Università degli studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in
Scienze Statistiche

Tesi di Laurea

Uso del text mining per l'estrazione dello stadio tumorale da referti di anatomia patologica

Relatrice: Prof.ssa Giovanna Boccuzzo
Dipartimento di Scienze Statistiche

Correlatore: Prof. Bruno Scarpa
Dipartimento di Scienze Statistiche

Laureando: Pietro Belloni
Matricola 1155613

Anno Accademico 2018/2019

There's a self-congratulatory feeling in the air. We say things like "machine learning is the new electricity". I'd like to offer an alternative metaphor: machine learning has become alchemy.

Ali Rahimi
Research Department, Google Inc.

Indice

Introduzione

Le basi di dati dei sistemi sanitari contengono una grande quantità di informazioni utili. Molte di queste sono strutturate, e dunque facilmente reperibili per una semplice consultazione o una più sofisticata modellazione statistica. Un'altra parte di queste informazioni (si stima che sia circa il 40% del totale) risulta invece essere non strutturata ma contenuta all'interno di testi clinici (Dalianis, 2018): dalla necessità di reperirle nasce il *text mining* clinico, ovvero l'adattamento di uno strumento, il *text mining*, al contesto sanitario. L'obiettivo del *text mining* clinico è estrarre l'informazione contenuta nei testi clinici, colmando dunque il *gap* fra informazione strutturata e non strutturata (Spasic *et al.*, 2014) e permettendo l'accesso a una maggiore quantità di dati.

In questo lavoro di tesi si cercherà di implementare il *text mining* clinico su alcuni testi tratti da cartelle cliniche oncologiche fornite dal Registro Tumori del Veneto. In particolare, i testi messi a disposizione sono diagnosi facenti riferimento a casi di melanoma cutaneo incidenti per l'anno 2013. Le informazioni da estrarre sono la stadiazione TNM del tumore, la dimensione del tumore primitivo, il coinvolgimento di linfonodi e la presenza di metastasi.

Verrà dedicato un primo capitolo alla descrizione del *text mining*, delle sue applicazioni in campo clinico e dei suoi differenti approcci proposti in letteratura. Il secondo capitolo sarà dedicato al contesto in cui questa tesi prende forma: verrà descritto sia il melanoma cutaneo che il Registro Tumori del Veneto con le relative procedure di raccolta dei dati. Il terzo capitolo affronterà la struttura dei dati, ossia dei testi utilizzati per il *text mining*, focalizzandosi sulle problematiche che essi presentano. Nel quarto capitolo sarà esposta la procedura di *preprocessing*,

ovvero la prima fase del processo di *text mining* tramite cui, dai testi grezzi, è possibile estrarre delle variabili statistiche. Il quinto capitolo esporrà alcuni modelli statistici applicabili a queste variabili al fine di estrarre le informazioni relative ai tumori citate in precedenza. Il sesto e ultimo capitolo sarà dedicato a una analisi approfondita dei risultati ottenuti a cui seguirà una discussione sull'efficienza generale della procedura di *text mining* e ad alcune possibili implementazioni di cui questo studio potrà beneficiare in futuro.

Ognuno di questi argomenti sarà affrontato in parallelo sia dal punto di vista teorico che pratico. Tutte le applicazioni proposte (dalla costruzione del *dataset* al *preprocessing* fino ad arrivare alla stima dei modelli statistici) saranno infatti svolte tramite il software **R** (ver. 3.4.4 – *"Someone to Lean On"*), fatta eccezione per la stima delle reti neurali che sarà eseguita con **TensorFlow** (ver. 1.8.0).

I tempi di calcolo riportati sono stati ottenuti con un computer con le seguenti caratteristiche: processore Intel Core i5-2430, CPU 2.40GHz, RAM 3.85GB, sistema operativo a 64 bit.

Capitolo 1

Text mining in ambito clinico

La necessità di reperire informazioni in campo medico ha portato vari attori, quali clinici, statistici o ricercatori, a ricorrere all'utilizzo delle fonti più disparate: oggigiorno è comune estrarre informazioni da esami clinici, strumenti diagnostici o perfino immagini. I "testi liberi" (*free text*), ovvero quei testi scritti senza degli schemi strutturati, non fanno eccezione. In particolare, in campo medico è comune trovare testi liberi in cartelle cliniche, diagnosi, referti e schede: questi testi possono costituire una fonte di informazione importante a patto che siano analizzati correttamente. Essi contengono, ad esempio, informazioni sullo stato di salute dei pazienti, sui sintomi delle malattie o sui trattamenti consigliati. Se queste informazioni non sono state registrate separatamente, è necessario ricavarle dal testo. E se questa estrazione di informazioni viene fatta in maniera automatica, si ricorre al *text mining*.

1.1 Uno strumento preso in prestito

Il *text mining* è un'ampia classe di procedure statistiche, linguistiche e informatiche volte ad analizzare grandi insiemi di testi al fine di ricavarne informazioni. Spesso si fa riferimento al *text mining* come a una branca del *data mining*, ovvero l'estrazione di informazioni da grandi insiemi di dati, che nel caso specifico sono, per l'appunto, i testi. Sebbene i primi tentativi di analisi automatica dei testi risal-

gano agli anni '60, si può supporre che il *text mining* prenda forma dal *data mining* solo alla fine degli anni '90. Marti A. Hearst in un articolo del 1999 *Untangling Text Data Mining* definisce il text mining come una "disciplina nascente":

> *"The nascent field of text data mining (TDM) has the peculiar distinction of having a name and a fair amount of hype but as yet almost no practitioners. I suspect this has happened because people assume TDM is a natural extension of the slightly less nascent field of data mining."*

Come si nota dalla frase di Hearst, il *text mining* è una "estensione naturale" del *data mining*, con il quale condivide gran parte delle tecniche e dei modelli. Infatti, una volta fatto in modo che i testi non siano più una semplice stringa di caratteri ma delle vere e proprie variabili statistiche (o *features*, per usare un linguaggio tipico del *machine learning*), è naturale che la loro analisi sia svolta con un approccio di *data mining* piuttosto che con un approccio statistico classico.

La complicazione che il *text mining* aggiunge al *data mining* è insita in quel primo passaggio: fare in modo che i testi diventino variabili statistiche. Questo dipende principalmente da due cose: le caratteristiche del testo e il contesto in cui il testo è stato scritto. Le difficoltà riscontrate nell'analisi dei testi clinici sono proprio dovute a questi due fattori. In primo luogo, i testi clinici hanno delle caratteristiche peculiari, che verranno illustrate in seguito. In secondo luogo, il *text mining* non è cresciuto in ambito clinico, bensì in un altro contesto, quello legato ai *social network* e alla *sentiment analysis*, dunque il tipo di testo su cui gli strumenti di *text mining* sono stati sviluppati è totalmente diverso dal tipo di testo trattato in seguito.

Gli sviluppi dell'analisi dei testi nell'ambito dei *social network* e la ricerca fatta in quel contesto sono stati enormi negli ultimi anni, ma non verranno trattati in questa tesi (per una lettura in merito, si veda l'ottimo lavoro di Ceron, Curini e Iacus, *Social Media e Sentiment Analysis: L'evoluzione dei fenomeni sociali attraverso la Rete*, 2014). Si può addirittura affermare che l'utilizzo del *text mining* sia quasi totalmente dedicato al contesto web e che altri utilizzi vengano considerati di nicchia. Basti pensare che lo stesso Hearst, come conclusione del suo articolo

citato in precedenza, definisce il *text mining* come vera e propria analisi di grandi insiemi di testi online:

> *"I have attempted to suggest a new emphasis: the use of large online text collections to discover new facts and trends about the world itself."*

L'adattamento degli strumenti di *text mining* al contesto clinico è un campo di ricerca minore e la letteratura in merito è più contenuta. Questo porta a definire il *text mining* uno strumento "preso in prestito" dall'ambito *social*, non tanto perché esso sia nato nell'ambito *social*, ma piuttosto perché lì si trovano i suoi maggiori sviluppi e contributi.

1.2 Le caratteristiche dei testi clinici

I testi clinici differiscono dei testi solitamente trattati con il *text mining* per molti aspetti, in particolare perché sono scritti da professionisti quali medici, infermieri, radiologi, assistenti sanitari, ricercatori...Proprio per questo hanno la particolarità di essere altamente specifici riguardo al gergo, il quale risulta complicato o addirittura incomprensibile a chi non è del settore. Inoltre, va considerato che i testi sono scritti ad uso e consumo dei professionisti stessi (per esempio: una diagnosi può essere redatta da un patologo e letta da un oncologo) e non vengono diffusi al di fuori della cerchia di soggetti in grado di comprenderli.

Un'altra complicazione è data dal fatto che spesso i testi clinici sono scritti in fretta: si pensi alle diagnosi fatte da un medico che ha a disposizione pochi minuti per paziente, e contengono abbreviazioni, errori ortografici, frasi troncate, verbi sottintesi e acronimi (Allvin *et al.*, 2011; Patrick e Nguyen, 2011). Si stima che, in campo clinico, le parole frutto di abbreviazioni, acronimi o errori ortografici possano ammontare fino al 30% del testo, anche se questa percentuale varia notevolmente in base alla lingua (Pakhomov *et al.*, 2005). Sorprendentemente, il testo delle cartelle cliniche contiene errori ortografici e abbreviazioni in quantità perfino superiore ai testi della comune messaggistica *on line* che, nell'immaginario collettivo, rappresenta l'esempio di scrittura frettolosa e sgrammaticata (Tabella 1.1).

	Errori ortografici (sul totale delle parole)	**Abbreviazioni (sul totale delle parole)**
Testo da cartelle cliniche	10.00%	10.60%
Testo da messaggistica (chat, SMS)	5.00% - 6.00%	5.00%
Testo scritto al computer (Word)	0.20%	
Testo da articolo di giornale	0.05% - 0.44%	
Testo web	0.80%	
Testo scritto a mano	1.50% - 2.50%	

Tabella 1.1: Errori ortografici e abbreviazioni nelle cartelle cliniche (adattato da Ehrentraut *et al.*, 2012)

Queste caratteristiche dei testi in esame complicano notevolmente la loro analisi, sia essa automatica o manuale.

1.3 L'estrazione manuale delle informazioni dai testi come *gold standard*

L'estrazione di informazione dai testi clinici avviene per lo più a mano: un soggetto opportunamente qualificato legge il testo e registra in un database le informazioni richieste. Questo lavoro avviene di norma nei registri tumori, i quali stanno iniziando a dotarsi di strumenti di *text mining* solo di recente. L'estrazione manuale di informazione è molto dispendiosa in termini di tempo e costi, inoltre i lavoratori che se ne occupano (*data manager*) leggono i testi per diverse ore di fila e possono commettere errori. Per valutare questo errore sarebbe necessario condurre uno studio con più *data manager* che estraggono le medesime informazioni dagli stessi testi per poi valutare i risultati ottenuti con una misura di concordanza (come la k di Cohen). Nonostante non sia perfetta, l'estrazione manuale di informazioni viene comunemente considerata come un *gold standard* per valutare la bontà degli strumenti di *text mining* in ambito clinico, e sarà considerata il *gold standard* anche

in questa tesi.

1.4 I principali utilizzi del *text mining* in ambito clinico

L'estrazione di informazione dai testi clinici può avere svariati utilizzi. Velupillai e Kvist (2012) li riassumono in tre principali classi: controllo degli eventi avversi, supporto alla decisione clinica e riassunto automatico dei testi clinici. Alla luce della letteratura successiva (Warren *et al.*, 2012; Spasic *et al.*, 2014; Aalabdulsalam *et al.*, 2018), si è preferito modificare questa divisione nelle seguenti classi:

- Controllo degli eventi avversi: ovvero sfruttare i testi delle cartelle cliniche per portare alla luce quei pazienti che potrebbero essere propensi a peggiorare il loro stato di salute o riammalarsi. Qui l'informazione estratta tramite il *text mining* gioca il ruolo di *trigger*: se, ad esempio, un paziente viene considerato a rischio di peggiorare in base alla sua cartella clinica, si attivano le relative procedure di sorveglianza. È evidente come il *text mining* clinico in questo caso funga da classificatore del paziente tra le classi "a rischio" e "non a rischio", privilegiando ovviamente la massima sensibilità (in questo caso è meglio che un paziente sano venga erroneamente classificato come a rischio e che si mettano inutilmente in atto le procedure di controllo, piuttosto che un paziente a rischio venga erroneamente considerato stabile). Possiamo immaginare che questo utilizzo del *text mining* clinico sia il più semplice e il più immediato, dal momento che la classificazione in due sole classi è ampiamente trattata in ambito statistico e gli strumenti a disposizione per massimizzare la sensibilità siano molteplici. Un esempio di lavoro che sfrutta questo aspetto del *text mining* è quello di Ehrentraut *et al.*, 2012: gli autori analizzano alcune cartelle cliniche del Karolinska University Hospital in Svezia e classificano i relativi pazienti come a rischio o non a rischio di contrarre infezioni ospedaliere, ovvero quelle infezioni non presenti al momento dell'ingresso in ospedale ma contratte all'interno dell'ambiente ospedaliero stesso. I risultati ottenuti sono molto promettenti.

- Strutturazione dell'informazione: ovvero "incasellare" l'informazione contenuta nei *free text* in campi strutturati. Spesso accade che un medico di un dato reparto scriva un testo, per esempio una diagnosi, e dentro di essa indichi una serie di parametri relativi alla salute del paziente, per esempio lo stadio di un tumore o la sua pressione sanguigna. Per agevolare le analisi delle cartelle cliniche nel loro insieme può essere utile un database contenente tutti gli stadi tumorali e tutte le pressioni sanguigne indicate nei testi. Questo database, la cui creazione con il *text mining* sarebbe molto più rapida che tramite la lettura manuale dei testi, avrebbe una grande utilità decisionale in campo epidemiologico dato che riassumerebbe le caratteristiche mediche di una vasta popolazione. Questo utilizzo del *text mining* clinico è però più complesso del precedente: non si tratta di classificare un testo in due classi, ma di farlo in molteplici classi (tutti gli stadi tumorali) o addirittura di estrarre un valore continuo (la pressione sanguigna). Esempi relativi a questo contesto possono essere il lavoro di Aalabdulsalam *et al.* (2018) nonché l'analisi dei referti oncologici che verrà fatta successivamente in questa tesi.

- Estrazione automatica di informazioni dalla letteratura medica. Questo utilizzo del *text mining* in campo clinico è totalmente slegato dai precedenti, cionondimeno presenta una certa rilevanza. La mole di letteratura medica è impressionante e la sua analisi presenta una sfida per i clinici che devono prendere decisioni in tempi rapidi su che procedure seguire o che interventi effettuare: il *text mining* può essere usato come strumento per estrarre i concetti chiave da una vasta collezione di articoli su un determinato tema. Può anche fornire grande supporto ai ricercatori nello sviluppo di meta-analisi: dal momento che le banche dati contenenti articoli scientifici sono in forte crescita, la necessità di rendere le revisioni di *papers* sempre più veloci unita alla facilità con cui gli stessi possono essere recuperati ha aperto la via per uno sviluppo dell'analisi automatica dei testi anche in questo campo. Per una panoramica di questo utilizzo del *text mining*, si vedano i lavori di Zhu *et al.* (2013) e O'Mara-Eves *et al.* (2015).

Questi tre utilizzi del *text mining* clinico hanno come comune denominatore

l'estrazione di informazioni da un *free text* e il fine di aiutare gli addetti alla sanità (medici, infermieri, ricercatori...) a svolgere più velocemente operazioni che in precedenza erano effettuate prevalentemente a mano.

1.5 Due approcci al *text mining* clinico: un confronto

In generale, l'estrazione di informazioni da testo può avvenire principalmente in due modi (Aggarwal e Zhai, 2012): tramite un approccio basato su regole prestabilite (approccio *rule-based*) o tramite un approccio statistico (anche detto di *machine learning*). Per quanto il *text mining* nell'ambito dei *social media* sia ormai quasi totalmente basato sull'approccio statistico, in campo clinico persistono entrambi gli approcci.

1.5.1 Approccio *rule-based*

L'approccio *rule-based* al *text mining* è l'approccio più semplice e intuitivo, nonché il primo ad essere utilizzato: un primissimo esempio di trova in Pratt e Pacak (1969). Se si vuole classificare un testo all'interno di due o più classi, si stabilisce un set di regole che mettano in corrispondenza alcune parole del testo (*pattern*) con una data classe in cui il testo deve essere classificato. Se, ad esempio, siamo interessati a estrarre dal testo di una diagnosi oncologica il corrispondente stadio del tumore, l'operazione di *text mining* si può ricondurre a un problema di classificazione del testo all'interno delle classi corrispondenti ai possibili stadi del tumore. Se si adotta un approccio *rule-based*, il set di regole dovrà mettere in corrispondenza una data stringa di testo (ad esempio "tumore al seno di stadio II") con la classe corrispondente (in questo caso, la classe II). Come si può intuire, la definizione delle regole è un processo fondamentale che può rivelarsi molto dispendioso in termini di tempo, e quindi anche in termini di costi, ma se effettuato a dovere può garantire una classificazione estremamente accurata dei testi. Per alcuni esempi di *text mining* clinico con sistemi *rule-based* si vedano i lavori di Napolitano *et al.* (2010), Hanauer *et al.* (2007), Zhou *et al.* (2006) e Angelova *et al.* (2017). I primi tre affrontano proprio l'analisi di *free text* contenuto in cartelle

cliniche oncologiche scritte da patologi in lingua inglese, analisi che si proporrà in maniera simile nei capitoli successivi per testi in lingua italiana, L'ultimo è un singolare esempio di applicazione di *text mining* clinico per testi scritti con un alfabeto non latino ma cirillico.

L'approccio *rule-based* risente molto dei problemi relativi ai testi che sono stati esposti nei paragrafi precedenti: errori ortografici, sinonimi o abbreviazioni possono influenzare in maniera pesante il comportamento delle regole e quindi possono compromettere la successiva classificazione. Basti pensare a tutti i sinonimi del termine "tumore al seno": "tumore mammario", "tumore della mammella" o "tumore della ghiandola mammaria" per citarne alcuni. O, ancora, si può immaginare come "stadio II" possa essere indicato in un *free text*: "stad. II", "stadiazione II", "secondo stadio"... È difficile scrivere manualmente un set di regole che comprenda tutti i possibili sinonimi di tutti i termini contenuti nei testi, si ricorre quindi a dei software specifici che al loro interno contengono dei dizionari medici con l'elenco dei sinonimi, delle abbreviazioni e dei comuni errori ortografici. Tra questi si cita MetaMap (`https://metamap.nlm.nih.gov`) che risulta essere il più usato e probabilmente uno dei più efficienti. MetaMap è un software sviluppato dalla National Library of Medicine in grado di mettere in corrispondenza un testo di carattere medico con i termini medici standard UMLS (*Unified Medical Language System*) in modo da conoscerne i sinonimi e le abbreviazioni. MetaMap, così come gli altri software, è però sviluppato per testi in lingua inglese e non esiste un suo adattamento per la lingua italiana che possa considerarsi soddisfacente (Chiaramello *et al.*, 2016). Questa impossibilità di mappare in modo efficiente i sinonimi e le abbreviazioni risulta essere un grande svantaggio nell'utilizzo dell'approccio *rule-based* in contesti al di fuori della lingua inglese, che diventa quindi molto più lento e macchinoso.

1.5.2 L'approccio statistico

L'approccio statistico al *text mining* si rifà completamente alle tecniche di *data mining*. Per quanto sia ormai lo standard per l'analisi dei testi online, in campo clinico si è iniziato a utilizzare questo metodo solo di recente, ottenendo risultati

competitivi con l'approccio *rule-based* in tempi molto rapidi. Questo approccio richiede una prima fase di *preprocessing* dove il testo viene modificato e trasformato in variabili statistiche, segue poi una fase di classificazione dei testi effettuata con metodi di *data mining*. Entrambe le fasi verranno ampiamente discusse nei capitoli successivi. Per alcuni lavori di ricerca con utilizzo di *text mining* clinico con approccio statistico si veda, ad esempio, il già citato articolo di Ehrentraut *et al.* (2012) o quello di Martinez *et al.* (2013).

Il vantaggio dell'approccio statistico è senza dubbio la sua maggiore capacità di adattamento ai problemi discussi in precedenza: la classificazione è più robusta alla presenza di sinonimi, abbreviazioni e errori ortografici. Inoltre, mentre un set di regole è in grado di classificare solo i testi inerenti a un determinato ambito (se cambia l'ambito devono per forza cambiare le regole), le procedure di *data mining* alla base dell'approccio statistico possono essere applicate a ogni tipo di testo richiedendo solamente leggere modifiche. Il principale svantaggio di questo approccio sta nel fatto che, per ottenere un risultato competitivo, la classificazione deve essere di tipo supervisionato, ovvero deve essere presente un insieme di testi dei quali si conoscono già le informazioni da estrarre. In altre parole, di questo insieme di testi si deve conoscere la classificazione a priori se si vuole che le procedure statistiche siano in grado di classificare ulteriori testi. Questo implica che dietro ogni tentativo di *text mining* basato su un approccio statistico ci sia un lavoro manuale di estrazione delle informazioni dai testi, lavoro che richiede tempo e personale specializzato.

1.5.3 Comparazione fra i due metodi

Come detto in precedenza, entrambi gli approcci sopra esposti coesistono nelle applicazioni cliniche del *text mining*. La ricerca degli ultimi anni tende a favorire l'approccio statistico grazie alla spinta data dallo sviluppo delle tecniche di *data mining* e *machine learning*, ciononostante il personale clinico continua a prediligere l'approccio *rule-based* per via della sua semplicità interpretativa. Come riportato in Spasic *et al.*, (2014):

Approccio		
Rule-based	**Vantaggi**	Semplicità interpretativa; Assistenza con strumenti elettronici quali dizionari (principalmente per la lingua inglese); Alta precisione se le regole sono stabilite correttamente.
	Svantaggi	Sensibile alla scarsa qualità dei testi; Strettamente dipendente dal contesto di sviluppo; Necessità di lunghi tempi per la scrittura delle regole.
Statistico	**Vantaggi**	Robustezza alla scarsa qualità dei testi; Poca dipendenza dal contesto di sviluppo; Tempi di stima rapidi.
	Svantaggi	Difficoltà interpretativa (spesso l'interpretazione è impossibile); Necessità di un insieme di stima con testi già analizzati.

Tabella 1.2: Vantaggi e svantaggi dei due approcci al *text mining* clinico

> *"Clinician tend to prefer rule-based systems because of their explanatory power as opposed to alternative machine learning approaches (e.g. support vector machines) whose "black box" models do not provide insight or explanation into the reasons for a particular classification."*

I vantaggi e gli svantaggi dei due differenti approcci sono riassunti nella Tabella 1.2.

È importante notare come la distinzione tra l'approccio *rule-based* e l'approccio statistico sia meno netta di quanto possa apparire. Infatti, quello che avviene all'interno di un modello statistico di classificazione si può immaginare come la creazione automatica di regole che il modello stesso stabilisce in base agli algoritmi di stima. Le unità statistiche, ossia i testi, verranno poi classificate in base a quelle regole. Per illustrare meglio questo concetto si può prendere come esempio uno dei modelli più semplici ma allo stesso tempo più usati: l'albero di classificazione (Paragrafo 5.3). Un albero nella fase di stima stabilisce un set di regole (che altro non sono che partizioni dello spazio delle variabili statistiche) scegliendole tra

tutte le regole possibili. Il criterio di decisione di queste regole è automatico e si basa sulla minimizzazione di una funzione di perdita, ma il loro funzionamento è identico a quello delle regole stabilite manualmente nell'approccio *rule-based*. Altri modelli statistici che saranno descritti successivamente sono sicuramente più complessi dell'albero di classificazione, ma possono essere immaginati come fonte automatica di regole secondo le quali i testi vengono classificati.

1.6 Il *text mining* in ambito oncologico

Il *text mining* oncologico può essere inteso come l'applicazione del *text mining* clinico a testi di natura oncologica come, per esempio, diagnosi, schede di morte o commenti a materiale estratto con istologie. I testi che verranno analizzati nei capitoli successivi di questa tesi provengono da referti diagnostici scritti da patologi. Fortunatamente, la correttezza ortografica di questi testi può considerarsi superiore alla media dei testi clinici: come riporta Dalianis (2018) "generalmente, i report scritti da patologi sono scritti con maggior attenzione, frequentemente con una giusta ortografia." Il motivo sta nel fatto che i loro testi hanno spesso una semi-struttura che incanala il *free text* in un testo più ordinato e preciso. L'analisi di un testo oncologico resta comunque complicata, anche se il testo ha una qualità maggiore sono presenti errori ortografici e lessico settoriale.

Vengono qui riportate come esempio due delle diagnosi di anatomia patologica che saranno analizzate in seguito:

```
Frammenti polipoidi di mucosa di tipo respiratorio con flogosi cronica (A-B)
ad impronta erosiva, con metaplasia squamosaassociata a modificazioni di tipo
iperplastico-rigenerativo dell'epitelio.
```

```
Frammenti polipoidi di mucosa del grosso intestino con lieve flogosi cronica
aspecifica e con modesti e focali aspetti di iperplasia delle cripte (A).Lembi
di mucosa del grosso intestino con lieve flogosi cronica aspecifica e focale
fibrosi del chorion (B). Minuti e superficiali frammenti di mucosa del grosso
```

`intestino con lieve flogosi cronica aspecifica, iperplasia delle cripte efocale fibrosi del chorion (C).`

Il contesto da cui sono state estratte sarà illustrato nei capitoli successivi, ma una loro visione sommaria fa immediatamente capire come:

- Si noti la già citata semi-struttura tipica delle diagnosi di anatomia patologica: ci sono somiglianze che si ritrovano in alcuni testi. In particolare, la prima parte della diagnosi inizia allo stesso modo ("`Frammenti polipoidi di mucosa`") per poi divergere;
- Ci siano errori ortografici: nello specifico ci sono due parole scritte senza spazi tra una e l'altra ("`squamosaassociata`" sarebbe da sostituire con "`squamosa associata`"). Questo errore sarebbe difficile da individuare e correggere in maniera automatica senza l'aiuto di uno dei software specifici come MetaMap;
- Il lessico sia altamente tecnico.

Il numero di studi che applica il *text mining* in ambito oncologico con un approccio statistico è abbastanza contenuto. Quasi tutti questi studi (molti dei quali sono già stati citati) hanno come obiettivo l'estrazione di informazioni da testi di diagnosi e i loro risultati possono dirsi soddisfacenti. Purtroppo nessuno di questi è stato fatto su testi in lingua italiana, fatto rilevante dal momento che la lingua è una componente cruciale nelle procedure di *text mining*.

1.7 Riepilogo del capitolo

Il *text mining* è uno strumento la cui utilità ed efficienza sono ampiamente comprovate e il suo sviluppo può dirsi in grande crescita, in particolare in relazione all'analisi dei testi nei nuovi *social media*. In ambito clinico il *text mining* ha avuto uno sviluppo minore, ma comunque consistente. Le principali difficoltà sono dovute alle particolarità con cui i testi sono scritti, specialmente perché si tratta

di testi contenenti errori, abbreviazioni e linguaggio tecnico, dunque la loro analisi richiede particolare attenzione. Il maggior utilizzo del *text mining* in ambito clinico è l'estrazione di informazioni da un testo non strutturato, questa azione coincide con un'operazione di classificazione dei testi. I metodi più usati per eseguire questa classificazione sono due: l'approccio *rule-based* (che classifica i testi in base a regole prestabilite) e l'approccio statistico (che utilizza modelli di *data mining* per classificare i testi). Il primo metodo è quello correntemente più usato in ambito clinico, ma il secondo sta conoscendo una maggiore crescita nell'ultimo periodo. L'obiettivo dei capitoli seguenti di questa tesi è proprio quello di sfruttare l'approccio statistico per estrarre automaticamente delle informazioni da dei testi presenti in referti di anatomia patologica disponibili presso il Registro Tumori del Veneto.

Capitolo 2

Analisi del contesto: il Registro Tumori del Veneto e il melanoma cutaneo

L'obiettivo della tesi è estrarre informazioni da una collezione di *free text* applicando un approccio statistico con il fine di strutturare le informazioni contenute nei testi. In particolare, verranno usati testi di diagnosi oncologiche da cui estrarre lo stadio del tumore, la grandezza del tumore primitivo, il coinvolgimento dei linfonodi e la presenza di metastasi. I testi sono tratti da referti di anatomia patologica dei casi incidenti dell'anno 2013 raccolti dal Registro Tumori del Veneto per uno specifico tipo di tumore: il melanoma cutaneo.

Questo capitolo ha due obiettivi principali: presentare il contesto in cui sono stati reperiti i dati (ovvero il Registri Tumori del Veneto) e descrivere brevemente il melanoma cutaneo, le sue caratteristiche e la sua epidemiologia.

2.1 Il Registro Tumori del Veneto

Come riportato dall'Associazione Italiana Registri Tumori (AIRTUM):

> "I registri tumori sono strutture deputate alla raccolta e registrazione di tutti i tumori incidenti in un determinato territorio."

