UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

METODOLOGIE BASATE SUL MACHINE LEARNING PER LA PREDIZIONE DELLA GLICEMIA NELLA TERAPIA DEL DIABETE DI TIPO 1

Relatore: Prof. Andrea Facchinetti

Laureando: Alessandro Sandron

ANNO ACCADEMICO 2021/2022

Data di laurea: 20 settembre 2022

Sommario

Il diabete, una condizione metabolica cronica che rappresenta un onere sanitario globale in quanto risulta essere una delle malattie più diffuse. Si stima che il numero di persone diabetiche aumenterà notevolmente nei prossimi anni passando da un 8.8% ad un 10% della popolazione globale, causando un impatto dal punto di vista clinico, sociale ed economico. Questo elaborato si pone l'obbiettivo di descrivere due algoritmi utilizzati in dispositivi di monitoraggio glicemico per la generazione di allarmi di eventi glicemici.

Nel primo capitolo si andrà a descrivere la malattia diabetica, principalmente il diabete di tipo 1, soffermandosi sull'ambito patologico, sintomatologico e terapeutico. Si porrà particolare attenzione sui dispositivi per il monitoraggio in continua del glucosio, descrivendone pregi e difetti rispetto a dispositivi per l'automonitoraggio della glicemia precedentemente utilizzati.

Il secondo capitolo tratterà di Machine Learning, illustrando differenti aspetti, tra cui: varie metodologie che possono essere utilizzate per l'implementazione, le problematiche più frequenti che possono essere riscontrate durante la costruzione e diverse metriche per valutare l'efficienza predittiva dell'algoritmo.

In fine il terzo capitolo verterà sull'analisi di macchine di apprendimento automatico utilizzate per la predizione di eventi ipo/iperglicemici, in particolare Support Vector Machine e Random Forest, soffermandosi sulla logica e matematica che vi è alla base di questi metodi, dimostrando la loro efficacia predittiva riportando alcune applicazioni in ambito medico.

Indice

Capitolo 1

Diabete e Nuove tecnologie per la gestione della terapia

1.1. Diabete Mellito

Il diabete è la più nota malattia metabolica che può interessare il genere umano. Essa si caratterizza dall'incapacità da parte dell'organismo di regolare autonomamente i livelli di glucosio sanguigno. Il glucosio è fondamentale per l'organismo poiché è il nutriente essenziale per tutte le cellule che lo prelevano direttamente dal sangue. La principale fonte di glucosio sono gli alimenti, ma, in misura minore, esso può anche essere sintetizzato ex novo a partire da protidi e lipidi all'interno dell'organismo stesso. Il corpo umano possiede un sistema di regolazione intrinseco che consente di mantenere relativamente costante la glicemia durante l'arco della giornata.

Il controllo dell'indice glicemico è dovuto principalmente a due ormoni antagonisti prodotti delle isole di Langerhans del pancreas: insulina e glucagone. Il primo permette l'abbassamento del livello di glucosio nel sangue favorendo l'assorbimento del glucosio da parte delle cellule muscolari e adipose, mentre il secondo interagisce con il glicogeno, formato da monomeri di glucosio, causandone l'idrolisi e quindi il rilascio di glucosio.

L'insorgenza di tale malattia è legata all'insulina; per la precisione, può dipendere da una ridotta disponibilità di essa (la cui produzione non soddisfa le esigenze dell'organismo), dalla scarsa sensibilità all'ormone da parte dei tessuti bersaglio o, infine, da una combinazione di questi fattori. Il risultato è il conseguente incremento del livello di glucosio nel sangue o iperglicemia. [1]

Prendendo in considerazioni i dati del WHO, circa 422 milioni di persone nel mondo sono affette da diabete, la maggior parte delle quali vive in paesi a basso e medio reddito, e ogni anno 1,5 milioni di morti sono direttamente attribuibili al diabete. Nel 2019, il diabete è stato la causa diretta di 1,5 milioni di decessi e il 48% di tutti i decessi dovuti al diabete è avvenuto prima dei 70 anni. Nel 2021, in Europa, oltre 1,1 milioni di decessi sono stati causati dal diabete. Sia il numero di casi che la prevalenza del diabete sono aumentati costantemente negli ultimi decenni e per questo rappresenta la quarta causa di morte nell'Unione Europea.[2][3].

È possibile distinguere il diabete di tipo 1 e 2. In questo elaborato verrà trattato solo il primo.

1.2. Diabete di tipo 1 e Sintomatologia

Il diabete di tipo 1 rientra nella categoria di malattie autoimmuni, in quanto causata dalla produzione di anticorpi da parte dell'organismo che riconosce come estranee le cellule Beta, reputate alla produzione di insulina. Questo tipo di diabete viene suddiviso in due tipologie. Diabete mellito autoimmune (più conosciuto come "insulino-dipendente") caratterizzato dalla distruzione delle cellule Beta da parte degli anticorpi. Viene a manifestarsi maggiormente nell'età infantile, ma non sono rari episodi in età adulta dove si presenta in una forma più lenta e progressiva, definito come diabete autoimmune latente dell'adulto (LAD). Un'altra tipologia è il diabete mellito idiopatico, più raro, presente maggiormente nelle persone di etnia africana e asiatica. Quest'ultima, si presenta con una carenza di insulina permanente accompagnata da chetoacidosi, ma nessuna evidenza di autoimmunità.

Le cause che alimentano il manifestarsi di questa malattia non sono ancora del tutto apprese, ma alcuni fattori sono di sicuro determinanti per la sua comparsa, tra cui: fattori genetici (ereditari), fattori immunitari (legati ad una particolare difesa del nostro organismo), fattori ambientali.[5]

La malattia diabetica causa importanti complicanze a carico di cuore, vasi sanguigni, reni, occhio e nervi e, per questo motivo, richiede frequenti controlli periodici. Per quanto riguarda l'occhio, infatti, vi è la possibilità di recare danno alla retina in quanto i vasi che la irrorano possono essere danneggiati da alti livelli di glicemia. Tale complicanza è definita come retinopatia diabetica che, se non curata, può portare alla cecità. Inoltre, se il livello del glucosio non è adeguato aumenta la probabilità di aterosclerosi, con conseguente diminuzione del flusso sanguigno al cuore (infarto) o al cervello (ictus). Invece, per i reni, tale malattia può portare ad un aumento della pressione arteriosa con conseguente alterazione delle sue normali funzionalità, fino a raggiungere in alcuni casi l'insufficienza renale che deve essere curata con la dialisi o il trapianto di rene.

Diversi sono i sintomi dovuti al diabete, il primo che viene a presentarsi è dovuto al non assorbimento del glucosio da parte dell'organismo che comporta un consistente aumento del volume delle urie (poliuria), veicolo utilizzato per l'eliminazione del glucosio, con conseguente aumento della sensazione di sete (polidipsia), e un calo di peso improvviso (polifagia paradossa) dovuto al fatto che non vengono trattenute le sostanze nutritive.

Un altro sintomo che viene a presentarsi soprattutto in pazienti soggetti a diabete di tipo 1 è la chetoacidosi diabetica, complicanza causata dall'impossibilità di utilizzare il glucosio per la sintetizzazione dell'ATP, fonte di energia essenziale per il funzionamento cellulare. Tale

problematica principalmente dovuta alla non inibizione da parte dell'insulina della chetoneogenesi, la quale procede formando i corpi chetonici. I principali chetoacidi prodotti, l'acido acetoacetico e l'acido beta-idrossibutirrico, sono acidi organici forti che provocano acidosi metabolica. L'acetone, derivato dal metabolismo dell'acido acetoacetico, si accumula nel siero e viene lentamente eliminato con la respirazione, per questo motivo molti pazienti che soffrono di tale disturbo possiedono solitamente un alito fruttato. L'accumulo di queste sostanze nell'organismo, se non si interviene per tempo, può portare a conseguenze molto pericolose fino al coma e al decesso.

Per i motivi elencati precedentemente, le persone a rischio di sviluppare tale malattia, ad esempio coloro che presentano familiari affetti, devono sottoporsi a controlli regolari per diagnosticare la malattia e intervenire tempestivamente. Sono tre i possibili esami clinici che possono essere prescritti per accertare la presenza di diabete di tipo 1, e sono:

- Controllo della glicemia, eseguito solitamente attraverso glucometro, ma anche attraverso esame del sangue, con il quale si rileva il tasso di glucosio nel sangue. Valori di glicemia per soggetti sani durante l'arco della giornata variano tra 60 e 130 mg/dl. Se, invece, vengono rilevati valori al di sopra di 200 mg/dl si procede con il rilevamento a digiuno, se questo risulta superiore a 126 mg/dl il paziente è da considerarsi un potenziale affetto. si procede con altri esami per confermare la malattia;
- Test dell'emoglobinaglicata (HbA1c), esame che misura i livelli di glicemia degli ultimi due o tre mesi;
- Esami del sangue, utili per rilavare eventuali anticorpi delle cellule Beta del pancreas e confermare la presenza del diabete di tipo 1;
- Esami delle urine, misurano la quantità di glucosio e/o la presenza di chetoni, indicando lo stato di avanzamento della patologia nel momento dell'esame.

Il diabete di tipo 1 non è guaribile, ma è possibile monitorarlo e tenerlo sotto controllo con diversi metodi.[4][5]

1.4. Terapia per la gestione del diabete di tipo 1

I pazienti affetti da diabete di tipo 1, devono sottoporsi per tutta la vita alla terapia insulinica. L'insulina viene somministrata prevalentemente con iniezioni nel tessuto sottocutaneo da cui poi si diffonde a tutto l'organismo. Il compito dell'insulina esogena (somministrata tramite iniezione) è quello di simulare quanto più possibile l'azione dell'insulina endogena (prodotta dall'organismo) sia

per quanto riguarda l'insulinizzazione basale (costante nell'arco della giornata) e acuta (in seguito ai pasti), sia a digiuno che dopo aver mangiato. Tuttavia, tale terapia non può prescindere da una dieta sana e bilanciata e dalla pratica costante di attività fisica, entrambe fondamentali per la gestione di questa malattia.

La terapia farmacologica basata sull'iniezione di insulina può avvenire in due modi: attraverso l'iniezione sottocutanea, da svolgere più volte al giorno, con l'utilizzo di una "penna" che contiene una cartuccia di insulina e un piccolo ago, oppure, attraverso un microinfusore, o pompa insulinica, dispositivo programmabile che permette la somministrazione automatica di insulina durante l'arco della giornata. [6]

Queste nuove tecnologie, con l'aggiunta di dispositivi per il monitoraggio del glucosio, hanno permesso ai malatini di non rinunciare alle proprie abitudini, mantenendo una qualità della vita alta. Infatti, fino a pochi anni fa, la vita delle persone affette da diabete di tipo 1 era scandita dalla necessità di effettuare iniezioni di insulina in determinati momenti della giornata, limitando notevolmente la vita del malato, che doveva sottostare a queste regole stingenti, talvolta, difficili da seguire minuziosamente, soprattutto se si considera che l'incidenza maggiore di questa malattia è tra i bambini e gli adolescenti. Oggi, possono mangiare quando desiderano mantenendo comunque un ottimo controllo glicemico grazie alla valutazione quantitativa dei carboidrati, al controllo dei livelli di glicemia e agli eventuali aggiustamenti nei dosaggi di insulina.

1.5. Sensori per il monitoraggio in continua del glucosio

Inizialmente il monitoraggio della glicemia si basava principalmente su misuratori glicemia ad auto monitoraggio (SMBG), dispositivi portatili che consentivano la misurazione della concentrazione di glicemia, solitamente 3-4 volte al giorno, su una goccia di sangue capillare mediante l'ossidazione del glucosio.

Gli svantaggi principali di questi strumenti, però, risiedono nell'invasività della procedura per la lettura del valore di glucosio e nel fatto che essi consentono solamente poche misurazioni durante l'arco della giornata, non essendo quindi in grado di cogliere eventuali episodi di iperglicemia o ipoglicemia durante i periodi tra una misurazione e l'altra.

