UNIVERSITÀ DEGLI STUDI DI PAVIA

FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Bioingegneria

Apprendimento Computazionale in Biomedicina

Analisi del dataset 'Thyroid Gland Data'

Federica Commisso

A.A. 2022/2023

Indice

1 Problema e obiettivi dell'analisi

L'analisi proposta ha come focus la creazione di un modello in grado di classificare in modo accurato le diagnosi legate ai problemi tiroidei. I disordini tiroidei sono le disfunzioni endocrine più diffuse globalmente. L'Organizzazione Mondiale della Sanità riporta che circa un miliardo di persone soffrono di tali disordini in tutto il mondo.

Nello specifico, l'ipotiroidismo spontaneo colpisce l'1-2% delle donne, con un'incidenza che aumenta notevolmente con l'avanzare dell'età. Questo disturbo è dieci volte più frequente nelle donne rispetto agli uomini. Analogamente, l'ipertiroidismo colpisce lo 0,5-2% delle donne, con una prevalenza dieci volte superiore rispetto agli uomini [1].

Nelle sezioni successive, verrà presentato il problema oggetto di analisi. Inizieremo con una descrizione dettagliata della tiroide e delle sue funzioni, seguita da un'ampia panoramica sugli studi finora condotti in questo campo. Infine, concluderemo introducendo gli obiettivi posti da questa analisi.

1.1 Studio fisiologico: la tiroide

La tiroide è un'importante ghiandola endocrina che svolge un ruolo cruciale nel corpo umano. Essa è responsabile della produzione di ormoni essenziali che influenzano il metabolismo, la crescita e lo sviluppo del nostro organismo, oltre a regolare varie funzioni corporee mediante il rilascio costante di ormoni nel flusso sanguigno.

La ghiandola tiroidea sintetizza tre principali ormoni:

- **Triiodotironina**, T3
- **Tiroxina**, T4

- **Calcitonina**

I primi due sono noti come **ormoni tiroidei** e differiscono dal terzo per la loro composizione, che include iodio. Questo elemento deve essere introdotto nel nostro organismo con la dieta poiché non siamo in grado di produrlo, ma solo di trasportarlo attraverso il flusso sanguigno alla ghiandola tiroidea. Qui, lega chimicamente l'aminoacido tirosina per formare rispettivamente T3 e T4. La calcitonina, invece, è prodotta dalle cellule C (parafollicolari) della tiroide e contribuisce alla regolazione dei livelli di calcio e fosforo nell'organismo.

Gli ormoni T3 e T4 circolano nel sistema sanguigno in due forme: una legata a una proteina chiamata TBG, che facilita il loro trasporto nel corpo, e l'altra forma "libera", meno abbondante ma coinvolta in molteplici processi corporei.

La sintesi e il rilascio degli ormoni tiroidei sono rigorosamente controllati da meccanismi di feedback. Un importante regolatore di questa produzione è l'**ormone tireotropina** (TSH), la cui secrezione è stimolata dall'ormone ipotalamico TRH e controllata dagli ormoni tiroidei circolanti. L'ipofisi anteriore, situata alla base del cervello, è responsabile della produzione e del rilascio di TSH, che agisce direttamente sulle cellule follicolari della tiroide.

Quando la tiroide non funziona correttamente, si verificano **squilibri dei valori ormonali**. L'ipertiroidismo si verifica quando la tiroide produce eccessivamente ormoni tiroidei, mentre l'ipotiroidismo è causato da una produzione insufficiente di questi ormoni. Entrambi questi disturbi comportano una serie di sintomi clinici [2, 3].

L'ipertiroidismo può portare a presentare molti sintomi, tra cui sudorazione, aritmia (battito cardiaco irregolare), perdita di peso, occhi sporgenti e nervosismo. Al contrario, I sintomi dell'ipotiroidismo posso-

no includere stanchezza, aumento di peso, depressione, sviluppo osseo anomalo, crescita stentata e raucedine [4].