Dunque, il ruolo di un registro tumori è quello di raccogliere, registrare e studiare tutti i casi di tumore diagnosticati nell'area geografica di sua competenza. Sono molti i benefici portati dal lavoro svolto dai registri tumori: basti pensare agli ovvi vantaggi che ne trae il sistema sanitario ad avere a disposizione statistiche approfondite sull'incidenza delle varie malattie tumorali, statistiche che possono indirizzare le strategie di gestione delle risorse a disposizione delle aziende sanitarie locali. Quest'opera di archiviazione e analisi di dati deve essere svolta nel modo più uniforme e sistematico possibile al fine di fornire un quadro preciso del numero di nuovi casi, dei tipi di tumori e del decorso delle malattie. In questo senso, AIRTUM promuove la standardizzazione dei metodi di raccolta e registrazione dei dati nei diversi registri italiani e incentiva la condivisione dei dati a fini di ricerca.

Il territorio coperto da un registro è variabile: può coincidere con l'intera regione, con la provincia o con la zona coperta dall'azienda sanitaria locale. In Italia esistono 49 registri tumori accreditati presso l'AIRTUM che vanno a coprire circa il 70% della popolazione nazionale. A questi si aggiungono molte aree in cui è stata avviata un'attività di registrazione, ma i registri non sono ancora stati accreditati. Considerando anche questi enti, la percentuale di popolazione italiana coperta da un qualche registro sale al 98% (dati risalenti a ottobre 2017, `www.registri-tumori.it`). Il Registro Tumori del Veneto è il registro tumori più grande d'Italia: l'area di riferimento corrisponde alla regione Veneto e la popolazione censita ha raggiunto il 96% della popolazione regionale totale, ovvero 4689057 unità. L'ente ha completato la raccolta dei casi incidenti dall'anno 1987 all'anno 2010 (361121 casi) e la raccolta dei casi incidenti dell'anno 2013 (39751 casi) i quali sono anche seguiti con un *follow-up* aggiornato a ottobre 2017 (dati risalenti a marzo 2018, `www.registrotumoriveneto.it`).

2.1.1 Il processo di raccolta dei dati all'interno del Registro Tumori del Veneto

Per registrare un caso incidente, il Registro Tumori del Veneto utilizza i dati derivati principalmente da tre fonti:

1. I referti di anatomia patologica, ovvero i risultati di analisi istologiche, citologiche o di autopsie;

2. Le schede di dimissione ospedaliere (SDO), che contengono le diagnosi e altre informazioni relative al ricovero ospedaliero;

3. I certificati di morte, che contengono le cause di un eventuale decesso.

All'interno di questi documenti si trovano sia informazioni strutturate che informazioni non strutturate contenute nei testi liberi (come, ad esempio, le diagnosi). È quindi di grande utilità lo sviluppo di un sistema che porti a una strutturazione di quest'ultime informazioni per permettere analisi migliori.

Tutti i laboratori di anatomia patologica delle aziende sanitarie locali, delle aziende ospedaliere e delle strutture private accreditate nella Regione Veneto devono inviare al Registro Tumori gli archivi elettronici contenenti le fonti sopra riportate. Le informazioni provenienti da queste fonti sono incrociate con quelle contenute nell'anagrafe regionale per eliminare i casi che fanno riferimento ai cittadini non residenti nel territorio regionale. Successivamente, i dati raccolti sono processati in maniera automatica con una procedura di *record-linkage* che segnala casi dubbi sui quali vengono condotte delle verifiche manuali. Solo al termine di questo processo informatico (ed eventualmente manuale) un caso incidente viene registrato nelle basi di dati del Registro Tumori. Infine, i dati vengono messi a disposizione in forma aggregata sul sito del Registro Tumori, dove è possibile consultare direttamente le statistiche dell'incidenza dei tumori nel Veneto suddivisi per area geografica, sesso, età e tipo di tumore.

2.2 Il melanoma cutaneo

I testi che saranno analizzati con la procedura di *text mining* sono contenuti in referti di anatomia patologica e fanno riferimento a casi di melanoma cutaneo. Questi casi risultano incidenti nell'anno 2013 e sono stati raccolti dal Registro Tumori in un periodo tra il 2013 e il 2017.

2.2.1 Caratteristiche mediche del melanoma cutaneo

I tumori della pelle (cutanei) si raggruppano in due famiglie: i melanomi e le neoplasie cutanee di origine epiteliale, quali il carcinoma squamocellulare e il carcinoma basocellulare. I melanomi sono tumori maligni che hanno origine dai melanociti, ovvero le cellule responsabili della produzione di melanina (il pigmento a cui è dovuto il colorito della pelle), sono più rari degli altri tumori cutanei, ma sono molto più pericolosi perché possono espandersi in altre parti del corpo. I melanomi possono svilupparsi ovunque, ma è possibile trovarli più frequentemente sul busto (prevalentemente negli uomini), sulle gambe (prevalentemente nelle donne), sul collo e sul viso (Gallo e D'Amanti, 2018). Tipicamente le cellule del melanoma continuano a produrre melanina e dunque appaiono scure, ma non sono da escludere casi in cui questo non avvenga e il melanoma appaia chiaro, e quindi sia più difficile da identificare. I modi più efficaci per ridurre la mortalità del melanoma sono:

- La prevenzione di tipo primario, per far acquisire alla popolazione comportamenti atti a ridurre il rischio (uso di creme solari, riduzione dell'esposizione al sole durante le ore in cui è più forte...).
- La diagnosi precoce attraverso controlli periodici dei nei. Se un melanoma viene diagnosticato immediatamente può essere trattato prima che si espanda e, di conseguenza, la sopravvivenza del soggetto aumenta sensibilmente.
- Un trattamento adeguato, che tipicamente avviene tramite chirurgia (per i dettagli si veda il Paragrafo 2.2.4).

Una storia di ustioni solari ripetute, soprattutto in giovane età, aumenta sensibilmente il rischio di sviluppare il melanoma. Questo rischio è accresciuto da eventuali caratteristiche congenite del soggetto, quali la carnagione chiara, un elevato numero di nei e una familiarità con altri soggetti affetti da melanoma[1][2].

[1] "Tumori della pelle e melanomi", *Istituto Europeo di Oncologia*, `www.ieo.it`.
[2] "What Is Melanoma Skin Cancer?", *American Cancer Society*, `www.cancer.org`.

2.2.2 Epidemiologia del melanoma cutaneo

Dal momento che il contesto di studio è strettamente circoscritto alla Regione Veneto, di seguito verranno riportati alcuni dati sull'epidemiologia del melanoma cutaneo per la popolazione della regione di interesse. I dati sono forniti dal Registro Tumori del Veneto e sono liberamente consultabili all'indirizzo `https://gecoopendata.registrotumoriveneto.it`.

Incidenza

L'incidenza del melanoma cutaneo è in costante aumento dagli anni ottanta fino ai giorni nostri (Figura 2.1). Dal primo anno di disponibilità dei dati (il 1987) all'ultimo (il 2013) il tasso standardizzato di incidenza per 100000 abitanti è passato da 8.0 a 18.8 per le femmine e da 7.5 a 19.6 per i maschi. Questo aumento è dovuto sia alla crescente esposizione ai raggi solari e alle lampade abbronzanti sia al maggior ricorso allo screening dei nei.

Per l'anno d'incidenza 2013 il rischio cumulativo di sviluppare questo tipo di tumore era di 1 su 79 nelle donne e di 1 su 66 negli uomini. Questa diffusione ha fatto del melanoma il sesto tumore più diffuso in Veneto, pari al 3.9% di tutti i tumori. Tale dato aumenta sensibilmente se si considera la fascia d'età 0-49 anni, dove il melanoma è il terzo per diffusione nelle femmine (pari al 9.3% di tutti i tumori) e il primo per diffusione nei maschi assieme al tumore del testicolo (pari al 13.1% del totale). L'incidenza in Veneto è in linea con il dato nazionale (AIRTUM *working grup*, 2017), mentre l'Italia è uno degli stati la cui popolazione è più a rischio di contrarre questo tipo di tumore a causa dell'etnia caucasica tipica della maggioranza dei suoi abitanti.

Sopravvivenza

La sopravvivenza indica la probabilità di sopravvivere alla malattia ad un certo tempo dalla diagnosi di melanoma. Per le femmine, la sopravvivenza a 5 anni delle pazienti con diagnosi di melanoma è del 89.5% [Intervallo di Confidenza al 95%: 88.6 - 92.6] mentre il corrispettivo per i maschi è del 88.8% [I.C. 95%: 86.4 - 91.1].

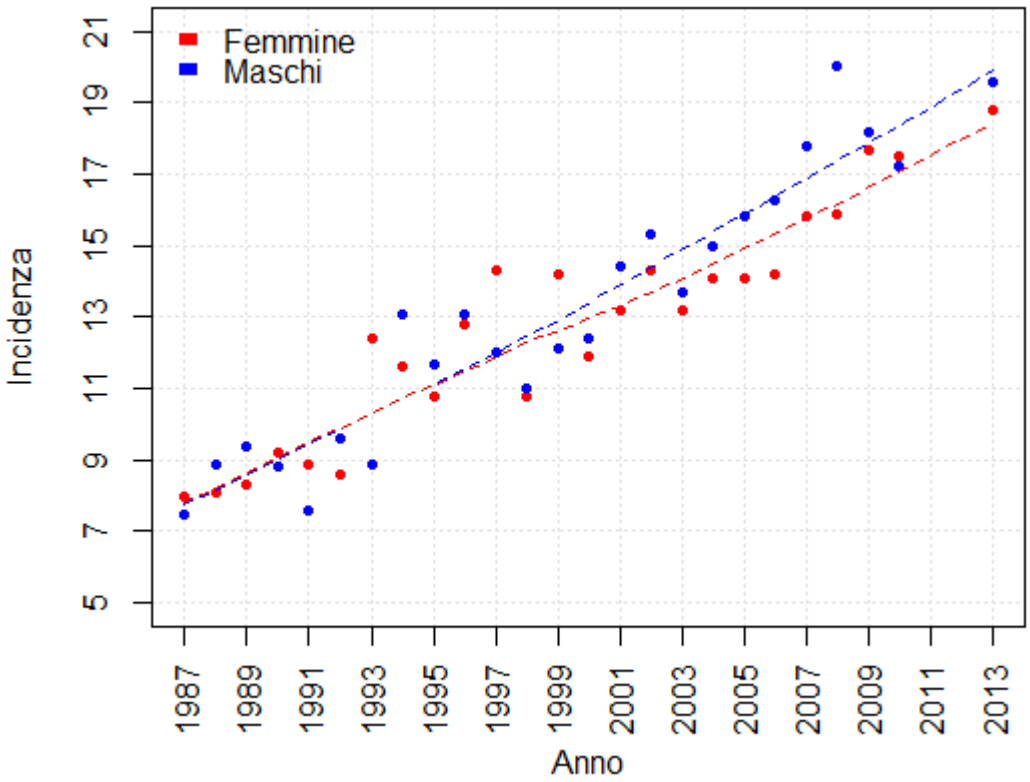

Figura 2.1: Andamento temporale dell'incidenza del melanoma cutaneo in Veneto (adattato da `https://gecoopendata.registrotumoriveneto.it`)

Questa sopravvivenza è alta se confrontata con altri tipi di tumore: per le femmine si tratta del secondo tumore con la sopravvivenza più alta (dopo il tumore alla tiroide) mentre per i maschi risulta essere il quarto (dopo i tumori al testicolo, prostata e tiroide).

In conclusione, il melanoma cutaneo può essere considerato un tumore con una diffusione medio-alta ma una mortalità bassa, a meno che non si presenti in uno stadio avanzato e si propaghi in altri organi.

2.2.3 Stadiazione del melanoma cutaneo

Al pari della maggior parte dei tipi di tumore, anche per il melanoma cutaneo è essenziale definire in maniera univoca lo stato di avanzamento alla diagnosi

	Parametro T	**Parametro N**	**Parametro M**
X	Spessore del tumore primitivo non misurabile	Impossibile stabilire il coinvolgimento dei linfonodi	Impossibile stabilire la presenza di metastasi
0	Nessun tumore primitivo	Nessun linfonodo coinvolto	Assenza di metastasi
1	Spessore del tumore primitivo inferiore a 1mm	Solo 1 linfonodo coinvolto	Presenza di metastasi
2	Spessore del tumore primitivo compreso fra 1mm e 2mm	Da 2 a 3 linfonodi coinvolti	-
3	Spessore del tumore primitivo compreso fra 2mm e 3mm	Più di 3 linfonodi coinvolti	-
4	Spessore del tumore primitivo superiore a 4mm	-	-

Tabella 2.1: Descrizione degli stadi T, N e M per il melanoma cutaneo

seguendo il sistema di classificazione TNM. La classificazione TNM descrive la grandezza del tumore primitivo (T), il coinvolgimento di eventuali linfonodi (N) e la presenza di metastasi a distanza (M) e da essa deriva la stadiazione del tumore. La Tabella 2.1 descrive i criteri utili a definire stadi T, N e M per il melanoma cutaneo. Esistono anche altri parametri con cui descrivere lo stato di avanzamento del melanoma (ad esempio il livello di Clark o lo spessore di Breslow), ma qui non verranno esposti per ragioni di semplicità.

Le informazioni ottenute dalla misura dei livelli T, N e M sono combinate per dare luogo alla stadiazione TNM del tumore (Tabella 2.2). In aggiunta, viene anche considerata la presenza di ulcere: l'ulcerazione è una rottura della pelle sopra il melanoma. La stadiazione TNM è di grande importanza per definire l'aggressività del tumore che si correla con sopravvivenza dei soggetti colpiti ed è anche utile per definire le linee guida del trattamento dello specifico tumore. Per questo motivo la sua diretta estrazione dai testi ha una grande utilità pratica. In aggiunta agli stadi precedenti, viene considerato uno "Stadio 0" di melanomi detti *in situ*, ovvero limitati allo strato superficiale della pelle. Questa classe di melanomi, la cui sopravvivenza è virtualmente pari al 100%, di seguito non sarà considerata poiché non rientra nella casistica disponibile. La stadiazione qui esposta è la meno speci-

Stadio	Descrizione	Sopravvivenza a 5 anni
I	T: 1 (o 2 in assenza di ulcere) N: 0 M: 0	89.0% - 95.3%
II	T: 2 (in presenza di ulcere), 3 o 4 N: 0 M: 0	67.7% - 77.4%
III	T: 1, 2, 3 o 4 N: 1, 2 o 3 M: 0	26.7% - 69.5%
IV	T: 1, 2, 3 o 4 N: 1, 2 o 3 M: 1	9.5% - 18.8%

Tabella 2.2: Stadiazione del melanoma cutaneo (adattato da Balch *et al.*, 2001)

fica: ne esistono di più precise ma per ragioni di semplicità si è preferito utilizzare la rappresentazione più grossolana. Per una descrizione accurata della stadiazione del melanoma cutaneo si veda Balch *et al.* (2001).

2.2.4 Trattamenti per il melanoma cutaneo

L'importanza di reperire e immagazzinare in database la stadiazione dei tumori è dovuta al fatto che non solo la stadiazione implica grandi differenze nella sopravvivenza, ma determina anche il tipo di trattamento utilizzato (AIRTUM *working group*, 2017; Italiano, 2018).

- Trattamento del melanoma di Stadio I:
  Il melanoma di Stadio I è trattato tipicamente con una operazione chirurgica di asportazione dell'area coinvolta detta "escissione ampia". Si asporta il melanoma più una parte circostante di pelle sana, questa viene analizzata per

assicurarsi che nessuna cellula tumorale sia stata lasciata ai bordi della pelle rimossa. Le linee guida indicano di non eseguire alcun trattamento adiuvante (trattamento successivo all'asportazione chirurgica del melanoma).

- Trattamento del melanoma di Stadio II:
  Anche in questo caso si effettua una escissione ampia. I pazienti con uno spessore del tumore maggiore di 1.5mm (o con un indice mitotico elevato) sono considerati a rischio di ricaduta e può essere indicato un trattamento adiuvante. Spesso si ricorre alla biopsia del linfonodo sentinella, ovvero il linfonodo che per primo è a rischio di essere coinvolto. Questo linfonodo viene asportato e analizzato, se non risulta infetto non è necessario proseguire con ulteriori trattamenti.

- Trattamento del melanoma di Stadio III:
  In questo caso si esegue la escissione ampia ma spesso non è sufficiente e bisogna ricorrere a ulteriori terapie. Frequentemente vengono asportati dei linfonodi e ne viene valutato lo stato di coinvolgimento. In molti casi l'operazione di asportazione del melanoma viene seguita da un trattamento adiuvante volto a prevenire la ricomparsa del tumore. Questo trattamento può consistere in una chemioterapia, in una elettrochemioterapia locoregionale (ovvero limitata alla zona colpita dal tumore), in una immunoterapia, in una terapia target (una terapia che agisce su precise caratteristiche biologiche del tumore) o in una combinazione delle precedenti. Può essere anche usata una radioterapia nell'area in cui sono stati rimossi i linfonodi, specialmente se molti di questi sono risultati intaccati dal tumore.

- Trattamento del melanoma di Stadio IV:
  Il melanoma di Stadio IV è il più aggressivo nonché il più complesso da curare, dal momento che è diffuso nei linfonodi e sotto forma di metastasi. Il tumore primitivo e i linfonodi coinvolti vengono rimossi chirurgicamente o trattati con una radioterapia. Se possibile, le metastasi negli organi interni sono rimosse chirurgicamente. Oltre la radioterapia si può ricorrere alla im-

munoterapia o alla terapia target, meno frequentemente alla chemioterapia, usata tipicamente se il paziente non risponde alle terapie già somministrate.

Come avviene per molti altri tipi di tumore, le terapie per il melanoma sono più efficienti se iniziate per tempo. La pericolosità del tumore aumenta con la sua diffusione, in particolare nei linfonodi e sotto forma di metastasi, quindi l'efficacia della terapia dipende gran parte dalla tempestività con cui si inizia la cura adeguata.

2.3 Riepilogo del capitolo

I testi che verranno usati per la procedura di *text mining* provengono da referti di anatomia patologica raccolti dal Registro Tumori del Veneto e fanno riferimento a casi di melanoma cutaneo incidenti per l'anno 2013. Il registro tumori del Veneto è il Registro Tumori più grande d'Italia e conta una copertura del 96% della popolazione regionale. Il Registro Tumori registra tutti i casi incidenti di tumore e raccoglie i dati usando diverse fonti, tra cui i referti di anatomia patologica, all'interno delle quali si trovano *free text* analizzabili con procedure di *text mining*.

Il melanoma cutaneo è un particolare tipo di tumore della pelle che ha origine dalle cellule responsabili della produzione di melanina. Nel territorio coperto dal Registro Tumori del Veneto il melanoma ha una incidenza in forte aumento. La sopravvivenza, d'altro canto, è anch'essa alta rendendolo uno dei tumori a più bassa letalità.

L'espansione del melanoma viene misurata attraverso alcuni parametri tra cui la grandezza del tumore primitivo (T), il coinvolgimento dei linfonodi (N) o la presenza di metastasi (M). Queste misure vengono combinate per dare origine alla stadiazione del tumore. La stadiazione ha molta importanza poiché guida la scelta del trattamento da riservare al paziente, dunque una procedura in grado di estrarre questa informazione dai testi avrebbe una grande utilità.

Capitolo 3

La struttura dei dati

Nel corso degli ultimi anni la statistica ha conosciuto un forte incremento dovuto principalmente alla crescita delle capacità di calcolo dei computer. Questo ha permesso la diffusione di tecniche in grado di analizzare dati strutturati nelle forme più disparate: immagini, reti o testi sono alcuni esempi di strutture di dati che si vanno affiancando alla classica misurazione di dati quantitativi/qualitativi effettuata su un campione (nonostante quest'ultima mantenga sempre una grande importanza). Perciò non ci si deve stupire se la parola "dati" in seguito indicherà un insieme di testi. Spesso un insieme di testi analizzati con *text mining* è detto anche *corpus* o *corpora*.

Come è già chiaro dai capitoli precedenti, i dati utilizzati in questo lavoro sono un insieme di testi di diagnosi tratte da referti di anatomia patologica facenti riferimento a casi di melanoma cutaneo incidenti per l'anno 2013. Gli obiettivi di questo capitolo sono due: descrivere questi testi e descrivere il *gold standard* necessario per la procedura di *text mining*.

3.1 I dati a disposizione

Si è utilizzata l'unione di 3 database forniti dal Registro Tumori del Veneto contenenti i referti di anatomia patologica facenti tutti riferimento ai casi di melanoma già citati. Il primo database conta 3065 referti raccolti in un periodo di

tempo compreso tra il 2013 e il 2015, il secondo database conta 81 referti raccolti nell'anno 2016 e il terzo conta 104 referti raccolti nell'anno 2017. Questi referti sono associati a 547 pazienti che hanno contratto il melanoma nell'anno 2013 e sono stati in cura in una struttura all'interno della Regione Veneto, questa struttura ha successivamente inviato i referti corrispondenti a ogni paziente al Registro Tumori. Si ha dunque una situazione in cui, ad ogni paziente, corrisponde spesso più di un referto.

Ogni referto ha al suo interno sia informazioni strutturate (ad esempio: il codice identificativo del paziente, il codice identificativi del referto, la data in cui è stato redatto...) che informazioni non strutturate sotto forma di *free text*. In particolare, sono contenute sotto forma di testo: la diagnosi, il risultato di un esame macroscopico del tessuto tumorale e uno di un esame microscopico dello stesso tessuto.

3.1.1 Selezione dei testi

Si è scelto di utilizzare solamente il testo della diagnosi per estrarre le informazioni sullo stato di avanzamento del tumore. Questa scelta è stata fatta perché, in accordo con il Registro Tumori del Veneto, si ritiene che la diagnosi sia il testo più esaustivo nel descrivere le informazioni che è di interesse estrarre. L'informazione contenuta nei testi di macroscopia e microscopia, a confronto, potrebbe risultare ridondante; inoltre analizzare con una procedura di *text mining* tre testi allo stesso tempo può portare a un costo computazionale molto elevato a fronte di un guadagno in termini di efficienza modesto, se non nullo. Una scelta analoga, ovvero l'utilizzo del testo della sola diagnosi per estrarre informazioni riguardo al tumore, si può ritrovare in letteratura nei lavori di McCowan *et al.* (2006; 2007) e Martinez *et al.* (2013), i quali usano proprio un approccio statistico al *text mining* simile a quello che sarà esposto in seguito.

Non tutti i referti hanno all'interno i testi, in alcuni casi i relativi campi sono stati semplicemente lasciati in bianco. In particolare, il 20.8% delle diagnosi risulta mancante, così come lo è il 32.1% delle macroscopie e ben il 94.7% delle microsco-

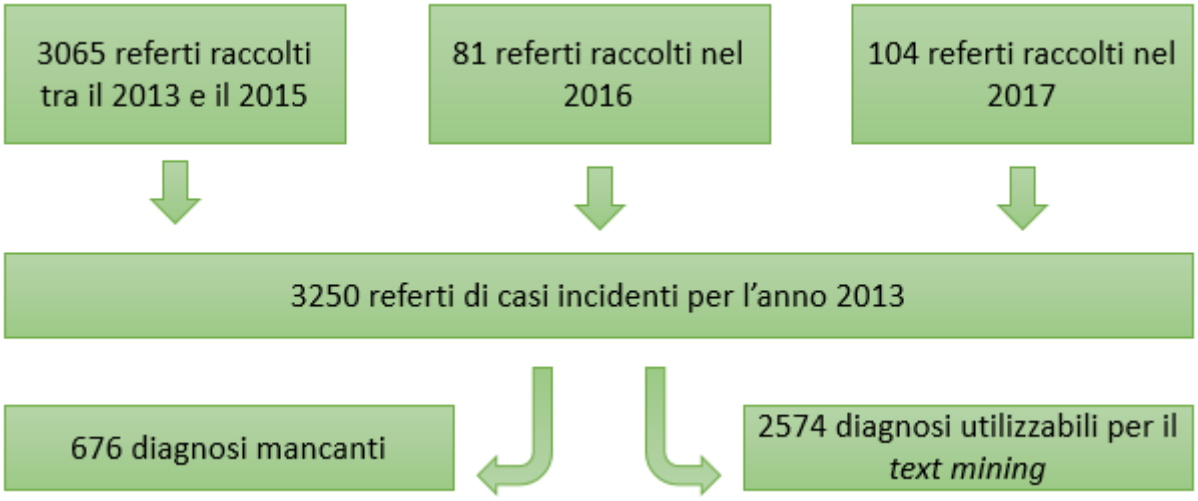

Figura 3.1: Selezione dei testi usati per la procedura di *text mining*

pie. I testi di diagnosi disponibili per la procedura di *text mining* vengono quindi ridotti a 2574 unità (Figura 3.1).

3.1.2 Le problematicità dei testi

Come ampiamente argomentato nel Capitolo 1, i testi contenuti nei referti clinici sono tipicamente problematici da analizzare tramite *text mining*, e i testi usati in questo contesto non fanno eccezione. Le problematiche legate ai testi sono quelle tipiche dei testi clinici elencate nel Paragrafo 1.2: abbreviazioni, errori ortografici e presenza di un linguaggio tecnico e variegato. A questi problemi se ne aggiungono altri che sono stati constatati nei testi in esame:

Testi fortemente eterogenei

Dal momento che il Registro Tumori raccoglie i referti da tutte le strutture sanitarie della regione, i dati provengono da fonti diverse. Questo ha portato a una grande eterogeneità nei testi, soprattutto nella loro forma e nella loro lunghezza: si osservano alcune diagnosi di poche parole e altre molto lunghe (Tabella 3.1) che possono arrivare ad avere al loro interno addirittura più paragrafi o elenchi puntati.

Minimo	1° qart.	Mediana	3° qart.	Massimo	Media	Dev. std.
7	129	216	432	3314	407.20	485.68

Tabella 3.1: Distribuzione del numero di caratteri delle diagnosi dei referti di anatomia patologica

Queste lunghezze aumentano la complessità computazionale in fase di stima dei modelli di *text mining*, mentre le formattazioni particolari verranno gestite nella fase di *preprocessing* dei dati.

Come paragone, si pensi a uno degli utilizzi più diffusi del *text mining*: l'analisi dei *tweet*. Come noto, i *tweet* contano un massimo di 140 caratteri (portati a 280 solo dal 2017) e al loro interno non presentano strutture testuali particolari quali possono essere gli elenchi puntati. È evidente come (considerando una pari numerosità campionaria) il *text mining* sui *tweet* sia un problema ben più semplice da affrontare.

Residui di formattazioni precedenti

Molti testi presentano simboli non pertinenti. Se ne riporta uno:

```
{\\rtf1\\ansi\\ansicpg1252\\uc1\\deff0{\\fonttbl??{\\f0\\fswiss\\fcharset0\\
fprq2Arial;}??{\\f1\\froman\\fcharset0\\fprq2 Times NewRoman;}??{\\f2\\froman
\\fcharset2\\fprq2Symbol;}}??{\\colortbl;\\red0\\green0\\blue0;\\red255
\\green255\\blue255;\\red0\\green0\\blue0;}??{\\stylesheet{\\s0\\itap0\\
nowidctlpar\\f0\\fs24[Normal];}{\\*\\cs10\\additive Default Paragraph Font;}}??
{\\*\\generatorTX_RTF3213.0.501.501;}??\\deftab1134\\paperw11905\\paperh16838\\
margl794\\margt0\\margr1247\\margb567\\widowctrl??{\\*\\background{\\shp{\\*\\
shpinst\\shpleft0\\shptop0\\shpright0\\shpbottom0\\shpfhdr0\\shpbxmargin\\
shpbxignore\\shpbymargin\\shpbyignore\\shpwr0\\shpwrk0\\shpfblwtxt1\\shplid1025
{\\sp{\\sn shapeType}{\\sv1}}{\\sp{\\sn fFlipH}{\\sv 0}}{\\sp{\\sn fFlipV}
{\\sv 0}}{\\sp{\\snfillColor}{\\sv 16777215}}{\\sp{\\sn fFilled}{\\sv 1}}{\\sp
{\\snlineWidth}{\\sv 0}}{\\sp{\\sn fLine}{\\sv 0}}{\\sp{\\sn fBackground}{\\sv
```

```
1}}{\\sp{\\snfLayoutInCell}{\\sv1}}}}}\\pard\\itap0\\nowidctlpar\\plain\\
f1\\fs24\\cf3 Lembi di mucosa gastrica di tipo antrale con lieve flogosi
cronica inattiva, follicolare, associata a atrofia ghiandolare, lieve e
a metaplasia intestinale, lieve, completa.\\par Ricerca Helicobacter pylori
negativa.}
```

In questo testo, portato come esempio, è chiaro che la parte d'interesse sia solamente "Lembi di mucosa gastrica di tipo antrale con lieve flogosi cronica inattiva, follicolare, associata a atrofia ghiandolare, lieve e a metaplasia intestinale, lieve, completa. Ricerca Helicobacter pylori negativa.". La presenza di altri caratteri di disturbo è verosimilmente dovuta al fatto che alcuni testi, prima di confluire nei database, avevano delle formattazioni e non erano semplice "testo piano". Dopo l'acquisizione dei testi dalle fonti e il suo salvataggio nei database, si è persa la formattazione del testo, ma non i caratteri che la definivano.

Questa conclusione è stata suggerita dalla presenza di parole che richiamano la formattazione dei testi (`rtf`, `ansi`...) o tipi di caratteri (`Arial`, `Times New Roman`...). La presenza di caratteri di disturbo di questo tipo dovuti a una vecchia formattazione si ritrova nel 13.3% dei testi. Fortunatamente, anche questo problema può essere risolto in fase di *preprocessing* dei dati.