Per far fronte a questi problemi, sono comparsi sul mercato, intorno agli anni 2000, dispositivi che permettono un monitoraggio (quasi) continuo della glicemia nei liquidi interstiziali, con una misura

ogni 1-5 minuti, detti dispositivi per il Monitoraggio del Valore Continuo del Glucosio: Continuous Glucose Monitoring (CGM). Essi consentono infatti di individuare un numero maggiore di rilevazioni rispetto ai convenzionali SMBG e di effettuare un'analisi retrospettiva dell'andamento glicemico stesso, in base alla quale si può intervenire opportunamente sulla terapia.

Il sensore può essere inserito autonomamente usando un dispositivo automatico fornito insieme al sistema oppure nel corso di una breve seduta ambulatoriale, in anestesia locale, mediante un'incisione di pochi millimetri a livello dell'addome o della parte superiore del braccio.
Oggi i pazienti affetti da diabete possono disporre, in particolare, di due tipologie di dispositivi per il monitoraggio continuo del glucosio. Si tratta del CGM in tempo reale e del CGM a rilevazione intermittente (*Flash glucose monitoring*).

Entrambi i dispositivi misurano e visualizzano in modo automatico la glicemia ad intervalli fissati, fornendo informazioni sui livelli glicemici pregressi, indicando, attraverso determinati algoritmi, la tendenza dei valori della glicemia e predicendo la glicemia futura. Nel caso del CGM real-time le informazioni vengono inviate senza bisogno di intervento da parte del paziente, mentre, con il CGM a rilevazione intermittente le informazioni vengono trasmesse ogni volta che l'utente effettua una scansione mediante il lettore del dispositivo oppure via app.

Un recente studio ha riportato che le barriere più comuni che dissuadevano dall'uso di CGM erano legate alla mancanza di copertura assicurativa e al fastidio di indossare dispositivi, ma i motivi più comuni che limitavano il CGM erano soprattutto il costo, l'inaffidabilità degli allarmi e l'imprecisione percepita del sensore. Ad oggi, grazie ai recenti sviluppi della tecnologia, in particolare con il rilascio di nuovi sensori più precisi con dimensioni ridotte e requisiti di calibrazione, la maggior parte di queste barriere sono state superate.[7]

La maggior parte dei sistemi CGM personali attualmente sul mercato, tra cui Dexcom G5 Mobile e G6, Medtronic Enlite e Guardian e Senseonics Eversense, forniscono avvisi glicemici alti e bassi che aiutano il paziente a rilevare eventi ipo/iperglicemici.

I dispositivi che risultano più performanti, prendendo come parametro di confronto il MARD (differenza media assoluta relativa) che misura l'accuratezza del dispositivo, sono sicuramente Dexcom G6 e Eversense XL.

Il sistema CGM Eversense XL fornisce un monitoraggio continuo del glucosio tramite un sensore impiantabile, applicabile attraverso una piccola incisione sottocutanea, da un professionista abilitato, che richiede di essere sostituita ogni 180 giorni. Il sensore viene ad interfacciarsi con un trasmettitore posto al di sopra di esso, che lo alimenta in modalità wireless per attivare il trasferimento delle

misurazioni del glucosio verso una apposita applicazione. Il sistema Dexcom G6 consente di inserire il sensore sotto la cute in maniera facile e in maniera meno invasiva attraverso uno specifico applicatore. Tale sensore posizionato sottocute misura continuamente e in modo accurato la quantità di glucosio interstiziale, inviando le letture. tramite Bluetooth. al dispositivo di visualizzazione. Tale dispositivo è indicato per qualsiasi fascia di età, mantenendo un MARD tra 7% e 10%. A differenza di Eversense XL, che per la sua modalità di applicazione è consigliato per persone con età superiore a 18 anni.

Figura A: Dexcom G6

Figura B: Eversense XL

I sistemi CGM sono ora accettati come strumenti standard per il controllo intensivo del glucosio nei pazienti con diabete mellito di tipo 1. Tuttavia, sono ancora presenti diverse limitazioni importanti. Infatti, i sensori elettrochimici a base di glucosio-ossidasi soffrono di diversi limiti come la loro risposta non lineare all'interno dell'intervallo biologico rilevante, la possibile interferenza con agenti attivi (ad esempio, paracetamolo) e, soprattutto, la loro dipendenza sia di sensibilità che di specificità dalla disponibilità enzimatica sulla superficie dell'elettrodo. Infatti, le aziende di sensoristica raccomandano ancora l'uso di dispositivi SMBG quando i sintomi non corrispondono alle letture del sensore o non viene visualizzata una freccia di tendenza. [8]

1.6 Predizione della glicemia futura

All'inizio del nuovo secolo, l'avvento delle apparecchiature per il monitoraggio continuo della glicemia ha compiuto un passo importante nel trattamento del diabete. Le informazioni continue sui livelli di glucosio nel sangue vengono utilizzate principalmente per regolare la terapia e per allertare in caso di un evento pericoloso. Tuttavia, è facile comprendere che piuttosto che intervenire quando si è verificato un episodio di ipoglicemia/iperglicemia, è senza dubbio più utile cercare di prevenire un tale episodio. I dati ricavati infatti si prestano ad essere elaborati per cercare di riconoscere con anticipo il verificarsi di eventi rischiosi, un anticipo tale da permettere di intervenire prima che questi eventi si verifichino, come ad esempio la semplice assunzione di glucosio per evitare un episodio di ipoglicemia. Per questo motivo negli ultimi anni ha assunto un peso sempre più rilevante la ricerca nell'ambito della predizione di serie temporali. Alcuni dispositivi CGM sono già dotati di un semplice algoritmo di proiezione, che rileva la tendenza degli ultimi campioni e genera un'allerta sulla base del valore di glucosio predetto. È chiaro, però, come sarebbe preferibile avere a disposizione meccanismi di predizione più raffinati, che siano in grado di dare con sufficiente anticipo un'informazione più precisa sull'aspettazione del valore glicemico, in modo da offrire la possibilità di intervenire correttamente e in tempo, evitando quindi i danni a breve e a lungo termine.

In letteratura sono stati proposti diversi algoritmi per la previsione in tempo reale di eventi ipoglicemici e iperglicemici dai dati CGM. Esistono modelli autoregressivi, reti neurali artificiali, modelli neurali artificiali e metodi basati sul kernel per la previsione del glucosio utilizzando solo la storia passata del segnale CGM come input. Poiché la cinetica del glucosio è influenzata dalla quantità di carboidrati ingeriti, insulina iniettata, attività fisica, ecc., esistono algoritmi di previsione del glucosio che considerano alcuni (o anche tutti) questi segnali, come i modelli autoregressivi di media mobile con input esogeni, il Random Forest, Support Vector Machine, processi gaussiani, Reti neurali. [21]. Questo elaborato si prefigge lo scopo di approfondire due algoritmi predittivi, in particolare Random Forest e Support Vector Machine, concentrandosi sulla parte logico-implementativa del modello, mostrando la loro efficacia in ambito diabetologico.

Capitolo 2

Machine learning

2.1. Introduzione

Nel 1956 un gruppo di scienziati informatici propose l'idea di programmare computer che potessero pensare e ragionare. Essi descrissero tale principio come Intelligenza Artificiale (AI), campo che si concentra nell'automatizzare tutti quei processi intellettuali normalmente svolti dall'uomo. Per raggiungere tale scopo vengono utilizzati diversi metodi, tra cui il machine learning.
Il Machine Learning (ML) consente ai computer di "auto-apprendere" dai dati di addestramento e di migliorare nel tempo, senza essere esplicitamente programmati. Gli algoritmi di apprendimento automatico sono in grado di elaborare modelli nei dati e di imparare da essi, al fine di fare le proprie previsioni, imparando attraverso l'esperienza rilevata dai dati raccolti.

Nella programmazione tradizionale, un informatico scrive una serie di istruzioni che indicano al computer come trasformare i dati in ingresso in un risultato desiderato. Le istruzioni sono per lo più basate su una struttura "If-Then": quando si verificano determinate condizioni, il programma esegue un'azione specifica. Il Machine Learning, invece, è un processo automatizzato che consente alle macchine di risolvere i problemi con pochi o nessun input umano.

L'aspetto più importante del ML è la ripetitività, perché più i modelli sono esposti ai dati, più sono in grado di adattarsi in modo autonomo. I computer imparano da elaborazioni precedenti per produrre risultati e prendere decisioni che siano affidabili e replicabili.

Grazie alle nuove tecnologie di elaborazione, il ML di oggi non è lo stesso del passato. Questa scienza non è nuova, ma sta acquisendo un nuovo slancio e miglioramento. E sebbene molti algoritmi di ML siano in circolazione da molto tempo, l'applicazione in abito medico è uno sviluppo più recente.[9]

2.2. Metodologie di apprendimento

Per capire come funziona l'apprendimento automatico, è necessario esplorare diversi metodi e algoritmi di apprendimento automatico, che sono fondamentalmente insiemi di regole che le

macchine utilizzano per prendere decisioni. Di seguito sono riportati i quattro tipi di tecniche di apprendimento automatico più comuni e più utilizzati.

Tra i diversi tipi di apprendimento sembra che il più conosciuto e utilizzato sia l'apprendimento supervisionato (Supervised Learning), una modalità di addestramento dove i dati utilizzati sono etichettati; ad ogni input si conosce il rispettivo output che viene utilizzato per insegnare all'algoritmo le regole del modello. I dati in input detti anche caratteristiche (features), "x", sono mappate sull'obiettivo (target), "Y", apprendendo la funzione di mappatura," f", così che in futuro il modello possa prevedere l'obiettivo partendo dalle caratteristiche usando Y = f(x).
L'apprendimento supervisionato oltre a essere il più conosciuto è forse il più laborioso in quanto prevede diverse fasi.

- La preparazione dei dati di addestramento dove a ogni insieme di dati di input è associato il corretto output. L'assegnazione dell'output viene fatta con la supervisione umana, ma quasi sempre con sistemi di raccolta massiva dei dati con cui si acquisiscono sia le caratteristiche sia il corrispondente output.
- La divisione dell'intero dataset in training e test in base alla strategia di addestramento
- L'addestramento dell'algoritmo sul training dataset; in questa fase si apprendono le relazioni tra features e output tramite cui sarà possibile fare previsioni con input diversi da quelli di addestramento. Questa fase può andare oltre il singolo addestramento e comprende cicli di apprendimento e validazione in cui il dataset di training è suddiviso in un numero intero di parti e a rotazione una di queste non è utilizzata per l'addestramento ma per la validazione.
- Il test, la fase in cui si prova il modello sul dataset di test, che sono dati di input ignoti al momento dell'addestramento. Con utilizzo di diverse metriche si può valutare la qualità del modello confrontando l'output vero e quello predetto. In base al contesto, alla tipologia di applicazione e se l'errore è ritenuto sufficientemente basso, l'addestramento si conclude e il modello può passare in produzione ovvero ricevere nuovi dati di input e fornire l'output in base alle relazioni apprese in addestramento.

Diversi sono gli aspetti che influiscono sull'efficienza dell'algoritmo finale, come la qualità, quantità e variabilità dei dati di addestramento, della scelta degli iperparametri che determinano l'ottimizzazione delle prestazioni del modello, dalla modalità di validazione e test e della scelta di un algoritmo tra i possibili applicabili per la soluzione del problema.

Gli algoritmi di apprendimento supervisionato sono divisi in due grandi categorie: classificatori e regressori. I classificatori separano i dati in due o più classi. Quando fornisco un esempio al

classificatore, l'algoritmo mi restituisce la classe a cui potrebbe appartenere. Questi si possono dividere come: lineari, semplici e veloci, oppure non lineari, più precisi, ma più lenti da elaborare. I regressori, invece, si basano sull'interpolazione dei dati per associare tra loro due o più caratteristiche. Quando fornisco all'algoritmo una caratteristica in input, il regressore mi restituisce l'altra caratteristica in base al modello implementato.

