retta. In ingresso alla rete vengono presentati i punti da approssimare con la linea retta e si lascia alla rete il compito di apprendere.

Possiamo usare una rete (rappresentata nella seguente Figura 4.14) costituita da un neurone di ingresso ed un neurone di uscita con funzione di attivazione lineare.

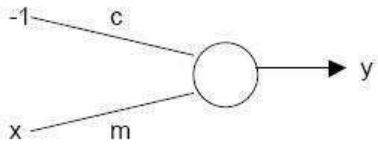


Figura 4.14 - *Esempio di una rete per approssimare una retta.*

Dato che la rete deve stimare m e c, questi ne costituiscono i pesi. Tali pesi saranno inizializzati in modo del tutto casuale. Gli esempi che costituiscono il *training set* sono le coppie (x, y) delle coordinate dei punti da approssimare con la linea retta; ovvero l'ascissa rappresenta l'ingresso e l'ordinata l'uscita desiderata. Il peso c è la soglia, quindi è associato ad un ingresso costantemente uguale a -1.

La rete viene addestrata usando la delta rule con un certo *learning rate* (ad esempio, $\eta = 0{,}1$). I punti del *training set* saranno presentati un certo numero di volte, finché l'errore commesso dalla rete non scende al di sotto di una soglia prestabilita.

L'uscita della rete produrrà una retta del tutto simile a quella generata col metodo dei minimi quadrati.

Diremo che la rete ha imparato perché dandole in ingresso l'ascissa di un punto non usato durante l'addestramento la rete produrrà in uscita l'ordinata corrispondente.

4.1.5.3 Classificazione

In molte applicazioni si incontrano problemi di classificazione di un insieme di oggetti, cioè occorre associare ogni oggetto alla classe corretta.

Supponiamo di voler classificare in due classi distinte degli oggetti rappresentati mediante punti nel piano. Se le due classi sono linearmente separabili, possiamo usare una rete neurale che approssima una retta di separazione tra le due classi. Un oggetto sarà quindi classificato rappresentandolo come punto nel piano ed assegnandolo a quella delle due classi individuata dal semipiano in cui cade il punto.

Tale classificazione può essere facilmente ottenuta addestrando una rete neurale con due ingressi ed un solo neurone di uscita con funzione di attivazione a soglia, come rappresentato in Figura 4.15.

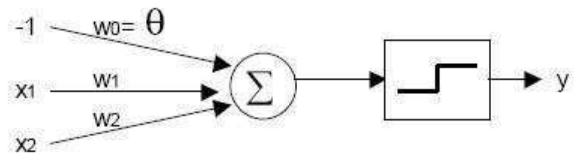


Figura 4.15 – *Esempio di perceptron*

Tale rete di Figura 4.15 è un esempio semplice di *perceptron*, costituito da più ingressi confluenti in un neurone di uscita con funzione di attivazione a soglia.
Consideriamo ad esempio la Figura 4.16. Possiamo associare la classe C_1 all'insieme di stimoli per cui la rete risponde con $y=1$ e C_2 all'insieme di stimoli per cui la rete risponde con $y=0$, cioè:

$x \in C_1 \quad se \quad y=1$
$x \in C_2 \quad se \quad y=0$

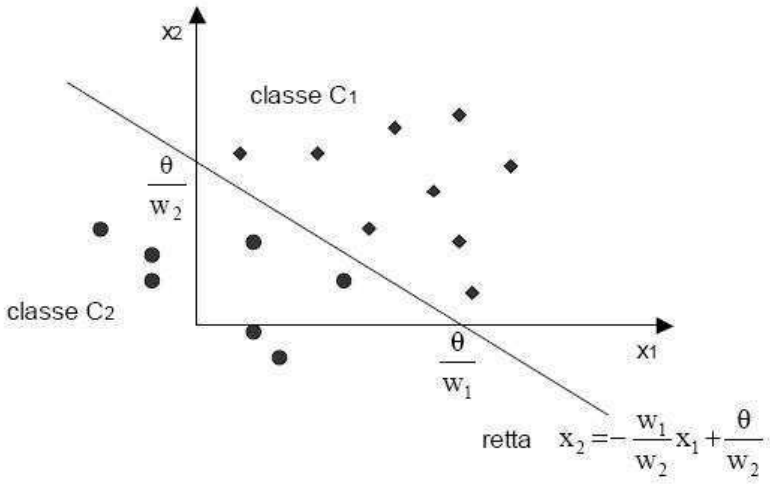


Figura 4.16 - *Esempio di utilizzo del perceptron.*

Nel piano (x_1, x_2) degli ingressi della rete, le classi C1 e C2 sono rappresentate da due semipiani separati dalla retta:

$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{\vartheta}{w_2} \tag{4.14}$$

Vediamo come è possibile addestrare il perceptron in modo che sia in grado di classificare correttamente i punti del piano. Dopo aver predisposto un opportuno *training set*, si fa uso della delta rule eseguendo i seguenti passi:

1. si inizializzano i pesi w_i con valori casuali;

2. si presenta alla rete un ingresso x_k insieme al valore t_k desiderato in uscita;
3. si calcola la risposta y_k della rete e si aggiornano i pesi mediante la delta rule;
4. si ripete il ciclo dal passo 2, finché la risposta della rete non risulti soddisfacente.

Con riferimento all'ultima Figura 4.16, osserviamo che il processo di apprendimento, modificando i pesi w_1, w_2 e ϑ, non fa altro che modificare la posizione e la pendenza della retta di separazione tra le due classi. Il processo termina quando la retta separa correttamente le due classi.

Infine, osserviamo che la rete considerata nell'esempio è un perceptron con due ingressi perché gli oggetti da classificare sono rappresentati come punti in $\Re^2$. Tali punti sono separati da una retta. In generale, se gli oggetti da classificare sono vettori $n-$dimensionali, gli ingressi del perceptron sono n e le due classi sono separate da un iperpiano in $\Re^n$.

4.1.5.4 Aggiornamento dei pesi e convergenza della Delta Rule.

Consideriamo la rete rappresentata in Figura 4.17, con n ingressi e p uscite.

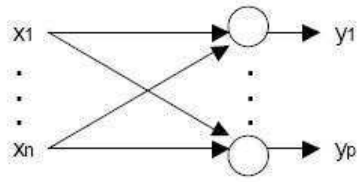


Figura 4.17 – *Rete neurale ad uno strato con n ingressi e p uscite.*

Siano t_j e o_j rispettivamente, l'uscita desiderata e l'uscita effettiva del neurone j. L'errore E_k commesso dalla rete sull'esempio k, può essere definito come:

$$E_k = \frac{1}{2}\sum_{j=1}^{p}(t_j - o_j)^2 \tag{4.15}$$

per cui l'errore globale commesso dalla rete su tutto il *training set* costituito da m esempi è:

$$E = \sum_{k=1}^{m} E_k \tag{4.16}$$

Per illustrare il razionale della delta rule, verrà preso in esame un caso semplice di comportamento lineare; anche se analiticamente tale situazione può essere affrontata con metodi diretti, faremo riferimento alla procedura iterativa della delta rule.

Consideriamo allora un singolo neurone, rappresentato nella successiva Figura 4.18, con due ingressi, soglia pari a 0 e funzione di attivazione lineare, quindi l'uscita coincide con l'ingresso: $f(a) = a$.

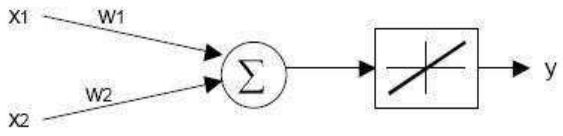


Figura 4.18 - *Esempio di singolo neurone con funzione di attivazione lineare.*

Tale neurone può modellare una qualunque linea retta passante per l'origine.
Per un neurone lineare ed un singolo esempio, l'errore diventa:

$$E = \frac{1}{2}(t-a)^2 \tag{4.17}$$

Considerando che $a = x_1 w_1 + x_2 w_2$ e svolgendo i calcoli otteniamo:

$$E = \frac{1}{2}(t^2 - 2ta + a^2) = \frac{1}{2}\left[t^2 - 2t(x_1 w_1 + x_2 w_2) + x_1^2 w_1^2 + 2x_1 w_1 x_2 w_2 + x_2^2 w_2^2\right] \tag{4.18}$$

Abbiamo ricavato che l'errore E , nel caso di neuroni lineari, è un paraboloide nello spazio dei pesi. In generale, per altri tipi di neuroni, sarà un'ipersuperficie nello spazio dei pesi (Figura 4.19).

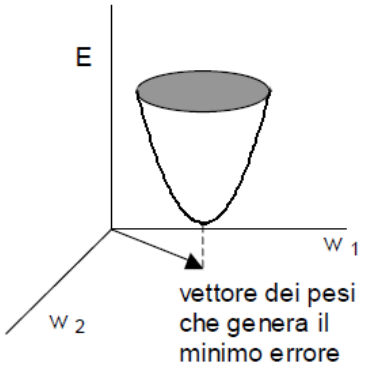


Figura 4.19 – *Paraboloide dell'errore nello spazio dei pesi.*

Prima di iniziare l'addestramento, i pesi sono inizializzati a valori casuali, quindi il punto che rappresenta lo stato iniziale della rete può trovarsi ovunque sulla superficie dell'errore (in generale non coinciderà con il punto di minimo di tale superficie). Durante l'addestramento i pesi dovranno essere modificati in modo da far muovere lo stato della rete lungo una direzione, che la delta rule individuerà essere quella di massima pendenza, della superficie dell'errore in modo da minimizzare l'errore globale.
Ricordiamo la definizione della delta rule con cui aggiustare i pesi:

$$\Delta w_{ij} = \eta \delta_j x_i \qquad \delta_j = (t_j - o_j)$$

dove t_j è l'uscita desiderata dal neurone j, o_j l'uscita effettiva, x_i il segnale proveniente dal neurone i, η il *learning rate* e Δw_{ij} la variazione del peso sulla connessione da i a j.

Vogliamo dimostrare che, aggiornando i pesi mediante la delta rule, l'apprendimento converge verso una configurazione dei pesi che minimizza l'errore quadratico globale.

Osserviamo che per dimostrare la convergenza della delta rule basterebbe dimostrare che tale regola è riconducibile alla forma:

$$\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}} \tag{4.19}$$

la quale varierebbe i pesi in modo da favorire la diminuzione dell'errore.

Infatti:

• se E cresce all'aumentare di w_{ij} (cioè $\partial E / \partial w_{ij} > 0$) allora w_{ij} viene diminuito per contrastare la crescita di E ($\Delta w_{ij} < 0$)

• se E diminuisce all'aumentare di w_{ij} (cioè $\partial E / \partial w_{ij} < 0$) allora w_{ij} viene aumentato per favorire la diminuzione di E ($\Delta w_{ij} > 0$)

Per semplicità, consideriamo un neurone lineare con uscita definita da:

$$o_j = \sum_j x_i w_{ij} \tag{4.20}$$

Esprimiamo la derivata dell'errore rispetto ad un peso come prodotto di due quantità, la prima delle quali esprime il cambiamento dell'errore in funzione dell'uscita di un neurone, la seconda riguarda il cambiamento dell'uscita rispetto ad un peso:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial w_{ij}} \tag{4.21}$$

Dall'equazione (4.15) e dalla definizione di δ_j otteniamo:

$$\frac{\partial E}{\partial o_j} = -\delta_j \tag{4.22}$$

Inoltre $\partial o_j / \partial w_{ij} = x_i$. Di conseguenza:

$$-\frac{\partial E}{\partial w_{ij}} = \delta_j x_i \tag{4.23}$$

A questo punto, inserendo il *learning rate*, otteniamo proprio la delta rule.

Con riferimento alla Figura 4.19, che mostra la superficie dell'errore, osserviamo che il processo di apprendimento può essere interpretato come una discesa su tale superficie lungo la linea di massima pendenza, individuata appunto dal gradiente:

$$-\Delta E = -\frac{\partial E}{\partial w_{ij}} \tag{4.24}$$

Il *learning rate* η rappresenta quindi la rapidità di discesa di E sulla superficie. È importante scegliere il valore giusto per η: un valore troppo piccolo può comportare un apprendimento troppo lento, mentre un valore troppo elevato può provocare oscillazioni dell'errore intorno al minimo. La soluzione tipicamente adottata è quella di stabilire un valore alto di η (prossimo a 1) all'inizio dell'addestramento e diminuire tale valore mano a mano che si procede con l'apprendimento.

Osservazione

La delta rule, che abbiamo ricavato riferendoci per semplicità ai neuroni lineari, è in realtà valida per qualsiasi tipo di neurone. Si parla, in effetti, di delta rule nel caso di neuroni lineari e di delta rule generalizzata per altri tipi di neuroni, tenendo però conto dell'eventuale forma della funzione dell'errore da generalizzare.

4.1.5.5 Problemi lineari e non lineari

Un problema di classificazione in cui si devono separare in due classi i punti appartenenti ad un certo insieme, si dice lineare se una linea (in due dimensioni) od un iperpiano (in n dimensioni) possono separare correttamente tutti i punti. In caso contrario, il problema si dice non lineare.

Abbiamo visto che un problema lineare può essere risolto usando un perceptron. Infatti, un perceptron con n ingressi è in grado di rappresentare un iperpiano $n-$dimensionale. Quindi, un perceptron è in grado di risolvere problemi linearmente separabili in cui gli ingressi devono essere catalogati in due differenti classi separabili tramite una retta (perceptron a due ingressi), un piano (perceptron a tre ingressi), o un iperpiano (perceptron a n ingressi).

Un tipico problema non lineare è l'or esclusivo (*XOR*). Tale operatore, infatti, produce 1 in uscita solo quando uno solo degli ingressi vale 1, altrimenti dà 0. Non esiste alcuna retta che separi i punti (0, 1) e (1, 0) dai punti (0, 0) e (1, 1). Per risolvere questo problema si hanno due possibilità:

- si ricorre a particolari funzioni di uscita non lineari;
- si usano reti con più strati.

Nel primo approccio si utilizza una rete con n ingressi ed un neurone di uscita la cui funzione di uscita sia scelta in modo appropriato. Un esempio di funzione di uscita non lineare adatta per i nostri scopi è la seguente:

$$y = (x_1 - x_2)^2 = \begin{cases} 1 & se\ x_1 \neq x_2 \\ 0 & se\ x_1 = x_2 \end{cases} \tag{4.25}$$

Ovviamente, questo approccio può essere difficile da perseguire qualora la funzione non lineare richiesta sia difficile da individuare.
Nel secondo approccio, si usa una rete con uno o più strati nascosti in modo da modellare due o più rette per separare i dati. Tale rete è detta *multilayer perceptron*.
Nel caso dello *XOR* la rete può essere quella già vista precedentemente e contenente due neuroni nascosti che rappresentano due rette ed un neurone di uscita che combina le informazioni prodotte dalle due rette.
Con riferimento alla rete già vista per risolvere lo XOR, consideriamo la seguente Tabella 4.2 che riporta gli ingressi ai neuroni nascosti e le relative uscite.

Tabella 4.2 - *Tabella logica XOR*

x1	x2	Neurone 1 ingressi strato nascosto	Neurone 2 ingressi strato nascosto	Neurone 1 uscita strato nascosto	Neurone 2 uscita strato nascosto
1	1	-0.5	-1.5	0	0
1	0	0	-0.5	1	0
0	1	0	-0.5	1	0
0	0	0	1	1	1

Possiamo osservare che gli ingressi alla rete risultano trasformati in uscita dallo strato nascosto: il primo livello di pesi (tra lo strato di ingresso e lo strato nascosto) ha spostato il punto originale (0, 1) nel punto (1, 0). Notiamo anche che (0, 0) e (1, 1) sono stati scambiati tra loro.
La Figura 4.20 rappresenta l'uscita dallo strato nascosto.

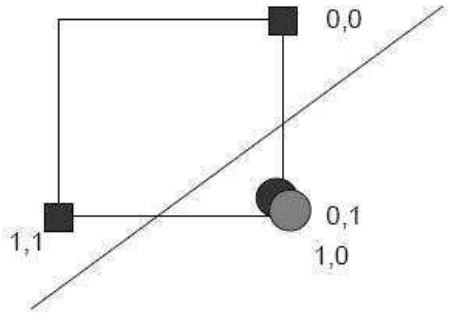


Figura 4.20 – *Rappresentazione dell'uscita dello strato nascosto.*

Anche il secondo livello di pesi (che connettono lo strato nascosto con lo strato di uscita) modella una retta. Affinché la rete produca gli output desiderati, occorre quindi che le configurazioni degli input al secondo livello di pesi siano separabili da una retta, come confermato dalla precedente Figura 4.20.

I pesi che definiscono la retta di separazione disegnata nella precedente Figura 4.20 sono quelli già visti quando abbiamo parlato per la prima volta del problema dello XOR.

Osservazione

Usando reti con uno strato nascosto è possibile formare regioni decisionali convesse nello spazio degli ingressi. Dato che ogni neurone di uno strato separa due regioni, il numero di lati della regione è minore od uguale al numero di neuroni dello strato nascosto. Ad esempio, possiamo ottenere le seguenti regioni in grigio (vedi Figura 4.21).