3.2 Il *gold standard*

L'approccio statistico al *text mining* corrisponde in buona sostanza a una classificazione supervisionata dei testi. In altre parole, la classe associata a ciascun testo corrisponde all'informazione da estrarre e si utilizza un modello che sia in grado di scegliere una delle classi sulla base del contenuto del testo. Questo modello va stimato utilizzando un gruppo di testi la cui classificazione è già nota a priori grazie alla presenza di un *gold standard*.

Si noti che senza il *gold standard* si sarebbe dovuto ricorrere a una classificazione non-supervisionata: una procedura nota come *text clustering* la cui resa è solitamente inferiore alle procedure supervisionate (Chaovalit e Zhou, 2005). Per una descrizione generale delle differenze tra metodi di classificazione supervisionata

e non-supervisionata si veda Hastie *et al.* (2013). Il *gold standard* più utilizzato nel *text mining* clinico è la classificazione manuale fatta da un esperto adeguatamente formato.

3.2.1 Il "Progetto per la registrazione ad alta risoluzione del melanoma cutaneo"

Nel caso dei referti di anatomia patologica in esame, la classificazione manuale è stata fatta all'interno del "Progetto per la registrazione ad alta risoluzione del melanoma cutaneo" curato dalla Dott.ssa Irene Italiano e supervisionato dal Prof. Carlo R. Rossi (Istituto Oncologico Veneto) e dal Dott. Manuel Zorzi (Registro Tumori del Veneto). Questo progetto, frutto del coordinamento tra l'Istituto Oncologico Veneto e il Registro Tumori del Veneto, ha permesso la registrazione manuale in un database strutturato di molte informazioni riguardanti i casi di melanoma. Tra le altre, sono state registrati i parametri T, N e M del melanoma e la sua stadiazione generale. Questa classificazione manuale dei casi è un perfetto *gold standard* per i modelli di *text mining* che andranno a classificare in automatico i testi delle diagnosi.

3.2.2 Descrizione del *gold standard*

La procedura di *text mining* punta a estrarre lo stadio del tumore, il parametro T (grandezza del tumore primitivo), il parametro N (il coinvolgimento dei linfonodi) e il parametro M (la presenza di metastasi). Riconducendosi a un contesto statistico, ciò equivale a stimare quattro modelli di classificazione le cui risposte, o *outcome*, sono le quattro variabili del *gold standard* che indicano le suddette caratteristiche del tumore e i dati, o variabili indipendenti, sono tratte dai testi.

I 2574 testi a disposizione sono associati ciascuno a quattro *gold standard* (uno per ogni *outcome*):

1. Lo stadio del tumore: ha 6 classi, quelle descritte nella Tabella 2.2 più una che raccoglie i casi non definibili (Stadio X) e quelli mancanti;

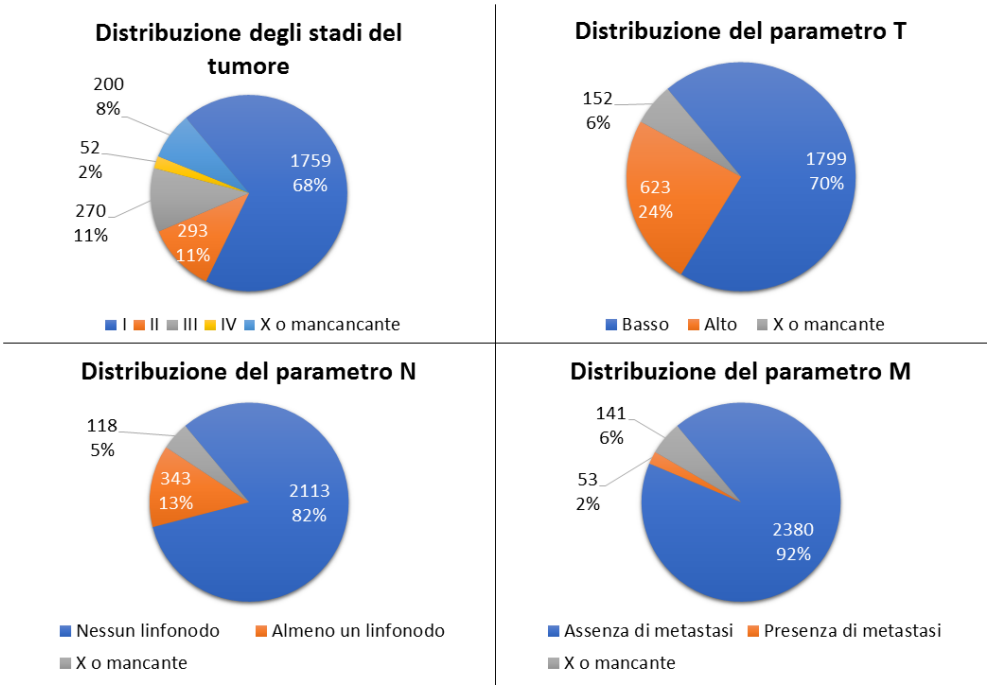

Figura 3.2: Distribuzione delle classi degli *outcome* nel *gold standard*

2. La grandezza del tumore primitivo (T): dal momento che il parametro T influisce sullo stadio del tumore quando è superiore a 2 (o è pari a 2 ma si è in presenza di ulcerazione) si è scelto di ridurre le classi a tre: "T basso" (T pari a 1 o a 2 senza ulcerazione), "T alto" (T maggiore di 2 o pari a 2 con ulcerazione) e "T X o assente" (T non definibile o mancante);

3. Il coinvolgimento dei linfonodi (N): similmente, dato che il parametro N influisce sullo stadio del tumore se maggiore di 0, si è scelto di ridurre le classi a tre anche in questo caso: "Nessun linfonodo" (N pari a 0), "Almeno un linfonodo" (N maggiore di 0) e "N X o assente" (N non definibile o mancante);

4. La presenza di metastasi (M): il parametro M ha tre classi: "Assenza di metastasi" (M pari a 0), "Presenza di metastasi" (M pari a 1) e "M X o assente" (M non definibile o mancante).

Una scelta analoga per l'accorpamento delle classi dei parametri T, N e M si ritrova in McCowan *et al.* (2006). Dalla distribuzione di tutti gli *outcome* si nota un forte sbilanciamento delle classi (Figura 3.2). Nel caso dello stadio del tumore il 68% del *gold standard* è stato classificato come Stadio I e il 24% come uno stadio superiore al I. Una situazione molto simile è quella colta dal parametro T, dove il 70% del *gold standard* presenta una grandezza del tumore primitivo bassa contro il 24% che la presenta alta. Ancora più sbilanciate sono le distribuzioni di N e M: l'82% non presenta un coinvolgimento linfonodale (contro il 13% che lo presenta), mentre il 92% non presenta metastasi (contro solo il 2% che le presenta).

Il forte sbilanciamento delle classi degli *outcome* crea non pochi problemi ai modelli statistici di classificazione che possono avere una buona capacità predittiva per le classi più frequenti ma una scarsa nelle classi meno frequenti. Lo studio di questo problema è stato a lungo affrontato in letteratura (Chawla, 2003; Chawla *et al.*, 2004; Cieslak, 2008) e le soluzioni abbondano (si veda, ad esempio, Menardi e Torelli, 2014). Nel caso in esame il problema del forte sbilanciamento del *gold standard* verrà affrontato in fase di stima dei modelli.

3.3 Riepilogo del capitolo

I dati utilizzati in per la procedura di *text mining* sono sotto forma di testo e consistono in 2574 diagnosi contenute in altrettanti referti di anatomia patologica. Questi presentano le criticità tipiche dei testi clinici già esposte nei capitoli precedenti, in aggiunta hanno delle problematicità legate all'origine stessa dei testi. Le problematicità saranno gestite nella fase di *preprocessing* dei testi che avrà luogo nel capitolo successivo.

La procedura di *text mining* può essere ricondotta a una classificazione supervisionata dei testi basata su un *outcome* e su una serie di variabili ricavate dai dati (ovvero i testi a disposizione). Nel caso in esame, gli *outcome* sono forniti da un *gold standard* ricavato dal "Progetto per la registrazione ad alta risoluzione del melanoma cutaneo" che ha permesso a un esperto di classificare manualmente i

testi delle diagnosi. Questa classificazione manuale verrà utilizzata per stimare i modelli di classificazione che costituiscono il cuore della procedura di *text mining*.

Capitolo 4

Il *preprocessing*

Nel capitolo precedente si è discussa l'origine e la struttura dei dati disponibili sotto forma di testo. Inoltre, si è discusso di come un problema di *text mining* sia riconducibile a un problema di classificazione supervisionata dei testi dove le variabili esplicative (o *features*) sono estratte dai testi stessi. L'obiettivo di questo capitolo è fornire una spiegazione adeguata di come da un testo grezzo si possa passare a un formato utilizzabile per la stima di un modello di classificazione attraverso una fase di *preprocessing*. Più che a una fase preliminare, si deve pensare al *preprocessing* come la vera e propria prima fase del *text mining*, a cui seguirà una seconda fase di stima dei modelli di classificazione.

Questo capitolo è organizzato in modo che ogni paragrafo sia dedicato a uno dei passaggi eseguiti in fase di *preprocessing*: per ognuna di queste fasi verrà descritta l'idea che ne sta alla base, l'algoritmo utilizzato, dove presente, e la sua applicazione sulla collezione dei testi delle diagnosi di anatomia patologica.

4.1 La normalizzazione dei testi

Nel Paragrafo 1.2 è emerso come i testi siano un formato di dati estremamente complesso: le frasi non sono semplici insiemi di parole, ma tra le parole stesse sussistono delle relazioni ben precise che a volte ne modificano il significato. Inoltre, in una frase sono presenti anche punteggiatura, simboli e numeri. Per fare in modo

che una frase sia "digeribile" da un algoritmo di stima bisogna spogliarla da tutte le parole e i simboli che ne aumentano la complessità e fare in modo che restino solo le parole chiave che da sole sono in grado di contenere gran parte dell'informazione (Kwartler, 2017).

4.1.1 Rimozione della punteggiatura, dei simboli e trattamento delle lettere maiuscole

È ben noto che un approccio statistico richieda sempre un compromesso: si rinuncia a modellare parte dell'informazione contenuta nei dati in modo che il modello sia più semplice e acquisisca un adattamento più flessibile. Nel caso in esame, gran parte degli elementi di una frase portano con sé almeno un minimo di informazione, ma siccome tenere conto di ognuno di questi elementi diventerebbe troppo complesso, si sceglie di eliminarne alcuni.

In primo luogo, è stata rimossa la punteggiatura. A volte la presenza di punteggiatura modifica il significato delle parole in una frase, dunque la sua rimozione coincide con una perdita di informazione. Nonostante ciò, la rimozione della punteggiatura è eseguita nella fase di *preprocessing* in ogni lavoro di *text mining* perché sarebbe estremamente complesso tener conto di decine di migliaia di punti e virgole durante la stima dei modelli. Analogamente, vengono rimossi tutti i simboli (\$, \, &...). Nell'insieme di testi trattato in questa tesi, la rimozione dei simboli è fondamentale: nel Paragrafo 3.1.2 si è visto come simboli superflui legati a una formattazione precedente siano assai diffusi. Si arrivano a osservare casi in cui gran parte del testo è composto da simboli di disturbo che vanno quindi rimossi affinché rimanga solo la parte di interesse.

Una ulteriore operazione che di solito segue la rimozione dei simboli è la trasformazione di tutte le lettere maiuscole in lettere minuscole. Anche in questo caso si tratta di un'operazione che comporta una piccola perdita di informazione, ma se non fosse eseguita l'algoritmo di stima tratterebbe due parole uguali come parole diverse in presenza di almeno una maiuscola, aumentando dunque il rumore nel testo.

Questa prima fase di rimozione della punteggiatura, dei simboli e delle lettere maiuscole è applicata senza grandi differenze in tutti i contesti in cui si opera con il *text mining*, dunque sono presenti molti strumenti per effettuarla. Nel contesto di questo lavoro, dove si è utilizzato il linguaggio R e il relativo software, era possibile scegliere tra una grande varietà di librerie in grado di implementare la normalizzazione dei testi. Si è scelto di utilizzare la libreria **TextWiller** (Finos *et al.*, 2016), poiché è stata concepita per le operazioni di *text mining* su testi in lingua italiana. Il codice R relativo al processo di normalizzazione è disponibile nell'Appendice A.

4.1.2 Rimozione delle *stopwords*

Successivamente si rimuovono dal testo una serie di parole, note come *stopwords*, che sono così comuni da portare un quantitativo di informazione trascurabile (a, tra, sì, qui...). Anche questa operazione riduce notevolmente il rumore nel testo, lasciando che abbiano più risalto le parole portatrici del senso delle frasi. Vengono comunemente trattate come *stopwords* gli articoli, le congiunzioni, i verbi ausiliari e le preposizioni. Inoltre, è comune che una lista di *stopwords* venga espansa a seconda del contesto in cui il *text mining* è applicato. Ad esempio: dal momento che la collezione di testi qui trattata è composta da diagnosi, la parola "diagnosi" non ha alcuna valenza nel testo pur essendo molto frequente. In molti casi (per la precisione il 32.20%) i patologi hanno scritto i testi come nell'esempio qui riportato:

```
DIAGNOSI: Nevo composto con displasia severa della componente giunzionale.
Lesione compresa nei limiti di exeresi (1,2).
```

Si noti come la parola "diagnosi" non sia importante (se non per esplicitare che il testo che segue è una diagnosi, cosa nota a priori) e dunque può essere aggiunta alla lista di *stopwords* senza che si perda informazione. A questa lista si aggiungono anche tutte le parole facenti riferimento alla vecchia formattazione dei testi (si

veda il Paragrafo 3.1.2) quali ad esempio quelle relative al carattere o alla codifica del testo (`arial`, `ansi`...).

Nella libreria **TextWiller** è contenuta una funzione per eliminare la lista di *stopwords* comunemente usate per la lingua italiana. A questa lista se ne è aggiunta un'altra creata appositamente per questa tesi contenente i termini superflui sopra citati (riportata nell'Appendice B). I termini di questa seconda lista sono stati eliminati dai testi con una semplice funzione. Anche in questo caso, il codice R relativo al processo è disponibile nell'Appendice A.

4.2 Lo *stemming* come base dell'approccio *bag-of-words*

Lo *stemming* è l'operazione successiva alla normalizzazione dei testi ed è il passo cruciale del *preprocessing* e uno dei principali in tutta la procedura di *text mining*. Prima di darne una descrizione, è necessario spiegare quale sia la finalità dell'applicazione dello *stemming*, in modo tale che sia chiara una delle idee chiave su cui si basa l'approccio al *text mining* qui utilizzato, ovvero il concetto di *bag-of-words*.

4.2.1 L'approccio *bag-of-words*

Si parla di *bag-of-words* quando non si considera più l'ordinamento delle parole in una frase ma si mantiene solo l'insieme dei termini scollegati tra loro. La perdita d'informazione che si ha smettendo di considerare le parole come ordinate è considerevole, sicuramente superiore alle altre perdite dovute alla normalizzazione dei testi (eliminazione della punteggiatura e delle *stopwords*). Ciononostante, anche questa operazione viene comunemente eseguita in tutti i lavori di *text mining* poiché permette di creare una lista di parole presenti in un testo, e queste parole possono essere utilizzate come variabili (o *features*) estratte dal testo. Come riportano Jurafsky e Martin in *Speech and Language Processing* (2008):

> *"Bag-of-words features are effective at capturing the general topic of the discourse"*

dunque, nonostante la perdita d'informazione, l'approccio *bag-of-words* rappresenta una buona strategia per estrarre i concetti chiave da un testo.

Per spiegare meglio il concetto di *bag-of-words* si procede con un esempio non inerente al contesto clinico (scelta dovuta al fatto che i testi clinici sono solitamente più lunghi e complessi):

> Si supponga di avere due testi:
>
> 1. `Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non interrotte di monti` [1]
> 2. `Ford si concentrò, cercando disperatamente di escogitare qualcosa, ma venne interrotto ancora una volta.` [2]
>
> Questi vengono poi normalizzati:
>
> 1. `ramo lago como volge mezzogiorno due catene interrotte monti`
> 2. `ford concentrò cercando disperatamente escogitare qualcosa venne interrotto ancora volta`
>
> Le due *bag-of-words* tratte da questi testi sono gli insiemi di termini:
>
> 1. `{ramo, lago, como, volge, mezzogiorno, due, catene, interrotte, monti}`
> 2. `{ford, concentrò, cercando, disperatamente, escogitare, qualcosa, venne, interrotto, ancora, volta}`

Nell'esempio riportato le parole "interrotte" e "interrotto" vengono trattate come due variabili differenti nonostante il loro significato sia lo stesso. Per evitare che il passaggio da frasi a *bag-of-words* generi un grande quantitativo di variabili tra cui molte simili e con uguale significato si ricorre proprio al processo di *stem-*

[1] Alessandro Manzoni, *I promessi sposi*, capitolo I, p. 9
[2] Douglas Adams, *Guida galattica per gli autostoppisti*, capitolo VII, p. 73

ming. Per una trattazione più rigorosa e matematica dell'approccio *bag-of-words* si rimanda a Zhang *et al.*, 2010.

4.2.2 Lo *stemming*

Lo *stemming* è il processo di riduzione di una parola alla sua radice, detta stilema o tema. Come radice non s'intende l'origine semantica della parola, bensì la troncatura della stessa: ad esempio le parole "tumore", "tumori", "tumorale" ecc. possono essere tutte associate allo stesso stilema "tumor". In questo modo il numero di variabili estratte da un testo si riduce e allo stesso tempo più testi avranno più variabili in comune.

L'operazione di *stemming* è completamente automatizzata e basata su degli algoritmi detti *stemmers*. Il primo *stemmer* è stato proposto da Lovins (1968), ma quello che a oggi risulta essere l'algoritmo più utilizzato è stato ideato da Porter (1980). Il primo algoritmo si basa semplicemente su una lista di suffissi e su una serie di regole per effettuare la troncatura delle parole. Il secondo considera invece i suffissi come composti da sotto-suffissi più piccoli e si dota di regole più complesse per la troncatura, applicate ai sotto suffissi, presentando in questo modo una maggiore efficienza (Jivani, 2011).

L'algoritmo di *stemming* di Porter è stato sviluppato da lui stesso nel progetto *Snowball* (`www.snowballstem.org`), una piattaforma il cui scopo è implementare l'algoritmo per più lingue possibili. Lo *stemming* differisce infatti a seconda della lingua in cui sono stati scritti i testi e grazie a *Snowball* è disponibile un algoritmo di *stemming* per la lingua italiana. Tornando all'esempio precedente:

> Si applica lo *stemming* con algoritmo di Porter alle frasi dopo la loro normalizzazione:
>
> 1. `ram lag com volg mezzogiorn due caten interrott mont`
> 2. `ford concentr cerc disperat escogit qualcos venn interrott ancor volt`
>
> Le due *bag-of-words* diventano:

1. `{ram, lag, com, volg, mezzogiorn, due, caten, interrott, mont}`
2. `{ford, concentr, cerc, disperat, escogit, qualcos, venn, interrott, ancor, volt}`

Si noti che ora lo stilema "interrott" faccia parte di entrambi gli insiemi *bag-of-words*. Può considerarsi dunque una variabile estratta da entrambi i testi (o una *feature* comune dei testi).

Si è applicato sull'insieme di testi di anatomia patologica in esame lo *stemming* di Porter implementato per la lingua italiana nella libreria R **tm** (Feinerer *et al.*, 2008). Il codice R è, anche in questo caso, all'interno dell'Appendice A. Tutte le parole sono state ridotte agli stilemi corrispondenti e i testi sono ora rappresentati dalle rispettive *bag-of-words*. Si riporta una delle diagnosi come esempio:

```
Melanoma: nodulare. PARAMETRI PROGNOSTICI: Fase di crescita: verticale. Livello
di Clark: V. Indice di Breslow: mm 9,8. Ulcerazione: presente. Indice mitotico:
>10 mitosi x mm. Infiltrato linfocitario: lieve, non brisk. Regressione: assente.
Invasione vascolare: non evidente. Tipo cellulare: prevalentemente a fisionomia
epiteliomorfa, con aree a morfologia fusata e a cellule chiare. Nevo associato:
assente. Margini di resezione chirurgica: esenti. Produzione di pigmento:
moderata.
```

Il testo viene normalizzato, vengono rimosse le *stopwords* e i simboli:

```
melanoma nodulare parametri prognostici fase crescita verticale livello clark
indice breslow mm ulcerazione presente indice mitotico 10 mitosi mm cup6 infiltrato
linfocitario lieve non brisk regressione assente invasione vascolare non evidente
tipo cellulare prevalentemente fisionomia epiteliomorfa aree morfologia fusata
cellule chiare nevo associato assente margini resezione chirurgica esenti
produzione pigmento moderata
```

Viene poi eseguito lo *stemming*:

```
melanoma nodular parametri prognostici fase crescita vertical livello clark
indic breslow mm ulcerazion present indic mitotico 10 mitosi mm cup6 infiltrato
linfocitario liev non brisk regression assent invasion vascolar non evident
tipo cellular prevalentement fisionomia epiteliom are morfologia fusata cellul
chiar nevo associato assent margini resezion chirurgica esenti produzion pigmento
moderata
```

Si ricava infine la *bag-of-words*:

```
{melanoma, nodular, parametri, prognostici, fase, crescita, vertical, livello,
clark, indic, breslow, mm, ulcerazion, present, indic, mitotico, 10, mitosi,
mm, cup6, infiltrato, linfocitario, liev, non, brisk, regression, assent,
invasion, vascolar, evident, tipo, cellular, prevalentement, fisionomia, epiteliom,
are, morfologia, fusata, cellul, chiar, nevo, associato, assent, margini,
resezion, chirurgica, esenti, produzion, pigmento, moderata}
```

Nella *bag-of-words* ogni stilema rappresenta un'informazione estratta dal testo e verrà trattato come una variabile statistica. Alcune di queste variabili saranno condivise da due o più testi, altre saranno presenti in un solo testo, altre ancora si ripeteranno in uno o più testi un certo numero di volte. È proprio individuando dei *pattern* nella presenza e nella frequenza degli stilemi che i modelli statistici di *text mining* classificano i testi.

4.2.3 I limiti dello *stemming*

Nonostante lo *stemming* sia una procedura che nel tempo ha acquistato sempre maggior precisione, rimane un passaggio limitante nel contesto del *text mining*. Le complicazioni legate a questa operazione sono evidenti, in particolare se ne riscontrano tre:

- Alcune parole si riconducono allo stesso stilema ma hanno significati diversi, ad esempio "foglio" e "foglia" hanno lo stesso stilema "fogl" e quindi sarebbero considerati come una stessa variabile nonostante il significato diverso.

- Alcune parole hanno lo stesso significato ma non la stessa radice, come per esempio "assente" ed "esenti", nell'esempio precedente. Il significato delle due parole è lo stesso ma gli stilemi a cui vengono ridotte sono differenti.

- Le regole di *stemming* cambiano a seconda della lingua utilizzata, quindi il fatto che in un testo siano presenti più lingue (cosa comune al giorno d'oggi) può rappresentare un problema.

Il *corpus* delle diagnosi di anatomia patologica non è esente dai difetti dello *stemming* (quantomeno dai primi due) e questo rappresenta un problema per l'efficienza generale della procedura di *text mining*.

4.3 Creazione della *document-term matrix*

Giunti a questo punto del *preprocessing* si ha una collezione di *bag-of-words* contenenti gli stilemi che rappresentano l'informazione estratta dai testi. Si procede costruendo una *document-term matrix*, ovvero una matrice che per ogni riga ha un testo (o, per comodità, l'identificativo di un testo), per ogni colonna ha uno stilema e per ogni cella ha un indicatore della presenza dello stilema in colonna nel testo in riga. Questo indicatore di presenza può prendere una delle seguenti tre forme (Miner *et al.*, 2012):

- Indicatore dicotomico:
  In ogni cella è presente un indicatore dicotomico (ad esempio: VERO o FALSO, 0 o 1...) che indica se lo stilema è presente nel testo. Questa rappresentazione è semplice ma efficace, solitamente viene scelta nel caso in cui la frequenza dei termini nel testo non sia di particolare interesse.

- Conteggio:
  In ogni cella è contenuto un numero intero che indica il conteggio del numero

di volte che lo stilema si trova nel testo. Questa rappresentazione è indicata quando si vuole tenere conto della frequenza con cui le parole si trovano nei testi.

- Conteggio pesato:
  In ogni cella è contenuto un numero naturale che rappresenta l'importanza (peso) di stilema all'interno del *corpus* dei testi: esistono diversi pesi utilizzabili per questo scopo, ma la scelta più comune ricade sul peso *tf-idf*. Solitamente il conteggio pesato è la migliore opzione per una *document-term matrix*.

È stata costruita una *document-term matrix* con i testi a disposizione (una volta normalizzati e sottoposti a *stemming*) scegliendo la terza opzione, ovvero il conteggio pesato. Il modo in cui il peso *tf-idf* è stato calcolato sarà esposto nel Paragrafo 4.3.2. La matrice è stata creata con una funzione della libreria R **tm** e arriva a contare 2574 righe (il numero dei testi) e 2631 colonne (il numero degli stilemi).

4.3.1 Utilizzo della matrice e riduzione della dimensionalità

La *document-term matrix* è utilizzabile come una comune matrice di regressione: ogni colonna rappresenta una variabile esplicativa estratta dal testo mentre il *gold standard* rappresenta le variabili risposta (gli *outcome*). Si hanno dunque tutte le componenti per la stima di un modello di classificazione supervisionata che sia in grado di prevedere le classi delle variabili risposta basandosi sulle informazioni estratte dai testi.

Solitamente una *document-term matrix* è una matrice sparsa (nel caso in esame si ha il 99.18% di elementi pari a 0) le cui dimensioni sono notevoli anche per piccole collezioni di testi; è dunque una scelta comune ridurre la dimensionalità della matrice eliminando tutti gli stilemi che compaiono un ridotto numero di volte. Solitamente, questa operazione riduce le dimensioni della matrice senza perdere una quantità significativa di informazione (Feinerer, 2018).

Questa operazione di riduzione della dimensionalità è stata eseguita anche nel caso della *document-term matrix* in esame sempre usando una funzione della li-

breria **tm**. Si è scelto di rimuovere dalla *document-term matrix* gli stilemi la cui sparsità fosse al 99%, ovvero erano presenti solo nell'1% dei testi. Dopo questa operazione le colonne della matrice sono diventate 316 (con il 94.02% di elementi pari a 0), dunque il numero di stilemi è stato ridotto drasticamente e la sparsità è leggermente diminuita. Questa nuova matrice ha delle dimensioni più consone a essere usata come matrice di regressione per un modello di classificazione. Ancora una volta ci si trova di fronte a una riduzione dell'informazione estratta dai testi per far fronte alla necessità di diminuire il rumore presente negli stessi. Nella matrice sottostante viene riportato un campione della *document-term matrix*, si noti che i testi delle diagnosi per comodità sono sostituiti da un indice:

Diagnosi	**Stilema**					
	exeres	infiltr	lesion	margin	melanom	nev
1429	1.85	3.91	0.00	2.42	0.00	0.00
1796	3.70	0.00	1.25	2.42	1.80	2.38
1836	1.85	1.95	3.75	2.42	3.60	3.59
2341	0.00	11.74	0.00	19.43	0.00	0.00
249	0.00	7.82	0.00	4.85	0.00	0.00
2503	0.00	0.00	0.00	0.00	0.00	0.00
2555	0.00	0.00	0.00	0.00	0.00	0.00
275	0.00	7.82	0.00	4.85	0.00	0.00
683	1.85	1.95	2.50	4.82	0.00	1.79

Una *document-term matrix* può anche essere usata per valutare le associazioni tra gli stilemi. Se si immaginano le colonne della matrice come vettori a sé stanti, si può calcolare la correlazione tra due di essi per valutare il grado di associazione fra i termini. Solitamente, in questo contesto una correlazione maggiore di 0.5 viene considerata ragionevolmente alta per ritenere due termini associati (Feinerer *et al.*, 2008). Come esempio nella Tabella 4.1 vengono riportate delle associazioni fra alcuni stilemi.

Stilema	Stilema associato	Correlazione
ulcer*azione*	cresc*ita*	0.83
	spessor*e*	0.81
	livell*o*	0.80
	superficial*e*	0.66
metastas*i*	linfonod*o*	0.85
	esent*e*	0.70
	esamin*ato*	0.58
	sentinell*a*	0.54
nev*o*	composto	0.70
	giunzional*e*	0.51
	displasi*a*	0.50

Tabella 4.1: Alcune associazione tra stilemi - la parte aggiunta in corsivo dopo uno stilema è uno dei possibili termini che ne hanno dato origine

4.3.2 I pesi *tf-idf*

I pesi *tf-idf* rappresentano l'importanza di uno stilema all'interno di un testo appartenente a un *corpus* di testi (Ramos, 2003). La sigla *tf-idf* abbrevia l'espressione *term frequency - inverse document frequency* e il suo calcolo si basa su una funzione di due differenti misure:

1. La frequenza di uno stilema in un documento (*term frequency*) pari al numero di volte in cui lo stilema appare in un documento.