Un'altra tecnica è l'apprendimento senza supervisione, o "non supervisionato" (Unsupervised Learning), che si differenzia dall' apprendimento supervisionato proprio perché avviene quando l'algoritmo deve scoprire eventuali relazioni esistenti durante il suo sviluppo, in quanto, i dati di input non sono etichettati o non hanno un corrispondente valore di output. Il suo utilizzo è adatto a cercare associazioni e modelli nascosti nei dati che il solo osservatore umano non riuscirebbe a individuare, sia perché oscurati da altre informazioni sia perché la quantità di dati è talmente grande da non poter essere osservata facilmente senza un ausilio computazionale.
Utilizzando l'apprendimento senza supervisione il modello non può essere addestrato su un insieme di dati preparati con il corretto output corrispondente, ma deve individuare autonomamente le differenze o le similitudini fra gli input identificandone le caratteristiche principali. Per questo motivo tali algoritmi realizzano operazioni molto più complesse rispetto al caso precedente, in quanto apprendono direttamente dai dati in ingresso e non da un dataset preimpostato di addestramento.
I principali algoritmi che utilizzano questa metodologia sono gli associativi, di clustering e riduzione delle anomalie. Il primo cerca la regolarità dei dati raggruppandoli in base alle loro caratteristiche; il secondo consente di associare variabili di grandi database attraverso parametri predefiniti e infine il terzo, elimina i dati non significativi e combina le informazioni ridondanti per concentrare l'analisi su quelli in cui emerge uno schema.

Inoltre, un'altra tecnica utilizzata per la costruzione di algoritmi è l'apprendimento semi-supervisionato (Semisupervised Learning), il quale si posiziona tra l'apprendimento supervisionato e non supervisionato. Tale metodologia viene utilizzata particolarmente per set di dati che non sono stati tutti etichettati.

L'ultima tecnica di apprendimento per il machine learning che si può utilizzare è l'apprendimento con rinforzo (Reinforcement Learning), dove, inizialmente si conosce l'obiettivo ma non si ha un dataset di esempi per fare l'addestramento, né una base di conoscenza pregressa. Si parte dall'osservazione dell'ambiente dove l'algoritmo compie delle decisioni che vengono valutate da una funzione di rinforzo, che misura il grado si successo di ciascuna azione, distribuendo premi o penalità in base a quanto la decisione si sia avvicinata all'obbiettivo. Tali informazioni vengono poi immagazzinate permettendo all'algoritmo di ripetere nel tempo le azioni più profittevoli ed evitare quelle in perdita.

Così facendo la macchina è meno legata al contenuto del training set e può prendere le decisioni con meno vincoli e un maggiore grado di libertà come nell'apprendimento non supervisionato, ma a differenza di questo, l'algoritmo non inizia il processo di apprendimento senza conoscenza pregressa, bensì la macchina può distinguere fin da subito le azioni positive e negative tramite una funzione di rinforzo che aiuta il processo di apprendimento. [10]

2.3. Addestramento: underfitting e overfitting

L'addestramento è la parte fondamentale per l'implementazione di un nuovo algoritmo. In questa determinata fase si procede, oltre all'inserimento di input (dati) nella macchina, anche alla validazione e controllo degli output, così da poter monitorare l'andamento del modello. Due sono gli errori che si possono riscontrare nell'addestramento di un modello, overfitting e underfitting. Il primo si presenta nel momento in cui la classificazione è molto sensibile ai dati di training (varianza alta) essendoci parametri troppo complessi, ad una non generalizzazione dell'algoritmo; il secondo, invece, si presenta con parametri di addestramento troppo semplici, portando ad un'alta discrepanza nella classificazione.

Una valutazione indipendente del modello condotta da esperti professionisti può ridurre i rischi associati alle nuove tecniche di modellazione. Nonostante la novità delle tecniche di machine learning, esistono diversi metodi per mitigare l'effetto dell'overfitting e di altre problematicità nel settaggio del modello previsionale. Il requisito più importante per la validazione del modello è valutazione della bontà di adattamento e la comprensione a pieno della corretta operatività dell'algoritmo. Se nella fase di validazione non si conosce la teoria e le ipotesi sottostanti alla base del modello o si ha un'esperienza limitata di queste tecniche, è probabile che non si sia in grado di eseguirne una valutazione efficace. Molto importante è la comparazione dei risultati stimati rispetto ai dati osservati.

L'analisi dei risultati rappresenta, dopo tutto, un approccio molto utile per comprendere e valutare le interazioni e le potenziali criticità del modello implementato. Valutare i valori della variabile indipendente in funzione sia del risultato osservato che di quello stimato, insieme al numero di osservazioni consente all'utente di osservare la relazione presente all'interno del modello e valutare il livello di overfitting prodotto. Per valutare le possibili interazioni, possono essere creati anche diagrammi incrociati valutando i risultati in due dimensioni anziché in una. Oltre le due dimensioni

diventa difficile valutare i risultati, ma osservando le interazioni univoche si acquisisce una prima comprensione del comportamento del modello rispetto alle singole variabili indipendenti utilizzate.

Per ottimizzare il livello di predizione del modello e diminuire l'errore di generalizzazione. Il dataset di addestramento viene solitamente suddiviso in un dataset di addestramento leggermente più piccolo e in un dataset di validazione separato. Il dataset di validazione ha lo scopo di imitare il dataset di test e aiutando a mettere a punto un algoritmo identificando quando un modello può generalizzarsi bene e funzionare in una nuova popolazione.

Una tecnica frequentemente utilizzata a tale scopo è la Cross Validation, la quale cerca di assicurarsi che un modello non produca un eccessivo livello di overfitting rispetto ai dati campionari. È stata utilizzata in passato per contribuire ad assicurare l'integrità di altri metodi statistici e con la crescente popolarità delle tecniche di Machine Learning, ha visto incrementarsi il suo utilizzo. Con tale tecnica, il modello viene costruito e adattato utilizzando solo una parte del campione di dati a disposizione, utilizzando la restante porzione per valutarne il potere predittivo. In condizioni ideali, il modello funzionerà ugualmente bene su entrambe le porzioni di dati. In caso contrario, è probabile la presenza di un livello di overfitting non trascurabile. Solitamente, i dati campionari vengono suddivisi in proporzione 80-20, in cui l'80% viene utilizzato per allenare il modello (training data set) e il restante 20% (validation data set) viene impiegato per la valutazione della capacità predittiva. Oltretutto, esistono anche approcci più rigorosi e performanti dalla Cross Validation, tra cui la k-fold Validation, che prevede che il processo sopra esposto venga ripetuto k volte con diverse suddivisioni dei dati campionari.

Le prestazioni del modello vengono monitorate sui set di dati di addestramento e di convalida, non intervenendo se il modello è in fase di apprendimento, ossia l'accuratezza del modello sul set di addestramento e sul set di validazione aumenta e converge dopo ogni iterazione di addestramento. Se entrambi convergono, ma non aumentano, il modello non sta apprendendo e potrebbe essere in una situazione di "underfitting", cioè potrebbe non aver appreso abbastanza della relazione tra le caratteristiche e gli obiettivi in modo tale da poter generalizzare. Infine, se le prestazioni dell'addestramento aumentano molto di più di quelle del set di validazione (ad esempio, il modello ha un'accuratezza molto alta sui dati che gli sono stati forniti per l'addestramento, ma non altrettanto alta sui dati di validazione), il modello è in overfitting, cioè, ha appreso caratteristiche specifiche per il set di dati di allenamento a scapito dell'adattabilità dell'algoritmo su altre popolazioni. Tuttavia, il set di dati di validazione non viene utilizzato specificatamente per addestrare l'algoritmo, ma viene utilizzato iterativamente per l'algoritmo.

Sebbene il modello possa avere buone prestazioni sul dataset di validazione, non è necessariamente un indicatore affidabile delle prestazioni del modello, perché è probabile che non le abbia sul dataset di test, molto più grande. Poiché il dataset di validazione è un piccolo campione della vera popolazione (più ampia) e potrebbe non rappresentare accuratamente la popolazione stessa (cioè, non sarebbe un modello generalizzabile).[11]

2.4. Metodo di valutazione dei modelli predittivi

Una volta completata la fase di addestramento e superata la fase di validazione si ha idealmente allenato un modello generalizzabile, tuttavia, questo deve essere accertato attraverso un dataset di test contenente altri dati, non ancora utilizzati. Con il quale si verifica, attraverso diversi metodi di valutazione, l'efficacia e la correttezza del modello predittivo.

Le metriche di valutazione delle performance di un algoritmo si possono valutare in diversi modi, e variano in base alla classe utilizzata: classificatori o regressori.

Per i regressori la metrica più comune è la funzione di costo della regressione: l'errore quadratico medio (MSE). indica la discrepanza quadratica media fra i valori dei dati osservati ed i valori dei dati stimati. È definito come.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \overline{y}_i\right)^2 \qquad 1.1$$

Dove $\overline{y}$ è il modello che prevede un vettore di feature x come input e genera una previsione, che va da 1 a N, numero di campioni. La somma delle previsioni meno i valori reali, y , su N indica la media. L'elevamento al quadrato ci permette di eliminare il segno negativo dei dati e di dare più peso a differenze maggiori.
Un modello di regressione tenta di adattare i dati tracciando una linea che riduce al minimo la distanza tra i punti di dati reali e i punti della stessa linea. Più i valori sono vicini alla linea, migliore è il comportamento del modello per quel particolare punto. Pertanto, si cercherà di ottenere un MSE basso. Tuttavia, l'MSE è influenzato da valori anomali, ovvero valori di dati che sono anormalmente distanti dalla vera retta di regressione. Per definizione, l'errore al quadrato per punti così distanti sarà molto alto.

Un'altra metrica utilizzata per valutare la bontà di una predizione è lo scarto quadratico medio che misura l'errore assoluto in cui gli errori sono al quadrato per evitare che valori positivi e negativi si

annullino a vicenda (come nel MSE). Rappresenta la deviazione standard dei residui. Il termine residuo si riferisce alla distanza tra il punto previsto e il punto osservato. Essendo la deviazione standard, indica quanto sono distribuiti i residui attorno alla nostra linea di regressione

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \overline{y}_i\right)^2} \quad 1.2$$

Nellì analisi di un algoritmo di regressione tende ad utilizzare la 1.1 oppure la 1.2.

La metrica più utilizzata per stabilire la performance di un classificatore è la semplice accuracy. Tale metrica è definita come il numero di classificazioni corrette, tipicamente su un test set. L'accuracy è estremamente semplice da calcolare e confrontare, ma non mi permette di capire nulla sugli errori commessi dall'algoritmo, quindi, da solo, non è in grado di stabilire se un classificatore è buono oppure no, Per questo motivo insieme ad essa si utilizzano altri parametri per la valutazione delle performance.

$$accuracy = \frac{N_{TN} + N_{TP}}{N_P + N_N} \quad 1.3$$

$$recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad 1.4$$

$$specificity = \frac{N_{TN}}{N_{TN} + N_{FP}} \quad 1.5$$

$$precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad 1.6$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad 1.7$$

Per il calcolo di queste misure spesso si ricorre all'utilizzo di:

- Veri positivi (TP): campioni positivi correttamente identificati come positivi;
- Veri negativi (TN): campioni negativi correttamente identificati come negativi;
- Falsi positivi (FP): campioni negativi erroneamente classificati come positivi;
- Falsi negativi (FN): campioni positivi erroneamente classificati come negativi;

Utilizzando queste metriche è possibile valutare con una interpretazione probabilistica l'efficacia predittiva di un modello. La *precision* (1.6) stima gli eventi previsti correttamente (cioè i veri positivi) tra tutti gli eventi che l'algoritmo reputa corretti (compresi i falsi positivi). Può essere interpretata come la probabilità che un evento previsto si verifichi effettivamente nel prossimo futuro. La *specificity* (1.5) è definita come la percentuale di negativi effettivi che sono stati predetti come negativi. Ciò implica che ci sarà un'altra percentuale di negativi effettivi, che sono stati predetti come positivi e che potrebbero essere definiti come falsi positivi. La somma delle specificità e del tasso di falsi positivi sarà sempre pari a 1. Per questo motivo, viene spesso usata la definizione "$(1 - specificity)$" chiamata tasso di falsi positivi (FPR). Con lo stesso concetto si definisce *recall* (o *sensibility*) (1.4) che misura la percentuale di casi positivi effettivi che sono stati predetti come positivi, definita anche, come tasso di veri positivi (TPR). Un valore più alto di *specificity* indica un valore più alto di veri negativi e un tasso più basso di falsi positivi. Invece, un valore più alto di *sensibility* determina un valore più alto di veri positivi e un valore più basso di falsi negativi. La misura *F1* combina queste due metriche con una media armonica, fornendo un buon modo per confrontare i risultati di diversi algoritmi.