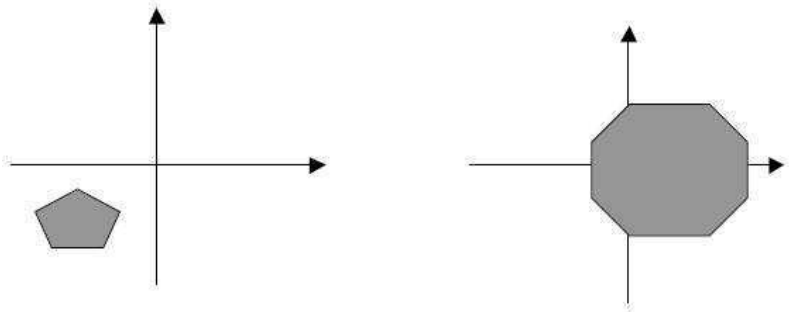


Figura 4.21 - *Regioni decisionali convesse ottenute con uno strato nascosto.*

Con due strati nascosti possiamo realizzare regioni decisionali complesse, ad esempio vedi Figura 4.22**.**

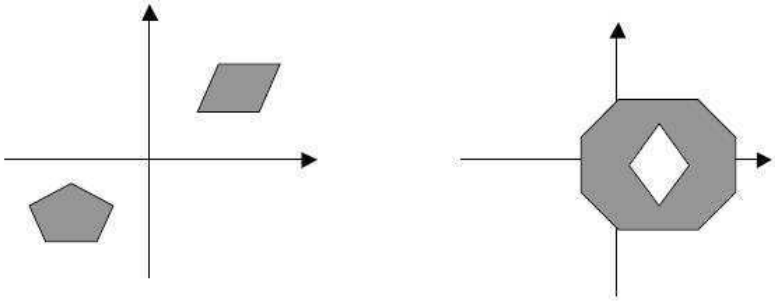


Figura 4.22 - *Regioni decisionali convesse ottenute con due strati nascosti.*

La maggior parte dei problemi è risolvibile adottando reti che prevedano al massimo due strati nascosti.

Osserviamo che la formazione di regioni decisionali complesse è possibile solo se si adottano funzioni di uscita non lineari. Infatti, una rete multistrato con neuroni lineari è equivalente ad una rete con uno strato di ingresso ed uno strato di uscita. Ad esempio, con riferimento ad una rete con uno strato nascosto e neuroni lineari, siano n, p e q il numero di neuroni dei tre strati di ingresso, nascosto ed uscita rispettivamente. Ricordando che, di solito, la matrice di connessione tra lo strato i e lo strato j si indica con W_{ij}, indichiamo con W_{21} e W_{32} le matrici dei pesi relative alla coppia di strati (ingresso-nascosto) e (nascosto-uscita), rispettivamente. W_{21} e

W_{32} hanno dimensioni $p \times n$ e $q \times p$, rispettivamente. L'uscita dallo strato nascosto è data da $Y = W_{32}X$, essendo X il vettore di ingresso. L'uscita dallo strato di uscita è $Z = W_{32}Y = W_{32}W_{21}X$. Definendo $W = W_{32}W_{21}$, otteniamo $Z = WX$, cioè la rete considerata è equivalente ad una rete senza strati nascosti con connessioni espresse da una matrice W, di dimensioni $q \times n$, che è il prodotto delle due matrici date.

4.1.6 Back Propagation

La rete multistrato che abbiamo adottato precedentemente per risolvere il problema del connettivo logico XOR è stata costruita definendo i pesi appropriati. Ovviamente, ciò che ci interessa trovare è un algoritmo di addestramento supervisionato che permetta alla rete multistrato di apprendere trovando autonomamente i pesi.
Il problema incontrato nell'addestramento delle reti multistrato è il seguente: volendo adottare un meccanismo di aggiornamento dei pesi simile alla delta rule (in cui l'errore è calcolato come differenza tra l'uscita desiderata e l'uscita effettiva di ciascun neurone) si riescono ad aggiornare solo i pesi relativi ai neuroni di uscita, ma non quelli relativi ai neuroni degli strati nascosti. Infatti, mentre per lo strato di uscita si conosce l'uscita desiderata (tale uscita viene data come secondo elemento delle coppie che costituiscono gli esempi del *training set*), niente si sa dell'uscita desiderata dei neuroni nascosti.
Questo problema è stato risolto, dopo molti anni di calo di interesse per le reti neurali dovuto proprio all'impossibilità di concretizzare l'addestramento, solo nel 1986, quando fu introdotto l'algoritmo di *back propagation*. Tale algoritmo prevede di calcolare l'errore commesso da un neurone dell'ultimo strato nascosto propagando all'indietro l'errore calcolato sui neuroni di uscita collegati a tale neurone. Lo stesso procedimento è poi ripetuto per tutti i neuroni del penultimo strato, e così via.

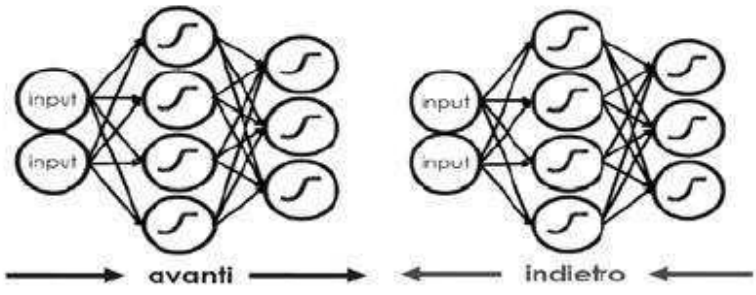


Figura 4.23 - *Le due fasi dell'algoritmo di back propagation.*

L'algoritmo di back propagation prevede che, per ogni esempio del *training set*, i segnali viaggino dall'ingresso verso l'uscita al fine di calcolare la risposta della rete. Vi è quindi una seconda fase durante la quale i segnali di errore vengono propagati all'indietro, sulle stesse connessioni su cui nella prima fase hanno viaggiato gli

ingressi, ma in senso contrario, dall'uscita verso l'ingresso (Figura 4.23). Durante questa seconda fase vengono modificati i pesi. I pesi sono inizializzati con valori casuali. Come funzione di uscita non lineare dei neuroni della rete si adotta in genere la funzione sigmoide (l'algoritmo richiede che la funzione sia derivabile). Tale funzione produce valori tra 0 ed 1.

L'algoritmo di back propagation usa una generalizzazione della delta rule.

Nel seguito useremo net per indicare la somma pesata degli ingressi in un neurone.

Possiamo esprimere la derivata dell'errore come segue:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} \tag{4.26}$$

Poniamo:

$$\delta_j = -\frac{\partial E}{\partial net_j} \tag{4.27}$$

Osserviamo che questa definizione coincide con la delta rule data nelle pagine precedenti. Lì, infatti, avevamo $\delta_j = -\partial E / \partial o_j$ perché consideravamo neuroni lineari, cioè $o_j = net_j$.

Riscriviamo l'equazione precedente:

$$\delta_j = -\frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j} \tag{4.28}$$

Poiché l'errore commesso sul k−esimo esempio del *training set* è:

$$E_k = \frac{1}{2}\sum_j (t_j - o_j)^2 \tag{4.29}$$

abbiamo:

$$\frac{\partial E}{\partial o_j} = -(t_j - o_j) \tag{4.30}$$

Per la funzione di attivazione f (di solito, la funzione logistica) l'uscita è:

$$o = f(net_j) \tag{4.31}$$

e quindi:

$$\frac{\partial o_j}{\partial net_j} = f'(net_j) \tag{4.32}$$

da cui:

$$\delta_j = (t_j - o_j) f'(net_j) \tag{4.33}$$

Essendo:

$$net_j = \sum_i x_i w_{ij} \tag{4.34}$$

si ha:

$$\frac{\partial net_j}{\partial w_{ij}} = x_i \tag{4.35}$$

per cui, tornando alla formula di partenza, risulta:

$$\frac{\partial E}{\partial w_{ij}} = -(t_j - o_j) f'(net_j) x_i = \delta_j x_i \tag{4.36}$$

Quindi, applicando il *learning rate*, abbiamo la seguente formula per la modifica dei pesi, sulla base della delta rule generalizzata:

$$\Delta w_{ij} = \eta \delta_j w_i \tag{4.37}$$

L'errore δ_j è calcolabile per un neurone di uscita, ma non per un neurone nascosto perché, come già detto, non conosciamo la sua uscita desiderata. Comunque, un neurone nascosto può essere adattato in modo proporzionale al suo contributo all'errore sullo strato successivo (verso l'uscita della rete).

Fissando l'attenzione su un neurone dell'ultimo strato nascosto, possiamo dire che l'errore commesso da tale neurone può essere calcolato come somma degli errori commessi da tutti i neuroni di uscita collegati a tale neurone nascosto. Il contributo di ciascuno di tali errori dipende, ovviamente, sia dalla dimensione dell'errore commesso dal relativo neurone di uscita, sia dal peso sulla connessione tra il neurone nascosto e il neurone di uscita. In altri termini, un neurone di uscita con un grosso errore contribuisce in maniera notevole all'errore di ogni neurone nascosto a cui è connesso con un peso elevato. Per un neurone nascosto l'errore è dato da:

$$\delta_j = f'(net_j) \sum_s \delta_s w_{js} \tag{4.38}$$

dove s è l'indice dei neuroni dello strato che trasmette all'indietro l'errore.

Come detto, una funzione spesso usata è la funzione logistica:

$$f(net_j) = \frac{1}{1 + e^{-net_j}} \tag{4.39}$$

La derivata di tale funzione è:

$$\begin{aligned} f'(net_j) &= \frac{e^{-net_j}}{(1 + e^{-net_j})^2} = \frac{1}{1 + e^{-net_j}} \left(1 - \frac{1}{1 + e^{-net_j}} \right) = \\ &= f(net_j)\left[1 - f(net_j)\right] \end{aligned} \tag{4.40}$$

Osservazione

Abbiamo già osservato, parlando del perceptron, che il *learning rate* η non deve avere valori troppo bassi né troppo alti per evitare, rispettivamente, tempi di addestramento troppo lunghi od oscillazioni dell'errore. Esistono due tecniche per risolvere questo problema. La prima è la stessa vista per il perceptron e prevede di variare η nel tempo. La seconda prevede di ridurre la probabilità di oscillazione dei pesi usando un termine α, detto *momentum*, che è una costante di proporzionalità (compresa tra 0 ed 1) alla precedente variazione dei pesi. La legge di apprendimento diventa quindi:

$$\Delta w_{ij}(n+1) = \eta \delta_j o_j + \alpha \Delta w_{ij}(n) \tag{4.41}$$

In questo modo, il cambiamento dei pesi per l'esempio $n+1$ dipende dal cambiamento apportato ai pesi per l'esempio n.

4.1.6.1 Algoritmo di back propagation

Dato un *training set* costituito da m esempi (X_k, T_k), l'addestramento di una rete multistrato, con funzioni di trasferimento sigmoidi, avviene tramite i seguenti passi:

1. si inizializzano i pesi con valori casuali (in genere con valori non troppo elevati);
2. si presenta un ingresso X_k e si calcolano le uscite:

   $$o_j = \frac{1}{1+e^{-net_j}} \tag{4.42}$$

   Di tutti i neuroni della rete.
3. dato T_k, si calcolano l'errore δ_j e la variazione dei pesi Δw_{ij} per ogni neurone dello strato di uscita:

   $$\delta_j = (t_j - o_j) f'(net_j) = (t_j - o_j) o_j (1 - o_j) \tag{4.43}$$

4. partendo dall'ultimo strato nascosto e procedendo all'indietro, calcolare:

   $$\delta_j = f'(net_j) \sum_s \delta_s w_{js} = o_j (1 - o_j) \sum_s \delta_s w_{js} \tag{4.44}$$

5. per tutti gli strati aggiornare i pesi:

   $$\Delta w_{ij}(n+1) = \eta \delta_j o_j + \alpha \Delta w_{ij}(n) \tag{4.45}$$

6. ripetere il processo dal punto 2 finché non si siano presentati tutti gli m esempi del *training set*;
7. si calcola l'errore medio sul *training set* (oppure l'errore globale); se l'errore è al di sotto di una soglia prefissata, l'algoritmo termina (si è raggiunta la

convergenza), altrimenti si ripete un intero ciclo di presentazione del *training set*.

Un ciclo di presentazione degli esempi del *training set* è detto *epoca*. Esistono due modalità di applicazione dell'algoritmo di back propagation: nella modalità *batch* i pesi sono aggiornati dopo aver presentato alla rete tutti gli esempi del *training set*, nella modalità *on-line* (o *incrementale*) i pesi sono aggiornati dopo la presentazione di ogni esempio. Nell'algoritmo precedente si è fatto riferimento a quest'ultima modalità.

4.1.7 Possibili problemi durante il training

Uno dei problemi in cui si può incorrere durante il *training* è chiamato *overfitting*. Questo si verifica quando l'errore che compie la rete fornendole il *training set* è molto piccolo, ma diviene grande quando le si presentano nuovi dati. La rete ha memorizzato gli esempi forniti dall'insieme di allenamento, ma non ha imparato a generalizzare ed a dare buoni risultati in situazioni nuove. Uno dei motivi di *overfitting* può essere l'utilizzo di un numero troppo elevato di epoche di addestramento.

Un metodo per accrescere la generalizzazione è usare una rete che non sia più complessa di quanto richiesto dal problema. Questo in termini di architettura e di ordine della rete utilizzata, di numero di strati e di numero di neuroni. Aumentando la complessità e, quindi, il numero dei parametri di un modello, cresce, infatti, anche il suo numero di gradi di libertà: ciò significa che è possibile modificare i parametri in modo da far aderire maggiormente il comportamento del modello ai dati disponibili. Si creano, così, due necessità opposte: da un lato risulta evidente che, all'aumentare della complessità della rete, aumenta la probabilità di *overfitting*; dall'altro non si possono utilizzare neanche reti troppo semplici in quanto queste non sarebbero in grado di modellare in maniera adeguata il sistema. Purtroppo, però, è difficile conoscere a priori quanto dovrebbe essere complessa una rete per un'applicazione specifica. Esistono principalmente due altri metodi per incrementare la capacità di generalizzare di una rete di cui sia stata definita l'architettura: il metodo *Early Stopping* ed il metodo *Bayesian Regularization*, descritti, rispettivamente, nel paragrafo 4.1.7.1 e nel paragrafo 4.1.7.2 (Demuth *et al.*, 2009).

Per quanto riguarda, invece, la scelta dell'ordine e del tipo di rete, in letteratura esistono diversi modelli matematici che cercano di risolvere tale problema, ma sono complessi, computazionalmente costosi e non sempre immuni da errori. Un metodo alternativo consiste nel *trial and error approach*. Si considera, in prima analisi, la

rete più semplice possibile. Se questa non risulta adeguata, si considera una rete di complessità superiore e si ripete il procedimento finché non si trova una rete soddisfacente.

4.1.7.1 Early Stopping

L'*Early Stopping* è uno dei metodi per incrementare la capacità di generalizzare di una rete. Con questa tecnica i dati a disposizione sono divisi in tre insiemi. Il primo è l'insieme di allenamento o *training set*, che è il solo ad essere usato durante l'addestramento per calcolare il gradiente ed aggiornare il valore dei pesi e dei bias. Il secondo è l'insieme di validazione o *validation set*. Durante il processo di allenamento si controlla l'errore sul *validation set*. Questo, normalmente, decresce nella fase iniziale dell'allenamento, come l'errore sul *training set*, ma inizia ad aumentare quando la rete comincia a perdere la capacità di generalizzare. Quando l'errore sull'insieme di validazione cresce per un determinato numero di epoche, il *training* viene interrotto ed il valore dei pesi e dei *bias* è riportato a quello relativo al minimo dell'errore sull'insieme di validazione (vedi Figura 4.24). Il terzo insieme è quello di test, il *test set*, che non viene utilizzato durante l'allenamento, ma viene utilizzato per confrontare le prestazioni di reti diverse.

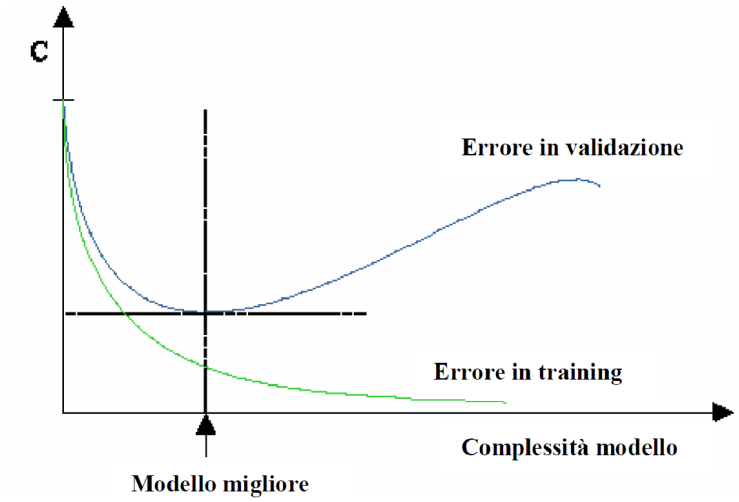


Figura 4.24 - *Processo di Early Stopping*

4.1.7.2 Bayesian Regularization

Il metodo *Bayesian Regularization* modifica la cifra di merito C, solitamente data dalla somma dei quadrati degli errori relativi ai dati del *training set*, ovvero:

$$C = MSE = \frac{1}{N}\sum_{i=1}^{N}(\varepsilon(i))^2 \tag{4.46}$$

con N numero degli output del *training set*.

Si modifica la cifra di merito aggiungendo il termine MSW, che consiste nella media della somma dei quadrati dei pesi e dei bias della rete, ottenendo:

$$MSE_{reg} = \gamma MSE + (1-\gamma) MSW \tag{4.47}$$

con:

$$MSW = \frac{1}{n}\sum_{j=1}^{n} w(j)^2 \tag{4.48}$$

n = numero di pesi e bias della rete;

γ = *performance ratio*.

L'uso di questa cifra di merito modificata porta la rete ad avere pesi e bias più piccoli e ciò si traduce in risposte più regolari e meno soggette ad overfitting. Con valori di γ troppo piccoli, la rete non riesce ad aderire in maniera sufficiente ai dati di *training*, mentre valori troppo grandi creano problemi di *overfitting*. Con il metodo Bayesian Regularization, il performance ratio γ è determinato in modo automatico ed ottimale attraverso metodi statistici. Inoltre la *Bayesian Regularization* permette di ottimizzare il numero dei parametri della rete realmente utilizzati, altro accorgimento che, come detto nel paragrafo 4.1.7, permette di evitare l'overfitting.