2. La frequenza del documento (*document frequency*) ossia il numero di documenti nel *corpus* in cui appare lo stesso stilema. Di questo valore si prende l'inverso, che dunque sarà basso quando un termine è comune a molti documenti e alto quando un termine è presente in pochi documenti.

L'indice *tf-idf* per la parola x nel documento y è:

$$\textit{tf-idf}_{x,y} = (N_{x,y}/N_{\cdot,y}) \cdot log(D/D_x)$$

Dove:

- $N_{x,y}$ è il numero di volte in cui lo stilema x appare nel testo D_y;
- $N_{\cdot,y}$ è il numero di stilemi nel testo D_y;
- $N_{x,y}/N_{\cdot,y}$ è la *term frequency*;
- D è il numero di testi nel *corpus*;
- D_x è il numero di testi in cui lo stilema x appare almeno una volta;
- D/D_x è la *inverse document frequency*.

L'idea dietro il peso *tf-idf* è che un termine con una alta frequenza all'interno di un testo dovrà ricevere una maggiore importanza a meno che non sia presente in un gran numero di testi. A quel punto si assume che quel termine sia estremamente comune fra i testi e quindi non sia in grado di discriminare un testo dall'altro.

Nel dataset in esame, il calcolo dei pesi *tf-idf* è stato eseguito assieme alla creazione della *document-term matrix*. Si può anche usare la frequenza degli stilemi e il peso *tf-idf* per avere una prima idea di quali termini possano essere più influenti nel processo di *text mining*. Si deve comunque tener presente che un peso *tf-idf* non corrisponde a uno stilema, bensì a uno stilema all'interno di un testo. In altre parole: non è possibile ordinare gli stilemi per *tf-idf* perché il peso assegnato allo stesso stilema può cambiare a seconda del testo in cui esso appare.

Dalle Tabelle 4.2 e 4.3 emergono delle prime conclusioni, sicuramente in linea con le aspettative:

- Termini come "lesione" (della cute), "nevo" (comunemente detto neo), "melanoma" o "exeresi" (operazione di asportazione del tessuto cellulare) sono molto comuni fra i testi e la loro importanza ci si aspetta che sia contenuta. Il motivo è perché la presenza di lesioni o nei è una caratteristica comuni a molti casi di melanoma cutaneo, quindi non rilevante nel distinguere i melanomi gravi da quelli meno gravi. Lo stesso termine "melanoma" è ovvio

Stilema	***Term-frequency***
lesion*e*	1081
nev*o*	741
melanom*a*	739
exeres*i*	714
infiltr*azione*	663
tip*o*	610
limit*ato*	581
malign*o*	575
ulcer*azione*	560

Tabella 4.2: Gli stilemi più frequenti nell'insieme dei testi - la parte aggiunta in corsivo dopo uno stilema è uno dei possibili termini che ne hanno dato origine

che appaia di frequente in questo genere di testi. Le stesse parole hanno, di conseguenza, dei pesi *tf-idf* che generalmente sono più bassi proprio a causa della loro alta frequenza.

- Al contrario dei precedenti, termini come "adenoma" (caratteristico tumore epiteliale benigno) o "iperplasia" (crescita di un tessuto) sono associati a *tf-idf* più alti, dunque non sono molto comuni tra i testi e sono in grado di discriminare più efficacemente i testi stessi.

Queste prime conclusioni sono puramente descrittive e si è scelto di non darvi troppo peso nelle conclusioni generali. Il motivo è dovuto al fatto che valutare i termini solo in base alla loro frequenza (o una funzione di essa) significa ignorare le relazioni che intercorrono fra i termini stessi. Sono proprio queste relazioni a essere determinanti nel processo di *text mining* e il modo migliore perché vengano colte è attraverso un modello statistico. Nei capitoli seguenti si affronterà la stima di tali modelli e sarà possibile trarre conclusioni più solide sull'importanza che i singoli termini hanno nell'associare il testo di una diagnosi a una particolare caratteristica del tumore.

Diagnosi	**Stilema**	***tf-idf***	**Diagnosi**	**Stilema**	***tf-idf***
1401	reatt*ive*	4.18	1836	lesion*e*	0.010
727	adenom*a*	4.00	1150	lesion*e*	0.013
38	epitel*iale*	3.56	683	lesion*e*	0.013
193	iperplas*ia*	3.25	1796	lesion*e*	0.014
662	iperplas*ia*	3.25	778	lesion*e*	0.014
1972	posit*ivo*	3.17	1836	nev*o*	0.015
2201	posit*ivo*	3.17	1836	melanom*a*	0.015
924	nodul*o*	2.78	801	lesion*e*	0.015
158	focal*e*	2.72	1464	lesion*e*	0.016
2408	localizz*ato*	2.66	1584	lesion*e*	0.016

Tabella 4.3: Gli stilemi all'interno dei testi il cui *tf-idf* è molto alto o molto basso - la parte aggiunta in corsivo dopo uno stilema è uno dei possibili termini che ne hanno dato origine

4.4 Aggiunta dei bigrammi alla *document-term matrix*

Una *bag-of-words* di stilemi non tiene conto dell'ordinamento delle parole all'interno dei testi. Questo può essere una limitazione: basti pensare alla differenza fra il considerare i termini "assenza" e "metastasi" come scollegati (cosa che avviene in una *bag-of-words*) o come collegati nell'espressione "assenza [di] metastasi". I bigrammi permettono di cogliere quelle espressioni composte da coppie di parole.

L'uso dei bigrammi è analogo a quello degli stilemi: si crea una matrice che ha per righe i testi e per colonne i bigrammi, gli elementi della matrice saranno pari a 0 nel caso in cui il bigramma sia assente dal testo e 1 nel caso in cui il bigramma sia presente nel testo (per semplicità si è scelto di non utilizzare i pesi *tf-idf* per i bigrammi). Una differenza con gli stilemi è rappresentata dal fatto che i bigrammi vengono creati senza che le parole siano soggette alla procedura di *stemming* in modo tale sia considerata la coppia ordinata di parole intere e non ridotte alla radice. Sempre al fine di ridurre la sparsità della *document-term matrix* si è deciso

Bigramma	**Frequenza**
fase crescita	463
limiti exeresi	462
compresa limiti	420
infiltrato linfocitario	398
lesione compresa	363
crescita verticale	303
completamente escissa	301
estensione superficiale	296
non ulcerato	241
invasione angiolinfatica	215

Tabella 4.4: I bigrammi più comuni e la loro frequenza

di tenere solo i bigrammi che compaiono almeno nell'1% dei testi: il loro numero ammonta a 379. Nella Tabella 4.4 sono elencati i bigrammi più frequenti all'interno dei testi delle diagnosi a disposizione.

La matrice originata dai bigrammi si giustappone a quella originata dagli stilemi andando a formare una matrice composta da 2574 righe (i testi) e 695 colonne (le variabili estratte dal testo sotto forma di stilemi o bigrammi). Questa matrice sarà la matrice di regressione per i modelli statistici di classificazione stimati nel capitolo successivo.

Ovviamente il concetto di bigramma può essere esteso ai trigrammi, quadrigrammi ecc. in modo che venga sempre più considerato l'ordinamento delle parole nella frase. Come però fanno notare Iacus *et al.* (2014):

> "In generale, considerare stilemi con tre o più parole non fornisce particolare aggiunta di informazione e non aumenta la qualità della classificazione"

per questo motivo si è scelto di limitare la procedura ai bigrammi.

4.5 Riepilogo del capitolo

Il *preprocessing* è l'insieme delle operazioni che portano a estrarre da un testo grezzo delle variabili che veicolano l'informazione contenuta in esso. Inizialmente tutti i testi sono sottoposti a una procedura di normalizzazione: vengono rimosse la punteggiatura, i simboli e le lettere maiuscole. Successivamente vengono rimosse le *stopwords*, ovvero quelle parole così comuni da non portare nessun tipo di informazione. Infine, viene effettuato lo *stemming* (riduzione alla radice, detta "stilema") delle parole e viene costruita una matrice che ha come righe i testi, come colonne gli stilemi e come elementi i pesi *tf-idf* degli stilemi nei testi. Questa matrice, detta *document-term matrix* può essere utilizzata come matrice di regressione nei modelli di classificazione dei testi ma conta ben 2631 colonne (contro 2574 righe) ed è estremamente sparsa, condizioni non favorevoli per la buona riuscita della stima dei modelli. Di conseguenza si è ridotta la dimensionalità eliminando gli stilemi la cui frequenza totale è bassa, in questo modo si ottengono 316 colonne e una sparsità inferiore (anche se comunque alta).

Utilizzando i pesi *tf-idf* contenuti nella *document-term matrix* è possibile trarre alcune conclusioni esplorative sull'importanza che certi termini hanno nel classificare i testi delle diagnosi (e quindi nel determinare le caratteristiche del tumore). In particolare, si nota che alcuni stilemi sono molto frequenti nei testi e quindi le caratteristiche a essi associate sono comuni a molti tumori (per esempio la presenza di lesioni o di nei). Probabilmente questi stilemi non aiuteranno molto a stabilire le caratteristiche del tumore proprio perché comuni a molti di essi. Altri stilemi, al contrario, sono presenti un cospicuo numero di volte ma in pochi testi, quindi ci sono buone possibilità che siano associati a caratteristiche peculiari del tumore in grado di stabilirne la gravità (ad esempio "adenoma" o "iperplasia").

Si è scelto infine di aggiungere alla *document-term matrix* i bigrammi, ovvero le coppie di parole più frequenti nei testi. Anche sei in maniera ridotta, i bigrammi sono d'aiuto per considerare l'ordinamento delle parole nel testo. Sono stati selezionati i 379 bigrammi più frequenti e si è creata una matrice che ha come righe i testi, come colonne i bigrammi e come elementi la presenza o l'assenza (codifi-

cata come 0 o 1) del bigramma nel testo. Questa matrice si giustappone a quella generata dagli stilemi andando a formare una matrice di 2574 righe (il numero dei testi) e 695 colonne (il numero degli stilemi e dei bigrammi) che sarà usata come matrice di regressione per i modelli di classificazione stimati nel capitolo seguente.

Capitolo 5

Stima dei modelli di classificazione

Con la creazione della *document-term matrix* e il suo allargamento ottenuto aggiungendo i bigrammi si ottiene una matrice di regressione adatta al problema di classificazione, che rappresenta il cuore della procedura di *text mining*. Tipicamente i problemi di classificazione collegati al *text mining* si affrontano con modelli statistici tipici del *data mining*. Questo è dovuto al fatto che le matrici di regressione hanno grandezze notevoli, dunque si necessita di modelli in grado di gestire alte dimensionalità anche a costo di sacrificare, del tutto o in parte, la loro interpretabilità.

Questo capitolo è organizzato nel modo seguente: dopo un paragrafo con alcune premesse, ne sarà dedicato uno per ogni modello stimato. Ogni modello verrà descritto e verranno mostrati i risultati prodotti per ognuna delle quattro classificazioni legate ai quattro *outcome* esposti nel Paragrafo 3.2.2. I modelli esposti sono largamente utilizzati nel campo del *data mining*, dunque si è scelto di descriverli solo brevemente: una loro descrizione accurata sarebbe molto impegnativa e va ben oltre gli obiettivi di questa tesi. Per una loro trattazione più approfondita si rimanda a Hastie *et al.* (2013) e ad Azzalini e Scarpa (2012): da questi due testi sono state tratte gran parte delle informazioni esposte in questo capitolo. Infine, i modelli saranno comparati e verrà affrontata una discussione sulla loro efficienza.

5.1 I modelli statistici più utilizzati nel *text mining* clinico e la valutazione del loro errore

Come riportato da Dalianis (2018), i principali modelli utilizzati nel campo del *text mining* clinico sono i modelli a *support vector machines*. Nei lavori più simili a questa tesi, ossia quelli dove si estraggono dai testi delle diagnosi lo stadio di un certo tipo di tumore usando solamente dei modelli statistici (ad esempio: McCowan *et al.*, 2007; Nguyen *et al.*, 2007; Martinez *et al.*, 2013), si fa uso proprio dei modelli a *support vector machines* ottenendo dei buoni risultati. Per questo motivo, il primo modello a cui si farà ricorso per la classificazione dei testi sarà quello a *support vector machines*, la cui efficienza e velocità di stima rappresentano un buon punto di partenza.

Verranno poi stimati altri modelli, in particolare alberi di classificazione, combinazioni di alberi di classificazione e reti neurali. Non sono stati trovati in letteratura casi di utilizzo di classificatori ad albero (o combinazioni di questi), dunque questa tesi può considerarsi un primo approccio al *text mining* clinico tramite l'uso di questa classe di modelli.

5.1.1 Valutazione dell'errore dei modelli

Per determinare la performance dei modelli si è utilizzata la ben nota tabella di errata classificazione, anche detta "matrice di confusione" (Stehman, 1997): questa tabella è comunemente utilizzata nella valutazione dei modelli di classificazione e ha come righe la classificazione effettuata dal modello, come colonne la vera classificazione (basata sul *gold standard*) e nelle celle il numero di testi ripartiti secondo la corrispondente riga e colonna.

Sommando gli elementi al di fuori della diagonale di questa matrice e dividendo per la numerosità totale si ottiene la proporzione di testi classificati incorrettamente, o "tasso di errata classificazione". Questa misura (detta anche "errore di classificazione") è utile per comparare più modelli fra loro.

Alcuni autori affiancano (o sostituiscono) l'errore di classificazione con la F-*measure*. La F-*measure* è una statistica alternativa all'errore di classificazione e si ottiene con una media armonica tra la sensibilità e la precisione della classificazione (Powers, 2007). L'interpretazione di questa misura è però più difficile rispetto al tasso di errata classificazione, in particolare per il personale medico. Per questo motivo si è scelto di utilizzare l'errore di classificazione, il quale ha un'interpretazione semplice e intuitiva.

Per evitare il sovradattamento del modello ai dati, il calcolo dell'errore di classificazione viene solitamente eseguito su una porzione di dataset che non è stata utilizzata nella stima del modello. Questa strategia porta a una divisione del dataset in due gruppi casualmente estratti dalla casistica disponibile: il primo, detto "insieme di stima", è composto da 1500 unità (circa il 60% del totale) e sarà usato per la stima del modello, mentre il secondo, detto "insieme di verifica", sarà usato esclusivamente per il calcolo dell'errore di classificazione.

5.2 Classificazione con *support vector machines*

I modelli a *support vector machines* (SVM) sono una classe di modelli sviluppati nel campo del *machine learning* e sono utilizzati principalmente per risolvere problemi di classificazione supervisionata. Inizialmente introdotte da Cortes e Vapnik (1995), sono stati affrontati da innumerevoli autori e sono tuttora uno dei temi di ricerca più attuali nell'ambito del *data mining* e del *machine learning*.

5.2.1 *Support vector machines*: descrizione del modello

Per una miglior comprensione dei modelli SVM verrà spiegato il loro funzionamento nel caso della classificazione di una variabile dicotomica, la generalizzazione al caso di classi multiple è reperibile nella letteratura citata in precedenza.

Si assuma di avere n coppie di dati (y_i, x_i) con $i = 1 \dots n$ all'interno dell'insieme di stima: x_i denota le variabili esplicative e y_i le variabili dicotomiche di riposta, che per comodità assumono i valori -1 e 1 a seconda della modalità assunta. Le osservazioni appartenenti alle due classi possono essere separate da un

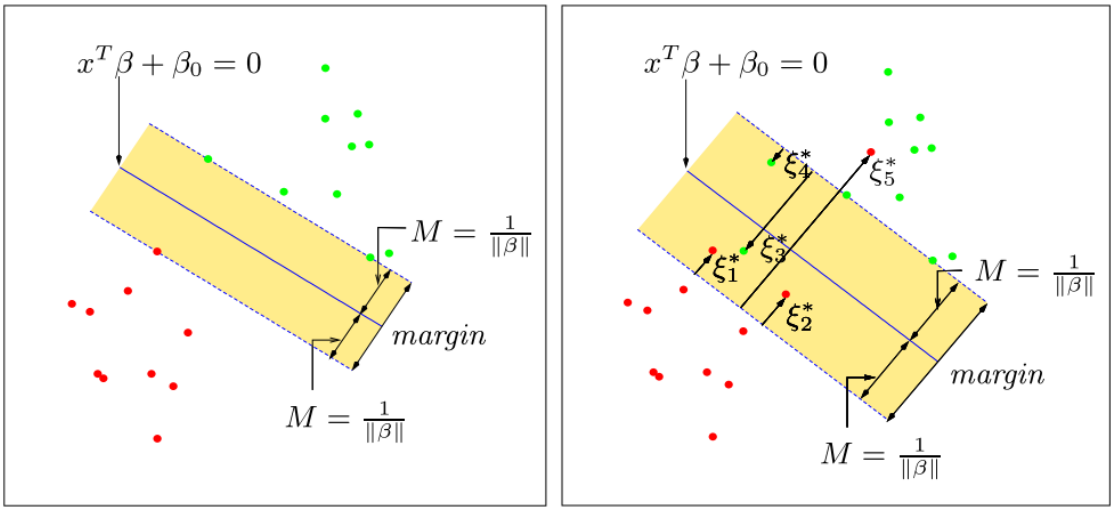

Figura 5.1: Rappresentazione geometrica in due dimensioni di una SVM: a sinistra il caso senza classi sovrapposte, a destra il caso con classi sovrapposte (tratto da Hastie *et al.*, 2013)

piano di equazione $\{f(x) = x^T\beta + \beta_0 = 0, \|\beta\| = 1\}$ che divide lo spazio delle variabili esplicative in due sotto-spazi. Questo piano è ricavato dal problema di ottimizzazione vincolata:

$$\max_{\beta,\beta_0,\|\beta\|=1} M \quad \text{s.v.} \quad M \leq y_i(x_i^T\beta + \beta_0), \ \forall i. \tag{5.1}$$

Si dimostra che il problema precedente è equivalente alla minimizzazione:

$$\max_{\beta,\beta_0} \|\beta\| \quad \text{s.v.} \quad y_i(x_i^T\beta + \beta_0) \geq 1, \ \forall i. \tag{5.2}$$

Nella Figura 5.1 si ha una rappresentazione in due dimensioni di un modello a SVM: i punti rossi e verdi sono i dati appartenenti alle due classi, la linea rappresenta il piano che li divide (nel primo caso in maniera netta, nel secondo si ha invece una sovrapposizione tra le due classi), mentre la banda gialla corrisponde al massimo margine individuato tra il piano e l'osservazione più vicina. Risolvere questa minimizzazione porta a individuare quel vettore β che massimizza l'ampiezza della banda gialla tra i due gruppi di osservazioni.

Il caso precedentemente esposto nelle Formule 5.1 e 5.2 assume però che le classi non siano sovrapposte, cosa che invece avviene regolarmente nei problemi di classificazione. In presenza di sovrapposizione tra classi (parte destra della Figura 5.1), non esiste un piano che ne permette la separazione netta, dunque si tollera che una o più osservazioni appartengano alla parte di sotto-spazio errata.

Si definisce l'insieme di variabili ξ_i con $i = 1 \dots n$ (nella Figura 5.1, $\xi_i^* = \xi_i M$). Se una osservazione appartiene al sotto-spazio corretto la variabile ξ_i associata vale 0, se invece un'osservazione appartiene al sotto-spazio errato la corrispondente variabile ne misura la distanza dal margine del gruppo di appartenenza (limite della banda gialla). Si modifica quindi il problema di minimizzazione vincolata tenendo conto delle variabili ξ_i nel seguente modo:

$$\max_{\beta,\beta_0,\|\beta\|=1} M \quad \text{s.v.} \quad \begin{cases} M(1-\xi_i) \leq y_i(x_i^T\beta + \beta_0) \\ \xi_i \geq 0 \qquad\qquad\qquad\qquad \forall i. \\ \sum_{i=1}^n \xi_i \leq \gamma \end{cases} \tag{5.3}$$

Attraverso il vincolo $\sum_{i=1}^n \xi_i \leq \gamma$ si sta imponendo un limite alla quantità totale di osservazioni che giacciono sul lato sbagliato dello spazio (quantità pesata per la distanza tra le osservazioni e il margine del gruppo). Così come nel caso dell'assenza di sovrapposizione, anche questo problema di minimizzazione può essere riscritto nella seguente forma:

$$\max_{\beta,\beta_0} \|\beta\| \quad \text{s.v.} \quad \begin{cases} y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i \\ \xi_i \geq 0 \qquad\qquad\qquad\qquad \forall i, \\ \sum_{i=1}^n \xi_i \leq \gamma \end{cases} \tag{5.4}$$

o, equivalentemente,

$$\max_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^n \xi_i \quad \text{s.v.} \quad \begin{cases} y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i \qquad \forall i. \\ \xi_i \geq 0 \end{cases} \tag{5.5}$$

Qui la costante C assume il ruolo di parametro di regolazione del modello.

Questa minimizzazione, risolta con il metodo dei moltiplicatori di Lagrange, ha come soluzione:

$$\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i y_i x_i, \tag{5.6}$$

dove α_i è diverso da 0 solo per quelle osservazioni, dette *support vectors*, che rispettano la condizione $y_i(x_i^T\beta + \beta_0) = 1 - \xi_i$, quindi la soluzione dipende solo da quest'ultime.

È comune ricorrere a una trasformazione delle variabili esplicative del tipo $h(x_i) = [h_1(x_i) \dots h_q(x_i)]^T$ al fine di modificare il piano separatore rendendolo non-lineare, e quindi più adattabile alla classificazione. Inserendo la Formula 5.6 nell'equazione del piano e tenendo conto della trasformazione delle x_i appena proposta, si ha la stima finale dell'equazione del piano:

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i y_i \langle h(x), h(x_i) \rangle, \tag{5.7}$$

dove $\langle \cdot, \cdot \rangle$ denota il prodotto interno.

La funzione $h(x)$ viene definita a seconda della cosiddetta "funzione *kernel*" $K(x, x') = \langle h(x), h(x') \rangle$, ossia una funzione che calcola il prodotto interno nello spazio delle variabili trasformate. La funzione *kernel* che sarà utilizzata in seguito è una delle più comuni: quella radiale $\exp(-d\|x - x'\|^2)$. La costante d va fissata a priori e costituisce anch'essa una parametro di regolazione del modello, assieme alla costante C. Entrambi i parametri saranno scelti minimizzando il tasso di errata classificazione calcolato sull'insieme di verifica.

Giunti a questo punto si ha la stima finale del piano e la classificazione di una certa osservazione x^* avviene secondo la regola:

$$\text{sign}(\hat{f}(x^*)) = \text{sign}\left(\hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha} y_i K(x^*, x_i)\right). \tag{5.8}$$

che sfrutta la codifica -1 e 1 delle classi della variabile risposta imposta in precedenza.

5.2.2 *Support vector machines*: stima del modello e risultato della classificazione

Il modello a SVM è stato stimato in R tramite la libreria **e1071** (Meyer *at al.*, 2018). Per far fronte allo sbilanciamento delle variabili di *outcome*, le righe della matrice di regressione sono state pesate con un peso inversamente proporzionale alla frequenza della classe a cui appartengono (Tan, 2005).

La scelta dei due parametri di regolazione del modello è stata fatta usando una "griglia" di valori, ossia stimando vari modelli a SVM facendo variare entrambi i parametri in un intervallo di valori plausibili e scegliendo quelli che garantivano al modello il minor errore di classificazione calcolato sull'insieme di verifica. Gli errori di classificazione del modello con i parametri ottimali e i suoi tempi di calcolo sono riportati per ogni *outcome* nella Tabella 5.1.

Outcome	TNM	T	N	M
Errore di classificazione	26.3%	18.7%	14.4%	6.5%
Tempo di calcolo (sec.)	8.9	7.9	6.1	4.0

Tabella 5.1: Errori di classificazione e tempi di calcolo per i modelli a SVM, TNM indica la stadiazione del tumore, T indica la dimensione del tumore primitivo, N indica il coinvolgimento dei linfonodi e M indica la presenza di metastasi

5.3 Classificazione con modello ad albero

I modelli ad albero sono una classe di modelli concettualmente più semplici rispetto ai modelli a SVM, ma con delle performance tradizionalmente più basse. Sono stati usati in questo contesto per due motivi:

1. la loro interpretazione è immediata, permettono infatti di costruire un albero binario che rappresenti visivamente la classificazione;

2. più alberi possono essere combinati per dare origine a modelli (detti anche "foreste") più complessi ed efficienti, dunque ci si serve di questo modello per introdurre i successivi più avanzati.

Quelli ad albero sono una classe di modelli trattata in modi diversi a seconda degli autori, di seguito si farà riferimento alla notazione introdotta da Breiman *et al.* (1984).

5.3.1 Modelli ad albero: descrizione del modello

I modelli ad albero si basano su progressive suddivisioni binarie (o *split*) dello spazio delle variabili esplicative: a ognuna di queste suddivisioni si associa una delle modalità della variabile risposta.

Si consideri per semplicità il caso in cui si hanno solo due variabili esplicative X_1 e X_2, usate per predire la variabile risposta Y: le due variabili esplicative generano uno spazio dentro il quale la modalità prevista per la variabile Y non è altro che la modalità più frequente. Su una delle due variabili viene poi scelta una soglia (detta "nodo" o "*split-point*"), questa soglia divide in due parti lo spazio e, in ognuno di questi sotto-spazi, la modalità della variabile Y verrà sempre predetta come la modalità più frequente all'interno del sotto-spazio. Si ripete poi il procedimento prendendo di nuovo una delle due variabili, scegliendo uno *split-point* e dividendo in due parti uno dei due sotto-spazi ottenuti dalla precedente divisione. Supponendo di ripetere il procedimento un certo numero di volte, si otterrà una suddivisione dello spazio delle variabili esplicative in aree rettangolari e, all'interno di ogni area, la variabile Y sarà predetta con la modalità più frequente all'interno dell'area stessa (a sinistra nella Figura 5.2).

La variabile scelta per lo *split* e lo *split-point* vengono determinati di volta in volta in modo da ottenere la miglior suddivisione possibile dello spazio. L'algoritmo comunemente utilizzato prevede che a ogni passo si considerino tutte le combinazioni di variabili e di *split-point* possibili, scegliendo quella che porta a una partizione che minimizza un indice di entropia. In altre parole, la divisione viene fatta cercando di minimizzare l'entropia all'interno dei due sotto-spazi che

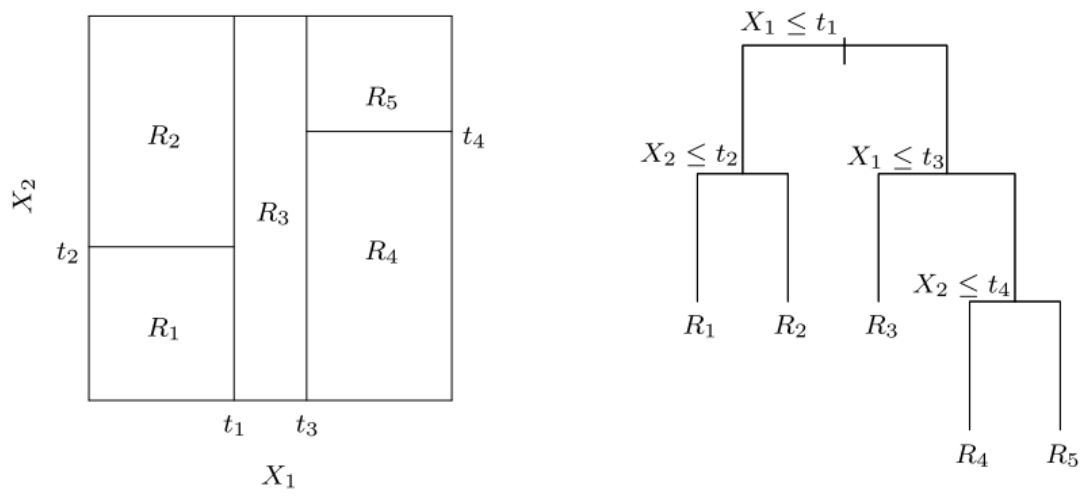

Figura 5.2: Modello di classificazione ad albero in presenza di due variabili esplicative X_1 e X_2: a sinistra la rappresentazione sul piano, a destra la rappresentazione come albero binario (tratto da Hastie *et al.*, 2013)

si andranno a creare dopo la divisione. Tra la vasta gamma di indici adatti a misurare l'entropia è stata scelto il noto indice di Shannon (Formula 5.10), perché considerato più adatto ai problemi di classificazione in presenza di $K > 2$ classi (l'indice di Gini sarebbe stato una valida alternativa).

Più formalmente: siano $R_1 \dots R_M$ le regioni in cui lo spazio delle variabili esplicative è stato diviso e siano K le modalità che assume la variabile risposta. Data la regione R_m, sia N_m il numero di osservazioni al suo interno. La previsione della modalità della variabile risposta nella regione R_m è data da:

$$k = \arg\max_k \hat{p}_{mk} \quad \text{con} \quad \hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \tag{5.9}$$

con $1 \leq k \leq K$. La variabile di *split* e lo *split-point* vengono scelti in modo da minimizzare l'entropia

$$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}. \tag{5.10}$$

Il numero totale di *split*, detto anche profondità dell'albero, è il parametro di regolazione per questo modello e viene scelto in modo tale da ottimizzare l'adattamento

complessivo dell'albero ai dati. Solitamente, l'albero viene fatto "crescere" al massimo, ossia si lascia che lo spazio venga diviso finché sia possibile ridurre l'entropia: quando non esiste più una divisione dello spazio in grado di ridurla, la crescita si ferma. A quel punto segue una fase detta "potatura" dell'albero: il percorso di crescita viene fatto a ritroso accorpando i sotto-spazi precedentemente divisi finché l'errore globale delle previsioni dell'albero calcolato sull'insieme di verifica non sia il minimo raggiungibile.