Spesso, per un'osservazione e una valutazione più grafica, si utilizza la matrice di confusione (*Confusion matrix)*; una matrice N x N, dove N è il numero di classi target. Essa permette di confrontare i valori effettivi (reali) del target con quelli previsti dal modello di apprendimento automatico. In sostanza, rappresenta un riepilogo tabellare del numero di previsioni corrette e scorrette fatte da un classificatore.

Per la valutazione più intuitiva delle performance del modello spesso si mettono in relazione TRP e FPR ponendo particolare attenzione all'area che sottende la curva (AUC). Tale curva viene valutata attraverso una gamma di soglie da 0 a 1 che funge da divisorio tra le due classi. Una caratteristica operativa curva è che inizia nel punto (FPR = 0, TPR = 0), che corrisponde a una soglia decisionale di 1 dove ogni campione viene classificato come negativo non essendoci, quindi, falsi o veri positivi. Termina nel punto (FPR = 1, TPR = 1), che corrisponde a una soglia di decisione pari a 0 dove, al contrario del caso precedente ogni campione è classificato come positivo e quindi tutti i punti sono

etichettati come veri o falsi positivi. I punti intermedi, che creano la curva, si ottengono calcolando il TPR e l'FPR per diverse soglie decisionali tra 0 e 1. [11]

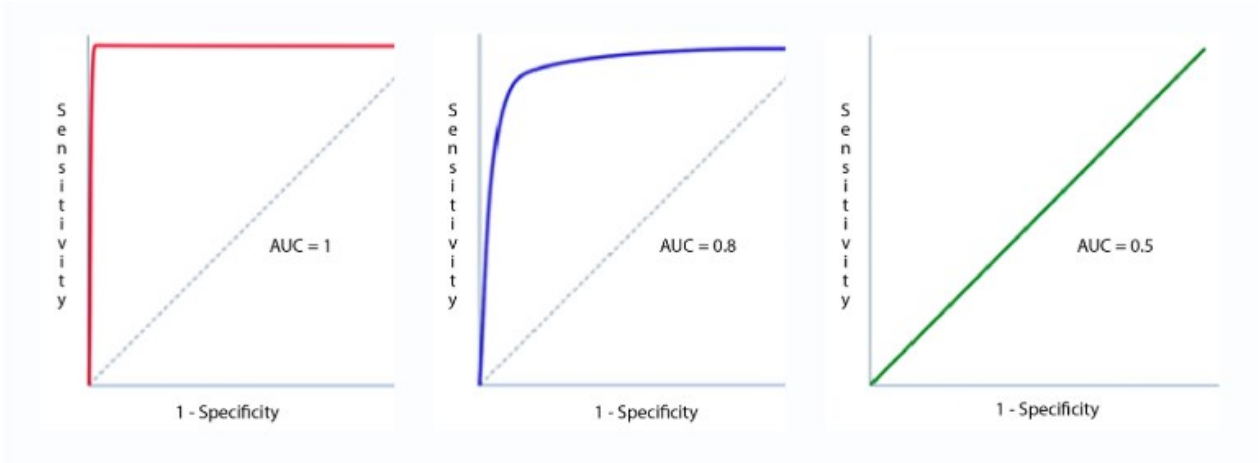

Figura 1: *differenti esempi di valori di AUC rispetto all'andamento della curva [12]*

Come si vede (*figura 1*), il primo modello fa un buon lavoro nel distinguere i valori positivi da quelli negativi. Pertanto, il punteggio AUC è pari a 0,9, poiché l'area sotto la curva ROC è ampia. Se invece osserviamo l'ultimo modello, le previsioni si sovrappongono completamente e il punteggio AUC è di 0,5. Ciò significa che il modello ha prestazioni scarse e le sue previsioni sono quasi casuali.

Pertanto, L'area sotto la curva (AUC) può essere calcolata e utilizzata come metrica per valutare le prestazioni complessive di un classificatore in quanto mi valuta il grado di separazione delle classi positive e negative, definendo la probabilità di errore su entrambe. [12]

Capitolo 3

Algoritmi predittivi di ML per la predizione di glicemia futura ed applicazioni

Dopo un'analisi delle diverse metodologie che si possono utilizzare per implementare e valutare le tecniche di Machine Learning si procederà, in questo capitolo, all'approfondimento di due tipi di algoritmi: Support Vector Machine e Random Forest. Questi verranno esaminati analiticamente dimostrando la loro efficacia attraverso studi applicativi.

3.1 Support Vector Machine (SVM)

La Support Vector Machine (SVM) è una delle tecniche di classificazione più diffuse che mira a minimizzare il numero di errori di classificazione diretti. Come con qualsiasi classificatore binario supervisionato, il compito di una macchina a vettore di supporto è individuare un confine di separazione (lineare o altro) in uno spazio di elementi in modo tale che le osservazioni successive possano essere automaticamente classificate in gruppi separati.

I dati in ingresso vengono suddivisi in due categorie attraverso un iperpiano di separazione. Gli iperpiani sono confini decisionali che aiutano a classificare i punti di dati. Tali punti, che si trovano su entrambi i lati dell'iperpiano, possono essere attribuiti a classi diverse e la dimensione dell'iperpiano dipende dal numero di caratteristiche. Se, per esempio il numero di caratteristiche in ingresso è 2, l'iperpiano è solo una linea. Se, invece, il numero di caratteristiche in ingresso è 3, l'iperpiano diventa un piano bidimensionale. Diventa difficile da immaginare quando il numero di caratteristiche è superiore a 3.

I dati per un problema di apprendimento a due classi consistono in oggetti etichettati con una delle due classi corrispondenti; per comodità si assume che le etichette siano +1 (esempi positivi) o 1 (esempi negativi). Nel seguito, x indica un vettore con componenti xi. La notazione xi indicherà l'iesimo vettore di un insieme di dati composto da n esempi etichettati (xi, yi), dove yi è l'etichetta associata a xi. Gli oggetti xi sono chiamati pattern o input.

L’iperpiano viene a definirsi tramite l’equazione:

$$\vec{w} \cdot \vec{x} + b = 0 \qquad 3.1$$

w viene definito come vettore peso, mentre b determina la traslazione dell’iperpiano dall’origine. L'iperpiano divide lo spazio in due sezioni e tramite il segno della funzione si può definire la posizione dei dati xi. Se ogni osservazione di addestramento è sopra o sotto l’iperpiano di separazione, secondo l’equazione geometrica che definisce il piano, la sua etichetta di classe associata sarà +1 o -1. Così abbiamo (potenzialmente) sviluppato un semplice processo di classificazione. Quindi, i punti x che soddisfano l’equazione (3.1) appartengo all’iperpiano, ma possiamo considerare altri punti:

$$\vec{w} \cdot \vec{x} + b < 0 \qquad 3.2$$

i quali si trovano al di sotto dell’iperpiano, o

$$\vec{w} \cdot \vec{x} + b > 0 \qquad 3.3$$

punti x che si trovano sopra di esso.

Per separare le due classi di punti dati, si possono scegliere molti iperpiani. Il nostro obiettivo è trovare un piano che abbia il massimo margine, cioè la distanza che separa la coppia più vicina di punti di dati appartenenti a classi opposte. Questi punti sono chiamati vettori di supporto, perché sono le osservazioni dei dati che "supportano", o determinano, il confine decisionale. La massimizzazione del margine fornisce un rinforzo in modo che i punti dati futuri possano essere classificati con maggiore sicurezza.

Per definire il margine dobbiamo ricorrere alla distanza punto piano *p*:

$$p = \frac{\|w \cdot x + b\|}{\|w\|} \qquad 3.4$$

Sa cui si ricava la dimensione del margine:

$$\mathrm{M} = \frac{2}{\|\mathrm{w}\|} \qquad 3.5$$

Per massimizzare il margine p bisogna minimizzare la sua inversa, ovvero

$$min\frac{1}{2}\|w\|^2 \qquad 3.6$$

Possiamo quindi scrivere il criterio del vettore di supporto unendo (3.3, 3.2, 3.5).

$$min\frac{1}{2}\|w\|^2$$

$$yi(w \cdot x_i \ + \ b) \geq \ 1, i \ = \ 1, \ldots, N, \qquad 3.7$$

Con N numero di feature.

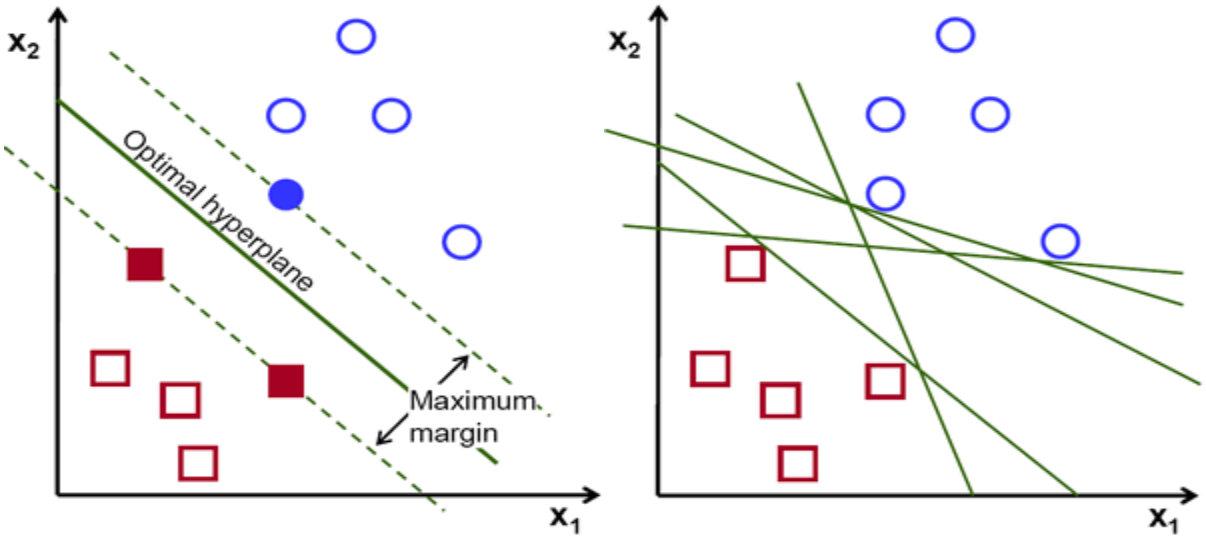

Figura 2: esempio di separazione perfetta di classi di dati ed esempio di Iperpiani di separazione multipli [16]

Chiaramente, la perfetta separabilità di un set di dati risulta possibile esclusivamente in un cotesto ideale, succede tutt'altro, considerando un data set reale, dove molto spesso alcuni dati delle due classi possono risultare mescolati non riuscendo a trovare un piano separatore, oppure essendo il margine dell'iperpiano troppo rigido, la sensibilità per l'introduzione di un singolo punto può cambiare radicalmente l'iperpiano, portando ad un errato adattamento dell'algoritmo. Per questo motivo, risulta efficace l'utilizzo di un margine morbido o SVC che ci permette un'approssimazione dell'iperpiano per la divisione delle classi.