4.1.8 Pregi e difetti delle reti neurali

L'interesse nelle reti neurali è dovuto alla capacità di individuare relazioni numeriche, tralasciando l'individuazione fisica o formale del modello, e quindi, l'assunzione di ipotesi "a priori" sul comportamento ed i legami tra le variabili.

Lo sviluppo di reti neurali presenta altresì il pregio di lavorare simultaneamente sui dati e quindi di essere in grado di trattare anche una mole rilevante di informazioni. Si tratta in sostanza di un sofisticato sistema di tipo statistico dotato di una buona immunità al rumore; se alcune unità del sistema dovessero funzionare male, la rete nel suo complesso avrebbe delle riduzioni di prestazioni ma difficilmente essa va incontro ad un blocco del sistema. Come per qualsiasi modellazione di processi, le reti neurali sono efficienti solo se le variabili in ingresso sono scelte con la dovuta cura. L'introduzione di input con alta correlazione con i dati in output facilita l'elaborazione statistica della rete, fornendo risultati migliori. Di rilevante importanza è la fase di addestramento della rete stessa; essa consta in un continuo confronto con degli output noti per la determinazione dell'influenza delle connessioni neurali attivate. Questa fase può richiedere molto tempo per determinare una condizione di convergenza accettabile, se il numero delle variabili analizzate risulta molto elevato.

Le reti neurali, oltre a richiedere una mole di dati molto elevata per la fase di apprendimento, pongono un rilevante problema di interpretazione dei risultati ottenuti, in quanto si rinuncia alla determinazione delle relazioni matematiche che legano le variabili in ingresso e quelle in uscita. Infatti, i modelli prodotti dalle reti neurali, anche se molto efficienti, non sono rappresentabili in linguaggio simbolico. I risultati vanno accettati "così come sono", da cui anche la definizione inglese delle reti neurali come tipica *black box*: in altre parole, a differenza di un sistema algoritmico, dove si può esaminare passo-passo il percorso che dall'input genera l'output, una rete neurale è in grado di generare un risultato valido, o comunque con un'alta probabilità di essere accettabile, ma non è possibile risalire a "come" e "perché" tale risultato sia stato generato.
Non esistono teoremi o modelli che permettano di definire la configurazione ottimale di una rete, quindi il progetto di una rete neurale dipende molto dall'esperienza.
Va infine tenuto conto del rischio di cadere in fenomeni di overfitting come descritto al paragrafo 4.1.7.

4.2 Random Forests

4.2.1 Classificazione e regressione ad albero, CART

Un modello per analizzare la relazione fra variabili è quello che va sotto il nome di "*classification and regression trees*" (CART). Sebbene l'idea di fondo nel caso di classificazione e nel caso di regressione sia la stessa, è preferibile analizzare separatamente gli alberi di classificazione da quelli di regressione.

4.2.1.1 Alberi di classificazione

Non sempre la variabile risposta y è continua: alle volte essa può essere una variabile nominale o ordinale, altre volte, pur essendo continua, può risultare più conveniente categorizzarla in due o più categorie. Per cui, formalmente: sia y la modalità della variabile risposta categoriale Y, con $y \in C = \{1,2,...,J\}$ dove $y = j$ indica che la Y assume il valore della j-esima modalità nominale o ordinale, mentre $\boldsymbol{x} \in D \subset R^d$.
Scopo dell'analisi è quello di costruire un classificatore, ossia una regola per cui a partire da un vettore di variabili esplicative $\boldsymbol{x}$ si possa associare un'etichetta $j \in C$ alla variabile d'interesse; quindi si definisce un classificatore come una funzione $d(\cdot)$, definita in $D \subset R^d$ e a valori in C, ossia $d: D \to C$, quindi $\forall \boldsymbol{x} \in D \; \exists! \; j \in C : d(\boldsymbol{x}) = j$.

Per costruire un classificatore è necessario considerare un campione di osservazioni $\mathcal{L} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, con la coppia $(\boldsymbol{x}_i, y_i)$ che indica i valori osservati sulla i-esima unità campionaria.
Dato un campione $\mathcal{L}$, si consideri una nuova unità $(\boldsymbol{x}, y)$, estratta dallo spazio $(D \times C)$ indipendentemente da $\mathcal{L}$. Si denota con la probabilità di errata classificazione, definita nel seguente modo:

$$R^*(d) = P\{d(\boldsymbol{x}) \neq y\} \tag{4.49}$$

con $d(\cdot)$ costruito per mezzo del campione $\mathcal{L}$.
Le tecniche per stimare $R^*(d)$ sono diverse, le principali sono: la stima per risostituzione $R(d)$, la stima tramite un *test set* $R^{ts}(d)$, la stima mediante *cross validation* $R^{cv}(d)$.
La stima per risostituzione $R(d)$, stima con lo stesso campione $\mathcal{L}$, per cui fornisce una stima distorta di tale parametro. Precisamente si ha:

$$R(d) = \frac{1}{n} \sum_{i=1}^{n} I_{\{d(\boldsymbol{x}_i) \neq y_i\}}. \tag{4.50}$$

La stima tramite *test set* è utilizzabile nel caso in cui si abbia un campione con molte osservazioni. Il campione si può allora suddividere in due sottoinsiemi: un sottoinsieme $\mathcal{L}_1$ di n_1 osservazioni (*training set*) usato per costruire $d(\cdot)$ ed un sottoinsieme $\mathcal{L}_2$ di n_2 osservazioni (*test set*) per stimare $R^*(d)$.
La stima di $R^*(d)$ attraverso il *test set* è:

$$R^{ts}(d) = \frac{1}{n_2} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{L}_2} I_{\{d(\boldsymbol{x}_i) \neq y_i\}} \tag{4.51}$$

Se il campione non è molto grande la stima mediante *test set* è preclusa, ma è possibile usare il metodo del *cross validation*: si suddivide il campione $\mathcal{L}$ in V sottoinsiemi disgiunti $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_V$ tali che $\bigcup \mathcal{L}_v = \mathcal{L}$; per ogni sottoinsieme $\mathcal{L}^{(v)} = \mathcal{L} - \mathcal{L}_v$ si costruisce $d^{(v)}(\cdot)$ e si definisce:

$$R^{ts}(d^{(v)}) = \frac{1}{n_v} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{L}_v} I_{\{d^{(v)}(\boldsymbol{x}_i) \neq y_i\}} \tag{4.52}$$

dove n_v è la numerosità di $\mathcal{L}_V$; per cui si stima $R^*(d)$ con $R^{cv}(d)$, ossia:

$$R^{cv}(d) = \frac{1}{V} \sum_{v=1}^{V} R^{ts}(d^{(v)}). \tag{4.53}$$

La classificazione ad albero o *classification tree* è una particolare tecnica per costruire una regola di classificazione $d(\cdot)$. Gli alberi di classificazione sono costruiti con ripetuti *splits* (divisioni) di sottoinsiemi di D in due sottoinsiemi discendenti, allo scopo di classificare le unità in gruppi omogenei al loro interno e quanto più

possibile differenziati. È possibile associare a tali modelli una rappresentazione grafica a forma di albero in cui, partendo dal nodo radice (dove l'insieme D non è stato ancora suddiviso), si diramano una serie di nodi e rami (Figura 4.25); ogni nodo h rappresenta un particolare sottoinsieme di D ed il nodo radice h_1 è uguale a D.

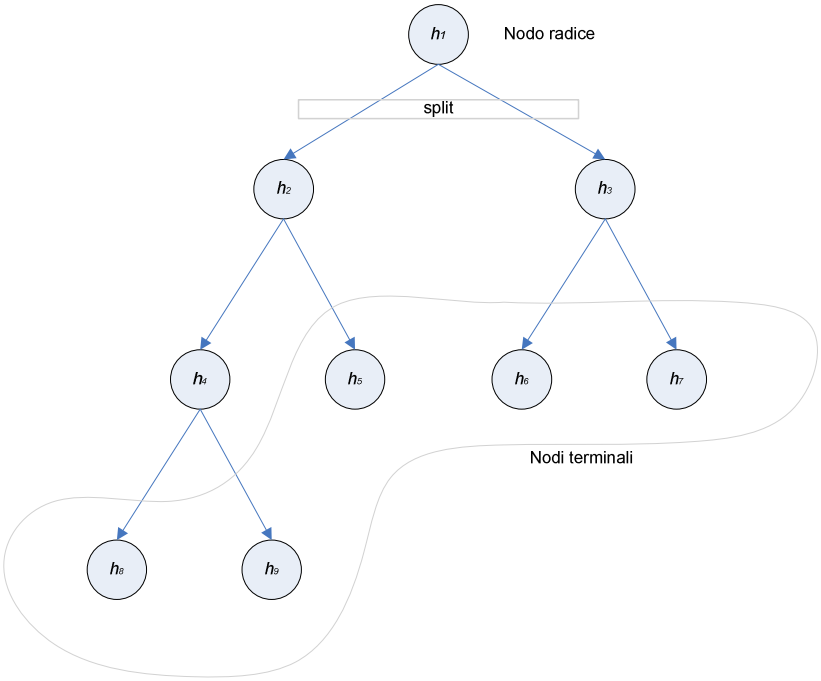


Figura 4.25 - *Rappresentazione grafica di un albero di classificazione.*

I nodi che non subiscono ulteriori *splits* sono detti nodi terminali dell'albero, essi formano una partizione di D e ad ogni nodo terminale è associata una modalità per y. I passi fondamentali per la costruzione di un albero di classificazione sono:

1. la selezione degli *splits*;
2. la decisione su quando dichiarare un nodo terminale o continuare a dividerlo;
3. l'assegnazione ad ogni nodo terminale di una modalità per y.

Partendo dal nodo radice h_1 si effettua una successione di *splits*; in ogni nodo si sceglie lo *split* che rende più omogenei i dati all'interno dei due nodi discendenti.
Per selezionare tale *split* si dà luogo alla seguente procedura:

passo 1: si definisce la probabilità che $y = j$ dato che $\boldsymbol{x} \in h$, tale che:

$$\sum_{j=1}^{J} P(j \mid h) = 1 \qquad \forall h; \tag{4.54}$$

se $n(h)$ è il numero di casi di $\mathcal{L}$ con $\boldsymbol{x}_i \in h \subset D$ e $n_j(h)$ è il numero di osservazioni con $\boldsymbol{x}_i \in h$ e $y_i = j$, allora si stima $P(j \mid h)$ con $p(j \mid h)$ definita:

$$p(j \mid h) = \frac{n_j(h)}{n(h)} \tag{4.55}$$

passo 2: si definisce quindi una misura $i(h)$ di *impurità per il nodo* h, come una funzione $\phi(\cdot)$ non negativa di $\{p(j \mid h)\}$, tale che:

$$i(h) = \phi(p(1 \mid h), p(2 \mid h), \ldots, p(J \mid h)) \tag{4.56}$$

con le caratteristiche:

a) $\phi(1/J, 1/J, \ldots, 1/J) = \max$;

b) $\phi(1,0,\ldots,0) = \phi(0,1,\ldots,0) = \ldots = \phi(0,0,\ldots,1) = 0$;

c) $\phi(\cdot)$ funzione simmetrica di $p(1 \mid h), \ldots, p(J \mid h)$;

ossia l'impurità è maggiore quando tutte le classi sono mischiate insieme e nulla quando il nodo contiene dati di una sola classe. L'indice di impurità più noto è quello di Gini, dove:

$$\begin{aligned} i(h) &= \phi(p(1 \mid h), p(2 \mid h), \ldots, p(J \mid h)) = \sum_{j \neq i} p(j \mid h)\, p(i \mid h) = \\ &= \left(\sum_j p(j \mid h) \right)^2 - \sum_j p^2(j \mid h) = 1 - \sum_j p^2(j \mid h). \end{aligned} \tag{4.57}$$

passo 3: si definisce un insieme S di *splits* binari s, per ogni nodo h, dove ogni *split* risponde alla domanda "$\boldsymbol{x} \in A$?", con $A \subset D$.

Nella metodologia CART (*Classification And Regression Trees*, Breiman *et al.*, 1984) ogni *split* dipende dal valore di una sola variabile esplicativa; se la variabile è continua allora lo *split* è determinato dalla domanda "$x_k < c$", con x_k componente di $\boldsymbol{x} \in D$ e $c \in R$; se invece x_k è categoriale con valori $\{b_1, b_2, \ldots, b_L\}$, allora la domanda è del tipo "$x_k \in S$?", con $S \in \mathcal{P}(\{b_1, b_2, \ldots, b_L\})$, cioè l'insieme di tutti i possibili sottoinsiemi di $\{b_1, b_2, \ldots, b_L\}$, detto insieme delle parti.

Ogni nodo h viene diviso dallo *split* s in h_L ed h_R, con p_L proporzione dei casi di h che va in h_L e la restante proporzione p_R in h_R, a seconda che $\boldsymbol{x}_i \in h$ risponda "SI" o "NO" alla domanda.

In ogni nodo h si sceglie quello split s^* che massimizza il decremento in impurità, cioè:

$$\Delta i(s,h) = i(h) - p_L\, i(h_L) - p_R\, i(h_R). \tag{4.58}$$

Dato un albero H e definito $\tilde{H}$ l'insieme dei nodi terminali di H, si pone $I(h) = i(h)\, p(h)$, dove:

$$p(h) = \sum_j p(j,h) = n(h)/n \tag{4.59}$$

è la proporzione delle n osservazioni che appartengono all'h-esimo nodo, quindi si definisce $I(H)$ l'*impurità dell'albero* H:

$$I(H) = \sum_{h \in \tilde{H}} I(h) = \sum_{h \in \tilde{H}} i(h)\, p(h)\,. \tag{4.60}$$

Considerato l'albero H, l'assegnazione di una modalità $j \in C \ \forall h \in \tilde{H}$ è effettuata mediante una funzione $j(h)$. Precisamente, se $p(j \mid h) = \max_i p(i \mid h)$, allora $j(h) = j$, ossia all'h-esimo nodo si assegna l'etichetta della variabile y più presente nel nodo considerato; se il massimo è raggiunto da più modalità, si sceglie arbitrariamente una fra queste.

Assegnata una modalità $j \in C$, per ogni nodo h, si può stimare la *probabilità di errata classificazione* dato che x appartiene all'h-esimo nodo, mediante la stima per risostituzione:

$$r(h) = 1 - \max_j p(j \mid h)\,, \tag{4.61}$$

quindi posto $R(h) = r(h)\, p(h)$, si determina la stima per risostituzione della *probabilità di errata classificazione dell'intero albero* H:

$$R(H) = \sum_{h \in \tilde{H}} R(h)\,. \tag{4.62}$$

Fondamentale è la proprietà in base alla quale più *splits* si considerano più piccolo diventa $R(H)$, in quanto per ogni *split* s di un nodo h in h_L ed h_R si ha:

$$R(h) \geq R(h_L) + R(h_R)\,. \tag{4.63}$$

Ciò implica che nel caso limite in cui $n(h) = 1 \ \forall h \in \tilde{H}$ risulti $R(H) = 0$; ma ciò vuol dire che l'albero è scarsamente generalizzabile, per cui il vero $R^*(H)$ tenderà ad aumentare se la cardinalità di $\tilde{H}$, $\left|\tilde{H}\right|$, è troppo elevata. La procedura per determinare una struttura dell'albero "generalizzabile" è quella di costruire un albero, detto $H_{\max}$, con tanti nodi terminali (ad esempio procedere a separare fin quando $n(h) \leq 5 \ \forall h \in \tilde{H}$ e successivamente "potare" l'albero. A tale scopo si considera una *funzione costo-complessità* $R_\alpha(H)$ da minimizzare, definita come:

$$R_\alpha(H) = R(H) + \alpha \left|\tilde{H}\right| \tag{4.64}$$

dove α è un parametro positivo che pesa la complessità del modello, espressa in termini di $\left|\tilde{H}\right|$; maggiore sarà il valore assunto da α, più semplice (cioè con meno nodi terminali e quindi *splits*) sarà l'albero H ottimale.

Per $\alpha = 0$, l'albero H che minimizza $R_\alpha(H)$ sarà $H_{\max}$; per $\alpha \to \infty$, l'albero minimizzante sarà $\{h_1\}$, ossia l'albero costituito dal solo nodo radice, senza nessuno *split*. Al variare di α fra i reali ci saranno diversi alberi che minimizzeranno $R_\alpha(H)$, precisamente $H(\alpha)$ è l'albero che minimizza $R_\alpha(H)$.