1.2 Revisione della letteratura

Salman, Sonuç [5]. Nello studio sono stati impiegati 8 diversi algoritmi di machine learning per classificare i disturbi della tiroide in tre categorie: ipotiroidismo, ipertiroidismo e normale. Il loro dataset era composto da 1250 casi, ciascuno con 19 attributi. Tra gli algoritmi testati, l'accuratezza migliore è stata ottenuta con l'uso del Multi-Layer Perceptron (MLP), che ha raggiunto un impressionante 96.4%. Gli altri algoritmi, in ordine decrescente di accuratezza, includono Decision Tree (90.13%), SVM (92.53%), Random Forest (91.2%), Naive Bayes (90.67%), Logistic Regression (91.73%), Linear Discriminant Analysis (83.2%), e K-Neighbors Neighbour (91.47%).

Razia, Kumar, Rao [6]. Gli autori hanno cercato di predire le malattie della tiroide utilizzando tecniche di data mining, tra cui modelli basati su alberi e modelli lineari. Il loro dataset comprendeva 9172 pazienti, ognuno con 31 caratteristiche, tra cui età, sesso, TSH, T3 e T4. Hanno utilizzato una 10-fold cross-validation per valutare le prestazioni degli algoritmi e hanno riportato un'accuratezza fino al 98% con l'algoritmo Random Forest. Hanno inoltre scoperto che gli algoritmi basati su modelli lineari non hanno prestazioni eccellenti a causa delle dimensioni ridotte del dataset e delle caratteristiche. Sia la Linear Regression che il Support Vector Machine hanno ottenuto un'accuratezza dell'85%.

Chaubey, Bisen, Arjaria, Yadav[7]. Nello studio è stato utilizzato il dataset 'Thyroid gland data' per rilevare la presenza o assenza di disfunzioni della tiroide, trasformando il problema in una classificazione binaria. Hanno selezionato solo gli attributi 'T3' e 'T4' e suddiviso il dataset in training (70%), test (15%) e validation (15%). Gli autori hanno te-

stato tre classificatori noti: Logistic Regression (81.25% di accuratezza), Decision Tree (87.5% di accuratezza) e k-Nearest Neighbour (96.875% di accuratezza), con quest'ultimo classificatore che si è dimostrato il migliore.

In sintesi, questi studi dimostrano l'efficacia di diversi algoritmi di machine learning nella diagnosi dei disturbi tiroidei. L'accuratezza varia tra gli studi, ma è evidente che alcuni algoritmi possono ottenere risultati molto promettenti in questa area di ricerca.

1.3 Studio del problema

L'analisi che è stata condotta aveva come obiettivo la **classificazione** del funzionamento della tiroide (normale, ipertiroidismo, ipotiroidismo) osservando gli esami di laboratorio condotti. Nel contesto clinico, le tecniche di data mining risultano essere un supporto fondamentale alle decisioni in medicina.

Il procedimento adottato nello studio è riportato in Figura 1.1:

1. Selezione del dataset
2. Separazione dei dati in training e test set
3. Bilanciamento e pre-processing del training set
4. Allenamento del modello con diversi classificatori
5. Scelta dei classificatori migliori
6. Apprendimento del modello sul test set
7. Valutatazione delle performance