La divisione dello spazio delle variabili esplicative è rappresentabile come un albero binario (a destra nella Figura 5.2), da qui il nome di "modello ad albero". Ogni nodo corrisponde a uno *split*, effettuato sulla variabile indicata sopra il nodo, e ogni foglia corrisponde alla modalità più frequente nella porzione di spazio generata dai nodi precedenti. Il vantaggio della rappresentazione ad albero rispetto a quella sul piano è che la prima può essere usata anche in presenza di più di due variabili esplicative.

5.3.2 Modelli ad albero: stima del modello e risultato della classificazione

Il modello ad albero è stato stimato R tramite la libreria **tree** (Ripley, 2018). Anche in questo caso, le righe della matrice di regressione sono state pesate con un peso inversamente proporzionale alla frequenza della classe a cui appartengono.

Per ognuna delle 4 classificazioni, l'albero viene fatto crescere utilizzando i dati dell'insieme di stima e viene poi potato tramite l'insieme di verifica. Così facendo si trova il numero ottimale di *split* in grado di ottimizzare l'adattamento del modello ai dati. Gli errori di classificazione e i tempi di calcolo sono riportati per ogni *outcome* nella Tabella 5.2.

Per ognuno dei modelli ad albero si ottiene la sua rappresentazione grafica. Un esempio è riportato nella Figura 5.3, dove si può osservare il processo che porta l'albero a classificare una diagnosi per l'*outcome* N. Sopra a ogni nodo è riportata la variabile scelta per lo *split* (corrispondente a uno stilema o a un bigramma) e il valore che questa deve assumere per appartenere a uno dei due sottospazi generati dallo *split*. Dalla Figura emerge il principale vantaggio dei modelli ad

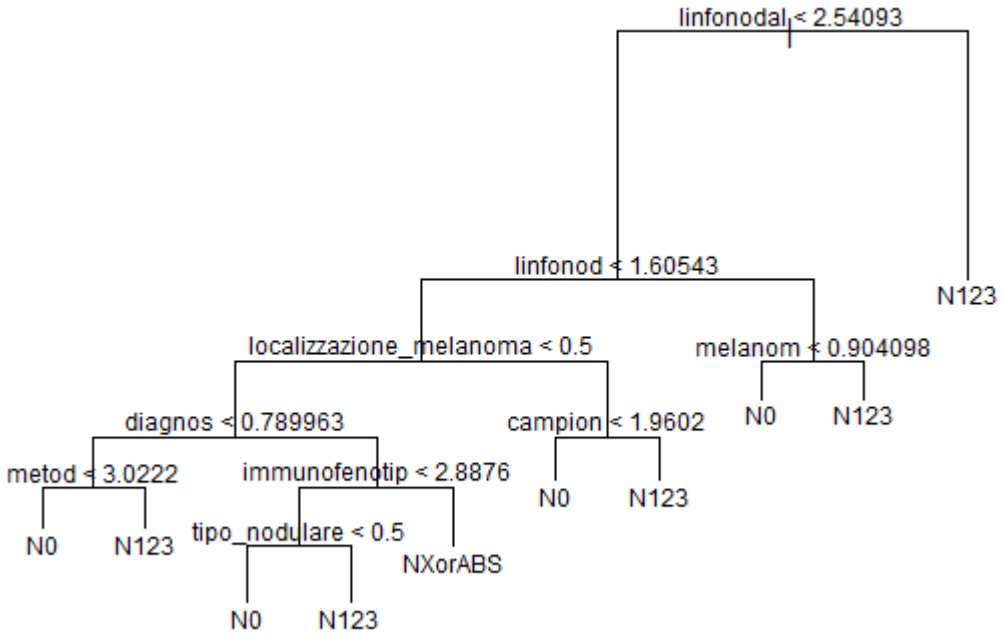

Figura 5.3: Albero di classificazione per l'*outcome* N: la classe `N0` indica l'assenza di linfonodi coinvolti, la classe `N123` indica la presenza di almeno un linfonodo coinvolto, la classe `NXorABS` indica lo Stadio X, ossia che il patologo non è stato in grado di stabilire se ci sono linfonodi coinvolti

Outcome	TNM	T	N	M
Errore di classificazione	28.5%	26.5%	16.2%	7.3%
Tempo di calcolo (sec.)	4.1	1.3	< 1	< 1

Tabella 5.2: Errori di classificazione e tempi di calcolo per i modelli ad albero, TNM indica la stadiazione del tumore, T indica la dimensione del tumore primitivo, N indica il coinvolgimento dei linfonodi e M indica la presenza di metastasi

albero: l'interpretabilità semplice e immediata anche per chi non ha conoscenze di statistica.

5.4 Classificazione con foreste casuali

Con il termine "foreste" solitamente si indicano tutti quei modelli basati su combinazioni di modelli ad albero. Ne esistono diversi: i più noti sono il *bagging* (Breimann, 1996), il *boosting* (Freund e Schapire, 1996) e le foreste casuali (Breimann, 2001), ma ne esistono anche altri, solitamente derivati da trasformazioni dei precedenti. Negli ultimi anni le foreste casuali hanno conosciuto un forte incremento di popolarità grazie alla maggiore semplicità di stima rispetto ad altri modelli simili, garantendo allo stesso tempo ottime performance.

5.4.1 Foreste casuali: descrizione del modello

L'idea sottostante a una foresta casuale è che, stimando più modelli ad albero sugli stessi dati[1] e prendendo come classificazione finale la media delle classificazioni ottenute, sia possibile ottenere previsioni la cui varianza è minore di quella dalle previsione fatta con un singolo albero.

Si supponga di avere B variabili indipendenti e identicamente distribuite con una certa varianza σ^2, la media di queste variabili sarà una variabile con varianza $\frac{\sigma^2}{B}$. Se le variabili non sono indipendenti ma solo identicamente distribuite, si dimostra che la varianza della loro media vale

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{5.11}$$

dove $\rho > 0$ è la correlazione a due a due tra le variabili. Al crescere di B, il secondo termine della 5.11 tende ad annullarsi mentre il primo rimane immutato.

Stimando molti alberi sugli stessi dati e prendendo la loro media come risultato finale si ricade esattamente in questa situazione: una parte di varianza diminuisce al crescere del numero di alberi, ma un'altra parte non cala dal momento che c'è correlazione fra alberi stimati con le stesse variabili esplicative. Le foreste casuali diminuiscono la correlazione fra gli alberi inserendo una componente di casualità

[1] Solitamente si preferisce non stimare ogni albero sui dati nel loro insieme ma su un campione *bootstrap* degli stessi, al fine di sfruttare l'errore *out-of-bag* così generato (Wolpert e MacReady, 1999). Questo dettaglio in seguito verrà ignorato per semplificare la descrizione del modello.

nella loro stima: dopo ogni *split* di ogni albero un sottoinsieme di variabili è scelto casualmente e soltanto queste vengono vagliate come possibili variabili per eseguire lo *split* successivo. Dopodiché, un altro gruppo di variabili viene estratto a caso e ognuna di esse verrà valutata per eseguire lo *split* ancora seguente. Andando avanti di questo passo si ottengono molti alberi diversi tra loro perché utilizzano variabili differenti, di conseguenza la correlazione si riduce abbattendo anche la prima parte della 5.11. La riduzione della varianza ottenuta in questo modo garantisce una classificazione più precisa sia rispetto al semplice albero, sia rispetto a una combinazione di alberi ottenuta a partire dalle stesse variabili. Il numero totale di alberi da combinare e il numero di variabili estratte a caso in ogni passaggio sono considerati i due parametri di regolazione del modello: questi parametri vengono quindi scelti in base all'errore globale calcolato sull'insieme di verifica. Per una trattazione più rigorosa della matematica dietro delle foreste casuali si veda il Capitolo 15 del volume di Hastie *et al.* (2013).

Al contrario del singolo modello ad albero, gli alberi che compongono le foreste casuali non vengono potati bensì vengono lasciati crescere al massimo della loro estensione. Nonostante gli alberi generati siano molto estesi, il modello nel complesso non corre il rischio di sovradattamento ai dati proprio perché frutto di una combinazione tra classificatori ottenuti da variabili diverse. Un altro grande vantaggio delle foreste casuali è la velocità: dal momento che, a ogni *split*, gli alberi devono considerare un ridotto numero di variabili, il tempo di stima è di gran lunga minore rispetto a una semplice media di alberi (quale è per esempio il modello *bagging*). Infine, le foreste casuali sono facilmente implementabili in algoritmi paralleli grazie ai quali il processo di stima si accorcia ulteriormente.

Lo svantaggio delle foreste casuali rispetto ai modelli ad albero è la perdita di interpretabilità del modello. Se nel modello ad albero è possibile visualizzare in che modo le variabili incidono sulla classificazione, nella foresta casuale ciò non avviene poiché la classificazione è il risultato di una media tra alberi diversi. Per sopperire a questa mancanza è possibile usare una misura di importanza delle variabili proposta in Breiman *et al.* (1984) che permette di ordinarle in base al loro contributo globale alla classificazione.

5.4.2 Foreste casuali: stima del modello e risultato della classificazione

I modelli a foreste casuali sono stati stimati in R tramite la libreria **randomForest** (Liaw e Wiener, 2002). Come nei casi precedenti, le righe della matrice di regressione sono state pesate con un peso inversamente proporzionale alla frequenza della classe a cui appartengono.

La scelta del numero di variabili da considerare a ogni *split* è stata fatta in modo da minimizzare l'errore calcolato sull'insieme di verifica. Il numero totale di alberi fatti crescere è stato fissato a 100 per tutti i modelli: si è osservato che già dopo 40 alberi l'errore si attesta al valore minimo, ma utilizzandone un numero maggiore non si incorre nel sovradattamento. Questo è dovuto alla particolare capacità delle foreste casuali nell'essere sostanzialmente immuni al sovradattamento con il crescere del numero di alberi (Breimann, 2001). Gli errori di classificazione e i tempi di calcolo sono riportati per ogni *outcome* nella Tabella 5.3.

Outcome	TNM	T	N	M
Errore di classificazione	25.2%	21.9%	13.9%	6.1%
Tempo di calcolo (sec.)	13.6	13.5	14.7	18.0

Tabella 5.3: Errori di classificazione e tempi di calcolo per i modelli a foreste casuali, TNM indica la stadiazione del tumore, T indica la dimensione del tumore primitivo, N indica il coinvolgimento dei linfonodi e M indica la presenza di metastasi

5.5 Classificazione con *gradient boosting*

Il *boosting* è un metodo per combinare più classificatori tra loro in modo da aumentarne l'efficienza, in particolare si utilizzeranno ancora una volta classificatori ad albero. Freund e Schapire hanno introdotto *AdaBoost*, il primo algoritmo per il *boosting*, nel 1996. Successivamente questo modello è stato modificato da diversi

autori e specialmente grazie ai contributi di Breiman (1997) e Friedman (2001; 2002) il *boosting* è stato migliorato nel più efficiente *gradient boosting*. Quest'ultimo a sua volta è oggetto di ricerca nel campo del *machine learning*: numerosi articoli su questo tema sono stati scritti di recente. Fra questi spicca il lavoro di Chen e Guestrin (2016) che ha dato origine all'algoritmo *XGBoost*, un'implementazione del *gradient boosting* che costituisce lo stato dell'arte dei modelli di *machine learning*.

5.5.1 *Gradient boosting*: descrizione del modello

Il *boosting*

Si consideri, per semplicità, un problema di classificazione simile a quello usato per presentare i modelli a *support vector machines*: si vuole classificare una variabile dicotomica Y che può assumere i valori -1 e 1 usando un insieme di variabili esplicative X. Dato un set di osservazioni (x_i, y_i) appartenenti all'insieme di stima, con $i = 1 \dots n$, si definisce $G(x)$ la classificazione prodotta in base ai dati da un certo classificatore $G(X)$ (nel caso in esame sarà un modello ad albero). Inizialmente, si assume che i dati abbiano lo stesso peso nel generare la classificazione: $w_i = 1/n$, con $i = 1 \dots n$.

Si supponga ora di stimare una sequenza di classificatori $G_m(x)$, con $m = 1 \dots M$, e di combinarla in una media ponderata che produce la classificazione finale

$$G(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right). \tag{5.12}$$

L'idea dietro al *boosting* è di:

1. Modificare in maniera iterativa i pesi $w_1 \dots w_n$ associati alle singole osservazioni in modo da dare una maggiore importanza alle osservazioni classificate incorrettamente;

2. Stabilire i pesi $\alpha_1 \dots \alpha_M$ associati ai classificatori in modo da dare un'importanza maggiore ai classificatori più accurati nella media ponderata finale.

Algoritmo 5.1 *AdaBoost*

1 Si inizializzano i pesi delle osservazioni $w_i = 1/n$, con $i = 1 \dots n$.

2 Per m da 1 a M:

2.a Si stima il classificatore $G_m(x)$ usando w_i.

2.b Si calcola il suo errore di classificazione: $\text{err}_m = \dfrac{\sum_{i=1}^{M} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{M} w_i}$.

2.c Si calcolano i pesi da attribuire ai classificatori: $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.

2.d Si aggiornano i pesi da attribuire alle osservazioni: $w_i \leftarrow w_i \exp(\alpha_m I(y_i \neq G_m(x_i)))$.

3 La classificazione finale è: $G(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$.

Nello specifico, le formule con cui gli α_m vengono calcolati e i w_i aggiornati sono descritte nell'Algoritmo 5.1: al generico passo m, il classificatore $G_m(x)$ viene stimato in base ai pesi ricavati dal passo precedente. Darà quindi più importanza a quelle osservazioni che precedentemente hanno avuto un errore di classificazione maggiore. In questo modo l'algoritmo "impara" dai dati concentrandosi man mano sugli elementi del campione più problematici da classificare.

Questa caratteristica rende il *boosting* un modello molto appetibile per i problemi di classificazione come quello in esame, ovvero con una variabile risposta le cui classi sono sbilanciate. Si può immaginare che gli elementi delle classi meno frequenti abbiano un errore di classificazione maggiore a causa della loro sotto-rappresentazione. Dopo alcuni passi, l'algoritmo *AdaBoost* assegnerà un peso maggiore a queste osservazioni tendendo a controbilanciare il loro scarso numero. Infine, lo stesso algoritmo assegnerà alle classificazioni dei primi passi (quelle dove le osservazioni sotto-rappresentate non hanno ancora acquisito un peso maggiore) un'importanza minore nella classificazione finale.

Molti autori fanno notare che la forza del *boosting* risiede nel sistema di aggiornamento dei pesi dati alle osservazioni e non nell'accuratezza del singolo classificatore $G_m(X)$ (Figura 5.4) . Nell'algoritmo *AdaBoost* non è necessario che i classificatori siano alberi particolarmente profondi, piuttosto è importante garantire un alto numero di iterazioni in modo da assicurarsi che i pesi w_i vengano

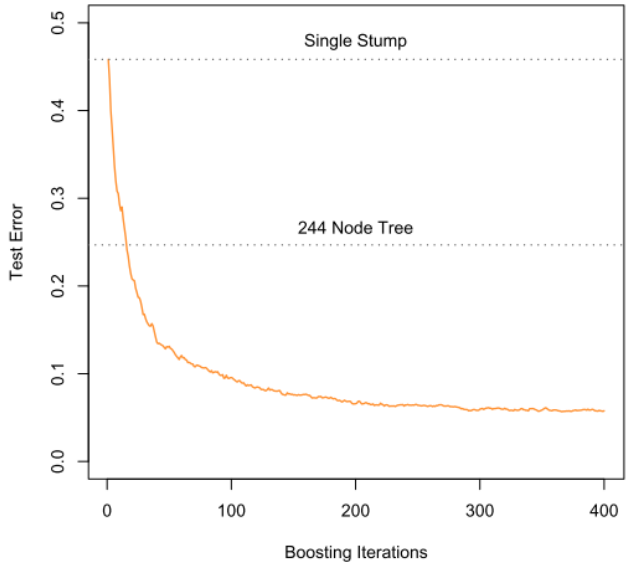

Figura 5.4: Errore percentuale sull'insieme di stima per dati simulati: vengono confrontati un albero *stump*, un albero profondo con 244 nodi e il *boosting* in funzione del numero di iterazioni dell'algoritmo di stima (tratto da Hastie *et al.*, 2013).

definiti nel modo più preciso possibile. Per questo motivo si prediligono alberi molto semplici che risultano veloci da stimare e permettono di effettuare un notevole numero di iterazioni dell'algoritmo in tempi brevi, spesso vengono utilizzati i cosiddetti *stumps*, alberi con solo un nodo e due foglie. Il numero totale di classificatori ad albero stimati M è il parametro di regolazione del modello e viene scelto in modo che minimizzi l'errore globale calcolato sull'insieme di verifica.

Analogamente alle foreste casuali, anche con il modello *gradient bosting* si perde l'immediata interpretazione del modello che caratterizza il singolo albero. Cionondimeno è possibile utilizzare la misura di importanza delle variabili introdotta da Breiman in modo analogo all'utilizzo che se ne fa con le foreste casuali.

Algoritmo 5.2 *Stagewise Additive Modeling*

1. Si inizializza $f_0(x) = 0$.

2. Per m da 1 a M:

2.a Si calcolano β_m e γ_m: $\arg\min_{\beta,\gamma} \sum_{i=1}^{M} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma))$.

2.b Si definisce: $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.

Il *boosting* come modello additivo e il *gradient boosting*

Si supponga di generalizzare la 5.12 nella seguente forma:

$$f(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m), \tag{5.13}$$

questo modello è riconducibile a un modello additivo[2] dove $f(x)$ prende il posto del classificatore finale, le funzioni $b(x, \gamma_m)$ sono i singoli classificatori e b_m è il peso dato a ciascuno di loro. In γ_m sono contenuti tutti i parametri che il modello di classificazione stima dai dati. In questo modello è possibile stimare i parametri d'interesse β e γ congiuntamente minimizzando una funzione $L(y_i, f(x_i))$ che misura la differenza fra i valori predetti e i valori stimati, detta funzione di perdita:

$$\min_{\beta,\gamma} \sum_{i=1}^{n} L\left(y_i, \sum_{m=1}^{M} \beta_m b(x_i; \gamma_m)\right). \tag{5.14}$$

Spesso questa minimizzazione congiunta risulta computazionalmente onerosa e si preferisce approssimarla nella maniera seguente: si stima l'm-esimo classificatore minimizzando la sua funzione di perdita $L(y_i, \beta_m b(x_i; \gamma_m))$ e lo si aggiunge alla somma dei classificatori stimati in precedenza $f_{m-1}(x)$. Si produce quindi la stima di $f_m(x)$, si ripete poi il processo in maniera iterativa. Questa procedura si traduce nell'Algoritmo 5.2, detto anche "modellazione *stagewise*", che di fatto generalizza l'Algoritmo 5.1 permettendo di stimare il modello *boosting* attraverso le funzioni di perdita dei singoli classificatori.

[2]Un modello additivo generalizza il modello di regressione lineare utilizzando, al posto delle semplici covariate, delle funzioni di covariate. Per ulteriori dettagli si veda il Paragrafo 4.5 di Azzalini e Scarpa (2012).

Si riprenda ora il singolo albero di classificazione descritto nel Paragrafo 5.3.1, questo può essere scritto come:

$$T(x;\Theta) = \sum_{j=1}^{J} \gamma_j I(x \in R_j), \tag{5.15}$$

dove: R_j, con $j = 1 \dots J$, sono le regioni dello spazio delle variabili esplicative partizionate dall'albero, γ_j è la classe stimata per la j-esima regione e Θ rappresenta l'insieme tutti i parametri che il modello ad albero stima dai dati (le variabili usate per gli *split*, gli *split point* e la classificazione stimata per ogni foglia). È possibile stimare l'insieme dei parametri minimizzando la funzione di perdita $L(y_i, \gamma_j)$:

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{j=1}^{J} \sum_{x \in r_j} L(y_i, \gamma_j), \tag{5.16}$$

nel caso in esame, la funzione di perdita è rappresentata dall'entropia (Formula 5.10), ma può essere utilizzata qualunque altra funzione di perdita purché sia differenziabile.

A questo punto si unisce questa rappresentazione alternativa dell'albero con il modello additivo *stagewise* appena visto: applicando l'Algoritmo 5.2 alla rappresentazione dell'albero 5.15 si ottiene la somma di alberi

$$f_M(x) = \sum_{m=1}^{M} T(x;\Theta_m). \tag{5.17}$$

A ogni passo **2.a** dell'Algoritmo vengono stimati i parametri tramite la minimizzazione della funzione di perdita:

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{n} L\left(y_i, f_{m-1}(x_i) + T(x_i;\Theta_m)\right) = \arg\min_{\Theta_m} \sum_{i=1}^{n} L\left(y_i, f_m(x_i)\right) \tag{5.18}$$

dove $f_{m-1}(x_i)$ è la somma degli alberi stimati fino al passo precedente.

Questo problema di minimizzazione si affronta tramite il metodo di discesa del gradiente, da cui il nome *gradient boosting*. Il metodo di discesa del gradiente permette di minimizzare funzioni complesse in maniera molto efficiente, per questo motivo si utilizza frequentemente nel contesto del *machine learning* e del *data mining*. In maniera essenziale, consiste nel:

1. Scegliere arbitrariamente un punto di partenza x_k;

2. Presa la funzione di perdita calcolata in x_k, la direzione in cui la funzione discende più velocemente verso il minimo è quella determinata dall'opposto del suo gradiente valutato in quel punto: $\rho_k = -\nabla L\left(y, f_m(x_k)\right)$;

3. Si aggiorna x_k iterativamente: $x_{k+1} = x_k + \alpha_k \rho_k$, dove α_k è un parametro detto "passo di discesa", che controlla la velocità con cui il processo fa scendere il punto di minimo lungo la direzione opposta al gradiente;

4. Dopo un adeguato numero di iterazioni h del secondo e del terzo passo della procedura, x_{k+h} si troverà sul punto di minimo della funzione di perdita.

Tramite questa procedura i parametri dei singoli alberi vengono stimati più velocemente e con più precisione. Per questo motivo ricondursi a un modello additivo e sfruttare il metodo di discesa del gradiente garantisce al *boosting* una performance maggiore rispetto alla sua stima effettuata con l'algoritmo *AdaBoost* (Friedman, 2001).

5.5.2 *Gradient boosting*: stima del modello e risultato della classificazione

Il modello *gradient boosting* è stato implementato in R tramite l'algoritmo *XGBoost* (*eXtreme Gradient Boosting*). *XGBoost* è disponibile nella libreria **xgboost** (Chen *et al.*, 2018) e apporta alcune modifiche al *gradient boosting* al fine di rendere l'algoritmo più veloce grazie al calcolo parallelo e a una migliore gestione della memoria del calcolatore.

Per ognuna della 4 classificazioni è stato calcolato l'errore del modello *gradient boosting* tramite una convalida incrociata all'interno dell'insieme di stima diviso in 10 parti. Grazie alla convalida incrociata si è definito il parametro di regolazione del modello, ossia il numero di iterazioni dell'algoritmo. Infine, utilizzando il numero di iterazioni così ricavato, il modello è stato stimato nuovamente su tutti i dati dell'insieme di stima e il suo errore è stato calcolato su tutti i dati dell'insieme

di verifica. Nella Tabella 5.4 sono riportati gli errori di classificazione e i tempi di calcolo.

Outcome	TNM	T	N	M
Errore di classificazione	20.6%	13.7%	10.1%	5.3%
Tempo di calcolo (sec.)	98.7	85.3	77.0	79.2

Tabella 5.4: Errori di classificazione e tempi di calcolo per i modelli *gradient boosting*, TNM indica la stadiazione del tumore, T indica la dimensione del tumore primitivo, N indica il coinvolgimento dei linfonodi e M indica la presenza di metastasi

5.6 Classificazione con reti neurali

Le reti neurali sono la classe di modelli più ampia, studiata e dibattuta del *machine learning*. La loro origine risale agli anni sessanta, ma conobbero una grande crescita negli anni ottanta con lo sviluppo dell'algoritmo di stima *back-propagation*. Grazie al rapido progresso dell'intelligenza artificiale iniziato nello scorso decennio (e tutt'ora in corso), le reti neurali stanno vivendo un periodo di auge: sono largamente utilizzate per risolvere gran parte dei problemi di *data mining* e *machine learning*, compresi quelli legati al *text mining*.

Un altro motivo della recedete popolarità delle reti neurali è la disponibilità di strumenti di calcolo sempre più potenti: fino a qualche decennio fa la stima di una rete a più strati sarebbe stata possibile solo su un ristretto numero di computer, invece al giorno d'oggi la stessa rete è stimabile su un comune *laptop* grazia all'avanzamento delle strutture *hardware* e l'affinamento dell'algoritmo di *back-propagation*.

Nel seguente paragrafo non si ha l'intenzione di fornire una descrizione dettagliata del funzionamento delle reti neurali ma solo una breve introduzione, si rimanda alla letteratura per ulteriori dettagli. Una buona panoramica su questi modelli si può trovare in Ripley (2007) o in Goodfellow *et al.* (2016).

5.6.1 Reti neurali: descrizione del modello

Si supponga di avere a disposizione delle osservazioni su $X_1 \dots X_p$ variabili esplicative (dette anche "di *input*") e una variabile risposta Y ("di *output*") con $k = 1 \dots K$ modalità. Una rete neurale è un modello che collega le prime alle seconde per mezzo di una regressione non lineare a più stadi organizzata come segue:

- Le variabili esplicative sono disposte in un primo strato (*layer*) situato all'inizio della rete, all'interno dello strato si fa corrispondere ogni variabile a un nodo;
- La variabile risposta è codificata come $Y_1 \dots Y_K$ variabili dicotomiche, ognuna delle quali rappresenta la presenza/assenza delle osservazioni in quella determinata classe. Queste variabili sono disposte in uno strato finale;
- Nel mezzo vengono posti uno o più strati latenti (*hidden layers*) che collegano lo strato di *input* a quello di *output*. Ognuno di questi strati avrà al suo interno un certo numero di nodi ($Z_1 \dots Z_M$, $W_1 \dots W_T$, e così via);
- I nodi di due strati consecutivi sono collegati da una funzione di regressione (detta anche "funzione di attivazione") i cui parametri prendono il nome di "pesi".

Ad esempio, se si fa riferimento alla rete rappresentata nella Figura 5.5, i nodi dei 4 strati (uno iniziale, uno finale e due latenti) saranno collegati nel seguente modo:

$$Z_m = f_0\left(\sum_{p\to m} \alpha_{pm} X_p\right) \qquad W_t = f_1\left(\sum_{m\to t} \beta_{mt} Z_m\right) \qquad Y_k = f_2\left(\sum_{t\to k} \gamma_{tk} W_t\right)$$

dove α, β, γ sono i parametri e f_0, f_1, f_2 sono le tre funzioni di attivazione. Queste funzioni possono essere tra loro uguali o differenti a seconda della struttura che si sceglie di dare alla rete. In un contesto di classificazione le scelte più frequenti per le funzioni di attivazione sono: la funzione sigmoidea[3] $f(u) = 1/(1 + e^{-u})$, la funzione ReLU $f(u) = \max(0, u)$ e la funzione *softplus* $f(u) = \log(1 + e^u)$.

[3]La funzione sigmoidea non è altro che un caso particolare della funzione logistica.

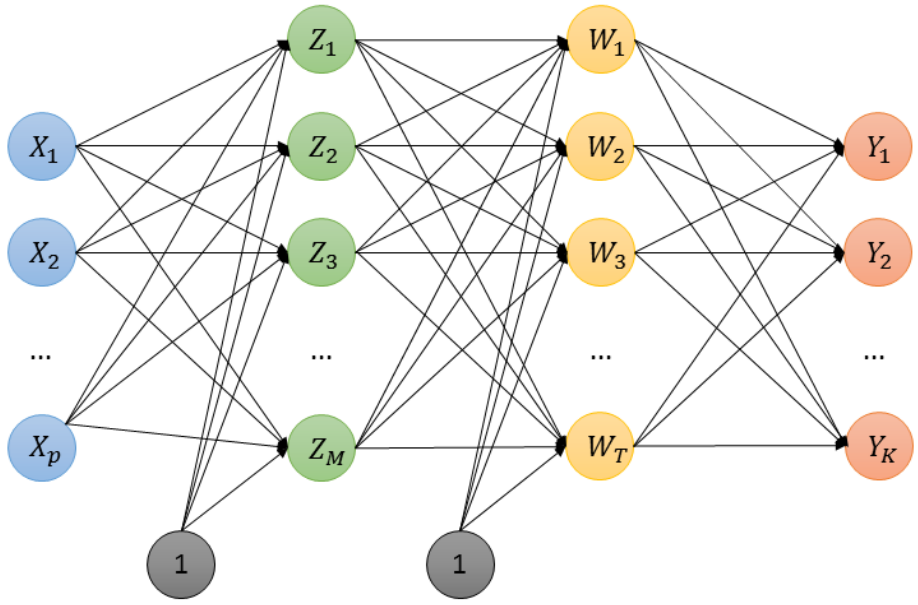

Figura 5.5: Schema di una rete neurale a due strati: i nodi in grigio rappresentano una costante pari a 1 aggiunta per generare l'intercetta, ogni arco che connette due nodi rappresenta un parametro

Si nota immediatamente come il numero totale di parametri diventi estremamente alto al crescere del numero dei nodi e degli strati. Questa massiccia parametrizzazione è allo stesso tempo il vantaggio e lo svantaggio dell'utilizzo delle reti neurali: da una parte permette una grande flessibilità, tanto che si può dimostrare che questa classe di modelli rientra fra gli approssimatori universali di funzioni (Hornik *et al.*, 1989), dall'altra rappresenta un problema in fase di stima perché richiede l'impiego di un algoritmo sofisticato e computazionalmente impegnativo.