Anche se α ha una cardinalità del continuo, esisterà comunque un numero finito di alberi $H(\alpha)$; per cui $H(\alpha)$ minimizza $R_\alpha(H)$ per il valore di α, ma continuerà a minimizzarla per valori contigui di α, fino a quando sarà raggiunto un valore α' tale per cui $H(\alpha') \neq H(\alpha)$ minimizza $R_{\alpha'}(H)$.
Breiman (1984) ha dimostrato che esiste una sequenza di sottoalberi (cioè ottenuti per mezzo di successive potature) $H_1, H_2, H_3, \ldots, h_1$ che minimizzano $R_\alpha(H)$ al variare di α; precisamente che $\{\alpha_k\}$ è una successione crescente, cioè $\alpha_k < \alpha_{k+1}$, dove $\alpha_1 = 0$ e che per $\alpha_k < \alpha < \alpha_{k+1}$ si ha $H(\alpha) = H(\alpha_k) = H_k$.
Una volta ottenuta la sequenza di sottoalberi $H_1, H_2, H_3, \ldots, h_1$ che minimizzano $R_\alpha(H)$ al variare di α, si deve scegliere l'albero "migliore" fra questi.
L'ottimalità è definita mediante una stima della probabilità di errata classificazione diversa dalla stima per risostituzione, per cui questa si dovrà ottenere o attraverso l'uso di un *test set* o utilizzando il metodo di *cross validation*. Fra gli alberi $H_1, H_2, H_3, \ldots$ si sceglie l'albero H_{k_0} che ha il minimo valore di $\hat{R}(H_k)$, con $\hat{R}(\cdot) = R^{ts}(\cdot)$ o $\hat{R}(\cdot) = R^{cv}(\cdot)$.
La metodologia CART è una tecnica non parametrica, poiché non richiede la specificazione di una forma funzionale e presenta diversi vantaggi: non richiede che le variabili esplicative più rilevanti siano selezionate a priori, in quanto le variabili meno rilevanti non saranno considerate negli *splits* e non influenzeranno quindi l'analisi; i risultati saranno invarianti rispetto a trasformazioni monotone delle variabili esplicative, infatti se $\varphi(\cdot)$ è una funzione monotona, gli eventi $\{x_k \leq c\}$ e $\{\varphi(x_k) \leq \varphi(c) = c'\}$ sono identici e danno luogo allo stesso *split*; gli outliers fra le variabili esplicative non influenzano i risultati, in quanto gli splits avverranno per valori delle osservazioni diversi dai dati anomali.
Nel caso in cui siano presenti dati mancanti, la metodologia CART utilizza *splits* surrogati: cioè se non è possibile effettuare un determinato *split* su una certa variabile esplicativa, in quanto tale valore è mancante in una determinata unità, si effettua lo *split* su un'altra variabile, scelta in modo tale che i due *splits* abbiano un indice di similarità massimo in quel determinato nodo.

4.2.1.2 Regressione ad albero

Nel caso in cui la variabile y sia una variabile casuale continua, non si parla più di *classification trees*, ma di regressione ad albero o *regression trees*.
In tal caso l'obiettivo è quello di stimare la relazione fra la variabile y ed il vettore di variabili indipendenti $\underline{x}$, a partire da un campione $\mathcal{L}$; l'obiettivo non è più costruire una regola di classificazione $d(\cdot)$, ma stimare la funzione $f(\cdot)$, che lega y ed $\underline{x}$.

La regressione ad albero costruisce un albero H a partire dal nodo radice h_1, effettuando una successione di *splits* dell'insieme D, in modo da rendere più omogenee, in termini di y, le unità.
Gli *splits*, come nella classificazione ad albero, sono determinati in base alla risposta ad una domanda del tipo "$\{x_i \le c\}$?" e per ogni nodo h, si sceglierà lo *split* s^*, fra l'insieme S degli *splits* considerati, che massimizzerà il valore:

$$\Delta R(s,h) = R(h) - R(h_L) - R(h_R) \tag{4.65}$$

dove

$$R(h) = \frac{1}{N(h)} \sum_{y_t \in h} (y_t - \bar{y}_h)^2 , \tag{4.66}$$

con

$$\bar{y}_h = \frac{1}{N(h)} \sum_{y_t \in h} y_t \tag{4.67}$$

ed $N(h)$ pari al numero di osservazioni appartenenti al nodo h.
Si definisce quindi $R(H)$ la funzione d'errore dell'albero H:

$$R(H) = \sum_{h \in \tilde{H}} R(h) = \frac{1}{N} \sum_{h \in \tilde{H}} \sum_{y_t \in h} (y_t - \bar{y}_h)^2 . \tag{4.68}$$

La procedura di determinazione dell'albero "ottimale" è la stessa di quella vista nel paragrafo 4.2.1.1, in cui partendo da $H_{\max}$ si considera una funzione di costo-complessità $R_\alpha(H) = R(H) + \alpha \left| \tilde{H} \right|$ e si determina una sequenza di sottoalberi $H_1, H_2, H_3, \ldots$ che la minimizzano al variare di α; quindi, fra questi, si sceglie quell'albero H_k che minimizza la funzione d'errore $R(h_k)$ calcolata su un *test set* $\mathcal{L}_2$ (oppure col metodo *cross validation*), ossia:

$$R^{ts}(H_k) = \frac{1}{N_2} \sum_{(\underline{x}_t, y_t) \in \mathcal{L}_2} \left(y_t - \hat{f}_k(\underline{x}_t) \right)^2 \tag{4.69}$$

con $\hat{f}_k(\cdot)$ stima di $f(\cdot)$ considerando l'albero H_k.
L'albero H attribuisce ad ogni nodo terminale un valore per la variabile y ottenuto mediante la media aritmetica delle y delle osservazioni appartenenti a quel nodo.
L'albero di regressione considera:

$$f(\underline{x}) = \sum_{m=1}^{M} \beta_m I_{\{\underline{x} \in R_m\}} = \sum_{m=1}^{M} \beta_m \phi(\underline{x}, \alpha_m) \tag{4.70}$$

dove M è il numero di nodi terminali dell'albero e la funzione $I_{\{\cdot\}}$ è la funzione indicatrice, che assume valore 1 se si verifica l'argomento $\{\cdot\}$ e zero altrimenti.

Poiché i sottoinsiemi $R_1, R_2, \ldots, R_M$ costituiscono una partizione dell'insieme D, allora la funzione $f(\underline{x})$ implica che: $f(\underline{x}) = \beta_m$ se $\underline{x} \in R_m$.
Come detto prima, si ha:

$$\hat{\beta}_m = E(y \mid \underline{x} \in R_m) = \frac{\sum_{t:\underline{x}_t \in R_m} y_t}{\#\{t : \underline{x}_t \in R_m\}} \tag{4.71}$$

dove la funzione $\#\{\cdot\}$ è la funzione conteggio.

4.2.1.3 Pregi e difetti dei modelli CART

I metodi di classificazione e regressione ad albero, fondati su un approccio non parametrico, si sono rivelati, sin dalla loro introduzione negli anni 80, un utile strumento di analisi in processi di scoperta della conoscenza e di apprendimento supervisionato dai dati.
Il loro notevole successo è da ricercarsi in alcuni fattori chiave che rispondono in maniera efficace alle esigenze proprie di un processo di analisi dei dati:

- la capacità di trattare dati eterogenei, sia quantitativi che qualitativi;
- la capacità di analizzare *dataset* di grandi dimensioni;
- la relativa semplicità nell'implementazione dell'algoritmo di generazione degli alberi;
- la capacità dell'algoritmo di convergere in un numero finito di passi;
- la semplicità di interpretazione dei risultati, che sono rappresentati attraverso un grafico ad albero;
- assenza di ipotesi sulla forma distributiva sia della variabile risposta che dei predittori;
- invarianza rispetto a trasformazioni monotone dei predittori;
- trattamento dei dati mancanti mediante split surrogati.

Il problema principale degli algoritmi come il CART è il fatto che essi non tengono conto dell'influenza che la scelta di una particolare divisione ha sui futuri divisori. In altre parole, la decisione della divisione avviene ad ogni nodo dell'albero, in un preciso momento durante l'esecuzione dell'algoritmo, e non è mai più riconsiderata in seguito. Dato che tutte le suddivisioni vengono scelte sequenzialmente e ognuna di esse di fatto dipende dalle precedenti, si ha che tutte le divisioni sono dipendenti dal nodo radice dell'albero; una modifica del nodo radice potrebbe portare alla costruzione di un albero completamente differente.
Un'altra limitazione è dovuta alla discontinuità della funzione di previsione che si ottiene nel caso di modelli di regressione.

4.2.2 Random Forests

La tecnica denominata *Random Forests*, estensione dell'approccio relativo alla costruzione degli alberi di classificazione, è stata recentemente proposta da Breiman (2001). Anche in *Random Forests* non esiste un modello definito che leghi le variabili predittive alla variabile risposta e pertanto non esistono parametri da stimare. Mediante opportuni algoritmi si cerca di associare direttamente i predittori con la variabile risposta al fine di ottenerne una stima. L'accuratezza ottenuta mediante questo approccio è notevole (Breiman, 2001; Berk, 2008a; Berk, 2008b; Berk *et al.* 2009).

L'algoritmo *Random Forests* può essere visto come un insieme di procedure statistiche preesistenti. Vengono utilizzati gli alberi di classificazione e regressione (Breiman *et al*., 1984), presentati al paragrafo 4.2.1, come mattone costruttivo e si prende spunto dalle tecniche di *bootstrap* (Efron e Tibshirani, 1993) e di *bagging* (Breiman, 1996).

4.2.2.1 L'algoritmo Random Forests

Consideriamo i seguenti passi che illustrano l'algoritmo *Random Forests*. Si consideri un campione di *training* composto da N osservazioni per le quali siano osservate la variabile risposta ed un insieme di p predittori. L'algoritmo può essere sintetizzato nei seguenti passi.

1. Estrarre casualmente un campione di dimensione N dai dati di *training*, con reinserimento. Le unità non selezionate nel campione sono accantonate ed identificate come insieme *out-of-bag* (OOB) ed utilizzate come dati di test per uno specifico albero. Mediamente circa un terzo del campione originario appartiene all'insieme OOB. Questo passo utilizza la tecnica di *bootstrap*; studi recenti suggeriscono di utilizzare un campione di dimensione leggermente inferiore ad N .
2. Estrarre casualmente un piccolo campione dei predittori, senza reinserimento. Il numero dei predittori estratti, indicato con m , deve essere molto più piccolo di p e rappresenta uno dei parametri dell'algoritmo.
3. Utilizzando le osservazioni del campione estratto al passo 1 e considerando le variabili predittive estratte al passo 2, suddividere i dati in due sottoinsiemi come avviene con l'algoritmo CART. Se la variabile risposta è categoriale, lo *split* è scelto in modo da minimizzare l'indice di Gini. Se invece la variabile risposta è quantitativa, lo *split* e scelto in modo da

minimizzare la somma dei quadrati dei residui. È il primo passo per la costruzione di un albero di classificazione o di regressione.

4. Ripetere i passi 2 e 3 per tutte le suddivisioni successive finché la partizione risultante continua a migliorare l'adattamento del modello ai dati. L'opinione corrente prevede che l'albero debba crescere fino alla sua dimensione massima, anche se su questo il dibattito è aperto.
5. Nel caso di albero di classificazione, assegnare come di consueto la modalità della variabile risposta ad ogni nodo foglia utilizzando la moda dei nodi terminali. Nel caso di albero di regressione considerare invece l'usuale media condizionata. In entrambi i casi i valori possono essere considerati come la previsione della variabile risposta effettuata dall'albero.
6. Utilizzare i dati OOB sull'albero costruito per ottenere stime della variabile risposta. Nel caso di variabile categoriale si considera la modalità associata al nodo terminale nel quale termina la singola osservazione. Nel caso di variabile quantitativa si utilizza invece la media condizionata associata al nodo. Si noti che i dati OOB non sono stati utilizzati per la costruzione dell'albero e che quindi possono essere utilizzati per valutare la bontà della previsione dell'albero.
7. Ripetere molte volte (qualche centinaio) i passi da 1 a 6 per costruire molti alberi di classificazione o di regressione. Il numero degli alberi costruiti, indicato con K, è il secondo parametro dell'algoritmo.
8. Nel caso di variabile risposta di tipo categoriale, classificare ciascuna osservazione utilizzando il "voto di maggioranza" considerando soltanto gli alberi nei quali quell'osservazione è ricaduta nell'insieme OOB. Nel caso di variabile risposta quantitativa, assegnare ad ogni osservazione la media condizionale ottenuta considerando soltanto gli alberi nei quali questa è ricaduta nell'insieme OOB. Si ottiene quindi, per ogni osservazione presente nel *test* set, una previsione della variabile risposta.

Da quanto ottenuto in uscita dall'algoritmo *Random Forests* è possibile calcolare varie misure della bontà della previsione ottenuta. Nel caso categoriale, la misura potrebbe essere la proporzione dei casi di errata classificazione. Per risposte quantitative, la misura potrebbe essere la media dell'errore quadratico. Nella pratica può essere utile controllare nel dettaglio come si realizzano gli errori di classificazione; ad esempio per variabile risposta di tipo dicotomico può essere utile scindere gli errori in falsi positivi e in falsi negativi.

La tecnica *Random Forests* gode di alcuni pregi rispetto ai modelli di regressione tradizionali. La metodologia costruttiva dei CART è molto flessibile e consente di

cogliere delle relazioni di tipo non lineare, non note a priori, tra variabile risposta e predittori desumendole per via induttiva dagli stessi dati. Tuttavia, presi singolarmente, i CART possono risultare molto instabili, come esplicitato nel paragrafo 4.2.1.3. Questo fenomeno viene neutralizzato in *Random Forests* grazie alla mediazione dei risultati di molti alberi.
Un altro pregio è dato dal fatto che campionando solo un sottoinsieme dei predittori, gli alberi calcolati sono più "indipendenti" l'un l'altro, in termini di valori stimati della variabile risposta. Anche ai predittori meno "importanti" viene quindi data possibilità di contribuire alla spiegazione della variabile risposta.
Una proprietà importante dei *Random Forests* è la consistenza delle stime prodotte all'aumentare del numero di alberi generati (Breiman, 2001). Questo implica che non vi è il rischio di ricadere in problemi di *overfitting* con l'aumento del numero di alberi, a differenza di quanto accade con altri metodi esplorativi.
Infine, per variabile risposta categoriale, con *Random Forests* è possibile introdurre una funzione di costo relativa differenziata per falsi positivi e falsi negativi direttamente nell'algoritmo costruttivo. Molti altri modelli di regressione assumono invece che il costo sia il medesimo.

4.2.2.2 Ottimizzazione dei parametri

A dispetto della complessità dell'algoritmo *Random Forests* e dell'elevato numero dei parametri potenziali, nella pratica i valori di default danno buoni risultati. I parametri che solitamente necessitano di un'attività di regolazione sono i seguenti.

1. Dimensione del nodo terminale. A differenza di quanto accade nell'algoritmo CART, il numero delle osservazioni presenti nei nodi terminali può essere molto basso. L'elevata variabilità che ne consegue viene mitigata dall'operazione di mediazione tra gli alberi, insita nell'algoritmo. L'implementazione di quest'ultimo realizzata per il software R prevede come valori di default per la dimensione del nodo terminale: 1 nel caso di variabile risposta categoriale e 5 nel caso quantitativo.
2. Numero degli alberi generati. Per questo parametro bisogna considerare un valore che va da un minimo di alcune centinaia fino ad un massimo di alcune migliaia, un valore tipico è 500.
3. Numero dei predittori campionati. Questo sembrerebbe essere un parametro chiave per quanto riguarda la bontà dei risultati ottenuti dal modello. Sebbene possa sembrare sorprendente, è necessario campionare un numero molto basso di parametri ad ogni *split*. Utilizzando un numero molto elevato di alberi, ogni predittore ha comunque un'ampia possibilità di

contribuire alla realizzazione del modello, anche se solo poche variabili vengono selezionate ad ogni suddivisione. Un valore consigliato dallo stesso Breiman è pari alla radice quadrata del numero dei predittori. È possibile poi tentare con valori leggermente inferiori o leggermente superiori valutando le differenze di prestazioni. Nell'implementazione di R esiste la possibilità di cercare il valore ottimale valutando la bontà di ogni valore utilizzando i dati OOB. Dal punto di vista teorico questa procedura sembra molto valida, in pratica tuttavia la variazione dei risultati è modesta. Occorre inoltre prestare attenzione a non esagerare nella ricerca di della bontà dell'adattamento poiché si corre il rischio di ricadere nell'*overfitting*, fenomeno che la realizzazione di *Random Forests* cerca di evitare per costruzione.

Una delle caratteristiche che tipicamente contribuiscono alla variabilità dei risultati ottenuti è il peso che viene dato ai falsi positivi e ai falsi negativi. Anche se questo non deve essere considerato un parametro per la regolazione dell'adattamento ma invece un'espressione di un'esigenza del fenomeno osservato, va tuttavia tenuto in considerazione tali pesi influenzano considerevolmente il risultato ottenuto e pertanto vanno analizzati con la dovuta attenzione.

4.2.2.3 Importanza dei predittori

Sebbene il punto di forza di *Random Forests* sia la previsione della variabile risposta, vi è naturalmente un forte interesse a capire quale sia l'importanza dei singoli predittori nel concorrere alla bontà della stima. Questa informazione è desumibile attraverso un secondo algoritmo schematizzato di seguito.

1. Nel caso di variabile risposta di tipo categoriale, calcolare la modalità stimata per tutte le osservazioni, considerando solo gli alberi nei quali queste ricadono nell'insieme OOB. Nel caso di variabile risposta di tipo continuo, calcolare, per ciascuna osservazione, il valore medio delle stime della risposta considerando solo gli alberi nei quali queste ricadono nell'insieme OOB.
2. Nel caso di variabile risposta di tipo categoriale, calcolare la proporzione dei casi di errata classificazione sul totale dei casi. Per variabile risposta di tipo continuo calcolare l'errore quadratico medio. Questi valori vengono utilizzati come riferimento per misurare l'accuratezza delle stime.
3. Fissato un predittore, permutarne in modo casuale tutti i valori tra le osservazioni. Questa operazione ha lo scopo di rendere incorrelato il predittore fissato con gli altri predittori e con la variabile risposta.

4. Ripetere il passo 1 sull'insieme di dati modificati.
5. Ripetere il passo 2 sull'insieme di dati modificati. Questo fornisce una misura dell'accuratezza delle stime dopo la permutazione dei valori del predittore.
6. Calcolare l'aumento dell'errore di previsione confrontando i risultati del passo 5 con quelli del passo 2. L'aumento è una misura dell'importanza del predittore ai fini predittivi della variabile risposta.
7. Ripetere i passi dal 3 al 6 per ciascun predittore.

Tramite questo algoritmo è possibile attribuire a ciascun predittore un fattore di importanza misurato in termini di aumento percentuale dell'errore di stima.