3.1.1 SVC lineare o margine morbido

In applicazioni reali non sempre esiste un margine, ovvero non sempre le classi sono linearmente separabili nello spazio dei features attraverso un iperpiano. Il concetto alla base del margine morbido (*Soft Margin)* permette di ovviare a questo limite, introducendo una variabile ε aggiuntiva per ogni campione, in modo da rilassare il vincolo sul margine.

$$yi(w \cdot x_i \quad + \ b) \geq \ 1 - \varepsilon_i \qquad 3.5$$

Il parametro ε rappresenta la rilassatezza (*slackness)* associata al campione. Quando $0 < \ \varepsilon < 1$ il campione è correttamente classificato ma è all'interno dell'area di margine. Quando $\varepsilon > 1$ il campione entra nello spazio di decisione della classe opposta e perciò verrà classificato in maniera errata. Per cercare ancora un iperpiano di separazione in qualche modo ottimo, la funzione costo da minimizzare deve considerare anche la distanza tra il campione e il margine.

$$min \frac{1}{2} \|w\|^2 + \sum \varepsilon_i \qquad 3.6$$

Dove C, il budget, è un parametro di "penalizzazione" non negativo. La costante C. Il parametro C è un grado di libertà del problema per indicare quanto un campione deve pagare la violazione del vincolo, stabilendo l'importanza relativa della massimizzazione e minimizzazione del margine. Verrà meglio discusso nel capitolo 3.1.3

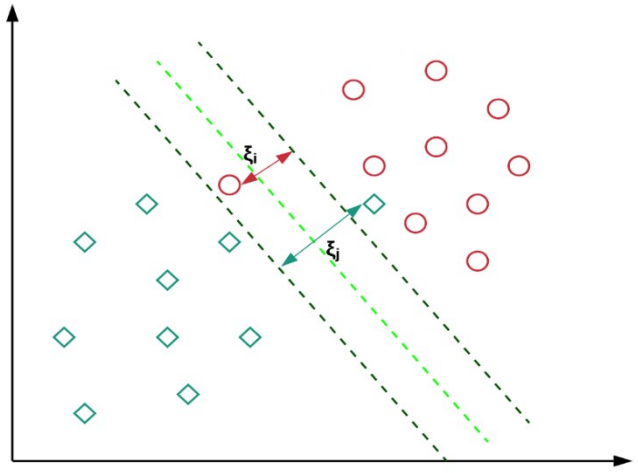

Figura 3: esempio di penalità ε data ad un set di dati a causa della loro posizione rispetto al margine [16]

La differenza tra un margine rigido e un margine morbido nelle SVM sta nella separabilità dei dati. Se i dati sono linearmente separabili, si opta per un margine rigido. In presenza di punti di dati che rendono impossibile trovare un classificatore lineare, dovremo essere più indulgenti e lasciare che alcuni punti di dati vengano classificati in modo errato. In questo caso, un SVM a margine morbido è appropriato.
A volte, i dati sono linearmente separabili, ma il margine è così piccolo che il modello diventa incline all'overfitting o troppo sensibile. Anche in questo caso, si può optare per un margine più ampio utilizzando SVM a margine morbido, per aiutare il modello a generalizzarsi meglio.[15][22]. In applicazioni reali molto spesso si osservano dati non lineari e difficili da separare con l'utilizzo del solo margine morbido; per risolvere questo tipo di problema viene spesso utilizzato SVM non lineare.

3.1.2 SVM non lineare

Nel momento in cui si ha bisogno di elaborare dati che si distribuiscono attraverso un confine non lineare risulta difficile intervenire utilizzando i metodi descritti precedentemente. Per risolvere tale problema si applicano delle trasformazioni, che mappano le caratteristiche dei dati dallo spazio originale in uno spazio di dimensione maggiore. L'obiettivo è quello di raggiungere uno spazio di dimensione superiore, in modo tale che le classi siano linearmente separabili, potendo, quindi, con più facilità inserire un confine decisionale per separare le classi e fare previsioni.

Di conseguenza per addestrare un classificatore a vettori di supporto e ottimizzare la nostra funzione obiettivo, dovremmo eseguire operazioni con i vettori a più alta dimensione nello spazio delle caratteristiche. Nelle applicazioni reali, le caratteristiche dei dati potrebbero essere molte e l'applicazione di trasformazioni che coinvolgono molte combinazioni polinomiali di queste caratteristiche porterebbe avere costi computazionali estremamente elevati e poco pratico. A tal proposito ci vengono in aiuto il metodo kernel (kernel trick) il quale rende molto meno laboriosa la mappatura dei dati a una dimensione superiore.

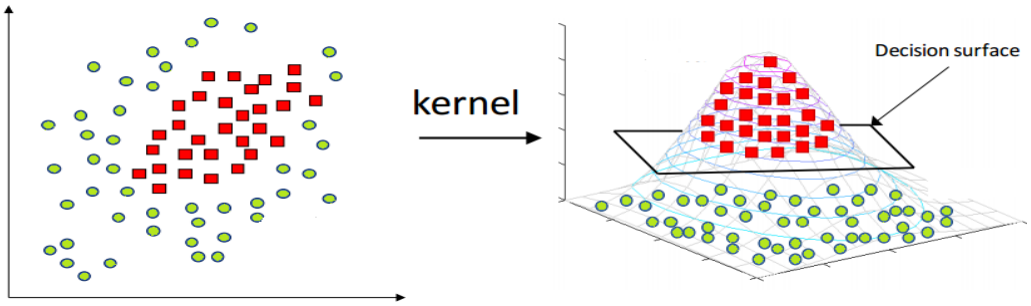

Figura 4 esempio di mappatura dei dati attraverso una funzione kernel, evidenziando l'iperpiano di separazione [14]

Questo metodo consiste in una trasformazione non lineare $\phi: x \rightarrow F$ che trasforma lo spazio delle feature di input *X* nello spazio delle feature *F* dove l'iperpiano di separazione permette di discriminare meglio le categorie. La funzione discriminante nello spazio *F* è

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b \qquad 3.7$$

Per permettere la separazione, normalmente lo spazio *F* è di dimensioni maggiori dello spazio *X*. Questo aumento di dimensioni provoca un aumento della complessità computazionale del problema e la richiesta di risorse. I metodi Kernel risolvono questo problema.

Il vettore $\mathbf{w}$ è una combinazione lineare dei campioni di addestramento (i *support vector)*:

$$\mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x}_i) \qquad 3.8$$

La funzione discriminante assume pertanto la forma

$$\begin{aligned} f(\mathbf{x}) &= \textstyle\sum_i \alpha_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}) + b \\ &= \textstyle\sum_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b \end{aligned} \qquad 3.9$$

con la valutazione della funzione kernel $k(x, x_i)$.
Questo metodo rappresenta i dati attraverso un insieme di confronti di similarità a coppie tra le osservazioni originali dei dati x. Infatti, invece di applicare esplicitamente le trasformazioni ϕ(x) e rappresentare i dati con queste coordinate trasformate nello spazio delle caratteristiche di dimensione superiore, si possono ottenere le coordinate dei dati nello spazio, calcolando, piuttosto, il prodotto interno tra le immagini di tutte le coppie di dati nello spazio *F*. Per capire come la funzione kernel sia uguale al prodotto del punto dei vettori trasformati, si può considerare che ogni coordinata del vettore trasformato *ϕ(x)* non è altro che una funzione delle coordinate del corrispondente vettore dimensionale inferiore *X*. [15][22]

3.1.3 Ottimizzazione con parametri

Come visto in precedenza il modello SVM risulta dipendente da differenti parametri, attraverso i quali è possibile ottimizzare il modello in base alla disposizione dei dati posseduti.

Uno dei principali parametri, visto nel cap. 3.1.2, è il Soft Margin "C", utilizzato come parametro per definire la penalità di un determinato in base alla posizione posseduta rispetto al margine originario, suggerendo al modello quali di questi punti possono essere considerati come vettori di supporto. Definendo, quindi, il grado di classificazione corretta che l'algoritmo deve soddisfare o il grado di ottimizzazione che SVM deve soddisfare.
Come si può notare della figura 5, se C è grande, il modello sceglie più punti di dati come vettore di supporto, ottenendo una varianza (errore dovuto alla sensibilità a piccole fluttuazioni nel set di addestramento) più alta e un bias più basso (errore dovuto ad assunzioni errate nell'algoritmo di apprendimento), che può portare al problema dell'overfitting.

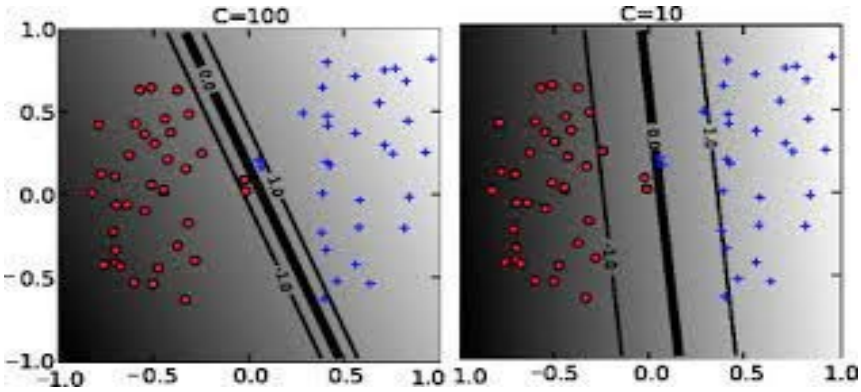

Figura 5: effetto del parametro "C" sulla previsione. A destra, Un basso "C" determina un margine più alto dando poca importanza ai punti vicini all'iperpiano, mentre a sinistra il contrario. [17]

Un altro parametro utilizzato per adattare meglio l'algoritmo ai dati sono le funzioni kernel. Queste possono essere di diverso tipo; lineare polinomiale, radiale e sigmoidee. Il grado del kernel polinomiale e il parametro di larghezza (γ) del kernel radiale controllano la flessibilità del classificatore risultante, più è alto il grado più flessibile è il confine decisionale adattandosi meglio ai dati. La polinomiale di grado più basso è il kernel lineare, che non è sufficiente quando esiste una relazione non lineare tra le caratteristiche. una polinomiale di grado 2, invece, è già abbastanza flessibile da discriminare tra le due classi con un margine considerevole. Figura 6a. [16]

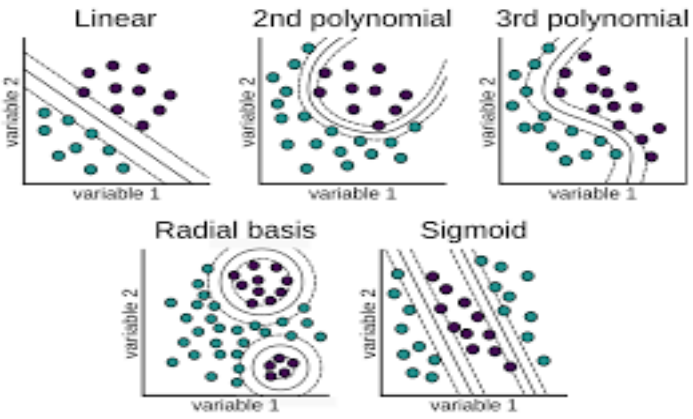

Figura 6a: effetto dei diversi tipi di kernel rispetto all' iperpiano di separazione.[18]

Per ultimo il parametro gamma, strettamente legato al kernel di tipo radiale ($k(x, x') = \exp(-\gamma\|x - x'\|^2)$), indica, intuitivamente, quanto lontano arriva l'influenza di un singolo punto nelle decisioni del modello: come mostrato nella figura 6b, valori di gamma elevati tendono a creare limiti di decisione "frastagliati" in quanto molto vicini ai punti appartenenti ad una classe; al contrario valori di gamma piccoli danno maggior peso ai punti lontani dal confine di decisione che assomiglierà più ad una retta. Valori troppo elevati di gamma generano il cosiddetto problema dell'overfitting, [17]

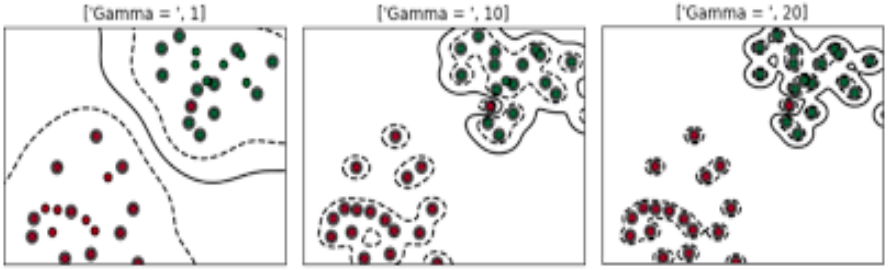

Figura 6b: effetto del parametro gamma sull'iperpiano di piano di separazione [18]

3.1.4 Applicazioni

L'algoritmo SVM di fronte alla sua versatilità risulta essere molto performante per la previsione di eventi glicemici e non solo. Questo viene dimostrato da differenti studi.