L'algoritmo usato per la stima delle reti neurali è detto *back-propagation*. Il *back-propagation* in questa tesi non verrà affrontato ma presenta molte affinità con la procedura di discesa del gradiente impiegata per il modello *gradient boosting* (Paragrafo 5.5.1).

Il numero di strati latenti e il numero di nodi per strato latente vengono con-

siderati i due parametri di regolazione del modello, la loro scelta è guidata sia dall'ottimizzazione dell'errore globale calcolato sull'insieme di verifica sia dalla vasta teoria delle reti neurali, che spesso suggerisce la struttura ottimale della rete in base al tipo di problema per cui la si usa.

5.6.2 Reti neurali: stima del modello e risultato della classificazione

Per stimare le reti neurali si è fatto ricorso a **keras** (Chollet e Allaire, 2018). **keras** è una libreria di alto livello, scritta in Python ma disponibile anche su R, specializzata per i modelli di *deep learning* (ovvero: reti neurali a più strati, reti neurali convoluzionali, reti neurali sequenziali...). **keras** non esegue le operazioni su R, bensì funge da interfaccia fra R e un *machine learning framework* scelto dall'utente, detto anche "libreria di *back-end*". Il *back-end* utilizzato per tutti i modelli è la libreria **TensorFlow** (Google Brain Team, 2018).

La struttura delle reti stimatesi basa su quanto indicato nel Paragrafo 3.4 di Chollet e Allaire (2018) dove gli autori suggeriscono una serie di conformazioni per reti neurali che più si adattano per affrontare alcuni tipi di problemi. Per i problemi di *text mining* simili a quello esposto in questa tesi è suggerita una rete con due strati latenti da 16 nodi ciascuno, dunque in tutto si avranno 4 strati: uno di *input* contenente le variabili esplicative, un primo strato latente collegato allo strato di *input* per mezzo della funzione di attivazione ReLU, un secondo strato latente collegato anch'esso con il precedente per mezzo della funzione ReLU e, infine, un ultimo strato di *output* contenente un numero di nodi pari al numero di classi dell'*outcome* del modello e collegato al secondo strato latente per mezzo della funzione sigmoidea. Successivamente, il numero di nodi negli strati latenti è stato portato a 32, scelta guidata dalla diminuzione dell'errore di classificazione che si ottiene sull'insieme di verifica. Infine è stato aggiunto un *drop-out* del 20% ad ogni strato latente: il *drop-out* consiste nel forzare a 0 una certa percentuale di nodi estratti casualmente durante la fase di stima della rete. Questa tecnica è comunemente utilizzata per prevenire il sovradattamento, come dimostrato da Srivastava *et al.* (2014).

Outcome	TNM	T	N	M
Errore di classificazione	24.5%	18.0%	12.4%	5.9%

Tabella 5.5: Errori di classificazione per le reti neurali, TNM indica la stadiazione del tumore, T indica la dimensione del tumore primitivo, N indica il coinvolgimento dei linfonodi e M indica la presenza di metastasi

La stima delle 4 reti restituisce gli errori di classificazione indicati nella Tabella 5.5. In questo caso non sono stati riportati i tempi di calcolo perché non sarebbero confrontabili con gli altri modelli dal momento che R non svolge le operazioni bensì effettua un richiamo a **TensorFlow**. Questo processo di richiamo richiede parecchi secondi, dunque è stato impossibile stabilire in che misura i tempi di calcolo siano dovuti alla vera e propria stima della rete o siano dovuti all'interfaccia tra R e **TensorFlow**. Ad ogni modo, viste le somiglianze fra il *back-propagation* e l'algoritmo di discesa del gradiente, ci si aspetta che i tempi di stima per le reti neurali siano simili a quelli dei modelli *gradient boosting*.

5.7 I modelli a confronto

Nella Tabella 5.6 sono stati messi a confronto gli errori di classificazione dei modelli per ogni *outcome*. È riportato anche un *baseline error*, ottenuto supponendo di classificare esclusivamente con la classe più frequente[4]. Per esempio, per l'*outcome* che esprime lo stadio del tumore (TNM) il *baseline error* si ottiene classificando ogni testo come "Stadio I". Per quanto una classificazione di questo genere non abbia alcuna valenza, può servire per valutare quanto i modelli siano migliori rispetto a un approccio *naive*.

Si nota subito come il modello *gradient boosting* ottenuto con l'algoritmo *XGBoost* abbia l'errore più basso in tutte le classificazioni: questo risultato, per certi versi atteso, conferma l'efficienza del *gradient boosting* nel trattare i dati sbilanciati: in seguito si farà riferimento esclusivamente alla classificazione generata da

[4]Se le classi fossero state bilanciate, il *baseline error* sarebbe stato calcolato per mezzo di una classificazione casuale.

Outcome	*Baseline error*	*Support vector machines*	Singolo albero
TNM	31.7%	26.3%	28.5%
T	30.1%	18.7%	26.5%
N	17.9%	14.4%	16.2%
M	7.5%	6.5%	7.3%

Outcome	Foresta casuale	*Gradient boosting*	Rete neurale multistrato
TNM	25.2%	20.6%	24.5%
T	21.9%	13.7%	18.0%
N	13.9%	10.1%	12.4%
M	6.1%	5.3%	5.9%

Tabella 5.6: Errori dei modelli a confronto

questo modello per discutere le conclusioni finali. Si noti che per il *gradient boosting* non è stato necessario pesare le righe della *document-term matrix* come è stato fatto per gli altri modelli poiché, dopo pochi passi, l'algoritmo di stima vi assegna automaticamente un peso ottimale.

Le reti neurali, le foreste casuali e i modelli a SVM svolgono anch'essi una classificazione accurata, anche se meno efficiente del modello *gradient boosting*. I modelli a SVM richiedono poco tempo di stima per via della loro natura "geometrica" che permette di eseguire un minimizzazione di tipo convesso, al contrario degli altri modelli che si basano su procedure iterative. Dunque se si fosse interessati a ottenere una classificazione veloce anche a costo di sacrificarne l'efficienza, probabilmente il modello a SVM sarebbe la scelta più appropriata.

Della classificazione tramite *gradient boosting* non va valutato solamente l'errore, bensì vanno discusse le matrici di errata classificazione, gli errori commessi su ogni singola classe e va condotta un'analisi dell'importanza che hanno gli stilemi nel classificare i testi. Questi dettagli, assieme alle conclusioni che ne seguono, saranno trattati nel Capitolo 6.

5.8 Riepilogo del capitolo

Utilizzando la *document-term matrix* come una matrice di regressione è possibile stimare dei modelli statistici di classificazione supervisionata la cui risposta sono le variabili rese disponibili dal *gold standard*. Dal momento che la matrice di regressione è di grandi dimensioni ed è estremamente sparsa, sono preferibili modelli di *data mining* che permettono di gestire più facilmente le alte dimensionalità.

Sono state stimate 4 classificazioni, una per ogni *outcome*, per mezzo di vari modelli: modelli a *support vector machines*, modelli ad albero, foreste casuali, *gradient boosting* (nella sua variante *XGBoost*) e reti neurali multistrato. Ognuno di questi modelli è stato descritto dal punto di vista matematico e, successivamente, ne è stata affrontata la stima in R. Il modello *gradient boosting* è risultato essere il modello migliore, principalmente perché è in grado di gestire lo sbilanciamento delle classi degli *outcome*.

Capitolo 6

Discussione dei risultati e conclusioni

È emerso come il modello *gradient boosting* ottenuto con l'algoritmo *XGBoost* abbia performance migliori rispetto agli altri modelli stimati. In questo Capitolo si analizzeranno più a fondo i risultati ottenuti per mezzo di questo modello, confrontandoli con la letteratura disponibile. Si discuterà anche del perché la procedura di *text mining* abbia dei limiti di utilizzo pratico e, infine, verranno proposte alcune possibili vie di sviluppo per incrementarne l'efficienza.

6.1 Discussione dei risultati

Prima di procedere va fatta un'osservazione fondamentale: nel Capitolo 5 sono stati illustrati i modelli e le loro procedure di stima senza tener conto del contesto di applicazione. Rispetto ad altri contesti, in quello clinico sono richiesti errori molto bassi, dunque quelli ottenuti dal modello *gradient boosting* non sono soddisfacenti. Per esempio, un registro tumori che volesse adottare una procedura simile non potrebbe permettersi di estrarre dai testi lo stadio del tumore sbagliando nel 20% dei casi. Ciononostante si vuole sottolineare che, considerando la natura della matrice di regressione (estremamente sparsa) e della distribuzione molto sbilanciata degli *outcome*, il tasso di errata classificazione corrisponderebbe a un buon risultato in altri contesti, come per esempio quello della *sentiment analysis*.

6.1.1 Errori di classificazione e matrici di confusione

Nella Tabella 5.4 sono stati riportati solamente i tassi di errata classificazione ottenuti per mezzo del modello *gradient boosting*. Per comprendere meglio in che maniera è distribuito l'errore di classificazione, si riportano le intere matrici di confusione (Tabelle 6.1, 6.2, 6.3 e 6.4). Si nota facilmente che le classi dove l'errore è più alto corrispondono a quelle con una frequenza più bassa. Nonostante il modello *gradient boosting* sia in grado di gestire bene la presenza di classi sbilanciate, il livello di sbilanciamento in questo caso è tale da portare comunque a un errore di classificazione molto alto sulle classi poco frequenti. Al contrario, le classi più frequenti vengono classificate con errori bassi, rendendo possibile l'individuazione di un corretto stadio del tumore solo quando è di Stadio I.

Un altro problema che emerge dalle matrici di confusione è l'alto errore che deriva dalle classi X degli *outcome*. Nel Paragrafo 2.2.3 si è descritta la classe X come "stadio non misurabile", ossia corrispondente al caso in cui chi ha effettuato la stadiazione del tumore non è stato in grado di stabilire con sicurezza uno o più parametri T, N o M. Si può immaginare che un tumore di Stadio X sia in realtà appartenente a uno degli altri stadi, è dunque probabile che nel testo della relativa diagnosi siano contenute caratteristiche del tumore riconducibili ad altre classi. Questo porta lo Stadio X ad essere difficilmente classificabile.

Questo fenomeno si riscontra particolarmente per l'*outcome* legato alla presenza di metastasi. Nella Tabella 6.4 si osserva come il numero di casi realmente appartenenti allo Stadio X sia nettamente maggiore di quelli appartenenti alla classe che corrisponde alla presenza di metastasi ma, allo stesso tempo, l'errore facente riferimenti allo Stadio X è di poco superiore a quello facente riferimento alla classe che rappresenta la presenza di metastasi.

I risultati ottenuti posso essere comparati con alcuni lavori simili presenti in letteratura. Si è riscontrato che la maggior parte delle ricerche relative al *text mining* in ambito oncologico adottano un approccio *rule-based* (Paragrafo 1.5.1) e non statistico, dunque i lavori con cui questa tesi può essere confrontata non sono molti. Un esempio è dato da Nguyen *et al.* (2007), dove gli autori applicano il *text mining* clinico in maniera simile a quanto fatto in precedenza utilizzando testi

		Gold standard					
		I	II	III	IV	X	TOT
Previsione	I	1694	109	98	27	105	2033
	II	23	156	22	6	18	225
	III	25	17	134	12	15	203
	IV	3	1	2	2	4	12
	X	14	10	14	5	58	101
	TOT	1759	293	270	52	200	2574

Errore globale:	20,6% (31,7%)
Errori su singola classe:	
I	3,7%
II	46,8%
III	50,4%
IV	96,2%
X	71,0%

Tabella 6.1: Matrice di confusione per l'*outcome* corrispondente allo stadio del tumore (stadiazione TNM), l'errore riportato tra parentesi è il valore *baseline*

		Gold standard			
		Basso	Alto	X	TOT
Previsione	Basso	1726	176	82	1984
	Alto	68	444	20	532
	X	5	3	50	58
	TOT	1799	623	152	2574

Errore globale:	13,8% (30,1%)
Errori su singola classe:	
Basso	4,1%
Alto	28,7%
X	67,1%

Tabella 6.2: Matrice di confusione per l'*outcome* corrispondente alla grandezza del tumore primitivo (parametro T), l'errore riportato tra parentesi è il valore *baseline*

		Gold standard			
		Assenti	Presenti	X	TOT
Previsione	Assenti	2071	157	46	2274
	Presenti	39	182	11	232
	X	3	4	61	68
	TOT	2113	343	118	2574

Errore globale:	10,1% (17,9%)
Errori su singola classe:	
Assenti	2,0%
Presenti	46,9%
X	48,3%

Tabella 6.3: Matrice di confusione per l'*outcome* corrispondente alla presenza di linfonodi infetti (parametro N), l'errore riportato tra parentesi è il valore *baseline*

provenienti da referti relativi a 710 casi di tumore al polmone. Il classificatore utilizzato, noto come *hierarchical support vector machine* (Chen *et al.*, 2004), consiste in un albero che effettua ogni *split* tramite un modello a SVM, dunque le SVM impiegate attuano una classificazione binaria e non multi-classe. Ne risulta un tasso di errata classificazione del 35% per il parametro T (diviso però in 5 classi al posto

		Gold standard			
		Assenti	Presenti	X	TOT
Previsione	Assenti	2355	29	80	2464
	Presenti	2	22	0	24
	X	23	2	61	86
	TOT	2380	53	141	2574

Errore globale:	5,3% (7,5%)
Errori su singola classe:	
Assenti	1,1%
Presenti	58,5%
X	56,7%

Tabella 6.4: Matrice di confusione per l'*outcome* corrispondente alla presenza di metastasi (parametro M), l'errore riportato tra parentesi è il valore *baseline*

di 3) e del 18% per il parametro N (diviso in 4 classi al posto di 3), non si hanno invece risultati per la presenza di metastasi o per la stadiazione complessiva. Un altro esempio è McCowan *et al.* (2007), i quali utilizzano lo stesso *corpus* di testi dell'articolo precedente ma adottano direttamente un modello a SVM multi-classe (come quello stimato nel Paragrafo 5.2.2): con questo modello è stato estratto il parametro T con un tasso di errata classificazione pari al 26% e il parametro N con un tasso di errata classificazione pari al 16%. Si cita infine il lavoro di Martinez *et al.* (2013), dove gli autori utilizzano dei referti relativi al cancro del colon-retto e comparano i risultai di vari modelli, tra cui il classificatore *naive Bayes*, i modelli a SVM, le foreste casuali e le reti bayesiane. I migliori modelli risultano essere quelli bayesiani, in grado di classificare il parametro T con un errore del 24%, il parametro N con un errore del 19% e il parametro M con un errore pari a 11%.

Risulta comunque difficile fare paragoni precisi tra questi risultati e quelli ottenuti in questa tesi per i seguenti motivi:

1. I testi fanno riferimento a tumori diversi con caratteristiche patologiche diverse, di conseguenza i parametri T, N e M sono stati spesso accorpati diversamente da quanto fatto in precedenza (e se un accorpamento diverso si traduce in un numero di classi maggiore, l'errore di classificazione tenderà anch'esso a essere maggiore).

2. La lingua è una componente determinante nei processi di *text mining*. Il numero di parole utilizzabili in un certo contesto, come per esempio quello clinico, può variare molto con la lingua utilizzata e con esso può variare

anche il rumore presente nei testi. I lavori sopra citati utilizzano testi in lingua inglese, mentre i testi del *corpus* usato in precedenza sono in lingua italiana.

Anche tenendo in considerazione queste due problematiche, si può concludere che gli errori della procedura esposta in precedenza siano in linea con quelli presenti in letteratura, se non leggermente inferiori.

6.1.2 Gli stilemi più rilevanti nella procedura di *text mining*

Un modello ad albero, grazie alla sequenza iterativa di *split*, è in grado di selezionare le variabili rilevanti per la classificazione. Se, ad esempio, si considera l'albero della Figura 5.3, si nota che solo pochi tra i 695 stilemi e bigrammi contenuti nella *document-term matrix* sono stati utilizzati per la sua costruzione. Un processo analogo avviene per il modello *gradient boosting* il quale, essendo una combinazione di alberi, effettua una selezione molto più ampia degli stilemi e dei bigrammi considerandone un numero totale di gran lunga maggiore rispetto al singolo albero (nello specifico, tra i 250 e i 500 circa a seconda del modello). È quindi di interesse capire quali stilemi o bigrammi siano stati utilizzati dal modello *gradient boosting* e, fra questi, quali abbiano avuto un maggior peso nella procedura di classificazione.

Così come nel caso delle foreste casuali, anche per il *gradient boosting* è possibile definire una misura dell'importanza che le variabili hanno avuto nel processo di classificazione. Questa misura è definita *importance* (Breiman *et al.*, 1984) ed è nulla per le variabili che non sono state incluse in nessuno degli alberi stimati dal *gradient boosting*. Nella Figura 6.1 è rappresentata la funzione di densità dell'indice di *importance* per le variabili del modello relativo alla stadiazione TNM (il grafico è simile per gli altri tre *outcome*). Si nota che quasi tutte le variabili utilizzate dal modello hanno un'*importance* bassa (minore di 0.01), un ridotto numero di variabili ha un'*importance* medio-alta (tra 0.01 e 0.02) e solo pochissime variabili hanno un'*importance* molto alta (maggiore di 0.02). In particolare si notano, sulla coda della distribuzione, due variabili che hanno contribuito più di tutte al processo di classificazione con un indice di *importance* estremamente alto.

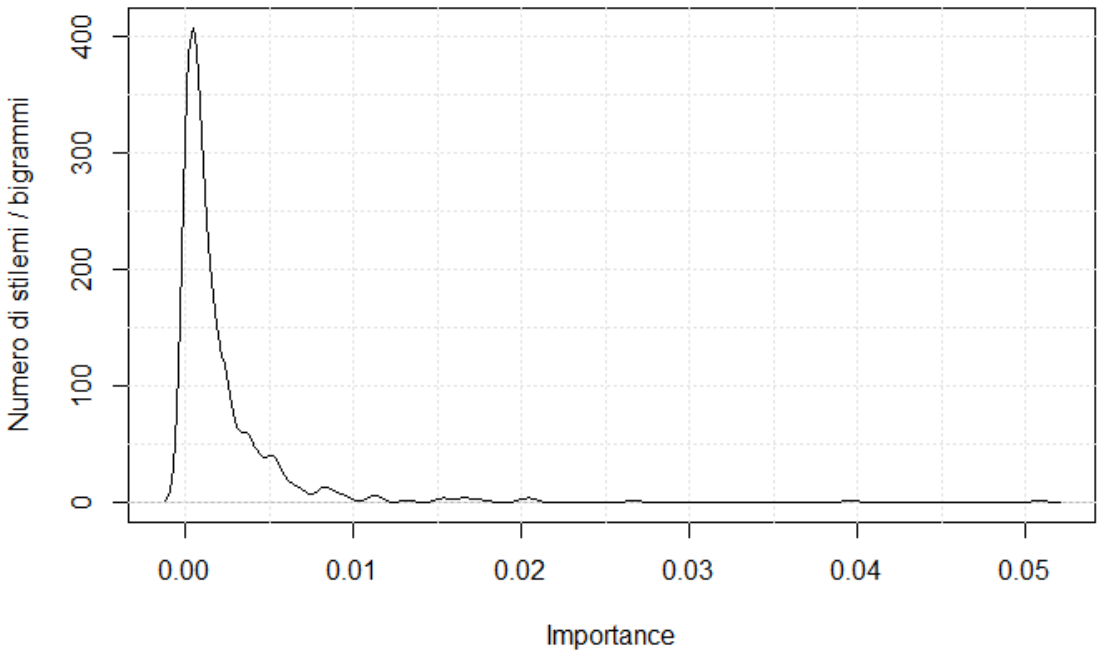

Figura 6.1: Densità dell'indice di *importance* delle variabili nel modello *gradient boosting* relativo alla stadiazione TNM

Nell'Appendice C sono riportate le rappresentazioni grafiche delle *importance* più alte per i quattro modelli: in ogni grafico si nota un piccolo *cluster* di una o due variabili che apportano da sole il maggior contributo alla classificazione:

- Per l'*outcome* relativo alla stadiazione TNM sono le variabili "urotelial" e "infonodal". Inoltre, sono di grande importanza "ulcerazione presente", "melanom", "cellule uroteliali", "metastas", "lesion", "malign" e "cellul".
- Per l'*outcome* relativo al parametro T sono "lesion" e "ulcerazione presente". Inoltre, sono di grande importanza "urotelial", "linfonodal", "melanom" e "metastas".
- Per l'*outcome* relativo al parametro N, la variabile più importante è, ovviamente, lo stilema "linofodal", seguito da "melanom".
- Per l'*outcome* relativo al parametro M sono "linfonodal" e "melanom". Inoltre, sono di grande importanza "lesion", "malign", "ipoderm" e "tess". Lo sti-

lema “metastas”, anche se non è nelle prime posizioni, è comunque tra i più importanti.

L’indice di *importance* di questi stilemi può essere confrontato con la Tabella 4.3, dove sono contenuti i pesi *tf-idf*. Si noti che stilemi con un basso *tf-idf* come, per esempio, “leison”, “nev” o “melanom”, siano stati comunque di grande importanza nel processo di classificazione. Al contrario, stilemi con un *tf-idf* molto alto come, per esempio, “adenom” o “nodul”, non hanno un indice di *importance* alto. Questo conferma la tesi avanzata nel Paragrafo 4.3.2: il solo peso *tf-idf* non è sufficiente per valutare l’importanza che ha uno stilema nel processo di *text mining*, lo si può solo fare attraverso un modello statistico.

6.2 Rimozione dello Stadio X e aumento della sensibilità

Nel Paragrafo precedente si è visto come lo Stadio X possa essere una potenziale fonte di rumore per la classificazione, in particolare per quanto riguarda l’*outcome* che esprime la presenza di metastasi. Per questo motivo, è stata ripetuta la stima dei modelli *gradient boosing* per gli *outcome* legati ai parametri T, N e M ignorando tutte le unità statistiche il cui *gold standard* fosse lo Stadio X. È evidente che questi nuovi modelli abbiano un’utilità pratica solo nelle situazioni in cui ci sia la certezza dell’assenza dello Stadio X, evento raro ma non escludibile. In assenza dello Stadio X l’errore globale diminuisce, sia perché in un qualunque problema di classificazione rimuovendo una classe si riducono i possibili casi di incorretta classificazione (ossia le celle al di fuori della diagonale della matrice di confusione), sia perché la classe rimossa è la più problematica da classificare, specialmente per l’*outcome* legato alla presenza di metastasi.

Nelle Tabelle 6.5, 6.6 e 6.7 si possono osservare le nuove classificazioni in assenza dello Stadio X. Il tasso di errata classificazione cala da 13.7% a 11.4%, da 10.1% a 7.0% e da 5.3% a 1.7% rispettivamente per gli *outcome* legati ai parametri T, N e M.

		Gold standard		
		Basso	Alto	TOT
Prev.	Basso	1709	186	1895
	Alto	90	437	527
	TOT	1799	623	2422

Errore globale:	11,4%	(19,1%)
Sensibilità:	70,1%	
Specificità:	95,0%	

Tabella 6.5: Matrice di confusione per l'*outcome* corrispondente alla grandezza del tumore primitivo (parametro T) in assenza dello Stadio X, l'errore riportato tra parentesi è il valore *baseline*

		Gold standard		
		Assenti	Presenti	TOT
Prev.	Assenti	2088	148	2236
	Presenti	25	195	220
	TOT	2113	343	2456

Errore globale:	7,0%	(14,0%)
Sensibilità:	56,9%	
Specificità:	98,8%	

Tabella 6.6: Matrice di confusione per l'*outcome* corrispondente alla di linfonodi infetti (parametro N) in assenza dello Stadio X, l'errore riportato tra parentesi è il valore *baseline*

		Gold standard		
		Assenti	Presenti	TOT
Prev.	Assenti	2378	39	2417
	Presenti	2	14	16
	TOT	2380	53	2433

Errore globale:	1,7%	(2,1%)
Sensibilità:	26,4%	
Specificità:	99,9%	

Tabella 6.7: Matrice di confusione per l'*outcome* corrispondente alla presenza di metastasi (parametro M) in assenza dello Stadio X, l'errore riportato tra parentesi è il valore *baseline*

Se, per queste tre classificazioni, si definiscono "sani" i soggetti con la condizione meno grave (tumore primitivo di dimensione bassa, assenza di linfonodi coinvolti o assenza di metastasi) e "malati" i rimanenti[1], si possono calcolare la sensibilità e la specificità della classificazione. In linea con le aspettative, la specificità della classificazione è altissima, ovvero la probabilità che un sano risulti classificato come tale è vicina al 100%. Questo è dovuto, ancora una volta, al forte sbilanciamento tra le classi in favore di quella associata ai sani. Per lo stesso motivo, la sensibilità della classificazione (ossia la probabilità che un malato sia classificato come tale) è molto più bassa.

Un altro vantaggio della rimozione dello Stadio X è ottenere una classificazione di tipo dicotomico e, di conseguenza, poter spostare la soglia con cui viene effettuata la classificazione (o valore di *cut-off*) al fine di privilegiare la sensibilità o la specificità (Fawcett, 2006). In un contesto clinico come quello in esame si tende a prediligere un'alta sensibilità: per esempio, se un soggetto ha sviluppato delle metastasi si vuole che la procedura di *text mining* associ il testo della sua diagnosi alla presenza di metastasi con un errore minimo, al costo di sbagliare maggiormente nel segnalare la presenza di metastasi quando queste non ci sono (ossia di generare un falso positivo).

Fino a questo momento il *cut-off* è sempre stato pari a 0.5, ovvero: il modello *gradient boosting* associa ad ogni soggetto una probabilità di appartenere al gruppo dei malati, se questa probabilità è maggiore di 0.5 il soggetto è classificato come malato, altrimenti è classificato come sano. Spostando il *cut-off* su valori più alti, è possibile aumentare la sensibilità a discapito della specificità (Tabella 6.8). Questo genera un beneficio pratico: supponendo di implementare in un registro tumori una procedura di *text mining* di questo genere, essa sarà in grado di riconoscere con grande precisione i testi associati a una dimensione bassa del tumore primitivo, alla assenza di linfonodi infetti o alla assenza di metastasi. Questi testi non serve vengano analizzati manualmente perché la probabilità che siano associati alla con-

[1]Il fine di questa dicitura è solamente quello di ricondursi ai classici concetti di sensibilità e specificità propri del campo epidemiologico, non si vuole sostenere che un soggetto rientrato nella casistica in esame, e quindi affetto da melanoma, sia sano dal punto di vista clinico.

Outcome	*cut-off*		
Dimensione del tumore primitivo	0.05	Sensibilità:	95.8%
		Specificità:	31.4%
Presenza di linfonodi infetti	0.01	Sensibilità:	98.5%
		Specificità:	11.6%
Presenza di metastasi	0.001	Sensibilità:	96.2%
		Specificità	7.5%

Tabella 6.8: Valori di *cut-off* che massimizzano la sensibilità

dizione opposta sono molto basse. Al contrario, i testi rimanenti dovranno essere comunque analizzati manualmente perché potrebbero rientrare in una qualunque delle due classi.

6.3 Conclusioni

Ciò che è emerso in questo lavoro porta a tre conclusioni: la prima riguardante la fattibilità del *text mining* clinico con approccio statistico supervisionato per testi in lingua italiana, la seconda riguardante la scelta del modello statistico più adatto e la terza riguardante i limiti di questa procedura.

1. In base alle ricerche fatte, questa tesi è il primo caso di *text mining* clinico su testi in lingua italiana condotto con un approccio di tipo statistico supervisionato (se ne ritrova uno, sempre in lingua italiana, con un approccio non supervisionato in Alicante *et al.*, Alicante et al. 2016). L'intera procedura, nel complesso, è risultata fattibile al pari di quelle eseguite su testi in lingua inglese e ha prodotto risultati simili. L'impostazione tenuta rispecchia il classico approccio al *text mining* che si adotta in presenza di un *gold standard*, dunque supervisionato.

2. Solitamente la scelta dei modelli da utilizzare in fase di classificazione ricade sulle *support vector machines*. Questo tipo di modelli ha dei grandi vantaggi,

primo tra tutti la velocità di stima, ma in questa tesi si conclude che un modello *gradient boosting* è preferibile per questo tipo di dati. In presenza di un matrice di regressione molto sparsa, di una forte componente di rumore e di un grande sbilanciamento delle classi della variabile risposta il *gradient boosting* è in grado di produrre risultati più precisi per via della natura del suo algoritmo di stima, che attribuisce pesi diversi alle unità statistiche a seconda della bontà con cui vengono classificate. A questo si aggiunge l'utilizzo dell'algoritmo *XGBoost*, che rende la stima del modello più veloce e accurata.