4.2.2.4 Funzione di previsione

La conoscenza del contributo dei singoli predittori nel concorrere alla stima della variabile risposta è certamente un'informazione utile ma può non essere sufficiente. È altrettanto importante conoscere come ciascun predittore influisca sulla variabile risposta quando tutti gli altri predittori rimangono costanti. Per ottenere questo risultato è necessario il terzo algoritmo illustrato di seguito.

1. Fissato un certo predittore che presenti V modalità, si costruiscono V insiemi di dati (*dataset*). In ciascuno di essi il predittore fissato assume su tutte le unità il valore costante v, mentre gli altri predittori mantengono i valori originari. Un esempio può chiarire meglio questo passaggio: supponiamo che il predittore fissato sia l'età e che le modalità siano 40 differenti anni (ad es. da 20 a 59). Si costruiscono 40 *dataset*, uno per ciascuna età, in ciascuno dei quali la variabile età sarà impostata al valore costante (per quel *dataset*) di ciascuna delle 40 modalità. Il primo *dataset* avrà l'età fissata a 20, il secondo a 21 e così via. Gli altri predittori avranno i valori originari, replicati tante volte quanti sono i *dataset*. In questo senso possiamo dire che gli altri predittori sono "costanti" tra un *dataset* e l'altro e che, considerando l'insieme di tutti i *dataset*, il predittore fissato è incorrelato con gli altri predittori.
2. Considerare uno dei *dataset* costruiti con una certa modalità v e su di questo calcolare la stima utilizzando il modello *Random Forests* per ciascun caso del *dataset*.
3. Calcolare la media per tutti i valori considerando tutti i casi. Questo risultato fornisce la risposta media per un certo v fissato del predittore. Nel caso di variabile risposta categoriale, questo è dato dalla proporzione

condizionata o da una sua trasformazione; nel caso di variabile risposta quantitativa, questo valore è la media condizionata.

4. Ripetere i passi dal 2 al 4 per ognuna delle V modalità.
5. Effettuare la rappresentazione grafica dei valori stimati al punto 4 per ogni v verso le V modalità del predittore considerato.
6. Ripetere i passi dall'1 al 5 per ogni predittore. Questo consente di creare una rappresentazione grafica della risposta parziale (*partial dependence plot*) che mostra la variazione della risposta media rispetto ai valori di quel predittore, mantenendo costanti gli altri.

4.2.2.5 Pregi e difetti di Random Forests

C'è un'evidenza crescente che *Random Forests* sia uno strumento statistico molto potente (Berk, 2008a). La sua bontà nella previsione è forse il suo punto di forza; il vantaggio ottenuto rispetto ad altre tecniche statistiche deriva in larga parte dalle seguenti caratteristiche del suo algoritmo:

- la possibilità di costruire un numero molto elevato di alberi, ottenendo al contempo valori bassi nella distorsione;
- l'utilizzo di campioni di *bootstrap* per la costruzione degli alberi;
- l'utilizzo di un campione di predittori per ogni suddivisione degli alberi;
- il calcolo dei valori stimati e degli indicatori statistici a partire dai dati OOB;
- il calcolo della media su tutti gli alberi costruiti.

D'altro canto, solo alcune delle proprietà di *Random Forests* sono state formalmente provate. Vi è inoltre il problema che, volendo conoscere la funzione $f(X)$, la stima ottenuta con *Random Forests* non è consistente. Su questi aspetti, viste anche le recenti origini del modello, c'è molto lavoro da fare per gli statistici teorici.

Appendice

In questa appendice si trovano i comandi dell'ambiente statistico R necessari alla realizzazione di alcune delle figure e dei modelli realizzati nei capitoli precedenti.

§3.2.3 - Statistiche descrittive

Figura 3.1 - *Distribuzione di frequenze del volume di acqua erogato (1995-2008)*

```
> hist(erogato0/1000,xlab="erogato0 (metri cubi x
      1000)",main="Distribuzione di frequenze del volume di acqua
      erogata", ylab="Frequenza",breaks=50,xlim=c(80,160))
> mean(pd.all$erogato0)
[1] 113720.1
> var(pd.all$erogato0)^0.5
[1] 9692.516
```

Figura 3.2 - *Normal probability plot della variabile* `erogato0`

```
> qqnorm(erogato0,main="erogato0: Normal Q-Q Plot")
> qqline(erogato0)
```

Figura 3.3 - *Valori della temperatura massima e volume di acqua erogato, valori giornalieri 1995-2008.*

```
> plot(temperatura,erogato0/1000,ylab="erogato0 (x1000)")
```

Figura 3.4 - *Boxplot del volume di acqua erogato per giorno della settimana. Dati giornalieri dal 1995 al 2008.*

```
> plot(giorno_settimana,erogato0/1000,xlab="Giorno della settimana
      (1=Domenica,2=Lunedì,...,7=Sabato)", main="Erogato per giorno della
      settimana", ylab="erogato0 (x 1000)")
> axis(2,10*8:15)
> abline(h=10*8:15,col="lightgray",lty=3)
```

Figura 3.5 - *Boxplot del volume di acqua erogato per giorno dell'anno (1…365). Dati giornalieri dal 1995 al 2008.*

```
> plot(giorno_anno,erogato0/1000,xlab="Giorno dell'anno", ylab="erogato0
      ( x 1000 )")
```

Figura 3.8 - *Boxplot del volume di acqua erogato per anno. Dati giornalieri dal 1995 al 2008.*

```
> plot(anno,erogato0/1000,xlab="Anno", ylab="erogato0 ( x 1000 )")
```

Figura 3.7 - *Boxplot del volume di acqua erogato per mese. Dati giornalieri dal 1995 al 2008.*

```
> plot(mese,erogato0/1000,xlab="Mese", ylab="erogato0 ( x 1000 )")
```

Figura 3.6 - *Boxplot del volume di acqua erogato per settimana. Dati giornalieri dal 1995 al 2008.*

```
> plot(settimana,erogato0/1000,xlab="Settimana", ylab="erogato0 (x
    1000)")
```

Figura 3.10 - *Boxplot del volume di acqua erogato per giorno festivo o feriale. Dati giornalieri dal 1995 al 2008.*

```
> plot(festivo,erogato0/1000,xlab="Festivo (0=NO, 1=SI)",
    ylab="erogato0 (x 1000)")
```

Figura 3.9 - *Boxplot della temperatura massima (°C) per settimana. Dati giornalieri dal 1995 al 2008.*

```
> plot(settimana,temperatura,xlab="Settimana", ylab="Temperatura")
```

§3.4.2 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti (lm.1)

```
> lm.1<- lm(erogato0~erogato1+erogato2+erogato3+erogato4+erogato5+
    erogato6+erogato7,data=df.train)
> lm.plot(lm.1,df.train,df.valid)
Call:
lm(formula = erogato0 ~ erogato1 + erogato2 + erogato3 + erogato4 +
    erogato5 + erogato6 + erogato7, data = df.train)
Residuals:
     Min       1Q   Median       3Q      Max
-17346.4  -2707.5   -217.9   2642.3  20951.9
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.600e+03  9.393e+02   4.898 1.00e-06 ***
erogato1     4.585e-01  1.358e-02  33.766  < 2e-16 ***
erogato2     1.741e-02  1.524e-02   1.142  0.25344
erogato3     4.875e-02  1.524e-02   3.200  0.00139 **
erogato4     1.388e-02  1.525e-02   0.910  0.36289
erogato5    -2.782e-02  1.524e-02  -1.826  0.06794 .
erogato6     8.591e-03  1.523e-02   0.564  0.57279
erogato7     4.405e-01  1.356e-02  32.483  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4683 on 4368 degrees of freedom
Multiple R-squared: 0.7739,     Adjusted R-squared: 0.7736
F-statistic:  2136 on 7 and 4368 DF,  p-value: < 2.2e-16
==== Errore Relativo (Validation set) ====
Media = 0.304778932320902
Std.Dev. = 4.22642861685425
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 19.15185  58.27633  22.57182
(-Inf,-5]    (-5,5]  (5, Inf]
  9.98632  78.38577  11.62791
==== Errore Relativo (Training set) ====
```

```
Media = 0.168073913288636
Std.Dev. = 4.10797511406015
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 19.46984  59.91773  20.61243
(-Inf,-5]    (-5,5]  (5, Inf]
 8.957952 81.192870  9.849177
```

§3.4.3 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e la temperatura (lm.2)

```
> lm.2<- lm(erogato0~erogato1+erogato2+erogato3+erogato4+erogato5+
     erogato6+erogato7+temperatura,data=df.train)
> lm.plot(lm.2,df.train,df.valid)
Call:
lm(formula = erogato0 ~ erogato1 + erogato2 + erogato3 + erogato4 +
    erogato5 + erogato6 + erogato7 + temperatura, data = df.train)
Residuals:
      Min        1Q    Median        3Q       Max
-17324.77  -2612.74    -60.53   2674.19  20359.14
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.709e+03  9.836e+02   8.855  < 2e-16 ***
erogato1     4.368e-01  1.347e-02  32.423  < 2e-16 ***
erogato2     1.080e-02  1.500e-02   0.720  0.47167
erogato3     4.217e-02  1.500e-02   2.812  0.00495 **
erogato4     7.418e-03  1.501e-02   0.494  0.62125
erogato5    -3.224e-02  1.499e-02  -2.151  0.03154 *
erogato6     3.013e-03  1.499e-02   0.201  0.84072
erogato7     4.378e-01  1.334e-02  32.825  < 2e-16 ***
temperatura  1.050e+02  8.623e+00  12.176  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4606 on 4367 degrees of freedom
Multiple R-squared: 0.7814,     Adjusted R-squared: 0.781
F-statistic:  1951 on 8 and 4367 DF,  p-value: < 2.2e-16
==== Errore Relativo (Validation set) ====
Media = 0.0100277529797556
Std.Dev. = 4.15779047741557
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 21.47743  60.05472  18.46785
(-Inf,-5]    (-5,5]  (5, Inf]
 11.08071  78.52257  10.39672
==== Errore Relativo (Training set) ====
Media = 0.163913374837697
Std.Dev. = 4.06660337528437
```

```
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 19.21846  60.55759  20.22395
(-Inf,-5]    (-5,5]  (5, Inf]
 8.135283 81.855576 10.009141
```

§3.4.4 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale (lm.3)

```
> lm.3<-
      lm(erogato0~erogato1+erogato2+erogato3+erogato4+erogato5+erogato6+
      erogato7+temperatura+rapp_settimana,data=df.train_r)
> lm.plot(lm.3,df.train_r,df.valid)
Call:
lm(formula = erogato0 ~ erogato1 + erogato2 + erogato3 + erogato4 +
    erogato5 + erogato6 + erogato7 + temperatura + rapp_settimana,
    data = df.train_r)
Residuals:
      Min        1Q    Median        3Q       Max
-17451.70  -2568.95    -72.04   2659.16  20342.23
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.091e+04  1.556e+03   7.016 2.66e-12 ***
erogato1        4.326e-01  1.408e-02  30.729  < 2e-16 ***
erogato2        8.334e-03  1.564e-02   0.533   0.5942
erogato3        3.876e-02  1.564e-02   2.478   0.0133 *
erogato4        8.498e-03  1.565e-02   0.543   0.5872
erogato5       -3.247e-02  1.563e-02  -2.077   0.0379 *
erogato6       -7.302e-05  1.564e-02  -0.005   0.9963
erogato7        4.365e-01  1.392e-02  31.360  < 2e-16 ***
temperatura     1.141e+02  1.031e+01  11.059  < 2e-16 ***
rapp_settimana -3.706e+04  8.711e+04  -0.425   0.6706
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4589 on 4008 degrees of freedom
Multiple R-squared: 0.7682,     Adjusted R-squared: 0.7677
F-statistic:  1476 on 9 and 4008 DF,  p-value: < 2.2e-16
==== Errore Relativo (Validation set) ====
Media = 0.112977127334884
Std.Dev. = 4.1590632734865
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 19.56224  61.55951  18.87825
 (-Inf,-5]    (-5,5]  (5, Inf]
 10.25992  78.79617  10.94391
==== Errore Relativo (Training set) ====
Media = 0.159639835523831
```

```
Std.Dev. = 4.01751745567393
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 19.08910  60.67695  20.23395
 (-Inf,-5]    (-5,5]  (5, Inf]
 7.864609 82.429069  9.706322
```

§3.5.1 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti (nn.1)

```
>set.seed(100)
>nn.1<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n,data=df.train,size=4,
     linout=F,maxit=10000)
# weights:  37
initial  value 73.070548
iter  10 value 21.096435
iter  20 value 17.347476
iter  30 value 15.385201
. . .
iter 290 value 13.637815
iter 300 value 13.637639
final  value 13.637625
converged
>nn.plot(nn.1,df.train,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.206133068590533
Std.Dev. = 3.68724306232965
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.22709  67.57866  18.19425
(-Inf,-5]    (-5,5]  (5, Inf]
 4.924761 86.730506  8.344733
==== Errore Relativo (Training set) ====
Media = 0.141841459690026
Std.Dev. = 3.62207721487221
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.78519  68.21298  17.00183
(-Inf,-5]    (-5,5]  (5, Inf]
 5.484461 86.311700  8.203839
```

§3.5.2 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e la temperatura (nn.2)

```
>set.seed(100)
```

```
>nn.2<- nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n+temperatura,
     data=df.train,size=4,linout=F,maxit=10000,skip=F)
# weights:  41
initial  value 484.291401
iter  10 value 52.800239
iter  20 value 18.020316
iter  30 value 17.549888
iter  40 value 16.916272
. . .
iter 340 value 13.342484
final  value 13.342483
converged
>nn.plot(nn.2,df.train,df.valid)
==== Errore Relativo (Validation set) ====
Media = -0.0172595233597254
Std.Dev. = 3.67115668152527
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.23666  66.62107  16.14227
(-Inf,-5]    (-5,5]  (5, Inf]
  5.06156  87.41450   7.52394
==== Errore Relativo (Training set) ====
Media = 0.130999771896357
Std.Dev. = 3.56456206935237
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.00823  69.92687  16.06490
(-Inf,-5]    (-5,5]  (5, Inf]
 4.958867 86.997258  8.043876
```

§3.5.3 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale (nn.3)

```
>set.seed(100)
>nn.3<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n+temperatura+rapp_settimana,
     data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  45
initial  value 76.923897
iter  10 value 30.507099
iter  20 value 20.246174
. . .
iter1410 value 12.073498
iter1420 value 12.072197
iter1430 value 12.071141
```

```
final  value 12.070483
converged
>nn.plot(nn.3,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = -0.175799688675642
Std.Dev. = 3.72398436808820
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 19.83584  64.70588  15.45828
(-Inf,-5]    (-5,5]  (5, Inf]
 6.566347 86.046512  7.387141
==== Errore Relativo (Training set) ====
Media = 0.130904104480527
Std.Dev. = 3.54748465868084
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.03683  69.91040  16.05276
(-Inf,-5]    (-5,5]  (5, Inf]
 4.853161 87.406670  7.740169
```

§3.5.4 - Modello avente come predittore il solo coefficiente settimanale (nn.4)

```
>set.seed(100)
>nn.4<-nnet(erogato0_n~rapp_settimana,
     data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  13
initial  value 141.707703
iter  10 value 68.372703
iter  20 value 68.220211
iter  30 value 63.200602
. . .
iter 360 value 50.563767
iter 370 value 50.476832
final  value 50.471734
converged
>nn.plot(nn.4,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 3.67983227354438
Std.Dev. = 6.25832283666003
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 10.67031  39.94528  49.38440
(-Inf,-5]    (-5,5]  (5, Inf]
 6.019152 57.318741 36.662107
==== Errore Relativo (Training set) ====
Media = 0.504919091323601
```

```
Std.Dev. = 7.23957365007139
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 33.34993  34.32056  32.32952
(-Inf,-5]    (-5,5]  (5, Inf]
 20.63216  54.92782  24.44002
```

§3.5.5 - Modello avente come predittori la temperatura e il coefficiente settimanale (nn.5)

```
>set.seed(100)
>nn.5<-nnet(erogato0_n~temperatura+rapp_settimana,
     data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  17
initial  value 73.687814
iter  10 value 53.592610
iter  20 value 50.844306
. . .
iter 450 value 43.470894
iter 460 value 43.470061
iter 460 value 43.470061
final  value 43.470061
converged
>nn.plot(nn.5,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 2.29271557825601
Std.Dev. = 6.01974474559047
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.96306  43.77565  39.26129
(-Inf,-5]    (-5,5]  (5, Inf]
 7.113543 63.885089 29.001368
==== Errore Relativo (Training set) ====
Media = 0.455991126978205
Std.Dev. = 6.83691577442388
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 35.16675  33.59881  31.23444
(-Inf,-5]    (-5,5]  (5, Inf]
 20.93081  56.14734  22.92185
```

§3.5.6 - Modello avente come predittori la temperatura normalizzata e il coefficiente settimanale (nn.6)

```
>set.seed(100)
>nn.6<-nnet(erogato0_n~scale(temperatura)+rapp_settimana,
     data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
```

```
# weights:  17
initial  value 162.849762
iter  10 value 50.052233
iter  20 value 49.228716
. . .
iter 510 value 43.479532
iter 520 value 43.477091
final  value 43.474950
converged
>nn.plot(nn.6,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 2.28738270533879
Std.Dev. = 6.01343146141272
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.96306  44.04925  38.98769
(-Inf,-5]    (-5,5]  (5, Inf]
 7.113543 63.474692 29.411765
==== Errore Relativo (Training set) ====
Media = 0.455183624410915
Std.Dev. = 6.83726909934535
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 35.26630  33.49925  31.23444
(-Inf,-5]    (-5,5]  (5, Inf]
 20.90592  56.22200  22.87208
```