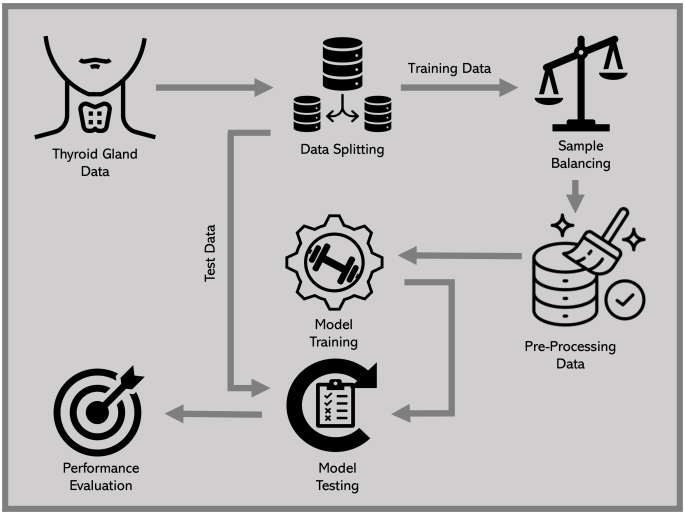


Figura 1.1: *Pipeline*

2 Comprensione e Preparazione dei Dati

Il dataset utilizzato in questo studio proviene dall'UCI Machine Learning Repository ed è specifico per le malattie della tiroide. Comprende 215 osservazioni, ciascuna delle quali è descritta da 5 attributi e la relativa classe di diagnosi (la Tabella 2.1 presenta la descrizione del dataset UCI). Nella Tabella 2.2, sono fornite le descrizioni degli attributi e i rispettivi tipi di dati.

La Figura 2.1 mostra le distribuzioni degli attributi e le relative statistiche, evidenziando le distinzioni tra le classi: classe 1 (rappresentata in azzurro), classe 2 (rappresentata in rosso) e classe 3 (rappresentata in verde). È importante notare che non sono presenti dati mancanti per nessun attributo.

Questi dati sono stati raccolti attraverso esami del sangue e analisi mediche per diagnosticare i casi di ipotiroidismo e ipertiroidismo. La distribuzione delle diagnosi è riportata nella Tabella 2.3, dalla quale emerge che il dataset presenta uno sbilanciamento tra le classi, con un numero significativamente maggiore di casi nella classe 1 rispetto alle altre classi.

Features	**Istanze**
5	215

Tabella 2.1: *Descrizione del dataset*

Feature Statistics

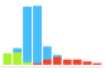
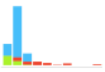
	Name	Distribution	Mean	Mode	Median	Dispersion	Min.	Max.	Missing
N	T3RU		109.60	105	110	0.12	65	144	0 (0 %)
N	T4		9.805	8.1	9.2	0.478	0.5	25.3	0 (0 %)
N	T3		2.050	1.8	1.7	0.691	0.2	10.0	0 (0 %)
N	TSH		2.880	1.0	1.3	2.119	0.1	56.4	0 (0 %)
N	DTSH		4.199	-0.1	2.0	1.918	-0.7	56.3	0 (0 %)
C	Diagnosis			1		0.821			0 (0 %)

Figura 2.1: *Distribuzione degli attributi*

Attributo	**Descrizione**	**Tipo**
T3RU	T3 uptake test, misura la percentuale di TBG legata	int
T4	concentrazione totale di Tiroxina	float
T3	concentrazione totale di Triiodotironina	float
TSH	ormone tireostimolante	float
DTSH	differenza massima di TSH dopo iniezione di 200 mg di TRH	float

Tabella 2.2: *Descrizione e tipo degli attributi*

Diagnosi	**Conteggio**	**Prevalenza**
Classe 1: (normal)	150	69.8%
Classe 2: (hyper)	35	16.3%
Classe 3: (hypo)	30	13.9%

Tabella 2.3: *Distribuzione delle classi*

2.1 Visualizzazione dei dati

L'analisi dei dati utilizzando le diverse widget di Orange per la visualizzazione dei dati ha portato a identificare il boxplot come il tipo di grafico più informativo. Da questo grafico, è emerso che l'**attributo 'T4'** sembra essere il più informativo per la classificazione delle diagnosi, poiché mostra chiare differenze nella distribuzione dei valori tra le diverse classi, ad eccezione di alcuni outliers.

La Figura 2.2 rappresenta il boxplot dell'attributo 'T4', suddiviso per classe di diagnosi. Si può osservare chiaramente come i valori di 'T4' varino significativamente tra le diverse classi, il che suggerisce che questo attributo potrebbe essere un forte indicatore per la diagnosi delle malattie tiroidee.