Lo studio [28], si pone l'obbiettivo di prevedere eventi di ipoglicemia notturna attraverso l'utilizzo dell'apprendimento supervionato. I modelli di previsione sono stati generati utilizzando due classificatori, tra cui SVM. Questa ricerca si sviluppa monitorando 10 pazienti durante 12 settimane in condizioni di vita libera, con l'obiettivo di raccogliere dati delle loro attività quotidiane. Durante questo periodo, ogni paziente ha utilizzato un CGM per monitorare la concentrazione di glucosio

interstiziale e ottenere i dati relativi all'attività fisica e ai periodi di sonno. Attraverso questi dati si è potuto creare una terapia specifica e personalizzata in grado di adattarsi al meglio alla fisiologia del paziente. A causa del numero limitato di istanze ottenute per alcuni pazienti, viene applicata k-fold la Cross Validation all'intero dei set di dati. Questo processo viene ripetuto un certo numero di volte per problemi legati alla dimensione del campione. Vengono scelti i vettori delle caratteristiche attraverso un opportuno metodo di selezione di features, e per ogni combinazione di caratteristiche viene eseguito un campionamento casuale stratificato, mantenendo così l'equilibrio tra le classi. Successivamente, viene condotta una k-fold Cross Validation (k = 5), dalla quale si ottengono i risultati di quella determinata iterazione. Tale processo viene ripetuto 100 volte per ogni vettore di caratteristiche. Il risultato finale viene poi ottenuto prendendo in considerazione la media aritmetica di queste misure, valuta attraverso tre metrice: specificity, accurancy e sensitivity.

Tabella 1: valori di predizione del SVM rispetto all'algoritmo di confronto.

Metrica	Sensitivity		Specificity		Accurancy	
Algoritmi	Alg. confronto	SVM	Alg. confronto	SVM	Alg. confronto	SVM
Media	69.52	78.75	78.98	82.15	77.98	80.77

Questi risultati confermano la fattibilità dell'approccio proposto. L'SVM ha ottenuto risultati migliori per quasi tutti i pazienti rispetto all'algoritmo confrontato, soprattutto nell'analisi di sensibilità. Considerando gli esiti mediani per l'intera coorte, quasi l'80% delle notti con ipoglicemia verrebbero evitate con questo algoritmo, mentre al tempo stesso, si otterrebbe una specificità superiore all'80%.

L'articolo [24] si concentra sulla previsione e prevenzione degli eventi ipoglicemici, utilizzando l'algoritmo SVM per valutare la risposta postprandiale. La previsione di questi eventi, quando un paziente annuncia un pasto, permette di valutare l'impatto dell'iniezione di insulina istantanea sulla risposta postprandiale consentendo di ottimizzarla per ottenere dosaggi più sicuri. Per questo programma viene utilizzato un kernel radiale risultato essere più performante, in questo caso, rispetto al kernel polinomiale. I dati utilizzati per l'implementazione vengono estratti da un dispositivo CGM e da una pompa insulinica, fornendoci differenti feature come: valore del BG, tasso di variazione della glicemia nei 30 minuti, valore medio glicemico nell'ultima ora, stima dell'insulina basale nelle ultime due ore, stima dell'insulina nelle successive 4 ore e stima dei carboidrati ingeriti.

Tabella 2: Valori predittivi dell'SVM per ipoglicemia postprandiale.

ipoglicemia	metrica	media	min	max
(A) BG < 70 mg/dL and BG ⩾ 54 mg/dL	sensitivity	69%	42%	82%
	specificity	80%	73%	97%
(B) BG < 54 mg/dL.	sensitivity	75%	60%	85%
	specificity	81%	60%	98%

Come mostra la Tabella 2, a sensibilità e la specificità sono rispettivamente del 69% e dell'80% per l'ipoglicemia A e del 75% e dell'81% per l'ipoglicemia B, potendo affermare che l'algoritmo risulta avere buone performance sia nella rilevazione di veri positivi e sia nella rilevazione di veri negativi. Questi risultati, ottenuti utilizzando una finestra di previsione di 4 ore dopo il consumo del pasto, risultano essere relativamente buoni permettendo di valutare e migliorare il dosaggio di insulina al paziente in concomitanza con i pasti.

Nella ricerca [21], in particolare, vediamo che i classificatori mostrano un miglioramento delle prestazioni quando si specializzano per rilevare un evento specifico (iperglicemia in questo caso). In questo studio, i dati utilizzati per l'implementazione dell'algoritmo vengono estrapolati solo attraverso le letture del monitoraggio in continua del glucosio (CGM), mentre gli input esogeni o qualsiasi altra informazione aggiuntiva non viene presa in considerazione, in quanto non disponibili nel dataset utilizzato. Per l'implementazione del SVM viene utilizzata la funzione kernel di tipo radiale, definendo come valore del parametro $\gamma = 1^{-7}$, portando l'iperpiano di separazione ad avere una sensibilità più alta nei confronti di punti lontani. Invece, per il parametro di penalizzazione viene utilizzato $C = 10^5$, con una conseguente maggiore adattabilità del piano ai dati considerati.

Tabella 3: valori per la valutazione predittiva del SVM

Metrica/evento	Iperglicemia	Iperglicemia grave
Recall	0.95 ± 0.06	0.93 ± 0.08
Precision	0.58 ± 0.13	0.60 ± 0.14
F1	0.71 ± 0.11	0.72 ± 0.12

L'SVM, pur avendo un F1 leggermente inferiore rispetto ad altri classificatori, risulta tuttavia possedere delle prestazioni molto migliori riguardo all'intervallo predittivo che va da 10 a 25 minuti.

Pertanto, anche se la misura F1 di SVM è leggermente inferiore, questo classificatore è in grado di prevedere gli eventi prima dei metodi di regressione, soprattutto nel momento in cui la classificazione riguarda una unica classe specifica. Ciò rende un classificatore specializzato la scelta migliore per gli algoritmi di previsione degli eventi di glucosio e un buon benchmark di prestazioni per gli sviluppi futuri.

3.2 Random Forest

Nel mondo del Machine Learning, i modelli Random Forest (RF) sono dei modelli non parametrici che possono essere utilizzati sia per la regressione che per la classificazione. Il Random Forest si basa su l'apprendimento ensemble, che cerca di migliorare la prestazione individuale di ciascun modello prendendo un insieme di essi. Questi metodi possono essere descritti come tecniche che utilizzano un gruppo di modelli deboli, al fine di crearne uno più forte e aggregato. Nel nostro caso, le foreste casuali sono un insieme di molti alberi decisionali individuali.

Classificatori basati su alberi di decisione hanno il vantaggio di essere semplici, di facile interpretazione ed efficienti a livello computazionale. Tuttavia, non sono in genere competitivi in termini di qualità delle predizioni con i migliori metodi di apprendimento supervisionati. Inoltre, se si cambiano di poco i dati di training, l'albero appreso potrebbe cambiare di molto, ossia manca stabilità (overfitting). In genere per ovviare a questi svantaggi è stato proposto il RF, implementato attraverso tecniche di ensemble, come il bagging. Mirando a costruire molteplici alberi che poi vengono combinati in una singola predizione. Questo può portare notevoli miglioramenti in termini di accuratezza delle predizioni, a svantaggio di una perdita parziale di interpretabilità del modello.

In una foresta di alberi le prestazioni sono di gran lunga migliori e non ci si deve preoccupare tanto della regolazione perfetta dei parametri della foresta a differenza dei singoli alberi.

3.2.1 Alberi decisionali e split

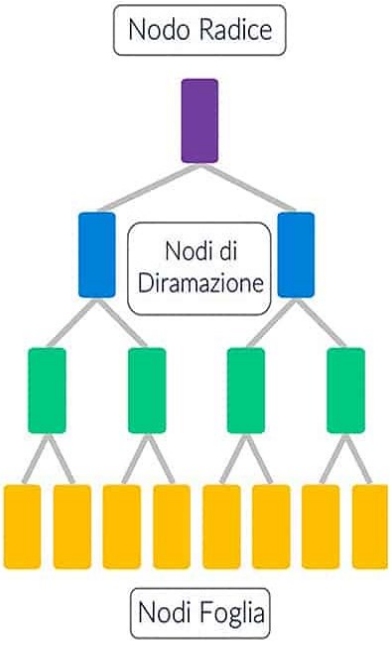

Figura 9: rappresentazione stilizzata di un albero decisionale

Per sapere come funziona un algoritmo di foresta casuale dobbiamo conoscere gli alberi decisionali, che sono un altro algoritmo di apprendimento automatico supervisionato utilizzato per problemi di classificazione e regressione. Gli alberi decisionali utilizzano un diagramma di flusso come una struttura ad albero per mostrare le previsioni, che risultano da una serie di suddivisioni basate sulle caratteristiche. Inizia con un nodo radice e termina con una decisione presa dalle "foglie". È costituito da tre componenti: il nodo radice, il nodo decisione e il nodo foglia. Il nodo da cui la popolazione inizia a dividersi è chiamato nodo radice. I nodi che si ottengono dopo la divisione del nodo radice sono chiamati nodi decisione e il nodo in cui non è possibile un'ulteriore divisione è chiamato nodo foglia. [20]

Un albero decisionale apprende dividendo i nodi in sotto nodi. Questo processo viene definito come *split* ed eseguito più volte durante il processo di addestramento fino al raggiungimento di soli nodi omogenei. Tale divisione può avvenire attraverso diverse metodologie in conforme si stia trattando problemi di classificazione o regressione.
Un metodo di divisione per trattare problemi di regressione ossia nel momento in cui la variabile target e continua, è la riduzione della varianza che mi definisce l'omogeneità del nodo stesso, se un nodo è completamente omogeneo la varianza è zero.

$$Varianza = \frac{\sum(x-\eta)^2}{N} \qquad 3.10$$

Definendo x come variabile, η la media dei valori nel nodo e N il numero di campioni totale. Quindi per ogni divisone passando da nodo padre a nodo figlio si cercherà di diminuire la varianza scegliendo lo split più efficiente.

Nel caso in cui la nostra variabile target è categoriale, trattando un problema di classificazione, valutare la varianza non è più sufficiente. Si ricorre al guadagno di informazioni che misura la quantità di "informazioni" che una caratteristica ci fornisce sulla classe, per questo viene utilizzato per decidere l'ordine degli attributi nei nodi di un albero decisionale ed è definito come:

$$information\ gain = 1 - H \qquad 3.11$$

Dove H è definita come entropia ed è.

$$\mathrm{H(x)} = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \qquad 3.12$$

Da cui x e la variabile, $P(xi)$ probabilità di un possibile risultato. L'entropia è la misura di impurità o disordine, di una distribuzione statistica (in questo caso un nodo), che controlla il modo in cui un albero decisionale decide di dividere i dati. Infatti, un nodo molto disordinato avrà un valore di entropia pari ad 1, mentre un nodo puro (ordinato) avrà un valore di entropia pari a zero. Intuitivamente L'addestramento del modello consisterà nel raggiungere le foglie dell'albero (nodi puri) utilizzando meno condizioni possibili, scegliendo lo split con un guadagno di informazione più alto.

Un altro modo attraverso cui si calcolare l'impurità di un nodo è attraverso l'indice di Gini:

$$Gini\ index = 1 - \sum_{i=1}^{n} (P_i)^2 \qquad 3.13$$

L'indice di Gini è una misura della eterogeneità (omogeneità) di una distribuzione statistica a partire dai valori delle probabilità relative associate alle differenti modalità di una generica variabile X. Se i dati sono distribuiti in modo eterogeneo l'indice di Gini è elevato. Indice raggiunge il suo massimo quando tutte le numerosità delle differenti classi sono uguali. Viceversa, in caso di distribuzione di probabilità omogenea (tutti gli individui appartengono ad una sola classe) l'indice sarà pari a 0. Questo viene spesso usato al posto del guadagno di informazione per decidere la divisione dei nodi, per il semplice motivo che non avendo logaritmi risulta essere più leggero a livello computazionale.