§3.5.7 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, valori normalizzati (nn.7)

```
>set.seed(100)
>nn.7<-nnet(erogato0_n~scale(erogato1)+scale(erogato2)+scale(erogato3)+
     scale(erogato4)+scale(erogato5)+scale(erogato6)+scale(erogato7),
     data=df.train,size=4,linout=F,maxit=10000,skip=F)
# weights:  37
initial  value 373.856602
iter  10 value 20.891771
iter  20 value 17.649704
iter  30 value 15.002955
. . .
iter 290 value 13.510774
iter 300 value 13.510222
final  value 13.510209
converged
>nn.plot(nn.7,df.train,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.0814942030242078
```

```
Std.Dev. = 3.66859046625827
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.73187  66.75787  17.51026
(-Inf,-5]    (-5,5]  (5, Inf]
 5.471956 86.867305  7.660739
==== Errore Relativo (Training set) ====
Media = 0.125858348325537
Std.Dev. = 3.6060555000115
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.53382  69.53839  15.92779
(-Inf,-5]    (-5,5]  (5, Inf]
 5.415905 86.928702  7.655393
```

§3.5.8 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e il coefficiente settimanale (nn.8)

```
>set.seed(100)
>nn.8<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
      erogato5_n+erogato6_n+erogato7_n+rapp_settimana,
      data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  41
initial  value 65.917471
iter  10 value 20.526014
iter  20 value 16.712551
. . .
iter 730 value 12.051328
iter 740 value 12.050947
iter 750 value 12.050797
final  value 12.050754
converged
>nn.plot(nn.8,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.345091669451611
Std.Dev. = 3.66155718647972
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 12.17510  70.31464  17.51026
(-Inf,-5]    (-5,5]  (5, Inf]
 5.198358 86.456908  8.344733
==== Errore Relativo (Training set) ====
Media = 0.123100182965238
Std.Dev. = 3.52292048004326
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
```

```
 13.91239  70.05973  16.02787
(-Inf,-5]    (-5,5]  (5, Inf]
 5.102041 86.933798  7.964161
```

§3.5.9 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale. Verifica overfitting (nn.9)

```
>set.seed(100)
>nn.9<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n+temperatura+rapp_settimana,
     data=df.train_r,size=18,linout=F,maxit=10000,skip=F)
# weights:  199
initial  value 209.150797
iter  10 value 46.702470
iter  20 value 16.844189
iter  30 value 16.044132
. . .
iter2400 value 9.451882
iter2400 value 9.451882
iter2400 value 9.451882
final  value 9.451882
converged
>nn.plot(nn.9,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.262726102920373
Std.Dev. = 5.32272656668176
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 18.33105  64.97948  16.68947
(-Inf,-5]    (-5,5]  (5, Inf]
 6.292750 86.320109  7.387141
==== Errore Relativo (Training set) ====
Media = 0.0985914476378814
Std.Dev. = 3.14366924388894
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 11.14983  75.21155  13.63863
(-Inf,-5]    (-5,5]  (5, Inf]
 3.509209 90.343454  6.147337
```

§3.5.10 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e il coefficiente settimanale. Molti neuroni nello strato nascosto (nn.10)

```
>set.seed(100)
```

```
>nn.10<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n+rapp_settimana,
     data=df.train_r,size=18,linout=F,maxit=10000,skip=F)
# weights:  181
initial  value 352.176498
iter  10 value 17.619033
iter  20 value 16.348470
iter  30 value 14.378068
. . .
iter3260 value 9.582428
iter3270 value 9.582074
final  value 9.582073
converged
>nn.plot(nn.10,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.436289217917664
Std.Dev. = 4.66072795506561
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 13.67989  69.22025  17.09986
(-Inf,-5]    (-5,5]  (5, Inf]
 5.882353 86.593707  7.523940
==== Errore Relativo (Training set) ====
Media = 0.097081129864265
Std.Dev. = 3.1457419943793
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 11.74714  74.78845  13.46441
(-Inf,-5]    (-5,5]  (5, Inf]
 3.608761 90.467894  5.923345
```

§3.5.11 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il coefficiente settimanale e il giorno festivo (nn.11)

```
>set.seed(100)
>nn.11<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n+temperatura+rapp_settimana+festivo,
     data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  49
initial  value 141.879529
iter  10 value 55.886343
iter  20 value 20.554892
. . .
iter 180 value 12.909551
iter 190 value 12.909282
iter 200 value 12.908480
```

```
final  value 12.908480
converged
>nn.plot(nn.11,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.213115400950559
Std.Dev. = 3.71854536406586
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.00547  65.38988  18.60465
(-Inf,-5]    (-5,5]  (5, Inf]
 5.335157 86.046512  8.618331
==== Errore Relativo (Training set) ====
Media = 0.129544553957578
Std.Dev. = 3.60009592075944
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.78347  68.04380  17.17272
(-Inf,-5]    (-5,5]  (5, Inf]
 5.400697 86.809358  7.789945
```

§3.5.12 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il coefficiente settimanale, il giorno festivo e il giorno della settimana (nn.12)

```
>set.seed(100)
>nn.12<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
     erogato5_n+erogato6_n+erogato7_n+temperatura+rapp_settimana+
     festivo+giorno_settimana,
     data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  73
initial  value 79.574088
iter  10 value 28.623384
iter  20 value 10.129989
. . .
iter 860 value 6.459892
iter 870 value 6.459594
iter 880 value 6.459504
final  value 6.459342
converged
>nn.plot(nn.12,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = -0.0168454060920882
Std.Dev. = 2.77517804521879
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 11.76471  76.06019  12.17510
```

```
(-Inf,-5]    (-5,5]  (5, Inf]
 2.462380 93.980848  3.556772
==== Errore Relativo (Training set) ====
Media = 0.065704061748138
Std.Dev. = 2.56068758526602
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 8.461921 80.662021 10.876058
(-Inf,-5]    (-5,5]  (5, Inf]
 2.065704 94.051767  3.882529
```

§3.5.13 - Previsione settimanale

```
####################################
# Estendo il dataframe di training per il modello
# settimanale (Reti Neurali = df.train_R).
# Ho bisogno delle variabili erogato8..erogato13
# per poter stimare i consumi dei
# giorni 0, +1, ,+2, ..., +6
####################################
L=length(df.train_r$erogato0)
erogato8_n=rep(NA, L)
erogato9_n=rep(NA, L)
erogato10_n=rep(NA, L)
erogato11_n=rep(NA, L)
erogato12_n=rep(NA, L)
erogato13_n=rep(NA, L)
for (i in 2:L){
  erogato8_n[i]<-df.train_r$erogato7_n[i-1]
}
for (i in 3:L){
  erogato9_n[i]<-df.train_r$erogato7_n[i-2]
}
for (i in 4:L){
  erogato10_n[i]<-df.train_r$erogato7_n[i-3]
}
for (i in 5:L){
  erogato11_n[i]<-df.train_r$erogato7_n[i-4]
}
for (i in 6:L){
  erogato12_n[i]<-df.train_r$erogato7_n[i-5]
}
for (i in 7:L){
  erogato13_n[i]<-df.train_r$erogato7_n[i-6]
}
df.train_r$erogato8_n<-erogato8_n
df.train_r$erogato9_n<-erogato9_n
```

```
df.train_r$erogato10_n<-erogato10_n
df.train_r$erogato11_n<-erogato11_n
df.train_r$erogato12_n<-erogato12_n
df.train_r$erogato13_n<-erogato13_n
rm(erogato8_n,erogato9_n,erogato10_n,erogato11_n,erogato12_n,erogato13_n,
     L,i)

####################################
# Creo un dataframe di validation per il modello
# settimanale. (Reti Neurali)
# Ho bisogno delle variabili
# erogato8..erogato13 per poter verificare le stime dei consumi dei
# giorni 0, +1, ,+2, ..., +6
####################################
L=length(df.valid$erogato0)
erogato8_n=rep(NA, L)
erogato9_n=rep(NA, L)
erogato10_n=rep(NA, L)
erogato11_n=rep(NA, L)
erogato12_n=rep(NA, L)
erogato13_n=rep(NA, L)
for (i in 2:L){
  erogato8_n[i]<-df.valid$erogato7_n[i-1]
}
for (i in 3:L){
  erogato9_n[i]<-df.valid$erogato7_n[i-2]
}
for (i in 4:L){
  erogato10_n[i]<-df.valid$erogato7_n[i-3]
}
for (i in 5:L){
  erogato11_n[i]<-df.valid$erogato7_n[i-4]
}
for (i in 6:L){
  erogato12_n[i]<-df.valid$erogato7_n[i-5]
}
for (i in 7:L){
  erogato13_n[i]<-df.valid$erogato7_n[i-6]
}
df.valid$erogato8_n<-erogato8_n
df.valid$erogato9_n<-erogato9_n
df.valid$erogato10_n<-erogato10_n
df.valid$erogato11_n<-erogato11_n
df.valid$erogato12_n<-erogato12_n
df.valid$erogato13_n<-erogato13_n
rm(erogato8_n,erogato9_n,erogato10_n,erogato11_n,erogato12_n,erogato13_n,
     L,i)
```

§3.5.13.1 - Modello di previsione a 1 giorno (nn.S0)

```
> set.seed(100)
> nn.S0<-nnet(erogato0_n~erogato1_n+erogato2_n+erogato3_n+erogato4_n+
      erogato5_n+erogato6_n+erogato7_n+temperatura+rapp_settimana+festivo+
      giorno_settimana,data=df.train_r,size=4,linout=F,maxit=10000,skip=F)
# weights:  73
initial  value 79.574088
iter  10 value 28.623384
iter  20 value 10.129989
. . .
iter 870 value 6.459594
iter 880 value 6.459504
final  value 6.459342
converged
> nn.plot(nn.S0,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = -0.0168454060920882
Std.Dev. = 2.77517804521879
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 11.76471  76.06019  12.17510
(-Inf,-5]    (-5,5]  (5, Inf]
 2.462380 93.980848  3.556772
==== Errore Relativo (Training set) ====
Media = 0.065704061748138
Std.Dev. = 2.56068758526602
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 8.461921 80.662021 10.876058
(-Inf,-5]    (-5,5]  (5, Inf]
 2.065704 94.051767  3.882529
```

§3.5.13.2 - Modello di previsione a 2 giorni (nn.S1)

```
> set.seed(100)
> nn.S1<-nnet(erogato0_n~erogato2_n+erogato3_n+erogato4_n+erogato5_n+
      erogato6_n+erogato7_n+erogato8_n+temperatura+rapp_settimana+
      festivo+giorno_settimana,
      data=df.train_r[!is.na(df.train_r$erogato8_n),],size=4,linout=F,
      maxit=10000,skip=F)
# weights:  73
initial  value 79.241310
iter  10 value 22.448469
iter  20 value 14.255389
. . .
iter 480 value 8.767905
iter 490 value 8.766543
```

```
iter 500 value 8.765811
final  value 8.765546
converged
> nn.plot(nn.S1,df.train_r[!is.na(df.train_r$erogato8_n),],
     df.valid[!is.na(df.valid$erogato8_n),])
==== Errore Relativo (Validation set) ====
Media = 0.000185054161770606
Std.Dev. = 3.28465740914865
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.61644  70.00000  14.38356
(-Inf,-5]    (-5,5]  (5, Inf]
 4.383562 88.630137  6.986301
==== Errore Relativo (Training set) ====
Media = 0.0891540284325554
Std.Dev. = 2.97841756122906
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 11.67538  75.03112  13.29350
(-Inf,-5]    (-5,5]  (5, Inf]
 3.559871 90.988300  5.451830
```

§3.5.13.3 - Modello di previsione a 3 giorni (nn.S2)

```
> set.seed(100)
> nn.S2<-nnet(erogato0_n~erogato3_n+erogato4_n+erogato5_n+erogato6_n+
     erogato7_n+erogato8_n+erogato9_n+temperatura+rapp_settimana+
     festivo+giorno_settimana,
     data=df.train_r[!is.na(df.train_r$erogato9_n),],size=4,linout=F,
     maxit=10000,skip=F)
# weights:  73
initial  value 79.215241
iter  10 value 25.353308
iter  20 value 16.160139
. . .
iter 560 value 10.596709
iter 570 value 10.595291
iter 580 value 10.593395
final  value 10.591283
converged
> nn.plot(nn.S2,df.train_r[!is.na(df.train_r$erogato9_n),],
     df.valid[!is.na(df.valid$erogato9_n),])
==== Errore Relativo (Validation set) ====
Media = 0.0833086985623247
Std.Dev. = 3.46417541097263
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
```

```
 16.04938  67.90123  16.04938
(-Inf,-5]    (-5,5]  (5, Inf]
 6.447188 86.145405  7.407407
==== Errore Relativo (Training set) ====
Media = 0.105644358632952
Std.Dev. = 3.27494228915177
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 13.72012  70.79183  15.48805
(-Inf,-5]    (-5,5]  (5, Inf]
 4.755976 88.222112  7.021912
```

§3.5.13.4 - Modello di previsione a 4 giorni (nn.S3)

```
> set.seed(100)
> nn.S3<-nnet(erogato0_n~erogato4_n+erogato5_n+erogato6_n+erogato7_n+
      erogato8_n+erogato9_n+erogato10_n+temperatura+rapp_settimana+
      festivo+giorno_settimana,
      data=df.train_r[!is.na(df.train_r$erogato10_n),],size=4,linout=F,
      maxit=10000,skip=F)
# weights:  73
initial  value 79.282899
iter  10 value 26.790320
iter  20 value 17.825372
. . .
iter 690 value 12.799532
iter 700 value 12.799470
iter 710 value 12.799455
final  value 12.799453
converged
> nn.plot(nn.S3,df.train_r[!is.na(df.train_r$erogato10_n),],
      df.valid[!is.na(df.valid$erogato10_n),])
==== Errore Relativo (Validation set) ====
Media = 0.189375978624159
Std.Dev. = 3.64801937490333
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.38462  69.09341  15.52198
(-Inf,-5]    (-5,5]  (5, Inf]
 7.005495 84.478022  8.516484
==== Errore Relativo (Training set) ====
Media = 0.128032244980513
Std.Dev. = 3.61084556921461
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.69116  67.47198  16.83686
(-Inf,-5]    (-5,5]  (5, Inf]
```

```
 5.828144 85.429639  8.742217
```

§3.5.13.5 - Modello di previsione a 5 giorni (nn.S4)

```
> set.seed(100)
> nn.S4<-nnet(erogato0_n~erogato5_n+erogato6_n+erogato7_n+erogato8_n+
      erogato9_n+erogato10_n+erogato11_n+temperatura+rapp_settimana+
      festivo+giorno_settimana,
      data=df.train_r[!is.na(df.train_r$erogato11_n),],size=4,linout=F,
      maxit=10000,skip=F)
# weights:  73
initial  value 79.364523
iter  10 value 23.309253
iter  20 value 18.340528
. . .
iter 710 value 14.689634
iter 720 value 14.689449
iter 730 value 14.689318
final  value 14.689177
converged
> nn.plot(nn.S4,df.train_r[!is.na(df.train_r$erogato11_n),],
      df.valid[!is.na(df.valid$erogato11_n),])
==== Errore Relativo (Validation set) ====
Media = 0.272665804704436
Std.Dev. = 3.87293376266193
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.05640  63.82393  19.11967
(-Inf,-5]    (-5,5]  (5, Inf]
 7.702889 81.843191 10.453920
==== Errore Relativo (Training set) ====
Media = 0.155228005646439
Std.Dev. = 3.86348872509223
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.11510  64.29995  18.58495
(-Inf,-5]    (-5,5]  (5, Inf]
 6.477329 83.731938  9.790732
```

§3.5.13.6 - Modello di previsione a 6 giorni (nn.S5)

```
> set.seed(100)
> nn.S5<-nnet(erogato0_n~erogato6_n+erogato7_n+erogato8_n+erogato9_n+
      erogato10_n+erogato11_n+erogato12_n+temperatura+rapp_settimana+
      festivo+giorno_settimana,
      data=df.train_r[!is.na(df.train_r$erogato12_n),],size=4,linout=F,
      maxit=10000,skip=F)
```

```
# weights:  73
initial  value 79.163815
iter  10 value 28.002121
iter  20 value 19.928804
iter  30 value 18.665934
. . .
iter1380 value 14.650731
iter1390 value 14.650050
iter1400 value 14.649621
final  value 14.649616
converged
> nn.plot(nn.S5,df.train_r[!is.na(df.train_r$erogato12_n),],
     df.valid[!is.na(df.valid$erogato12_n),])
==== Errore Relativo (Validation set) ====
Media = 0.171145400568954
Std.Dev. = 3.95153001166905
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.07989  64.18733  18.73278
(-Inf,-5]    (-5,5]  (5, Inf]
 8.953168 80.991736 10.055096
==== Errore Relativo (Training set) ====
Media = 0.149780591093085
Std.Dev. = 3.86513525855872
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.89509  64.76452  18.34039
(-Inf,-5]    (-5,5]  (5, Inf]
 7.151757 83.055071  9.793172
```

§3.5.13.7 - Modello di previsione a 7 giorni (nn.S6)

```
> set.seed(100)
> nn.S6<-nnet(erogato0_n~erogato7_n+erogato8_n+erogato9_n+erogato10_n+
     erogato11_n+erogato12_n+erogato13_n+temperatura+rapp_settimana+
     festivo+giorno_settimana,
     data=df.train_r[!is.na(df.train_r$erogato13_n),],size=4,linout=F,
     maxit=10000,skip=F)
# weights:  73
initial  value 79.023431
iter  10 value 29.018280
iter  20 value 20.837726
. . .
iter1310 value 15.432887
iter1320 value 15.432354
iter1330 value 15.432040
final  value 15.432038
```

```
converged
> nn.plot(nn.S6,df.train_r[!is.na(df.train_r$erogato13_n),],
     df.valid[!is.na(df.valid$erogato13_n),])
==== Errore Relativo (Validation set) ====
Media = 0.067573890058632
Std.Dev. = 4.02463381481279
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 18.06897  63.72414  18.20690
(-Inf,-5]    (-5,5]  (5, Inf]
 9.103448 81.931034  8.965517
==== Errore Relativo (Training set) ====
Media = 0.162016486906861
Std.Dev. = 3.94428732834437
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.67198  63.31007  19.01795
(-Inf,-5]    (-5,5]  (5, Inf]
 7.701894 82.402792  9.895314
```