3. Così come per gli analoghi lavori su testi il lingua inglese, anche quello qui presentato ha dei forti limiti di utilizzo pratico. L'errore con cui si estrae l'informazione dai testi è troppo alto perché questa procedura possa essere utilizzata così com'è in una struttura quale un registro tumori. Ciò non toglie che questo approccio sia perseguibile: nel Paragrafo 6.4 verranno esposte diverse strade percorribili al fine di migliorarne l'efficienza.

Concludendo, l'approccio adottato per trattare i testi in esame si è rivelato funzionale ma non abbastanza efficiente. Esiste uno spazio di miglioramento che, se opportunamente perseguito, potrebbe portare l'intera procedura ad essere utilizzabile nel contesto clinico.

6.4 Possibili sviluppi

In seguito verranno proposte due diverse strategie perseguibili per aumentare l'efficienza complessiva della procedura di *text mining* clinico. La prima consiste nell'integrare l'approccio statistico con opportuni passaggi tipici dell'approccio *rule-based*, ottenendo di fatto un approccio misto; la seconda consiste invece nell'abbandonare l'utilizzo delle *bag-of-words* in favore dell'approccio *word embedding*.

6.4.1 Approccio misto statistico e *rule-based*

Si riconsideri ora il Paragrafo 1.5, dove vengono esposti i due approcci al text mining clinico: *rule-based* e statistico; questa tesi si è concentrata sul secondo approccio descritto, ossia quello prettamente statistico. Per quanto questo approccio sembri concettualmente distante da quello *rule-based*, si può adottare una combinazione tra i due in modo da sfruttarne i rispettivi vantaggi.

Per comprendere il funzionamento di questo approccio, si prenda come esempio il lavoro di Kovačević *et al.* (2013): in questo caso gli autori estraggono, da testi contenuti in cartelle cliniche, la presenza di alcuni eventi accaduti ai pazienti (problematicità in fase di cura, test diagnostici o trattamenti eseguiti) affiancando un modello di *machine learning* a un sistema basato su regole prestabilite (Figura 6.2). Per prima cosa, nella fase di *preprocessing* dei testi vengono impiegati due software: cTAKES (`http://ctakes.apache.org`) e il già citato MetaMap. cTAKES è in grado di riconoscere il ruolo delle singole parole all'interno della frase (*part-of-speech tagging*), in particolare lavora sulla presenza di negazioni (*negation detection*). MetaMap, invece, associa ogni termine a un determinato concetto medico grazie all'uso di un dizionario medico integrato: in questo modo vengono identificati i sinonimi, gli errori ortografici e gli acronimi. Di conseguenza le variabili estratte dal testo tramite le *bag-of-words* sono di numero minore e più rappresentative dell'informazione contenuta nei testi, quindi la *term-document matrix* avrà una dimensionalità più contenuta e sarà meno sparsa. Successivamente, viene eseguita una classificazione supervisionata utilizzando un modello di *machine learning* noto come *conditional random fields* (Lafferty *et al.*, 2001), parallelamente viene definito un set di regole che, sfruttando nuovamente dei dizionari clinici, rende la classificazione più robusta. Le regole generate dal modello e quelle stabilite "a priori" vengono integrate a vicenda e il risultato è una classificazione molto accurata. Altri esempi di approcci misti si possono trovare in Nassif *et al.* (2009), Liu *et al.* (2012) e Aalabdulsalam *et al.* (2018), per un trattato sull'utilizzo di MetaMap e cTAKES si veda invece Reátegui e Ratté (2018).

Questa prima strada per incrementare la precisione della procedura di *text mining* clinico non è attualmente perseguibile per i testi di lingua italiana: software

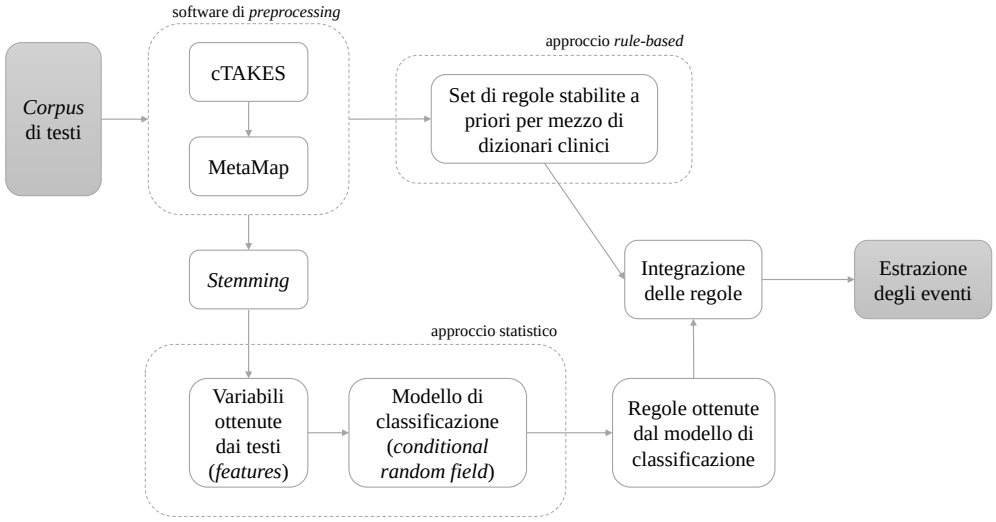

Figura 6.2: Schema dell'approccio misto seguito in Kovačević *et al.* (2013)

quali cTAKES o MetaMap non sono disponibili in altre lingue oltre l'inglese. Un tentativo di adattare MetaMap a testi di lingua italiana è stato condotto in Chiaramello *et al.* (2016) traducendo i testi dall'italiano all'inglese con un traduttore automatico, ma non ha portato a risultati soddisfacenti. Se in futuro venisse sviluppato un software con un dizionario clinico integrato anche per la lingua italiana probabilmente l'approccio misto porterebbe a una maggiore efficienza rispetto al singolo approccio statistico.

Un'altro modo per incrementare il risultato della procedura di *text mining* facendo uso di informazioni a priori è l'aggiunta di variabili esogene al testo. Per esempio, si può supporre che i testi in esame siano stati scritti da un certo numero di medici e che ogni medico abbia uno stile di scrittura differente. Creando un indicatore relativo al medico e includendolo nella matrice di regressione si aggiunge una variabile che, interagendo con quelle estratte dai testi, potrebbe aumentare l'accuratezza in fase di classificazione.

6.4.2 Approccio *word embedding*

Rimanendo all'interno dell'ambito statistico, è possibile abbandonare le *bag-of-words* come mezzo per trasformare il testo in variabili statistiche e adottare invece un approccio basato sulla vettorizzazione delle parole, detto *word embedding*. L'approccio *word embedding* è stato introdotto in Milkolov *et al.* (2013) e costituisce lo stato dell'arte della ricerca sul *text mining*. Lo scopo è ottenere una rappresentazione più compatta ed efficiente rispetto alla *document-term matrix* ottenuta con le *bag-of-words*, rappresentando le parole di un testo come dei vettori reali densi (non sparsi) di piccole dimensioni (solitamente nell'ordine delle decine o centinaia) detti "*word vectors*".

Utilizzando le *bag-of-words* non si considera il significato delle parole ma solo la loro presenza all'interno di un testo, è dunque impossibile stabilire la somiglianza fra due termini dal punto di vista del loro significato. I *word vectors*, al contrario, sono dei vettori che puntano ad associare le parole tra loro: parole più simili avranno dei vettori più vicini. La distanza che viene utilizza per esprimere questa "vicinanza" è quella basata sul coseno, o *cosine similarity*, definita come:

$$sim(A, B) = \cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||},$$

con θ angolo tra i vettori A e B. Per esempio, le parole "melanoma" e "metastasi" avranno dei *word vectors* vicini tra loro rispetto a "melanoma" e "aspirapolvere", poiché le prima coppia di parole condivide lo stesso contesto (l'oncologia), mentre la seconda no.

Ogni dimensione dei *word vectors* è il valore che le parole assumono rispetto a una variabile latente che rappresenta un certo concetto comune alle parole in esame. Detto in altri termini, le parole sono dei vettori dove ogni componente cattura una dimensione del significato di quella parola. Per fare un altro esempio: date le parole "regina", "re" e "suddito" si sceglie di considerare due variabili latenti: lo stato sociale (alto per "regina" e "re", basso per "suddito") e il genere (femminile per "regina", maschile per "re" e "suddito"). "Regina" e "re" avranno quindi un valore simile nella prima componente del vettore, quella legata allo stato sociale, mentre "suddito" avrà una componente distante dalla loro. Al contrario, "re" e "suddito"

avranno un valore simile nella seconda componente del vettore, legata al genere, ma distante da quello di "regina".

Il problema diventa dunque la scelta di quali e quante variabili latenti utilizzare. Nella libreria per il *word embedding* più utilizzata, **word2vec** (Mikolov, 2013), il problema viene affrontato stimando una rete neurale a due strati il cui *input* è il *corpus* di testi e l'*output* sono le variabili latenti. La rete costruisce un vocabolario con tutti i termini contenuti nei testi e poi crea i *word vectors* di una parola basandosi sulla ricorrenza della stessa all'interno delle frasi.

Sicuramente il *word embedding* costituisce un approccio più difficile dal punto di vista matematico e più oneroso dal punto di vista computazionale delle *bag-of-words*, cionondimeno i risultati sono promettenti e un suo sviluppo nel *text mining* clinico potrebbe essere una svolta per l'efficienza dell'intera procedura. Per una spiegazione esaustiva dell'approccio *word embedding* si vedano i lavori di Milkolov *et al.* (2013) e Pennington *et al.* (2014), per le implementazioni in R si veda il Capitolo 5 di Kwarter (2017) e il Capitolo 6 di Chollet e Allaire (2018), per un adattamento al contesto clinico si veda Wu *et al.* (2015). Se invece si cerca un'introduzione semplice al *word embedding* senza entrare nei dettagli matematici, si vedano "Introduction to Word Embedding and Word2Vec" (`http://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa`), "Introduction to Word Embeddings" (`http://hunterheidenreich.com/blog/intro-to-word-embeddings`) e "Introduction to Word Vectors" (`http://medium.com/@jayeshbahire/introduction-to-word-vectors-ea1d4e4b84bf`).

6.5 Riepilogo del capitolo

Le classificazioni ottenute tramite il modello *gradient boosting* presentano un errore di classificazione ancora troppo alto per permettere all'intera procedura di *text mining* di essere utilizzata in una struttura sanitaria quale un registro tumori. Nonostante ciò, questi errori risultano simili a quelli ottenuti in diversi lavori presenti in letteratura.

Osservando le matrici di errata classificazione si deduce che l'errore sia causato, in buona parte, da due fattori: lo sbilanciamento degli *outcome* e la difficoltà nel classificare alcuni testi, soprattutto quelli appartenerti allo Stadio X. Questo fenomeno è accentuato per l'*outcome* relativo alla presenza di metastasi. Per questo motivo le tre classificazioni sono state ripetute escludendo lo Stadio X. Fatto ciò, l'errore di classificazione cala, sia come conseguenza della rimozione di una classe, sia come conseguenza dell'esclusione delle unità più difficilmente classificabili. Inoltre, la classificazione ottenuta è di tipo dicotomico, quindi è possibile spostare il *cut-off* in modo da favorire una massima sensibilità, obiettivo da preferire in campo clinico.

Le conclusioni tratte alla fine di questo lavoro di tesi sono tre: la fattibilità della procedura di *text mining* clinico con approccio statistico supervisionato per testi in lingua italiana, la scelta del *gradient boosting* come miglior modello per la fase di classificazione e i limiti di utilizzo pratico legati a un errore ancora troppo alto.

Si espongono infine alcuni possibili sviluppi per migliorare l'intera procedura. Il primo è l'utilizzo di un approccio misto statistico e *rule-based*, il quale però fa uso di alcuni software con dizionari medici integrati. Questi software sono in grado di rintracciare abbreviazioni, accorpare sinonimi e correggere errori ortografici, dunque riducono la complessità e il rumore dei testi. Sfortunatamente, non esistono ancora adattamenti per la lingua italiana, quindi per il momento questa strada non è perseguibile. Il secondo è un approccio consiste nell'abbandonare la tecnica *bag-of-words* in favore del *word embedding* per estrarre con più efficienza le variabili dai testi. Questo approccio rappresenta lo stato dell'arte nel campo *text mining* e, specialmente se combinato con le reti neurali, potrebbe garantire una svolta per problemi come quello esposto in questa tesi.

Bibliografia

Aalabdulsalam, Abdulrahman K et al. (2018). «Automated Extraction and Classification of Cancer Stage Mentions from Unstructured Text Fields in a Central Cancer Registry». In: *AMIA Summits on Translational Science Proceedings*, pp. 16–25.

Aggarwal, Charu C e ChengXiang Zhai, cur. (2012). *Mining text data*. Springer Science e Business Media.

AIRTUM (2017). *I numeri del cancro in Italia*. Il pensiero scientifico editore.

Alicante, Anita et al. (2016). «Unsupervised entity and relation extraction from clinical records in Italian». In: *Computers in Biology and Medicine* 72, pp. 263–275.

Allvin, Helen et al. (2011). «Characteristics of Finnish and Swedish intensive care nursing narratives: a comparative analysis to support the development of clinical language technologies». In: *Journal of Biomedical Semantics* 2.3, pp. 1–11.

Angelova, Galia, Svetla Boytcheva e Ivelina Nikolova (2017). «Mining Association Rules from Clinical Narratives». In: *Proceedings of Recent Advances in Natural Language Processing*, pp. 130–138.

Azzalini, Adelchi e Bruno Scarpa (2012). *Data Analysis and Data Mining*. Oxford University Press.

Balch, Charles M et al. (2001). «Final version of the American Joint Committee on Cancer staging system for cutaneous melanoma». In: *Journal of Clinical Oncology* 19.16, pp. 3635–3648.

Breiman, Leo (1996). «Bagging predictors». In: *Machine Learning* 24.2, pp. 123–140.

— (1997). *Arcing the edge*. Rapp. tecn. Statistics Department, University of California.

— (2001). «Random Forests». In: *Machine Learning* 45.1, pp. 5–32.

Breiman, Leo et al. (1984). «Classification and regression trees». In: *Wadsworth International Group*.

Ceron, Andrea, Luigi Curini e Stefano Maria Iacus (2014). *Social Media e Sentiment Analysis: L'evoluzione dei fenomeni sociali attraverso la Rete*. Vol. 9. Springer Science & Business Media.

Chaovalit, Pimwadee e Lina Zhou (2005). «Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches». In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, p. 112.

Chawla, Nitesh V (2003). «C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure». In: *Proceedings of the ICML*. Vol. 3, p. 66.

Chawla, Nitesh V, Nathalie Japkowicz e Aleksander Kotcz (2004). «Special issue on learning from imbalanced data sets». In: *ACM Sigkdd Explorations Newsletter* 6.1, pp. 1–6.

Chen, Tianqi e Carlos Guestrin (2016). «Xgboost: A scalable tree boosting system». In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, pp. 785–794.

Chen, Yangchi, M.M. Crawford e J. Ghosh (2004). «Integrating support vector machines in a hierarchical output space decomposition framework». In: *IEEE International Geoscience and Remote Sensing Symposium*. Vol. 2, pp. 949–952.

Chiaramello, Emma et al. (2016). «Attempting to Use MetaMap in Clinical Practice: A Feasibility Study on the Identification of Medical Concepts from Italian Clinical Notes». In: *Studies in health technology and informatics* 228, pp. 28–32.

Chollet, François e Joseph J Allaire (2018). *Deep Learning with R*. Manning Publications Company.

Cieslak, David A e Nitesh V Chawla (2008). «Learning decision trees for unbalanced data». In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 241–256.

Cortes, Corinna e Vladimir Vapnik (1995). «Support-vector networks». In: *Machine Learning* 20.3, pp. 273–297.

Dalianis, Hercules (2018). *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer, p. 192.

Ehrentraut, Claudia et al. (2012). «Detection of Hospital Acquired Infections in sparse and noisy Swedish patient records». In: *Sixth Workshop on Analytics for Noisy Unstructured Text Data*, pp. 1 –8.

Fawcett, Tom (2006). «An introduction to ROC analysis». In: *Pattern Recognition Letters* 27.8, pp. 861–874.

Feinerer, Ingo (2018). *Introduction to the tm Package*.

Feinerer, Ingo, Kurt Hornik e David Meyer (2008). «Text Mining Infrastructure in R». In: *Journal of Statistical Software* 25.5.

Freund, Yoav e Robert E Schapire (1996). «Experiments with a new boosting algorithm». In: *Icml* 96, pp. 148–156.

Friedman, Jerome H (2001). «Greedy function approximation: a gradient boosting machine». In: *Annals of statistics*, pp. 1189–1232.

— (2002). «Stochastic gradient boosting». In: *Computational Statistics & Data Analysis* 38.4, pp. 367–378.

Gallo, Pietro e Giulia D'Amanti (2018). *Anatomia patologica. La sistematica*. Edra.

Goodfellow, Ian, Yoshua Bengio e Aaron Courville (2016). *Deep learning*. Vol. 1. MIT press Cambridge.

Hanauer, David A. et al. (2007). «The Registry Case Finding Engine: An Automated Tool to Identify Cancer Cases from Unstructured, Free-Text Pathology Reports and Clinical Notes». In: *Journal of the American College of Surgeons* 205.5, pp. 690–697.

Hastie, Trevor, Robert Tibshirani e Jerome H Friedman (2013). *The Elements of Statistical Learning*. Springer series in statistics. New York: Springer.

Hearst, Marti A. (1999). «Untangling text data mining». In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pp. 3–10.

Hornik, Kurt, Maxwell Stinchcombe e Halbert White (1989). «Multilayer feedforward networks are universal approximators». In: *Neural networks* 2.5, pp. 359–366.

Italiano, Irene (2018). *Progetto per la registrazione ad alta risoluzione del melanoma cutaneo*. Rapp. tecn. Istituo Oncologico Veneto.

Jivani, Anjali Ganesh (2011). «A comparative study of stemming algorithms». In: *Int. J. Comp. Tech. Appl* 2.6, pp. 1930–1938.

Jurafsky, Dan e James H Martin (2008). *Speech and language processing*. Pearson London.

Kovačević, Aleksandar et al. (2013). «Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives». In: *Journal of the American Medical Informatics Association* 20.5, pp. 859–866.

Kwartler, Ted (2017). *Text mining in practice with R*. John Wiley & Sons.

Lafferty, J., A. McCallum e F. Pereira (2001). «Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data». In: *Proc. of ICML* 2001.June, pp. 282–289.

Liu, Hongfang et al. (2012). «Clinical decision support with automated text processing for cervical cancer screening». In: *Journal of the American Medical Informatics Association* 19.5, pp. 833–839.

Lovins, Julie Beth (1968). «Development of a stemming algorithm». In: *Mech. Translat. & Comp. Linguistics* 11.1-2, pp. 22–31.

Martinez, David, Lawrence Cavedon e Graham Pitson (2013). «Stability of text mining techniques for identifying cancer staging». In: *Louhi, The 4th International Workshop on Health Document Text Mining and Information Analysis*.

McCowan, Iain, Darren Moore e Mary-Jane Fry (2006). «Classification of cancer stage from free-text histology reports». In: *Engineering in Medicine and Biology Society*, pp. 5153–5156.

McCowan, Iain et al. (2007). «Collection of Cancer Stage Data by Classifying Free-text Medical Reports». In: *Journal of the American Medical Informatics Association* 14.6, pp. 736–745.

Menardi, Giovanna e Nicola Torelli (2014). «Training and assessing classification rules with imbalanced data». In: *Data Mining and Knowledge Discovery* 28.1, pp. 92–122.

Mikolov, Tomas et al. (2013). «Efficient Estimation of Word Representations in Vector Space». In: pp. 1–12.

Miner, Gary, John Elder IV e Thomas Hill (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press.

Napolitano, Giulio et al. (2010). «Pattern-based information extraction from pathology reports for cancer registration». In: *Cancer Causes & Control* 21.11, pp. 1887–1894.

Nassif, Houssam et al. (2009). «Information Extraction for Clinical Data Mining: A Mammography Case Study.» In: *International Conference on Data Mining*, pp. 37–42.

Nguyen, Anthony N. et al. (2007). «Multi-class Classification of Cancer Stages from Free-text Histology Reports using Support Vector Machines». In: *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5140–5143.

O'Mara-Eves, Alison et al. (2015). «Using text mining for study identification in systematic reviews: a systematic review of current approaches». In: *Systematic Reviews* 4.1, p. 5.

Pakhomov, Serguei, Ted Pedersen e Christopher G Chute (2005). «Abbreviation and Acronym Disambiguation in Clinical Discourse». eng. In: *AMIA Annual Symposium Proceedings*. Vol. 2005, pp. 589–593.

Patrick, Jon e Dung Nguyen (2011). «Automated proof reading of clinical notes». In: *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*.

Pennington, Jeffrey, Richard Socher e Christopher Manning (2014). «Glove: Global vectors for word representation». In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Porter, Martin F (1980). «An algorithm for suffix stripping». In: *Program* 14.3, pp. 130–137.

Powers, David (2007). «Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation». In: *Journal of Machine Learning Technologies* 2.December, p. 24.

Pratt, Arnold W e Milos G Pacak (1969). «Automated processing of medical English». In: *Proceedings of the 1969 conference on Computational linguistics*. Association for Computational Linguistics, pp. 1–23.

Ramos, Juan (2003). «Using tf-idf to determine word relevance in document queries». In: *Proceedings of the first instructional conference on machine learning*. Vol. 242, pp. 133–142.

Reátegui, Ruth e Sylvie Ratté (2018). «Comparison of MetaMap and cTAKES for entity extraction in clinical notes». In: *BMC Medical Informatics and Decision Making* 18.3, p. 74.

Ripley, Brian D (2007). *Pattern recognition and neural networks*. Cambridge university press.

Spasic, Irena et al. (2014). «Text mining of cancer-related information: review of current status and future directions.» In: *International journal of medical informatics* 83.9, pp. 605–623.

Srivastava, Nitish et al. (2014). «Dropout: a simple way to prevent neural networks from overfitting». In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Stehman, Stephen V. (1997). «Selecting and interpreting measures of thematic classification accuracy». In: *Remote Sensing of Environment* 62.1, pp. 77–89.

Tan, Songbo (2005). «Neighbor-weighted k-nearest neighbor for unbalanced text corpus». In: *Expert Systems with Applications* 28.4, pp. 667–671.

Velupillai, Sumithra e Maria Kvist (2012). «Fine-grained certainty level annotations used for coarser-grained e-health scenarios». In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 450–461.

Warrer, Pernille et al. (2012). «Using text-mining techniques in electronic patient records to identify ADRs from medicine use». In: *British Journal of Clinical Pharmacology* 73.5, pp. 674–684.

Wolpert, David H e William G MacReady (1999). «An Efficient Method To Estimate Bagging's Generalization Error». In: *Machine Learning* 35.1, pp. 41–55.

Wu, Yonghui et al. (2015). «A Study of Neural Word Embeddings for Named Entity Recognition in Clinical Text». In: *AMIA Symposium*. Vol. 2015. American Medical Informatics Association, pp. 1326–1333.

Zhang, Yin, Rong Jin e Zhi-Hua Zhou (2010). «Understanding bag-of-words model: a statistical framework». In: *International Journal of Machine Learning and Cybernetics* 1.1-4, pp. 43–52.

Zhou, Xiaohua et al. (2006). «Approaches to text mining for clinical medical records». In: *Proceedings of the 2006 ACM symposium on Applied computing*. August 2014, p. 235.

Zhu, Fei et al. (2013). «Biomedical text mining and its applications in cancer research». In: *Journal of Biomedical Informatics* 46.2, pp. 200–211.

Sitografia

"Registro Tumori Veneto." www.registrotumoriveneto.it. [Consultato: 06-Nov-2018].

"Associazione Italiana Registri Tumori." www.registri-tumori.it. [Consultato: 06-Nov-2018].

"Ministero della Salute." www.salute.gov.it. [Consultato: 10-Nov-2018].

"Istituto Europeo di Oncologia." www.ieo.it. [Consultato: 12-Nov-2018].

"American Cancer Society." www.cancer.org. [Consultato: 07-Dic-2018].

"Snowball." www.snowballstem.org. [Consultato: 17-Dic-2018].