L'ultimo metodo che si può utilizzare per valutare l'efficienza di uno split in un albero decisionale è il chi-square, definito come:

$$Chi - Square = \sqrt{\frac{(valore\ vero - valore\ previsto)^2}{valore\ previsto}} \qquad 3.14$$

Dove si indica per "Valore Previsto" il valore atteso per una classe su un nodo figlio in base alla distribuzione delle classi sul nodo padre mentre, "Valore Vero" indica il valore reale di una classe in un nodo figlio. Tale indice mi identifica la differenza che viene a presentarsi tra padre e figlio durante lo split, più questo sarà alto maggiore sarà l'omogeneità nel nodo figlio. Durante la ricerca della divisione più efficiente bisognerà calcolare la somma dei chi-square per tutte le classi presenti nei nodi figli e prendere in considerazione tra i vari split quello con valore maggiore.[22]

3.2.2 Apprendimento nel Random Forest

Una volta definito i componenti di un Random Forest, gli alberi decisionali, possiamo andare a definire meglio come questi vengono addestrati per sfruttare efficacemente un insieme decisionale di questo tipo. Per la fase di addestramento si utilizzano metodi di insieme definiti come ensemble che usano modelli multipli per ottenere una migliore prestazione predittiva rispetto ai modelli da cui è costituito. Questi, solitamente, raggiungono prestazioni elevatissime, soprattutto se rapportati ai classificatori singoli. Naturalmente richiedono molto più tempo di addestramento, in quanto al posto di un solo classificatore devono essere addestrati centinaia o migliaia di classificatori. La tecnica che viene utilizzata maggiormente è il bagging che mira a creare un insieme di classificatori aventi la stessa importanza. All'atto della classificazione, ciascun modello voterà circa l'esito della predizione e l'output complessivo sarà la classe che avrà ricevuto il maggior numero di voti.

Nel momento in cui si addestra un una foresta utilizzando un determinato set di dati, si presenta il rischio che sui componenti, gli alberi decisionali, si adattino molto ai dati in input. Per evitare ciò, si tenta di rendere indipendente ciascun albero, così da non cadere in overfitting. Per rendere l'algoritmo il più generalizzabile possibile vengono utilizzate due tecniche, le quali caratterizzano parte del nome stesso dell'algoritmo, "random".

Una di queste è, proprio, il bagging è una tecnica di ri-campionamento statistico che prevede il campionamento casuale di un insieme di dati con sostituzione, (figura 7). Viene spesso utilizzato come mezzo per quantificare l'incertezza associata a un modello di apprendimento automatico. Il bagging è estremamente utile perché consente di generare nuovi campioni da una popolazione senza dover raccogliere ulteriori dati di addestramento. L'idea consiste nel campionare ripetutamente i dati con sostituzione dall'insieme di formazione originale per produrre più insiemi di formazione separati, così da diminuire il più possibile la correlazione tra gli alberi. Solitamente, la maggior parte delle implementazioni di Random Forest seleziona il campione dei dati di addestramento, utilizzato per ogni albero, in modo che abbia le stesse dimensioni del set di dati originale per consentire ai metodi di ridurre la varianza delle loro previsioni, migliorando così notevolmente le loro prestazioni predittive.[22]

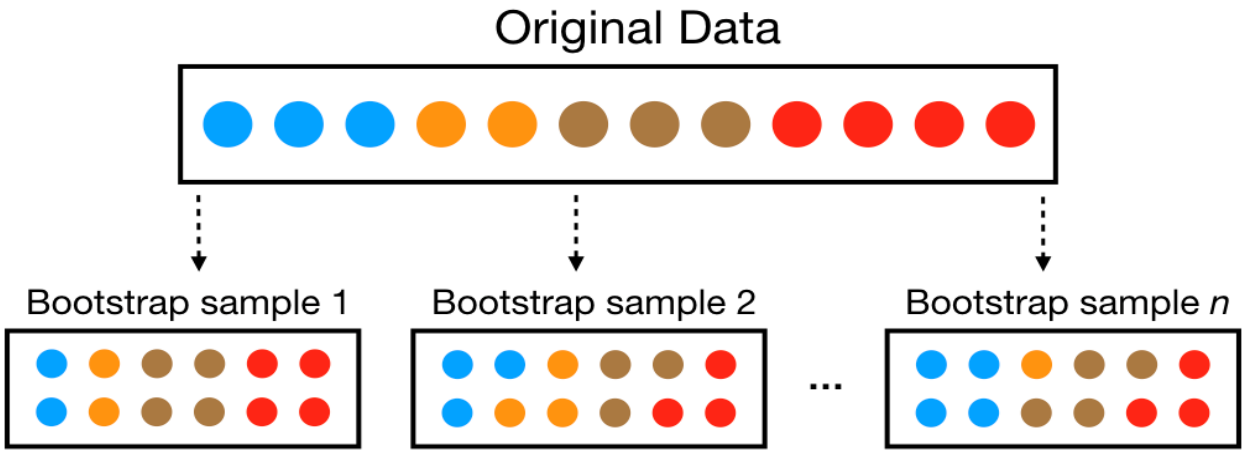

Figura 7: esempio grafico dell'esecuzione della tecnica bagging [23]

Un altro metodo che si può utilizzare, per aumentare maggiormente l'indipendenza di ciascun albero, è la selezione di caratteristiche casuali (feature randomness), che consiste nello scegliere sempre casualmente alcune caratteristiche dei campioni iniziali del data set originale e implementarli sui rispettivi alberi decisionali. Ciò può sembrare controintuitivo, in quanto spesso si desidera includere inizialmente il maggior numero possibile di caratteristiche per ottenere il maggior numero di informazioni per il modello. Tuttavia, ha lo scopo di evitare deliberatamente caratteristiche predittive molto forti che portano a suddivisioni simili negli alberi (e quindi aumentano la correlazione) aumentandone l'accuratezza predittiva. [22]

Osservando la successiva figura 8, l'albero decisionale tradizionale (in blu) può scegliere tra tutte e quattro le caratteristiche per decidere come dividere il nodo. Avendo a disposizione tutto il database, si decide di scegliere la caratteristica 1 (nera e sottolineata), che divide i dati in gruppi il più possibile separati. Invece, ponendo l'attenzione solo su due alberi della foresta (in verde) scopriamo che può considerare solo le caratteristiche 2 e 3 (selezionate in modo casuale) per la suddivisione dei nodi. Sappiamo dal nostro albero decisionale tradizionale (in blu) che la caratteristica 1 è la migliore per la divisione, ma l'albero 1 non può vedere la caratteristica 1, quindi è costretto a scegliere la caratteristica 2 (nera e sottolineata). L'albero 2, invece, può vedere solo le caratteristiche 1 e 3, quindi è in grado di scegliere la caratteristica 1. Alla fine, nella nostra foresta casuale, ci ritroviamo con alberi che non solo sono addestrati su diversi set di dati (grazie al bagging), ma che utilizzano anche caratteristiche diverse per prendere decisioni, dando vita così ad alberi non correlati.

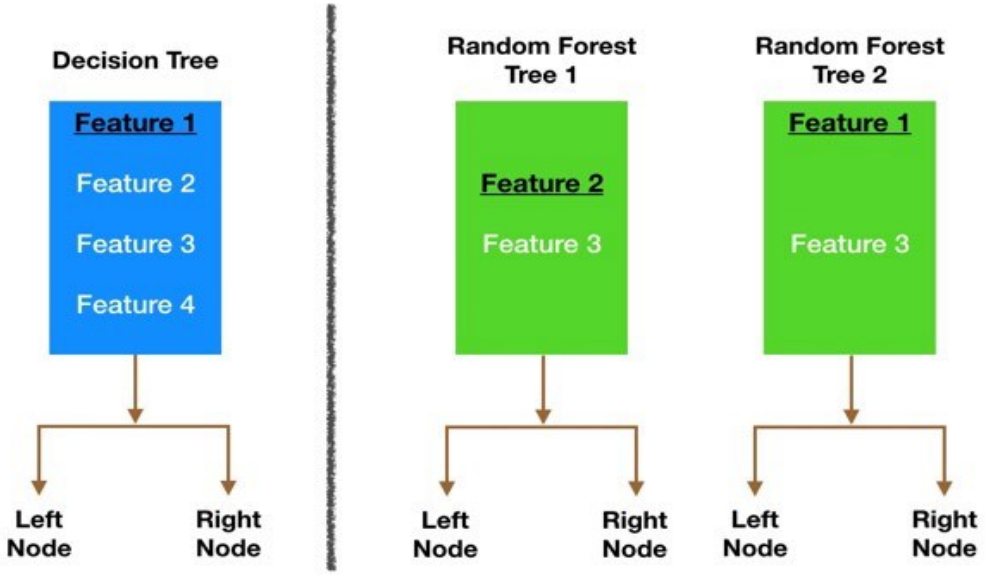

Figura 8: esempio di feature randomness comparando un albero decisionale con un Random Forest.[19]

Successivamente, una volta allenati singolarmente gli alberi decisionali che compongono la foresta, ciascuno di essi produrrà un suo specifico output, caratterizzato dal processo di bagging e feature randomness, che verranno utilizzati per definire la predittività della Random Forest attraverso una media di tutti i differenti output o una selezione per maggioranza in conforme si stia trattando di un algoritmo classificatore o regressore. [19]

3.2.3 Implementazione e bagging

Come accennato nel precedente paragrafo l'idea fondamentale alla base del bagging è quella di calcolare la media (o maggioranza) di molti modelli contenenti errore, ma approssimativamente non distorti, in modo da ridurre la varianza. Gli alberi sono candidati ideali per il bagging, poiché possono catturare strutture di interazione complesse presenti nei dati. Poiché gli alberi sono notoriamente soggetti ad errore, possono beneficiare in maniera importante dal calcolo della loro media. Perdipiù, ogni albero generato nel bagging proviene da una distribuzione identica (i.d.: identicamente distribuito), quindi il valore atteso di una media di B alberi sarà lo stesso valore atteso di uno qualunque di loro, e la sola opzione di miglioramento sarà tramite la riduzione della varianza. Infatti, una media di variabili B casuali i.d., ciascuna con varianza σ^2, ha varianza $\frac{1}{B}\sigma^2$. Se le variabili sono semplicemente i.d. (identicamente distribuite, ma non necessariamente indipendenti) con correlazione a coppie ρ positiva, la varianza della media è

$$p\sigma^2 + \frac{1-p}{B}\sigma^2 \qquad 3.15$$

Al crescere di B, il secondo termine della somma si riduce, mentre il primo rimane, e quindi il valore della correlazione di coppie di alberi limita i benefici del calcolo della media. Di conseguenza lo scopo da raggiungere per la costruzione di un Random Forest efficiente è quello di migliorare la riduzione della varianza del bagging riducendo la correlazione tra gli alberi, Questo è raggiunto nel processo di crescita degli alberi attraverso la selezione casuale delle variabili e caratteristiche in input

In particolare, quando si addestra un albero su un dataset "bagging", in ogni split dell'albero si seleziona causalmente m<p delle variabili di input come candidate per lo split. I quali valori di m, solitamente oscillano tra $\sqrt{p}$ e 1. Dopo aver fatto crescere B alberi $T_b(x, \theta_b)$ in questo modo, il predittore basato su foresta casuale sarà:

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x, \theta_b) \qquad 3.16$$

Dove θ_b caratterizza il b-mo albero della foresta casuale in termini di variabili di split, punto di split in ogni nodo, valori del nodo terminale; $T_b(x, \theta_b)$ è la previsione del b-mo albero della foresta casuale. Intuitivamente, la riduzione di m produrrà una riduzione della correlazione tra coppie di alberi nell'insieme, e quindi ridurrà la varianza della media.[22]
L'algoritmo del Random Forest può quindi essere riassunto come:

1. Per b=1…Bb=1…B:
    a. Estrai un campione bagging Z di dimensione N dai dati di training.
    b. Fai crescere un albero T_b della foresta casuale su dati "bootstrapped", ripetendo ricorsivamente i seguenti passaggi per ciascun nodo terminale dell'albero, fino al raggiungimento della dimensione minima dei nodi n.
        i. Seleziona mm variabili a caso tra le pp variabili.
        ii. Seleziona la migliore variabile e punto di split tra le mm.
        iii. Dividi (split) il nodo in due nodi figli.
2. Ritorna l'insieme degli alberi $\{T_b\}$ [22]

3.2.4 Applicazioni

In ambito medico il ruolo dell'apprendimento automatico sta avendo sempre di più un ruolo fondamentale nella cura e nella terapia di diverse malattie. Una di queste, il diabete mellito negli

ultimi anni ha usufruito maggiormente di questa tecnologia, che grazie anche all' introduzione dei CGM, ha permesso lo sviluppo di nuovi approcci all'analisi dei dati offrendo l'opportunità di prevedere il livello di glucosio. Tra questi algoritmi, uno molto utilizzato è il Random Forest, il quale risulta essere efficacie per la previsione di eventi ipo/iperglicemici. Questo viene dimostrato in diversi studi.