§3.5.13.8 - Modello settimanale (nn.S)

```
####################################
#Calcolo consumo settimanale e stima settimanale
# sul dataset di training (RETI NEURALI)
####################################
cs.nn_train=rep(NA, 574)   #consumo settimanale (1...574
                            # settimane intere nel df.train_r)
ss.nn_train=rep(NA, 574)   #stima settimanale (1...574 settimane
                            # intere nel df.train_r)
tmp0<-dati.orig(predict(nn.S0))
tmp1<-dati.orig(predict(nn.S1))
tmp2<-dati.orig(predict(nn.S2))
tmp3<-dati.orig(predict(nn.S3))
tmp4<-dati.orig(predict(nn.S4))
tmp5<-dati.orig(predict(nn.S5))
tmp6<-dati.orig(predict(nn.S6))
for (n in 1:574) {
  i<-(n-1)*7+1   # riga 1, 8, 15, ..., 721
  cs.nn_train[n]<-
     dati.orig(df.train_r$erogato0_n[i])+dati.orig(df.train_r$erogato0_n[
     i+1])+dati.orig(df.train_r$erogato0_n[i+2])+dati.orig(df.train_r$ero
     gato0_n[i+3])+dati.orig(df.train_r$erogato0_n[i+4])+dati.orig(df.tra
     in_r$erogato0_n[i+5])+dati.orig(df.train_r$erogato0_n[i+6])
     ss.nn_train[n]<-tmp0[i]            # previsione giorno corrente (t=0)
  ss.nn_train[n]<-ss.nn_train[n]+tmp1[i]  # giorno t+1
  ss.nn_train[n]<-ss.nn_train[n]+tmp2[i]  # giorno t+2
```

```
  ss.nn_train[n]<-ss.nn_train[n]+tmp3[i]  # giorno t+3
  ss.nn_train[n]<-ss.nn_train[n]+tmp4[i]  # giorno t+4
  ss.nn_train[n]<-ss.nn_train[n]+tmp5[i]  # giorno t+5
  ss.nn_train[n]<-ss.nn_train[n]+tmp6[i]  # giorno t+6
}
####################################
# Calcolo consumo settimanale e stima settimanale sul
# dataset di validation (RETI NEURALI)
####################################
cs.nn_valid=rep(NA, 104)   #consumo settimanale (1...104 settimane
                           # intere nel df.valid)
ss.nn_valid=rep(NA, 104)   #stima settimanale (1...104 settimane
                           # intere nel df.valid)
tmp0<-dati.orig(predict(nn.S0,df.valid))
tmp1<-dati.orig(predict(nn.S1,df.valid))
tmp2<-dati.orig(predict(nn.S2,df.valid))
tmp3<-dati.orig(predict(nn.S3,df.valid))
tmp4<-dati.orig(predict(nn.S4,df.valid))
tmp5<-dati.orig(predict(nn.S5,df.valid))
tmp6<-dati.orig(predict(nn.S6,df.valid))
for (n in 1:104) {
  i<-(n-1)*7+1  # riga 1, 8, 15, ..., 721
     # consumi reali della settimana
  cs.nn_valid[n]<-
     dati.orig(df.valid$erogato0_n[i])+dati.orig(df.valid$erogato0_n[i+1]
     )+dati.orig(df.valid$erogato0_n[i+2])+dati.orig(df.valid$erogato0_n[
     i+3])+dati.orig(df.valid$erogato0_n[i+4])+dati.orig(df.valid$erogato
     0_n[i+5])+dati.orig(df.valid$erogato0_n[i+6])

  ss.nn_valid[n]<-tmp0[i]    # previsione giorno corrente (t=0)
  ss.nn_valid[n]<-ss.nn_valid[n]+tmp1[i+1]  # giorno t+1
  ss.nn_valid[n]<-ss.nn_valid[n]+tmp2[i+2]  # giorno t+2
  ss.nn_valid[n]<-ss.nn_valid[n]+tmp3[i+3]  # giorno t+3
  ss.nn_valid[n]<-ss.nn_valid[n]+tmp4[i+4]  # giorno t+4
  ss.nn_valid[n]<-ss.nn_valid[n]+tmp5[i+5]  # giorno t+5
  ss.nn_valid[n]<-ss.nn_valid[n]+tmp6[i+6]  # giorno t+6
}
rm(i,n,tmp0,tmp1,tmp2,tmp3,tmp4,tmp5,tmp6)

#################################
# Plot risultati modello settimanale (RETI NEURALI)
> nn.plot_s()
==== Errore Relativo (Validation set) ====
Media = -0.247012649691442
Std.Dev. = 2.37259577165358
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
```

```
12.500000 79.807692  7.692308
(-Inf,-5]    (-5,5]  (5, Inf]
 2.884615 94.230769  2.884615
==== Errore Relativo (Training set) ====
Media = 0.0733045879800118
Std.Dev. = 2.50665013901226
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
  9.23345  80.66202  10.10453
(-Inf,-5]    (-5,5]  (5, Inf]
 1.916376 93.728223  4.355401
```

§3.6.1 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti (rf.1)

```
>set.seed(100)
>rf.1<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
     erogato5+erogato6+erogato7,data=df.train,importance=TRUE)
No. of variables tried at each split: 2
Mean of squared residuals: 17280213
% Var explained: 82.15
>rf.plot(rf.1,df.train,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.344270092587455
Std.Dev. = 3.7437888668438
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 13.67989  66.75787  19.56224
(-Inf,-5]    (-5,5]  (5, Inf]
 6.019152 85.225718  8.755130
==== Errore Relativo (Training set) ====
Media = 0.161736066018013
Std.Dev. = 3.69447686899690
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.21938  68.57861  16.20201
(-Inf,-5]    (-5,5]  (5, Inf]
 5.553016 86.083181  8.363803
```

§3.6.2 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti e la temperatura (rf.2)

```
>set.seed(100)
>rf.2<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
     erogato5+erogato6+erogato7+temperatura,
     data=df.train,importance=TRUE)
Number of trees: 500
```

```
No. of variables tried at each split: 2
Mean of squared residuals: 16273841
% Var explained: 83.19
>rf.plot(rf.2,df.train,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.137267176362319
Std.Dev. = 3.7615930458483
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.82627  65.25308  17.92066
(-Inf,-5]    (-5,5]  (5, Inf]
 5.745554 86.456908  7.797538
==== Errore Relativo (Training set) ====
Media = 0.158456559241271
Std.Dev. = 3.6067370784379
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 14.09963  69.62980  16.27057
(-Inf,-5]    (-5,5]  (5, Inf]
 4.913163 86.791590  8.295247
```

§3.6.3 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il coefficiente settimanale (rf.3)

```
>set.seed(100)
>rf.3<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
     erogato5+erogato6+erogato7+temperatura+rapp_settimana,
     data=df.train_r,importance=TRUE)
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 15774647
% Var explained: 82.6
>rf.plot(rf.3,df.train_r,df.valid)
==== Errore Relativo (Validation set) ====
Media = 0.186786384656390
Std.Dev. = 3.72098141370602
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.55267  65.66347  17.78386
(-Inf,-5]    (-5,5]  (5, Inf]
 5.198358 86.046512  8.755130
==== Errore Relativo (Training set) ====
Media = 0.156517020038885
Std.Dev. = 3.52083832530600
Frequenze relative:
```

```
(-Inf,-3]    (-3,3]  (3, Inf]
 13.38975  70.75660  15.85366
(-Inf,-5]    (-5,5]  (5, Inf]
 4.654057 87.356894  7.989049
```

§3.6.4 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura e il giorno dell'anno (rf.4)

```
> set.seed(100)
> rf.4<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
      erogato5+erogato6+erogato7+temperatura+giorno_anno_n,
      data=df.train,importance=TRUE)
> rf.plot(rf.4,df.train,df.valid)
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 15508020
% Var explained: 83.98
==== Errore Relativo (Validation set) ====
Media = 0.0359758439463149
Std.Dev. = 3.64177985435745
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.86867  67.71546  16.41587
(-Inf,-5]    (-5,5]  (5, Inf]
 5.745554 87.004104  7.250342
==== Errore Relativo (Training set) ====
Media = 0.142489885828288
Std.Dev. = 3.51630845217145
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 13.32267  71.13803  15.53931
(-Inf,-5]    (-5,5]  (5, Inf]
 4.776051 87.317185  7.906764
```

§3.6.5 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il giorno dell'anno e il giorno festivo (rf.5)

```
> set.seed(100)
> rf.5<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
      erogato5+erogato6+erogato7+temperatura+giorno_anno_n+festivo,
      data=df.train,importance=TRUE)
> rf.plot(rf.5,df.train,df.valid)
Type of random forest: regression
```

```
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 14099283
% Var explained: 85.44
==== Errore Relativo (Validation set) ====
Media = 0.115058325956943
Std.Dev. = 3.43315615546007
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 13.81669  70.17784  16.00547
(-Inf,-5]    (-5,5]  (5, Inf]
 5.061560 87.961696  6.976744
==== Errore Relativo (Training set) ====
Media = 0.168248012064598
Std.Dev. = 3.31198848557746
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 12.56856  72.18921  15.24223
(-Inf,-5]    (-5,5]  (5, Inf]
 4.250457 88.574040  7.175503
```

§3.6.6 - Modello avente come predittori i volumi di erogato nei 7 giorni precedenti, la temperatura, il giorno dell'anno, il giorno festivo e il giorno della settimana (rf.6)

```
> set.seed(100)
> rf.6<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
      erogato5+erogato6+erogato7+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train,importance=TRUE)
> rf.plot(rf.6,df.train,df.valid)
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 10462520
% Var explained: 89.19
==== Errore Relativo (Validation set) ====
Media = 0.079039817823351
Std.Dev. = 2.99688843000801
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 12.58550  74.55540  12.85910
(-Inf,-5]    (-5,5]  (5, Inf]
 3.693570 90.834473  5.471956
==== Errore Relativo (Training set) ====
Media = 0.102365042938985
Std.Dev. = 2.85237596073924
Frequenze relative:
```

```
(-Inf,-3]    (-3,3]  (3, Inf]
 10.55759  76.34826  13.09415
(-Inf,-5]    (-5,5]  (5, Inf]
 2.490859 92.458867  5.050274
```

§3.6.7 - Previsione settimanale

```
>####################################
># Creo un dataframe di training per il modello settimanale.
># Ho bisogno delle variabili erogato8..erogato13 per poter stimare i
># consumi dei giorni 0, +1, ,+2, ..., +6
>####################################
>L=length(df.train$erogato0)
>erogato8=rep(NA, L)
>erogato9=rep(NA, L)
>erogato10=rep(NA, L)
>erogato11=rep(NA, L)
>erogato12=rep(NA, L)
>erogato13=rep(NA, L)
>for (i in 2:L){
  erogato8[i]<-df.train$erogato7[i-1]
}
>for (i in 3:L){
  erogato9[i]<-df.train$erogato7[i-2]
}
>for (i in 4:L){
  erogato10[i]<-df.train$erogato7[i-3]
}
>for (i in 5:L){
  erogato11[i]<-df.train$erogato7[i-4]
}
>for (i in 6:L){
  erogato12[i]<-df.train$erogato7[i-5]
}
>for (i in 7:L){
  erogato13[i]<-df.train$erogato7[i-6]
}
>df.train$erogato8<-erogato8
>df.train$erogato9<-erogato9
>df.train$erogato10<-erogato10
>df.train$erogato11<-erogato11
>df.train$erogato12<-erogato12
vdf.train$erogato13<-erogato13
>rm(erogato8,erogato9,erogato10,erogato11,erogato12,erogato13,L,i)
>####################################
># Creo un dataframe di validation per il modello settimanale.
># Ho bisogno delle variabili erogato8..erogato13 per poter verificare le
```

```
>#  stime dei consumi dei giorni 0, +1, ,+2, ..., +6
>###################################
>L=length(df.valid$erogato0)
>erogato8=rep(NA, L)
>erogato9=rep(NA, L)
>erogato10=rep(NA, L)
>erogato11=rep(NA, L)
>erogato12=rep(NA, L)
>erogato13=rep(NA, L)
>for (i in 2:L){
  erogato8[i]<-df.valid$erogato7[i-1]
}
>for (i in 3:L){
  erogato9[i]<-df.valid$erogato7[i-2]
}
>for (i in 4:L){
  erogato10[i]<-df.valid$erogato7[i-3]
}
>for (i in 5:L){
  erogato11[i]<-df.valid$erogato7[i-4]
}
>for (i in 6:L){
  erogato12[i]<-df.valid$erogato7[i-5]
}
>for (i in 7:L){
  erogato13[i]<-df.valid$erogato7[i-6]
}
>df.valid$erogato8<-erogato8
>df.valid$erogato9<-erogato9
>df.valid$erogato10<-erogato10
>df.valid$erogato11<-erogato11
>df.valid$erogato12<-erogato12
>df.valid$erogato13<-erogato13
>rm(erogato8,erogato9,erogato10,erogato11,erogato12,erogato13,L,i)
```

§3.6.7.1 - Modello di previsione a 1 giorno (rf.S0)

```
> set.seed(100)
> rf.S0<-randomForest(erogato0~erogato1+erogato2+erogato3+erogato4+
      erogato5+erogato6+erogato7+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train,importance=TRUE)
> rf.plot(rf.S0,df.train,df.valid)
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 10462520
% Var explained: 89.19
```

```
==== Errore Relativo (Validation set) ====
Media = 0.079039817823351
Std.Dev. = 2.99688843000801
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 12.58550  74.55540  12.85910
(-Inf,-5]    (-5,5]  (5, Inf]
 3.693570 90.834473  5.471956
==== Errore Relativo (Training set) ====
Media = 0.102365042938985
Std.Dev. = 2.85237596073924
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 10.55759  76.34826  13.09415
(-Inf,-5]    (-5,5]  (5, Inf]
 2.490859 92.458867  5.050274
```

§3.6.7.2 - Modello di previsione a 2 giorni (rf.S1)

```
> set.seed(100)
> rf.S1<-randomForest(erogato0~erogato2+erogato3+erogato4+erogato5+
      erogato6+erogato7+erogato8+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train[!is.na(df.train$erogato8),],
      importance=TRUE)
> rf.plot(rf.S1,df.train[!is.na(df.train$erogato8),],
      df.valid[!is.na(df.valid$erogato8),])
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 14049599
% Var explained: 85.47
==== Errore Relativo (Validation set) ====
Media = 0.067073552304436
Std.Dev. = 3.46308918366902
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.16438  69.04110  14.79452
(-Inf,-5]    (-5,5]  (5, Inf]
 5.890411 86.986301  7.123288
==== Errore Relativo (Training set) ====
Media = 0.120693356138838
Std.Dev. = 3.29472048055837
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 13.60000  70.90286  15.49714
(-Inf,-5]    (-5,5]  (5, Inf]
 4.160000 88.594286  7.245714
```

§3.6.7.3 - Modello di previsione a 3 giorni (rf.S2)

```
> set.seed(100)
> rf.S2<-randomForest(erogato0~erogato3+erogato4+erogato5+erogato6+
      erogato7+erogato8+erogato9+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train[!is.na(df.train$erogato9),],
      importance=TRUE)
> rf.plot(rf.S2,df.train[!is.na(df.train$erogato9),],
      df.valid[!is.na(df.valid$erogato9),])
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 16427345
% Var explained: 83.01
==== Errore Relativo (Validation set) ====
Media = 0.0541312346219256
Std.Dev. = 3.68480047678183
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.28395  66.52949  16.18656
(-Inf,-5]    (-5,5]  (5, Inf]
 6.858711 85.459534  7.681756
==== Errore Relativo (Training set) ====
Media = 0.144142235531769
Std.Dev. = 3.56024788538157
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 15.54641  67.67261  16.78098
(-Inf,-5]    (-5,5]  (5, Inf]
 5.601280 86.053955  8.344765
```

§3.6.7.4 - Modello di previsione a 4 giorni (rf.S3)

```
> set.seed(100)
> rf.S3<-randomForest(erogato0~erogato4+erogato5+erogato6+erogato7+
      erogato8+erogato9+erogato10+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train[!is.na(df.train$erogato10),],
      importance=TRUE)
> rf.plot(rf.S3,df.train[!is.na(df.train$erogato10),],
      df.valid[!is.na(df.valid$erogato10),])
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 18187261
% Var explained: 81.19
==== Errore Relativo (Validation set) ====
Media = 0.0810304450297565
Std.Dev. = 3.79630902513523
```

```
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.58242  66.34615  16.07143
(-Inf,-5]    (-5,5]  (5, Inf]
 7.554945 84.340659  8.104396
==== Errore Relativo (Training set) ====
Media = 0.161243501072696
Std.Dev. = 3.74366603621231
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.51040  65.35559  18.13400
(-Inf,-5]    (-5,5]  (5, Inf]
 6.219986 84.793048  8.986965
```

§3.6.7.5 - Modello di previsione a 5 giorni (rf.S4)

```
> set.seed(100)
> rf.S4<-randomForest(erogato0~erogato5+erogato6+erogato7+erogato8+
      erogato9+erogato10+erogato11+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train[!is.na(df.train$erogato11),],
      importance=TRUE)
> rf.plot(rf.S4,df.train[!is.na(df.train$erogato11),],
      df.valid[!is.na(df.valid$erogato11),])
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 19252312
% Var explained: 80.09
==== Errore Relativo (Validation set) ====
Media = 0.0735471901220202
Std.Dev. = 3.95954497007519
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 18.70702  64.78680  16.50619
(-Inf,-5]    (-5,5]  (5, Inf]
 8.803301 82.668501  8.528198
==== Errore Relativo (Training set) ====
Media = 0.154290226578098
Std.Dev. = 3.85290364281459
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 16.97164  64.54712  18.48124
(-Inf,-5]    (-5,5]  (5, Inf]
 6.907594 83.737420  9.354986
```