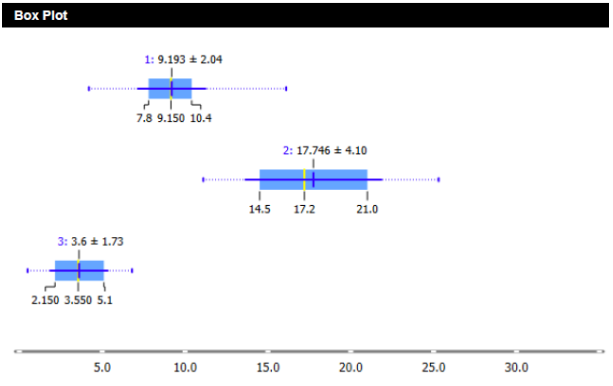


Figura 2.2: *Box Plot dell'attributo 'T4' raggruppato per classe*

Inoltre, è stata effettuata un'analisi di riduzione dimensionale utilizzando la widget *'t-SNE'* per visualizzare la distribuzione dei dati in uno spazio 2D. Questo metodo permette di convertire le somiglianze tra i dati in probabilità congiunte, riducendo le dimensioni e cercando di preservare la struttura dei dati ad alta dimensione. Nella Figura 2.3, i dati sono rappresentati in uno spazio bidimensionale, evidenziando la distribuzione dei punti. Tuttavia, si notano alcuni punti al di fuori della regione principale, il che potrebbe indicare la presenza di outliers o dati anomali.

In generale, sembra che il dataset possa essere gestito con relativa semplicità, ma è importante prestare attenzione agli outliers e considerare come trattarli durante l'analisi e la modellazione dei dati.

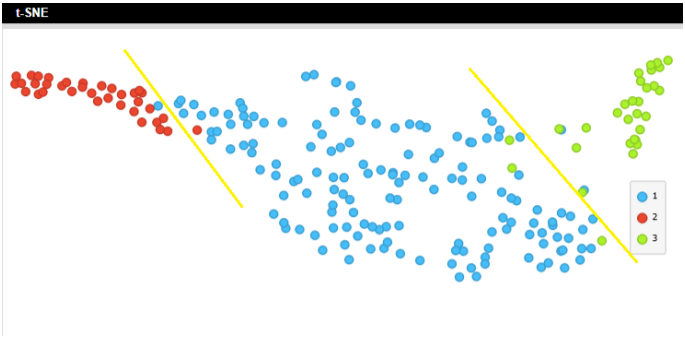


Figura 2.3: *t-SNE con PCA a 2 componenti*

2.2 Preparazione dei dati

Il data preprocessing ha come obiettivo principale l'ottimizzazione della qualità dei dati, la risoluzione delle possibili problematiche che potrebbero influire sull'analisi, e la creazione delle condizioni ideali per l'estrazione di informazioni rilevanti dai dati stessi. È importante notare che, come precedentemente affermato, il dataset a disposizione è privo di dati mancanti. Pertanto, dopo una fase di esplorazione e l'utilizzo di apposite visualizzazioni tramite la widget di Orange, ci si è focalizzati sulla rilevazione degli outliers.

In totale, sono stati individuati 20 outliers all'interno del dataset. Tuttavia, la loro percentuale rispetto all'intero dataset supera il 5%, pertanto, si è scelto di non procedere con la loro rimozione. Questa decisione è stata presa per evitare la perdita di una quantità significativa di dati, che potrebbe comportare la riduzione dell'informazione disponibile o la distorsione della distribuzione complessiva dei dati.