Uno di questi analizzando 50 registrazioni di CGM ottenute nel arco di 48 ore da pazienti affetti da diabete mellito di tipo 1 e di tipo 2. Sono state scelte varie finestre predittive per prevedere l'andamento del glucosio in questi pazienti diabetici. Concludendo che l'algoritmo Random Forest può essere utilizzato con successo per prevedere i livelli di glucosio dei pazienti diabetici sulla base dei dati del CGM. Nella finestra predittiva di 50-120 minuti (un periodo legato alla glicemia postprandiale) in cui il modello ha una buona accuratezza predittiva.[25]

La ricerca [26] pone l'attenzione su eventi ipoglicemici valutandoli attraverso due algoritmi: classificatore Random Foreste (RF) e regressione lineare (LR) (non trattato in questo elaborato). I set di dati CGM sono stati ottenuti da 112 pazienti che hanno utilizzato dispositivi CGM Dexcom G6 per un periodo di 90 giorni, con valori CGM in condizioni di vita normali. Vengo estratte In totale 26 caratteristiche dal dispositivo; queste si possono suddividere in diverse categorie: features a corto, medio e lungo periodo (valutano il livello del glucosio in diverse finestre temporali), demografiche, non lineare e di interazione (valuta le fluttuazioni del livello di glucosio), "Snowball "(valutano gli effetti della variabilità glicemica) e contestuali (come l'assunzione di carboidrati o l'iniezione di insulina). Questi dati, nel Random Forest, vengono implementati attraverso una metodologia differente (VIP), che viene utilizzata per ordinare le caratteristiche in base al loro impatto sulla classificazione, così facendo le caratteristiche con un impatto marginale possono essere escluse. A differenza del LR il quale necessità di tutte e 26 le feature per realizzare la previsione. Per il confronto dell'efficacia predittiva vengono utilizzate due misure: sensibility e specificity descritte precedentemente al paragrafo 2.4. Gli RF sembrano essere in grado di catturare i modelli complessi e non lineari che influenzano l'ipoglicemia meglio degli LR. È evidente, dai risultati (Tabella 4) che non solo si è stati in grado di prevedere con precisione i veri eventi ipoglicemici con durata inferiore, ma anche più lunghi, fino a 60 minuti, con una precisione superiore al 90%. Ciò ha implicazioni cliniche importanti dato che il tempo guadagnato con la previsione fornisce al paziente una maggiore flessibilità per rispondere.

Tabella 4: performance predittive di eventi glicemici: RL (regressore lineare), RF with VIP (Random Forest implementato con il metodo VIP), RF with VIP and carb/insul. (Random Forest implementato con VIP con dati di insulina e carboidrati).

PH	0-15 min		15-30 min		30-45		45-60	
	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
LR	94.27	97.32	84.19	94.06	68.08	92.90	53.87	92.88
RF with VIP	95.82	97.21	93.61	93.50	92.28	89.59	89.37	89.16
RF with VIP and carb/insul	98.35	97.75	97.04	95.23	96.92	94.89	96.21	95.73

Il modello RF ottimizzato per VIP richiede solo 9 features per raggiungere prestazioni elevate, mentre l'LR ne richiede 21. La maggior parte dei features utilizzate nel modello RF ottimizzato per VIP sono di durata da breve a media. Infine, si può notare che i dati sull'insulina e sui carboidrati hanno migliorato le prestazioni in generale lungo tutto l'arco temporale considerato, in particolar modo per le previsioni da 30 a 60 minuti. Questo studio evidenzia la necessità di disporre di questi dati in tempo reale per facilitare le previsioni ipoglicemiche a lungo termine e la possibilità di realizzare modelli predittivi per i nuovi pazienti senza raccogliere grandi quantità di dati.

In [27], viene presentato uno studio sistematico su Random Foster per la regressione con un set di dati multivariato per la previsione della concentrazione di glucosio nel diabete di tipo 1. Infatti, le variabili che vengono prese in considerazione in questa ricerca sono diverse: il profilo del glucosio (gl), la concentrazione plasmatica di insulina (Ip), concentrazione plasmatica di insulina (Ip), la velocità di comparsa del glucosio esogeno (derivato dal pasto) nel plasma (Ra), la quantità cumulativa di glucosio esogeno comparso nel plasma (SRa), l'ora del giorno (1 - 24) (h) e la spesa energetica cumulativa (SEE). Per valutare la capacitò predittiva, vengono esaminiate tre diversi casi di input. Nel primo caso, denominato Caso 1, la previsione del glucosio si basa solo sul profilo del glucosio passato (gl). Nel secondo caso, indicato come Caso 2, le variabili di ingresso $Ip, Ra, SR\alpha \; e \; h$ vengono aggiunte all'input della funzione predittiva. L'ultimo caso, ovvero il Caso 3, risulta dall'aggiunta della variabile SEE all'input del Caso 2. Infine, le previsioni vengono eseguite per quattro valori dell'orizzonte di previsione l, ossia l5, 30, 60 e 120 min. Per la valutazione delle

performance essendo un regressore viene utilizzata la tecnica Cross Validaton con l'RMSE (paragrafo 2.4. e 2.3)

Tabella 5: valori RMSE in base ai dati in input utilizzati

Orizzonte predittivo	15 min	30 min	60 min	120 min
Metodo di valutazione	RMSE	RMSE	RMSE	RMSE
Case 1	9.84	15.37	23.43	31.04
Case 2	6.99	8.98	11.00	12.38
Case 3	6.60	8.15	9.25	10.83

Osservando il Caso 1 si nota un errore sufficientemente basso per quanto riguarda le previsioni a breve termine (cioè per 15 e 30 minuti) della concentrazione di glucosio. Tuttavia, come indicato da entrambe le misure, il profilo del glucosio previsto si discosta in modo significativo da quello reale per gli orizzonti di previsione a medio e lungo termine (cioè per 60 e 120 minuti). L'introduzione delle variabili di input $Ip, Ra, SRa\ e\ h$ nel Caso 2 riduce di 10 volte l'RMSE medio associato alle variabili di input associato alle previsioni a 15 e 30 minuti. Ottenendo un miglioramento effettivo del 29% e del 42%, nei primi due orizzonti temporali, rispetto al Caso 1. Inoltre, le previsioni a 60 e 120 minuti sono migliorate rispettivamente del 53% e del 60% rispetto al caso 1. Con l'aggiunta della variabile SEE, nel Caso 3, si nota notevolmente il miglioramento rispetto al Caso 1. In Generale, l'algoritmo RF risulta avere delle buone capacità predittive nel breve e medio periodo, le quali possono essere notevolmente migliorate con l'introduzione di featurs reali specifiche.

Conclusioni

Il diabete risulta essere una delle malattie più diffuse a livello globale. Secondo le stime la percentuale di malati aumenterà drasticamente, portando con sé disagi a livello clinico, sociale ed economico. Una soluzione a questi problemi è stato il passaggio da dispositivi SMBG a dispositivi CGM, portando una diminuzione notevole dei costi terapeutici e soprattutto, un aumento della semplicità di utilizzo; dando la possibilità a qualsiasi persona affetta da diabete, di poter condurre una vita normale, in quanto tali dispositivi richiedono molta meno attenzione ed esperienza rispetto a quelli precedenti. Inoltre, l'introduzione di dispositivi CGM ha cambiato drasticamente il modo di approcciarsi allo studio del diabete, aprendo nuove prospettive, tra cui l'utilizzo di Machine Learning. Tale tecnica, attraverso l'identificazione automatica di specifici pattern all'interno dei dati, evidenzia delle correlazioni che consentono di esprimere delle "predizioni" con ragionamenti di tipo induttivo tipici della mente umana. In campo diabetologico, questa nuova tecnologia ha contribuito ad un ulteriore passo in avanti, dando la possibilità di intervenire su eventi ipo/iperglicemici prima che questi accadano, salvando il paziente da possibili complicanze. In questo elaborato sono stati presentati due algoritmi di apprendimento risultati molto efficaci per la previsione glicemica, dimostrando, con diverse applicazioni, la loro affidabilità e prestanza in questo ambito.

In futuro, la terapia del diabete si baserà sempre di più su tecniche di apprendimento automatico, ponendosi l'obbiettivo di aumentare l'efficacia e l'accuratezza della previsione utilizzando database più dettagliati e nuove tecniche di addestramento.

Bibliografia

[1] Pathophysiology of diabetes mellitus. Guthrie RA, Guthrie DW. Crit Care Nurs Q.

[2] WHO. https://www.who.int/news-room/fact-sheets/detail/diabetes

[3] https://www.salute.gov.it/

[4] https://www.diabete.net/caratteristiche-cause-e-sintomi

[5] Kaul K., Tarr J. M., Ahmad S. I., Kohner E. M. and Chibber R., Introduction to diabetes mellitus

[6] https://www.msdmanuals.com

[7] Cappon G., Acciaroli G., Vettoretti M., Facchinetti A. and Sparacino G., Wearable Continuous Glucose Monitoring Sensors: A Revolution in Diabetes Treatment

[8] https://www.dexcom.com/g6-cgm-system

[9] Introduction to Machine Learning, Neural Networks, and Deep Learning Rene Y Choi 1, Aaron S Coyner 2, Jayashree Kalpathy-Cramer 3, Michael F Chiang 1 2, J Peter Campbell 1

[10] Machine Learning in Medicine Rahul C Deo

[11] Prediction of Adverse Glycemic Events from Continuous Glucose Monitoring Signal, Matteo Gadaleta, Andrea Facchinetti, Enrico Grisan, Michele Rossi

[12] https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152

[13] https://www.baeldung.com/cs/svm-hard-margin-vs-soft-margin

[14] https://medium.com/@zxr.nju/what-is-the-kernel-trick-why-is-it-important-98a98db0961d

[15] The Elements of Statistical Learning, Data Mining, Inference, and Prediction Trevor Hastie Robert Tibshirani Jerome Friedman

[16] Non-Invasive Glucose Monitoring Using Optical Sensor and Machine Learning Techniques for Diabetes Applications Maryamsadat Shokrekhodaei1 [Graduate Student Member, IEEE], David P. Cistola2, Robert C. Roberts1 [Member, IEEE], Stella Quinones3

[17] A User's Guide to Support Vector Machines Asa Ben-Hur and Jason Weston

[18] https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/

[19] https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[20] Random Forests Leo Breiman

[21] Prediction of Adverse Glycemic Events from Continuous Glucose Monitoring Signal Matteo Gadaleta, Andrea Facchinetti, Enrico Grisan, Michele Rossi

[22] The Elements of Statistical Learning, Data Mining, Inference, and Prediction Trevor Hastie Robert Tibshirani Jerome Friedman

[23] https://bradleyboehmke.github.io/HOML/process.html

[24] Prediction and prevention of hypoglycaemic events in type-1 diabetic patients using machine learning Josep Vehí [1], Iván Contreras, Silvia Oviedo, Lyvia Biagi, Arthur Bertachi

[25] Glucose trend prediction in diabetic patients using Random Forests algorithm for analysis of CGM data K. Tabakov1, N. Myakina2; 1Department of Information Technology, Novosibirsk State University, Russian Federation, 2Laboratory of Endocrinology, Institute of Clinical and Experimental Lymphology, Novosibirsk, Russian Federation

[26] Feature-Based Machine Learning Model for Real-Time Hypoglycemia Prediction Article *in* Journal of Diabetes Science and Technology June 2020

[27] A Predictive Model of Subcutaneous Glucose Concentration in Type 1 Diabetes Based on Random Forests Eleni I. Georga, Vasilios C. Protopappas, Demosthenes Polyzos, and Dimitrios I. Fotiadis, Senior Member, IEEE

[28] Prediction of Nocturnal Hypoglycemia in Adults with Type 1 Diabetes under Multiple Daily Injections Using Continuous Glucose Monitoring and Physical Activity Monitor Arthur Bertachi , Clara Viñals , Lyvia Biagi , Ivan Contreras [1], Josep Vehí , Ignacio Conget , Marga Giménez