§3.6.7.6 - Modello di previsione a 6 giorni (rf.S5)

```
> set.seed(100)
> rf.S5<-randomForest(erogato0~erogato6+erogato7+erogato8+erogato9+
      erogato10+erogato11+erogato12+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train[!is.na(df.train$erogato12),],
      importance=TRUE)
> rf.plot(rf.S5,df.train[!is.na(df.train$erogato12),],
      df.valid[!is.na(df.valid$erogato12),])
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 20022893
% Var explained: 79.29
==== Errore Relativo (Validation set) ====
Media = 0.0631221719374927
Std.Dev. = 4.04590526995063
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 18.73278  63.36088  17.90634
(-Inf,-5]    (-5,5]  (5, Inf]
 9.504132 81.267218  9.228650
==== Errore Relativo (Training set) ====
Media = 0.160973460537974
Std.Dev. = 3.9342530974769
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.57035  63.80691  18.62274
(-Inf,-5]    (-5,5]  (5, Inf]
 7.275223 83.070236  9.654541
```

§3.6.7.7 - Modello di previsione a 7 giorni (rf.S6)

```
> set.seed(100)
> rf.S6<-randomForest(erogato0~erogato7+erogato8+erogato9+erogato10+
      erogato11+erogato12+erogato13+temperatura+giorno_anno_n+festivo+
      giorno_settimana,data=df.train[!is.na(df.train$erogato13),],
      importance=TRUE)
> rf.plot(rf.S6,df.train[!is.na(df.train$erogato13),],
      df.valid[!is.na(df.valid$erogato13),])
Type of random forest: regression
Number of trees: 500
No. of variables tried at each split: 3
Mean of squared residuals: 20682192
% Var explained: 78.6
==== Errore Relativo (Validation set) ====
Media = 0.0880705837953598
Std.Dev. = 4.1394787811781
```

```
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 18.20690  63.58621  18.20690
(-Inf,-5]    (-5,5]  (5, Inf]
  9.24138  81.37931   9.37931
==== Errore Relativo (Training set) ====
Media = 0.167423247879542
Std.Dev. = 3.99680519725571
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
 17.36842  63.59268  19.03890
(-Inf,-5]    (-5,5]  (5, Inf]
 7.665904 82.608696  9.725400
```

§3.6.7.8 - Modello settimanale (rf.S)

```
>####################################
>#Calcolo consumo settimanale e stima settimanale sul dataset di training
>####################################
>cs_train=rep(NA, 625)   #consumo settimanale (1...625 settimane intere
                         # nel df.train)
>ss_train=rep(NA, 625)   #stima settimanale (1...625 settimane intere
                         # nel df.train)
>tmp0<-predict(rf.S0)
>tmp1<-predict(rf.S1)
>tmp2<-predict(rf.S2)
>tmp3<-predict(rf.S3)
>tmp4<-predict(rf.S4)
>tmp5<-predict(rf.S5)
>tmp6<-predict(rf.S6)
>for (n in 1:625) {
  i<-(n-1)*7+1          # riga 1, 8, 15, ..., 721
  cs_train[n]<-df.train$erogato0[i]+df.train$erogato0[i+1]+
     df.train$erogato0[i+2]+df.train$erogato0[i+3]+
     df.train$erogato0[i+4]+df.train$erogato0[i+5]+df.train$erogato0[i+6]
  ss_train[n]<-tmp0[i]             # previsione giorno corrente (t=0)
  # giorno t+1: uso la riga successiva con il modello rf.S1.
  # Dovrei considerare l'indice i+1, ma poiché predict esclude
  # la prima riga devo togliere 1 --> punto a "i"
  ss_train[n]<-ss_train[n]+tmp1[i]
  # giorno t+2: uso la riga successiva con il modello rf.S2.
  # Dovrei considerare l'indice i+2, ma poiché predict esclude
  # le righe 1 e 2 devo togliere 2 --> punto a "i"
  ss_train[n]<-ss_train[n]+tmp2[i]
  # giorno t+3: uso la riga successiva con il modello rf.S3.
  # Dovrei considerare l'indice i+3, ma poiché predict esclude
  # le righe 1...3 devo togliere 3 --> punto a "i"
```

```
  ss_train[n]<-ss_train[n]+tmp3[i]
  # giorno t+4: uso la riga successiva con il modello rf.S4.
  # Dovrei considerare l'indice i+4, ma poiché predict esclude
  # le righe 1...4 devo togliere 4 --> punto a "i"
  ss_train[n]<-ss_train[n]+tmp4[i]
  # giorno t+5: uso la riga successiva con il modello rf.S5.
  # Dovrei considerare l'indice i+5, ma poiché predict esclude
  # le righe 1...5 devo togliere 5 --> punto a "i"
  ss_train[n]<-ss_train[n]+tmp5[i]
  # giorno t+6: uso la riga successiva con il modello rf.S6.
  # Dovrei considerare l'indice i+6, ma poiché predict esclude
  # le righe 1...6 devo togliere 6 --> punto a "i"
  ss_train[n]<-ss_train[n]+tmp6[i]
}
>####################################
>#Calcolo consumo settimanale e stima settimanale
>#sul dataset di validation
>####################################
>#consumo settimanale (1...104 settimane intere nel df.valid)
>cs_valid=rep(NA, 104)
>#stima settimanale (1...104 settimane intere nel df.valid)
>ss_valid=rep(NA, 104)
>tmp0<-predict(rf.S0,df.valid)
>tmp1<-predict(rf.S1,df.valid)
>tmp2<-predict(rf.S2,df.valid)
>tmp3<-predict(rf.S3,df.valid)
>tmp4<-predict(rf.S4,df.valid)
>tmp5<-predict(rf.S5,df.valid)
>tmp6<-predict(rf.S6,df.valid)
>for (n in 1:104) {
  i<-(n-1)*7+1           # riga 1, 8, 15, ..., 721
  # consumi reali della settimana
  cs_valid[n]<-
     df.valid$erogato0[i]+df.valid$erogato0[i+1]+df.valid$erogato0[i+2]+d
     f.valid$erogato0[i+3]+df.valid$erogato0[i+4]+df.valid$erogato0[i+5]+
     df.valid$erogato0[i+6]
  ss_valid[n]<-tmp0[i]    # previsione giorno corrente (t=0)
  ss_valid[n]<-ss_valid[n]+tmp1[i]  # giorno t+1: punto a "i"
  ss_valid[n]<-ss_valid[n]+tmp2[i]  # giorno t+2: punto a "i"
  ss_valid[n]<-ss_valid[n]+tmp3[i]  # giorno t+3: punto a "i"
  ss_valid[n]<-ss_valid[n]+tmp4[i]  # giorno t+4: punto a "i"
  ss_valid[n]<-ss_valid[n]+tmp5[i]  # giorno t+5: punto a "i"
  ss_valid[n]<-ss_valid[n]+tmp6[i]  # giorno t+6: punto a "i"
}
>rm(i,n,tmp0,tmp1,tmp2,tmp3,tmp4,tmp5,tmp6)
################################
# Plot risultati modello settimanale
```

```
> rf.plot_s()
==== Errore Relativo (Validation set) ====
Media = -0.393029532793454
Std.Dev. = 2.74339972551344
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
13.461538 76.923077  9.615385
 (-Inf,-5]     (-5,5]   (5, Inf]
 8.6538462 90.3846154  0.9615385
==== Errore Relativo (Training set) ====
Media = 0.113435322987502
Std.Dev. = 2.73347502718288
Frequenze relative:
(-Inf,-3]    (-3,3]  (3, Inf]
    10.40     77.28     12.32
(-Inf,-5]    (-5,5]  (5, Inf]
     2.56     93.44      4.00
```

Funzioni di utilità definite in R

```
dati.norm <- function (x) {
 return ((x-81648)/(154630-81648))
}
dati.orig <- function (x) {
 return ((154630-81648)*x+81648)
}
lm.plot <- function (lm,df,dfv) {
   windows(width=11,height=6)
   plot.new()
   par(ask=F)
   par(mfrow=c(1,2))
   plot(df$erogato0,predict(lm),xlab="valore reale",ylab="valore
     stimato",main="Training Set")
   abline(0,1)
   plot(dfv$erogato0,predict(lm,dfv),xlab="valore reale",ylab="valore
     stimato",main="Validation Set")
   abline(0,1)
   par(ask=T)
   plot(fitted(lm),residuals(lm),xlab="Valori
     stimati",ylab="Residui",main="Residui verso valori stimati")
   err_rel<-(predict(lm,dfv)-dfv$erogato0)*100/dfv$erogato0
   hist(err_rel,nclass=20,xlab="Errori
     relativi",ylab="Frequenza",main="Distribuzione errori relativi")
   par(ask=T)

   print(summary(lm))
   cat("==== Errore Relativo (Validation set) ====\n")
```

```
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    cat("\n==== Errore Relativo (Training set) ====\n")
    err_rel<-(predict(lm)-df$erogato0)*100/df$erogato0
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    rm(err_rel)
    rm(err_freq_ass)
 }
nn.plot <- function (nn,df,dfv) {
    windows(width=11,height=6)
    plot.new()
    par(ask=F)
    par(mfrow=c(1,2))
    plot(df$erogato0_n,predict(nn),xlab="valore reale",ylab="valore
      stimato",main="Training Set",xlim=c(0,1),ylim=c(0,1))
    abline(0,1)
    plot(dfv$erogato0_n,predict(nn,dfv),xlab="valore reale",ylab="valore
      stimato",main="Validation Set",xlim=c(0,1),ylim=c(0,1))
    abline(0,1)
    par(ask=T)
    plot(fitted(nn),residuals(nn),xlab="Valori
      stimati",ylab="Residui",main="Residui verso valori
      stimati",xlim=c(0,1))
    err_rel<-(dati.orig(predict(nn,dfv))-dfv$erogato0)*100/dfv$erogato0
    hist(err_rel,nclass=20,xlab="Errori
      relativi",ylab="Frequenza",main="Distribuzione errori relativi")
    par(ask=T)
    cat("==== Errore Relativo (Validation set) ====\n")
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    cat("\n==== Errore Relativo (Training set) ====\n")
```

```
    err_rel<-(dati.orig(predict(nn))-df$erogato0)*100/df$erogato0
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    rm(err_rel)
    rm(err_freq_ass)
 }
rf.plot <- function (rf,df,dfv) {
    windows(width=11,height=6)
    plot.new()
    print(rf)
    par(ask=F)
    par(mfrow=c(1,1))
    plot(rf,main="")
    par(ask=T)
    par(mfrow=c(1,1))
    varImpPlot(rf,main="")
    par(mfrow=c(1,2))
    plot(df$erogato0,predict(rf),xlab="valore reale",ylab="valore
      stimato",main="Training Set")
    par(ask=F)
    abline(0,1)
    plot(dfv$erogato0,predict(rf,dfv),xlab="valore reale",ylab="valore
      stimato",main="Validation Set")
    abline(0,1)
    par(ask=T)
    plot(predict(rf),df$erogato0-predict(rf),xlab="Valori
      stimati",ylab="Residui",main="Residui verso valori stimati")
    err_rel<-(predict(rf,dfv)-dfv$erogato0)*100/dfv$erogato0
    hist(err_rel,nclass=20,xlab="Errori
      relativi",ylab="Frequenza",main="Distribuzione errori relativi")
    par(ask=T)
    cat("==== Errore Relativo (Validation set) ====\n")
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))

    cat("\n==== Errore Relativo (Training set) ====\n")
    err_rel<-(predict(rf)-df$erogato0)*100/df$erogato0
```

```
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    rm(err_rel)
    rm(err_freq_ass)
}
nn.plot_s <- function () {
    windows(width=11,height=6)
    plot.new()
    par(ask=F)
    par(mfrow=c(1,2))
    plot(cs.nn_train,ss.nn_train,xlab="valore reale",ylab="valore
      stimato",main="Training Set")
    abline(0,1)
    plot(cs.nn_valid,ss.nn_valid,xlab="valore reale",ylab="valore
      stimato",main="Validation Set")
    abline(0,1)
    par(ask=T)
    plot(ss.nn_train,cs.nn_train-ss.nn_train,xlab="Valori
      stimati",ylab="Residui",main="Residui verso valori stimati")
    err_rel<-(ss.nn_valid-cs.nn_valid)*100/cs.nn_valid
    hist(err_rel,nclass=20,xlab="Errori
      relativi",ylab="Frequenza",main="Distribuzione errori relativi")
    par(ask=T)
    cat("==== Errore Relativo (Validation set) ====\n")
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    cat("\n==== Errore Relativo (Training set) ====\n")
    err_rel<-(ss.nn_train-cs.nn_train)*100/cs.nn_train
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    rm(err_rel)
    rm(err_freq_ass)
```

```
}
rf.plot_s <- function () {
    windows(width=11,height=6)
    plot.new()
    par(ask=F)
    par(mfrow=c(1,2))
    plot(cs_train,ss_train,xlab="valore reale",ylab="valore
      stimato",main="Training Set")
    abline(0,1)
    plot(cs_valid,ss_valid,xlab="valore reale",ylab="valore
      stimato",main="Validation Set")
    abline(0,1)
    par(ask=T)
    plot(ss_train,cs_train-ss_train,xlab="Valori
      stimati",ylab="Residui",main="Residui verso valori stimati")
    err_rel<-(ss_valid-cs_valid)*100/cs_valid
    hist(err_rel,nclass=20,xlab="Errori
      relativi",ylab="Frequenza",main="Distribuzione errori relativi")
    par(ask=T)
    cat("==== Errore Relativo (Validation set) ====\n")
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    cat("\n==== Errore Relativo (Training set) ====\n")
    err_rel<-(ss_train-cs_train)*100/cs_train
    cat(paste("Media =", mean(err_rel),"\n"))
    cat(paste("Std.Dev. =", sd(err_rel),"\n"))
    cat("Frequenze relative:")
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-3,3,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    err_freq_ass<-table(cut(err_rel,b=c(-Inf,-5,5,+Inf)))
    print(err_freq_ass*100/sum(err_freq_ass))
    rm(err_rel)
    rm(err_freq_ass)
}
```

Riferimenti bibliografici

Azzalini A., B. Scarpa (2004), *Analisi dei dati e data mining*, Springer Verlag

Berk R. A. (2008a), *Statistical Learning from a Regression Perspective*, Philadelphia, PA (USA), Springer Verlag (Springer Series in Statistics)

Berk R.A. (2008b), *An Introduction to Statistical Learning from a Regression Perspective*, PA (USA)

Berk R.A., L. Sherman, G. Barnes, E. Kurtz e A. Lindsay, A. (2009), Forecasting Murder within a Population of Probationers and Parolees: A High Stakes Application of Statistical Forecasting.", in *Journal of the Royal Statistical Society (Series A)*, Wiley Interscience, 191-211

Breiman L. (2001), Random Forests, in *Machine Learning*, Berkeley, CA (USA), 5-32

Breiman L., J.H. Friedman, R.A. Olshen e C.J. Stone (1984), *Classification and Regression Trees*, Wadsworth, CA (U.S.A.)

Cammarata S. (1997), *Dal perceptron alle reti caotiche e neuro-fuzzy*, Milano, Etas Libri

Campisano A., P. Cutore, Z. Kapelan, C. Modica e D. Savic (2007), La previsione stocastica dei consumi idrici urbani attraverso l'algoritmo SCEM-UA, in *Approvvigionamento e distribuzione idrica: esperienze, ricerca e innovazione*, Ferrara, 443-455

Da Peppo L. (2001), La realizzazione dell'acquedotto, in *Le sorgenti per Padova*, Padova, 101-143

Davanzo, F. (1982), Metodologia di previsione dei consumi a breve termine ai fini del risparmio energetico, in *Seminario di aggiornamento su: Il risparmio energetico nel servizio acquedottistico*, Padova, 151-177

Dell'Omodarme M. (2008), *Esercitazioni di statistica biomedica*, http://cran.r-project.org/doc/contrib/DellOmodarme-esercitazioni-R.pdf

Demuth, H., M. Beagle e M. Hagan (2009), *Neural Network Toolbox 6 – User's guide – The MathWorks*, Natick-MA (U.S.A.)

Efron, B. e R. Tibshirani (1993), *Introduction to the Bootstrap*, New York, Chapman & Hall

Kuhn S., B. Egert, S. Neumann and C. Steinbeck (2008), Building blocks for automated elucidation of metabolites: Machine learning methods for NMR prediction, *BMC Bioinformatics*, 9:400, (25/9/2008), Halle (Germany), http://www.biomedcentral.com/1471-2105/9/400

Lazzerini B. (2002), *Introduzione alle reti neurali*, Pisa, http://info.iet.unipi.it/~lazzerini/sisd/Reti.pdf

Maffei M. (2001), L'acqua potabile in città, in *Le sorgenti per Padova*, Padova, 9-11

Maindonald J. H., J. Braun (2007), *Data analysis and graphics using R: an example-based approach. Second Edition*., Cambridge (UK), Cambridge University Press

Nash J.E.,J.V. Sutcliffe, River flow forecasting through conceptual models, Part 1, A discussion of principles, *Journal of Hydrology*, 10: 282-290, 1970.

Ripley B.D. (1996), *Pattern Recognition and Neural Networks*, Cambridge (U.K.), Cambridge University Press

Verzani J. (2005), *Using R for introductory statistics*, Boca Raton FL (USA), Chapman&Hall/CRC