Successivamente, il dataset è stato diviso in due sottoinsiemi: un training set, rappresentante il 55% dei dati, destinato all'addestramento e alla valutazione dei classificatori, e un test set, rappresentante il 45% dei dati, utilizzato per testare i modelli di classificazione addestrati e selezionare il migliore tra essi. La scelta di queste percentuali, non consuete, è stata motivata dalla necessità di avere un numero sufficiente di dati nel test set, poiché percentuali inferiori avrebbero comportato un campione troppo esiguo. La divisione tra training e test set è stata effettuata utilizzando un campionamento stratificato, garantendo così che le proporzioni tra le diverse classi all'interno dei dati rimanessero invariate in entrambi i sottoinsiemi.

È importante notare che nessuna operazione è stata eseguita sul test set, poiché questo deve riflettere fedelmente la realtà dei dati. Tutte le operazioni descritte successivamente sono state applicate esclusivamente

al training set.

2.3 Bilanciamento delle classi

Come detto in precedenza il dataset è fortemente sbilanciato: il bilanciamento delle classi è necessario per evitare che i classificatori abbiano un "bias" verso la classe più numerosa, rischiando di avere un overfitting verso quella classe. Quindi, a seguito della divisione del dataset tra training e test set, le classi del training set sono state bilanciate utilizzando Synthetic Minority Oversampling TEchnique (**SMOTE**) implementato in Python, l'algoritmo prevede di generare dei dati sintetici partendo dagli esempi delle classi minoritarie:

1. Si seleziona un esempio della classe minoritaria;
2. Si identificano i k nearest neighbours (vicini più prossimi) appartenenti alla stessa classe rispetto all'esempio selezionato, utilizzando la distanza euclidea come metrica di similarità.
3. Per ciascun vicino, si genera un dato sintetico che si trova in una posizione interpolata tra l'esempio selezionato e il vicino stesso. Questo viene fatto calcolando la differenza tra le feature dell'esempio e del vicino, e quindi moltiplicando questa differenza per un valore casuale compreso tra 0 e 1.
4. Questi passaggi vengono ripetuti fino a raggiungere il numero desiderato di campioni sintetici.

La funzione SMOTE della libreria imbalanced-learn in Python di default seleziona 5 campioni vicini per generare dati sintetici. Tuttavia, a causa del numero molto limitato di campioni nelle classi 2 e 3 rispetto alla

classe 1, è stato scelto di ampliare il numero di campioni nelle classi 2 e 3 per ottenere una percentuale soddifacente di dati, senza esagerare nella creazione di dati sintetici. L'obiettivo era ottenere una distribuzione dei dati che fosse soddisfacente dal punto di vista percentuale, rispettando al contempo un approccio moderato alla generazione di dati sintetici. Di conseguenza, si è raggiunta una distribuzione delle classi con il 44.62% dei dati nella classe 1, il 27.96% nella classe 2 e il 27.42% nella classe 3.

2.4 Normalizzazione dei dati

La normalizzazione dei dati continui nell'intervallo tra 0 e 1 è un passaggio importante prima di procedere all'analisi quantitativa. Questo processo è stato effettuato utilizzando la widget di Orange *'Preprocess'*. La normalizzazione consente di portare tutte le caratteristiche continue dei dati in una scala comune, il che è cruciale per garantire risultati accurati e migliorare le performance degli algoritmi che verranno utilizzati in seguito.

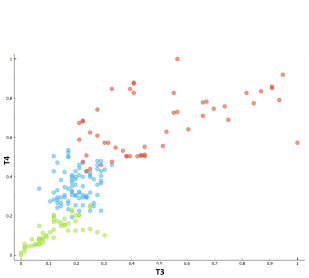
Questa pratica è fondamentale perché molti algoritmi di machine learning sono sensibili alle differenze di scala tra le variabili.