J. B. Ahire, "Introduction to Word Vectors." https://medium.com. [Consultato: 14-Feb-2019].

H. Heidenreich, "Introduction to Word Embeddings." http://hunterheidenreich.com. [Consultato: 11-Feb-2019].

D. Karani, "Introduction to Word Embedding and Word2Vec." https://towardsdatascience.com. [Consultato: 14-Feb-2019].

Appendice A

Codice R

In questa appendice è contenuto il codice R utilizzato. Per semplicità è riportato solo il codice relativo all'*outcome* legato alla stadiazione TNM: il codice per gli altri *outcome* ne è una semplice ripetizione.

```
#Carica alcune librerie
library("Cairo")
library("MASS")
library("Rcpp")
library("class")
library("caret")
library("e1071")
library("nnet")
library("tree")
library("randomForest")
library("tm")
library("tau")
library("TextWiller")
library("stringr")
library("dplyr")
library("tidytext")
```

```
library("tidyr")
library("tidytext")
library("haven")
library("tibble")
library("devtools")
library("iSAX")
library("Matrix")
library("topicmodels")
library("quanteda")
library("ada")
library("xgboost")
library("glue")
library("keras")
library("lsa")
library("readr")
library("keras")
library("reticulate")
library("purrr")
library("text2vec")
library("tidyr")

#Carica alcune funzioni
source("lift_roc_tab.R") #funzioni messe a dispisizione dal prof. Bruno Scarpa
    per l'eseme di Data Mining
source("funzioni_per_esame.R") #alcune funzioni scritte da Pietro Belloni e
    Sara Cozzolino

dati <- read.csv("dati.csv", he=T)[,-1] #in questo dataset sono contenuti i
    testi e i gold standard gia' associati
more_stopwords <- read.csv("stopwords.csv", he=F, encoding = "UTF-8") #carica
    la lista di stopwords personalizzate (appendice B)
my_stopwords <- c(1:NCOL(more_stopwords))
for(i in 1:NCOL(more_stopwords)){
```

```
        my_stopwords[i] <- sub(pattern = " ", replacement = "", x = as.
            character(more_stopwords[1,i]))
}


# Preprocessing ------------------------------------------------------------

#Codifica NA nella diagnosi
for (i in 1:nrow(dati)) {
        if(dati$diagnosi[i]==""|
                dati$diagnosi[i]=="NC"|
                dati$diagnosi[i]==" \n"|
                dati$diagnosi[i]=="\n\n"|
                dati$diagnosi[i]=="\n\n\n"|
                dati$diagnosi[i]=="N/A"){
                dati$diagnosi[i] <- NA
        }
}

#Rimozione delle diagnosi il cuo testo e' vuoto
naindex <- which(is.na(dati$diagnosi))
dati <- dati[-naindex,]

#Alcune statistiche sul numero di caratteri dei testi
nc <- nchar(as.character(dati$diagnosi), type = "chars")
summary(nc)
sd(nc)

#Rimozione stopwords
stopw1 <- normalizzaTesti(as.vector(dati$diagnosi)) #rimuove stopita default
for (i in 1:length(my_stopwords)) { #rimuove le stopwords personalizzate
        stopw1 <- gsub(pattern = my_stopwords[i], replacement = "", x = stopw1)
}
```

```
#Alcune correzioni manuali
stopw1 <- gsub(pattern = "ccparametri ", replacement = "parametri ", x = stopw1
    )
stopw1 <- gsub(pattern = "cdiagnosi ", replacement = "diagnosi ", x = stopw1)
stopw1 <- gsub(pattern = "cindice ", replacement = "indice ", x = stopw1)
stopw1 <- gsub(pattern = "cmargini ", replacement = "margini ", x = stopw1)
stopw1 <- gsub(pattern = "ccparametri ", replacement = "parametri ", x = stopw1
    )
stopw1 <- gsub(pattern = "cmelanoma ", replacement = "melanoma ", x = stopw1)
stopw1 <- gsub(pattern = "cspessore ", replacement = "spessore ", x = stopw1)
stopw1 <- gsub(pattern = "fcmelanoma ", replacement = "melanoma ", x = stopw1)
stopw1 <- gsub(pattern = "fframmento ", replacement = "frammento ", x = stopw1)
dati$diagnosi_stopw <- stopw1

#Creazione corpus:
corpus_diagnosi <- VCorpus(VectorSource(dati$diagnosi_stopw), readerControl =
    list(language="italian"))

#Creazione document-term matrix:
dtm <- DocumentTermMatrix(corpus_diagnosi,
        control = list(weighting=function(x) weightTfIdf(x, normalize = FALSE),
        stemming = T, readerControl = list(language="italian"),
        stopwords = F, minWordLength = 2,
        removeNumbers = T, removePunctuation = T,
        bounds=list(local = c(1,Inf))))
inspect(dtm)
dtm <- removeSparseTerms(dtm, sparse = 0.99) #riduzione della dimensionalita'
inspect(dtm)
dtm$dimnames$Terms #316 stilemi
regmat <- as.matrix(dtm) #conversione a formato matriciale

#Trova le associazioni tra parole:
```

```
findAssocs(dtm, "ulcer", 0.5)
findAssocs(dtm, "malign", 0.4)

#Calcolo pesi if-idf:
tdm <- TermDocumentMatrix(corpus_diagnosi,
        control = list(weighting=function(x) weightTfIdf(x, normalize = FALSE),
        stemming = T, readerControl = list(language="italian"),
        stopwords = F, minWordLength = 2,
        removeNumbers = T, removePunctuation = T,
        bounds=list(local = c(1,Inf))))
inspect(tdm)
tdm <- removeSparseTerms(tdm, sparse = 0.99)
inspect(tdm)
tidy_is <- tidy(tdm) #converte in oggetto tidy
# frequenze termini decrescenti
tidy_is %>%
count(term, sort = TRUE)
#alcune operazioni di conteggio e calcolo tf-idf
document_words <- tidy_is %>%
        count(document, term, sort = TRUE) %>%
        ungroup()
document_words <- document_words %>%
        bind_tf_idf(term, document, n)
total_words <- document_words %>%
        group_by(document) %>%
summarize(total = sum(n))
document_words <- left_join(document_words, total_words)
document_words #tf-idf totali
summary(document_words$tf_idf)
#termini con il piu' alto tf-idf
document_words %>%
        select(total) %>%
        arrange(desc(tf_idf)) %>%
```

```
top_n(15) #(primi 15 termini)

# Data frame per i modelli, aggiunta della variabile risposta
#(qui per l'outcome TNM, da modificare per gli altri autome)
regmat <- as.data.frame(regmat)
regmat$TNM <- dati$TNM

#Aggiunta bigrammi alla matrice di regressione
bigrams <- textcnt(dati$diagnosi_stopw, method="string", n=2L, split="[[:blank
    :]]")
sortedbigrams <- data.frame(freq=c(sort(bigrams, decreasing=TRUE)[1:424])) #
    questi sono i bigrammi che compaiono in almeno l'1% dei testi
sortedbigrams <- rownames_to_column(df=sortedbigrams, var = "bigrams")
regmat_bi <- matrix(0, nrow = NROW(regmat), ncol = NROW(sortedbigrams))
colnames(regmat_bi) <- sortedbigrams$bigrams
regmat_bi <- as.data.frame(regmat_bi)
for (i in 1:NROW(sortedbigrams)) {
        regmat_bi[grep(pattern = sortedbigrams[i,1], x = dati$diagnosi_stopw),i
            ] <- 1
}
colnames(regmat_bi) <- gsub(pattern=" ", x = sortedbigrams$bigrams, replacement
     = "_")
regmat <- cbind(regmat, regmat_bi)
regmat <- na.omit(regmat)



# Trattamento dati ----------------------------------------------------------

#Divisione insiemi di stima e verifica
set.seed(42)
acaso <- sample(1:NROW(regmat), 1500)
stima <- regmat[acaso,]
verifica <- regmat[-acaso,]
```

```
#Aggiunta pesi per i modelli
table(regmat$TNM)/NROW(regmat)
#     I     II    III     IV XorABS     TOT
#  1759    293    270     52    171    2545
prior <- 1-c(1759, 293, 270, 52, 171)/NROW(regmat)
prior <- prior/4
names(prior) <- c("I", "II", "III", "IV", "XorABS")
prior
pesi <- rep(1, NROW(stima))
pesi[stima$TNM=="I"] <- prior[1]
pesi[stima$TNM=="II"] <- prior[2]
pesi[stima$TNM=="III"] <- prior[3]
pesi[stima$TNM=="IV"] <- prior[4]
pesi[stima$TNM=="XorABS"] <- prior[5]



# Albero ------------------------------------------------------------------

m.alb <- tree(TNM~. ,
        data = stima,
        split=c("deviance"),
        weights=pesi, #bilancimaneto
        control = tree.control(nobs = NROW(stima),
                minsize = 2,
                mindev = 0.02))
plot(m.alb)
text(m.alb)
prune <- prune.tree(m.alb, newdata = verifica)
plot(prune)
J <- prune$size[which.min(prune$dev)]
m.alb.scelto <- prune.tree(m.alb, best=J)
plot(m.alb.scelto) #grafico albero
```

```
text(m.alb.scelto, cex=0.5)
p.alb <- predict(m.alb.scelto, newdata = verifica, type="class")
tabella.sommario(p.alb, verifica$TNM) #matrice di confusione



# Random forest ------------------------------------------------------------

m.rf <- randomForest(TNM~.,
        data = stima,
        nodesize = 1,
        classwt=prior,
        mtry = 44,
        ntree = 200)
plot(m.rf)
p.rf <- predict(m.rf, newdata = verifica, type = "response")
tabella.sommario(p.rf, verifica$TNM) #matrice di confusione

#Selezione parametro mtry:
err.tot.rf <- c(1:11)
k <- 1
for (i in 41:51) {
        m.rf <- randomForest(TNM~.,
                data = stima,
                nodesize = 1,
                mtry = i, #numero di covariate che usa ogni volta
                ntree = 200, #numero totale di alberi
                classwt=prior)
        plot(m.rf)
        p.rf <- predict(m.rf, newdata = verifica, type = "response")
        matr.conf <-  table(p.rf, verifica$TNM)
        err.tot.rf[k] <- 1-sum(diag(matr.conf))/sum(matr.conf)
        k <- k+1
        print(paste("Numero di features usate:", i, "Errore di stima:", err.tot
```

```
            .rf[k], collapse = ""))
}
plot(41:51, err.tot.rf, type="l", col=2) #top=44



# xgboost ----------------------------------------------------------------

#Ricodifica delle variabili di outcome:
train_labs <- as.numeric(stima$TNM) - 1
val_labs <- as.numeric(verifica$TNM) - 1
new_train <- model.matrix(~ . + 0, data = stima[, -319])
new_val <- model.matrix(~ . + 0, data = verifica[, -319])

#Ricodifica delle matrici di regressione:
xgb_train <- xgb.DMatrix(data = new_train, label = train_labs)
xgb_val <- xgb.DMatrix(data = new_val, label = val_labs)

#Xgboost con convalida incrociata:
xgbcv <- xgb.cv(params = list(booster = "gbtree", objective = "multi:softprob",
    num_class = 5, eval_metric = "merror"),
        data = xgb_train,
        nrounds = 50, #iterazioni
        nfold = 20, #numero di fold della convalida
        showsd = TRUE,
        stratified = TRUE,
        print_every_n = 1,
        early_stop_round = 10, #eventualmente stoppa dopo 10 uguali
        maximize = FALSE,
        prediction = TRUE)
xgb_train_preds <- data.frame(xgbcv$pred) %>%
        mutate(max = max.col(., ties.method = "last"), label = train_labs + 1)
xgb_conf_mat <- table(true = train_labs + 1, pred = xgb_train_preds$max)
t(xgb_conf_mat) #matrice di confusione
```

```
classification_error <- function(conf_mat) {
        conf_mat = as.matrix(conf_mat)
        error = 1 - sum(diag(conf_mat)) / sum(conf_mat)
        return (error)
}
cat("XGB Training Classification Error Rate:",
classification_error(xgb_conf_mat), "\n") #errore di classificazione


# SVM ----------------------------------------------------------------------

m.svm <- svm(TNM~.,
        data=stima[,1:318],
        cost=5, #par C
        class.weights=prior,
        gamma=0.00143472 #par d)
p.svm <- predict(m.svm, newdata = verifica)
tabella.sommario(p.svm, verifica$TNM) #matrice di confusione

#Scelta parametri di regolazione:
svm_tune <- tune.svm(TNM~., data = stima, cost=5, gamma=c(0.5, 1, 2))
print(svm_tune)
svm_tune$best.model


# Deep NN ------------------------------------------------------------------

#Ricodifica dati per keras:
x_train <- as.matrix(stima[,-319])
y_train <- to_categorical(c(stima[,319]))
x_test <- as.matrix(verifica[,-319])
y_test <- to_categorical(c(verifica[,319]))
```

```
#Definzione rete neurale:
model <- keras_model_sequential() %>%
        layer_dense(units = 32, activation = 'relu', input_shape = c(ncol(x_
            train))) %>%
        layer_dropout(rate = 0.2) %>%
        layer_dense(units = 32, activation = 'relu') %>%
        layer_dropout(rate = 0.2) %>%
        layer_dense(units = ncol(y_train), activation = 'softmax')
summary(model) #rete neurale

#Definizione ottimizzatore e funzione di perdita:
model %>% compile(
        optimizer = optimizer_adam(),
        loss = loss_categorical_crossentropy,
        metrics = metric_categorical_accuracy)
set.seed(42)

#Stima:
history <- model %>% fit(
        x_train,
        y_train,
        epochs = 25,
        batch_size = 512,
        validation_data = list(x_test, y_test))
plot(history)

#Verifica:
results <- model %>% evaluate(x_test, y_test)
1-results$categorical_accuracy #errore di classificazione
```

Appendice B

Lista aggiuntiva di *stopwords*

In questa appendice è contenuta la lista di termini che si è aggiunta alle comuni *stopwords* della lingua italiana:

allegato, diagnosi, 00b0, 00b2, rtf1, ansi, ansicpg1252, uc1, deff0, fonttbl, f0, fswiss, fcharset0, fprq2, arial, f1, froman, fcharset0, fprq2, times, new, roman, f2, fswiss, fcharset0, fprq2, tahoma, f3, froman, fcharset2, fprq2, symbol, colortbl, red0, gren0, blue0, red255, gren255, blue255, styleshet, s0, itap0, nowidctlpar, fs24, normal, cs10, additive, default, paragraph, font, generator, tx, rtf32, deftab1134, paperw11905, paperh16838, margl794, margt0, margr1247, margb567, widowctrl, background, shp, shpinst, shpleft0, shptop0, shpright0, shpbottom0, shpfhdr0, shpbxmargin, shpbxignore, shpbymargin, shpbyignore, shpwr0, shpwrk0, shpfblwtxt1, shplid1025, sp, sn, shapetype, sv, ffliph, fflipv, fillcolor, 16777215, ffilled, linewidth, fline, fbackground, flayoutincell, pard, itap0, nowidctlpar, tx850, tx1700, tx2550, tx3400, tx4250, tx5100, tx5950, tx6800, tx7650, tx8500, tx9350, tx10200, tx11050, tx11900, plain, fs24, cf3, ftx720, tx1440, tx2160, tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, tx7200, tx7920, tx8640, tx9360, tx10080, fs20, fs20, tx720, tx1440, tx2160, tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, tx7200, tx7920, tx8640, tx9360, tx10080, fs20, tx1440, tx2160, tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, tx7200, tx7920, tx8640, tx9360, tx10080, fs20, tx1440, tx2160,

tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, tx7200, tx7920, tx8640, tx9360, tx10080, fs20, tx1440, tx2160, tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, tx7200, tx7920, tx8640, tx9360, tx10080, fs20, tx1440, tx2160, tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, tx7200, tx7920, tx8640, tx9360, tx10080, tx1440, tx2160, tx2880, tx3600, tx4320, tx5040, tx5760, tx6480, 7200, tx7920, tx8640, tx9360, tx10080, par, cpar, qc, 13, 501, defwidctls16, cs17, i0, sbasedon10, emphasis, fs16, margl1134, margt283, margr850, margb283, paperh16832, margl850, margt850, margr850, margb850, margl1134, margt283, margr850, margb283, paperw12240, paperh15840, margl1134, margt283, margr567, margb283, margr567, margb283, red160, gren160, blue164, margl1134, margt283, margr567, margb283, 10789024, red192, gren192, blue192, margl1134, margt283, margr567, margb283, 12632256, def, fwidctltx708, tx1416, tx2124, tx2832, tx3540, tx4248, tx4956, tx5664, tx6372, tx7080, tx7788, tx8496, tx9204, tx9912, fnil, margl0, margr0, margb0, formshade, sectd, headery720, fotery720, pgwsxn11905, pghsxn16838, marglsxn794, margtsxn0, margrsxn1247, margbsxn567, fprq0, fcharset1, f4, margl14, e0, fqj.

Appendice C

Grafici del modello *gradient boosting*

In questa appendice sono contenuti i grafici relativi all'importanza che assumono i singoli stilemi o bigrammi nella classificazione ottenuta con il modello *gradient boosting*. L'asse "*features*" indica il nome dello stilema o del bigramma, l'asse "*importance*" indica l'importanza assunta dallo stilema o dal bigramma in fase di classificazione. i colori indicano i *cluster* in cui le *features* sono state raggruppate in base alla loro importanza.

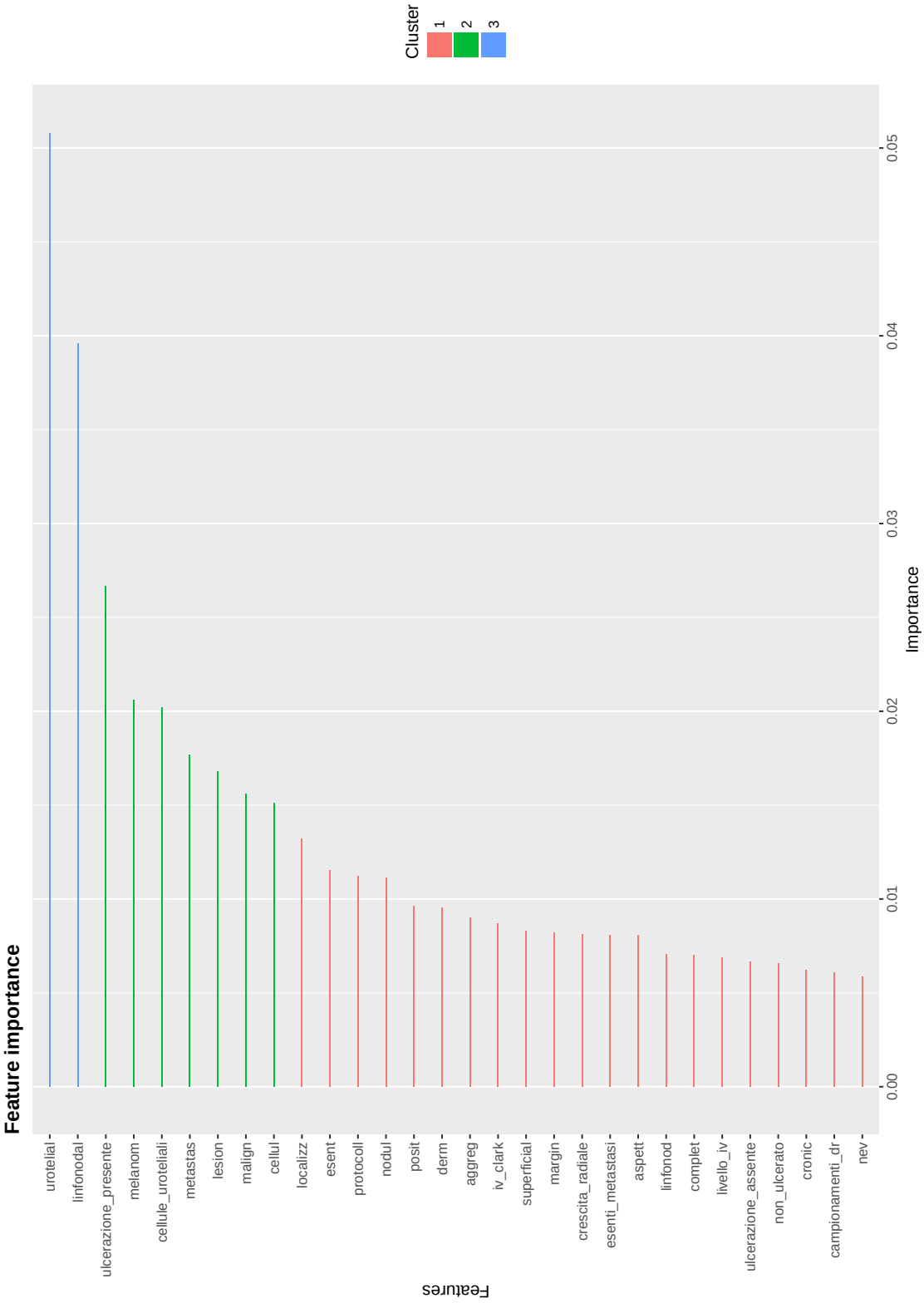

Figura C.1: *Features importance* per la stadiazione TNM

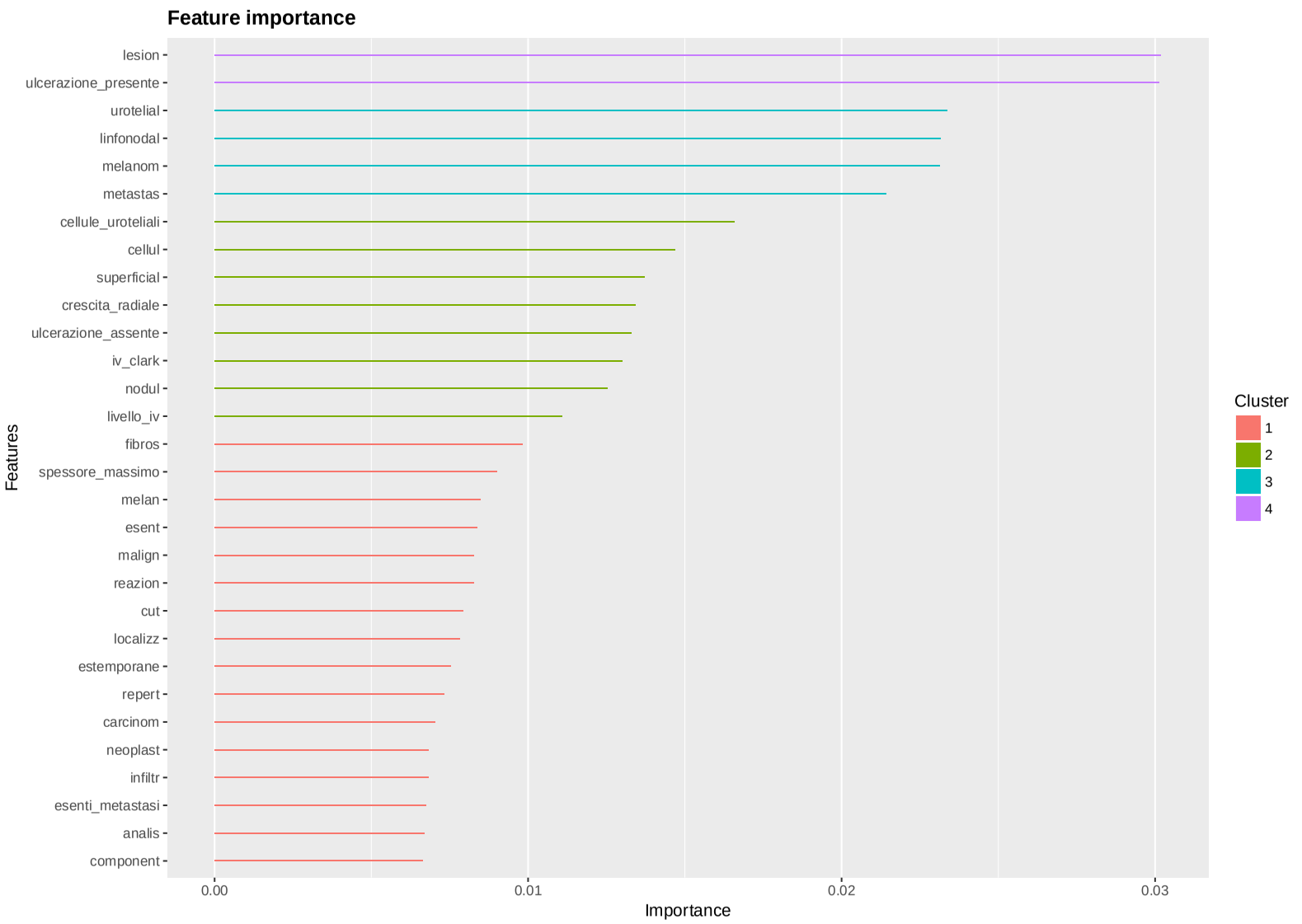

Figura C.2: *Features importance* per il parametro T

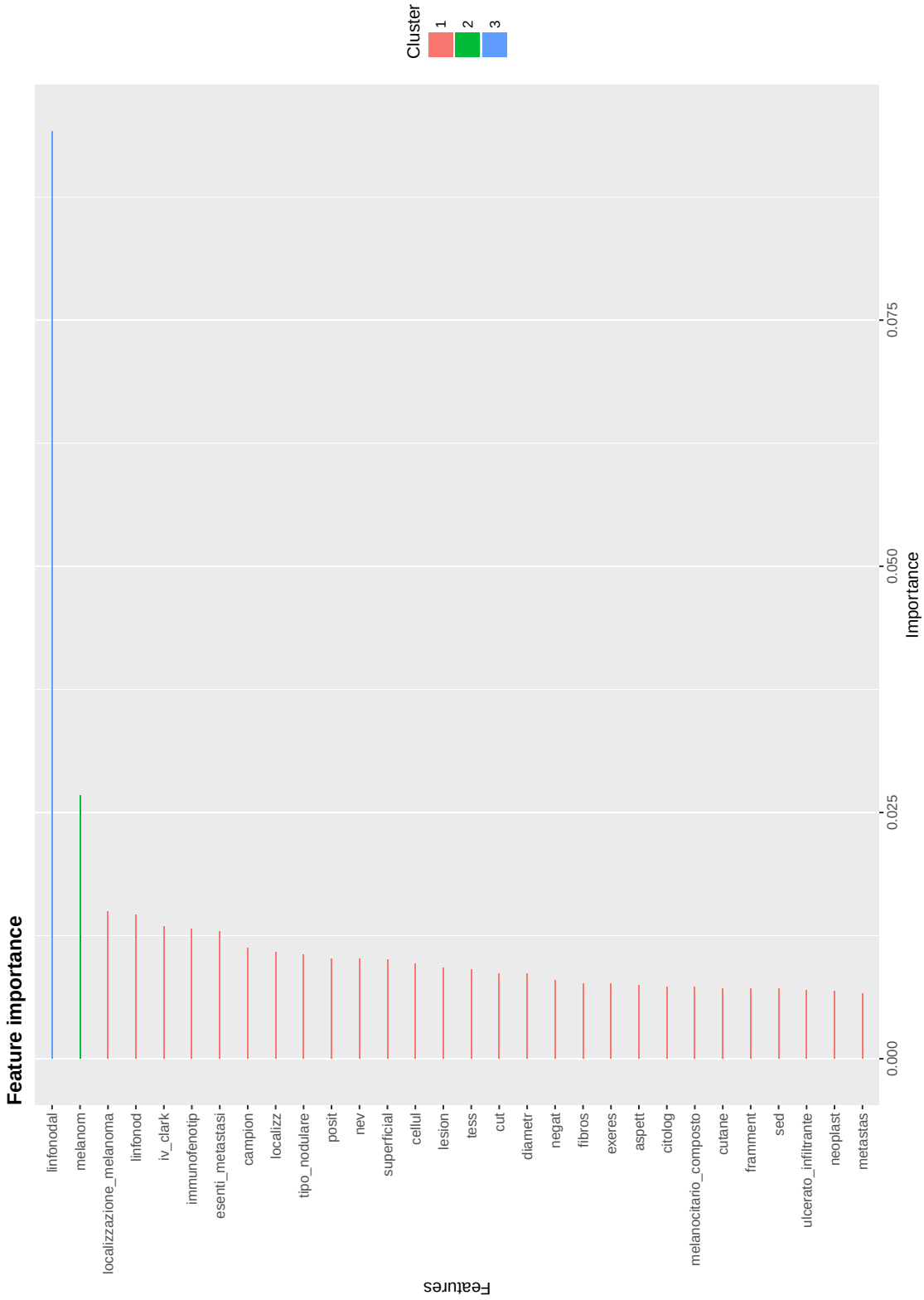

Figura C.3: *Features importance* per il parametro N

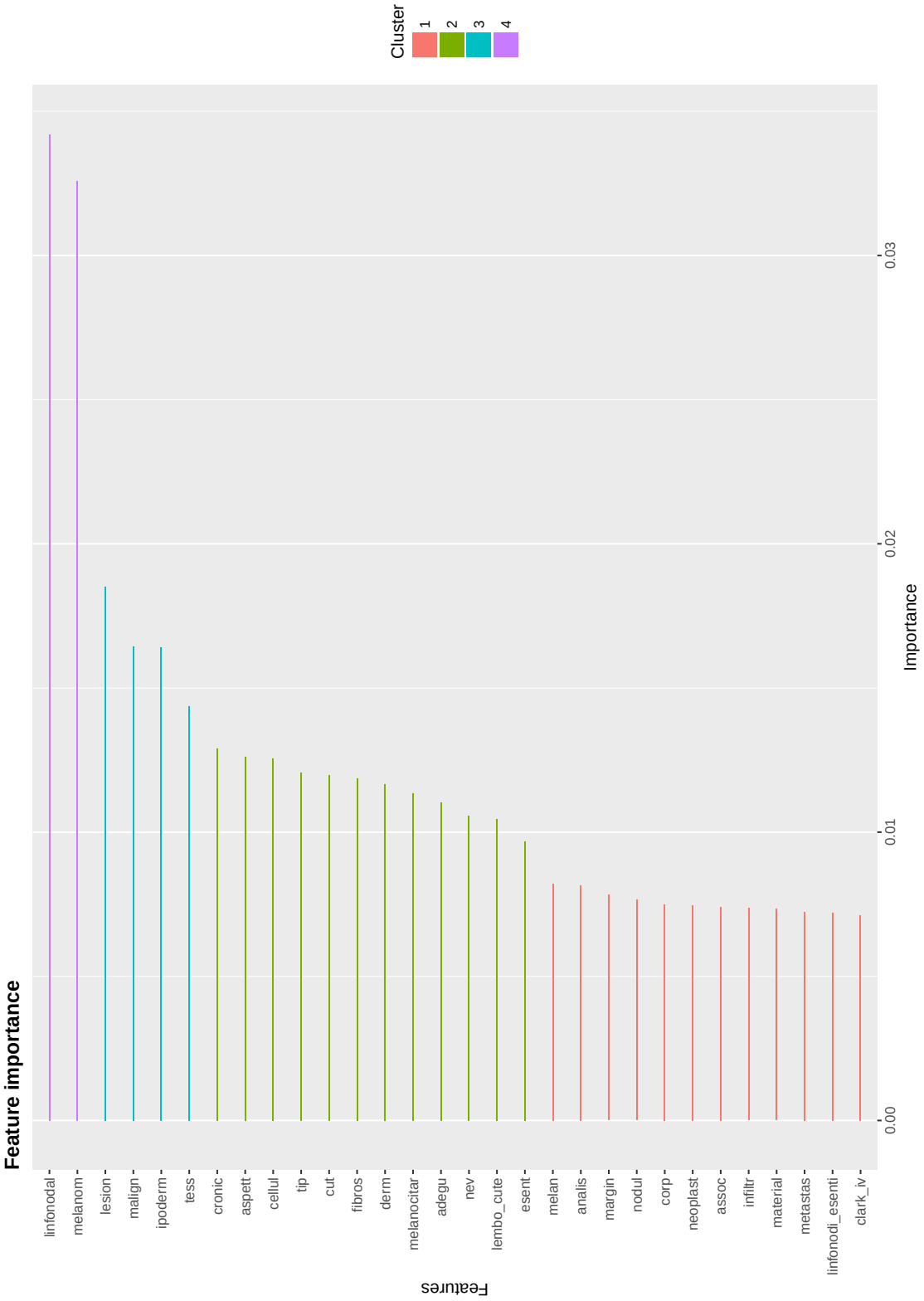

Figura C.4: *Features importance* per il parametro M

Ringraziamenti

Questo lavoro è il frutto di una collaborazione fra il Dipartimento di Scienze Statistiche dell'Università di Padova e il Registro Tumori del Veneto, parte dell'Azienda Zero; si è sviluppato nell'arco di circa 7 mesi e ha coinvolto diversi soggetti senza i quali non sarebbe stato possibile portarlo a termine, a ognuno di loro va un sentito ringraziamento. In primis, ci tengo a ringraziare la Prof.ssa Boccuzzo che ha ideato, coordinato e corretto l'intera tesi. Un ringraziamento anche al Prof. Scarpa per aver curato la parte relativa al *preprocessing* e alla modellistica, fornendo spunti e suggerimenti. Gran parte del lavoro è stato svolto presso il Registro Tumori, ci tengo dunque a ringraziare il Dott. Zorzi che non solo mi ha (letteralmente) ospitato all'interno dei suoi uffici, ma mi ha anche seguito con dedizione e pazienza per tutta la parte relativa agli aspetti clinici del melanoma e al funzionamento del Registro. Ringrazio con gratitudine anche tutti gli impiegati e le impiegate del Registro, in particolare il Dott. Guzzinati, che mi hanno accolto con grande professionalità fornendomi sempre tutto il supporto necessario. Il *gold standard* è stato fornito dal lavoro della Dott.ssa Italiano, alla quale va un altro sentito ringraziamento. Grazie anche al Prof. Sciandra, che per mezzo della sua grande esperienza nel campo del *text mining* ha saputo fornirmi dei preziosi consigli. Ringrazio infine tutti i professori e le professoresse delle università di Padova e Aarhus che mi hanno dato una lunga serie di strumenti senza i quali non sarei stato in grado di costruire questo lavoro di tesi.