2.5 Analisi e scelta degli attributi

Si è proseguito conducendo un'analisi approfondita degli attributi del dataset utilizzando le widget *'Scatter Plot'*, *'Correlations'* e *'Distance Matrix'*. I risultati di questa analisi sono riportati nella Figura 2.4. È emersa una forte correlazione tra la coppia di attributi T3 e T4. Questa correlazione, tuttavia, è una conseguenza naturale del processo di sintesi degli ormoni tiroidei nel corpo umano. Quando il T4 entra in circolo, subisce una conversione in T3 attraverso il processo di deionizzazione. In modo simile, le correlazioni tra T3-T3RU e T4-T3RU hanno una base

fisiologica. Come precedentemente spiegato, la misurazione di T3RU è fondamentale per calcolare la quantità di ormoni T3 e T4 liberi nel corpo.

È importante sottolineare che, considerando la natura fisiologica di queste correlazioni e il fatto che il dataset non contenga un gran numero di attributi, non è ritenuta necessaria una selezione delle caratteristiche. Ogni attributo nel dataset può contribuire in modo significativo all'analisi e alla classificazione delle malattie tiroidee. Tuttavia, durante il processo di modellazione, è comunque importante valutare attentamente la rilevanza di ciascun attributo per garantire risultati accurati e significativi.



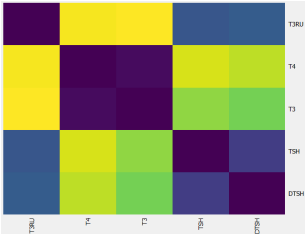
(a) Scatter Plot

Correlations

Pearson correlation

1	+0.788	T3	T4
2	-0.657	T3	T3RU
3	-0.644	T3RU	T4
4	-0.567	T4	TSH
5	+0.560	DTSH	TSH
6	-0.513	DTSH	T4
7	+0.431	T3RU	TSH
8	-0.417	T3	TSH
9	+0.394	DTSH	T3RU
10	-0.348	DTSH	T3

(b) Correlations



(c) Distance Matrix

Figura 2.4: *Correlazione degli attributi*

3 Modellizzione e Valutazione

Per questo problema di classificazione è stato utilizzato un approccio di apprendimento supervisionato. Considerando che il dataset è dotato

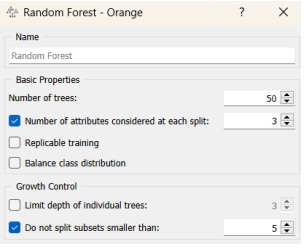
di 3 classi, è sbilanciato e non è particolarmente numeroso, per l'analisi sono stati scelti i seguenti classificatori:

- Random Forest
- Gradient Boosting
- Logistic Regression
- K-Nearest Neighbour
- SVM (linear)

A seguito della scelta dei classificatori, ne è stata valutata la performance

3.1 Classificatori

3.1.1 Random Forest



Random Forest - Widget

L'algoritmo Random Forest adotta un approccio che combina i risultati di diversi modelli per produrre previsioni più stabili e affidabili. Questo processo coinvolge la creazione di numerosi alberi decisionali, ognuno addestrato su campioni di dati estratti dal dataset originale. Inoltre, varia il numero di attributi considerati in ogni albero. Successivamente, ciascun albero emette un voto per la previsione di classe, e la classe che riceve la maggioranza dei voti viene selezionata come previsione finale del Random Forest.

L'approccio Random Forest è stato scelto in questo contesto poiché sfrutta la diversità e la complessità degli alberi decisionali per ottenere previsioni migliori e più affidabili rispetto a un singolo albero. Nonostante il dataset non contenga un gran numero di attributi, ho deciso di allenare 30 alberi per massimizzare la varietà e la robustezza delle previsioni.

3.1.2 Gradient Boosting

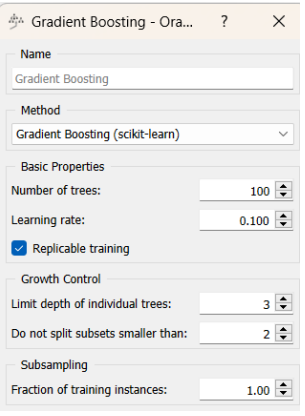
Il Gradient Boosting è un algoritmo che sfrutta l'assemblaggio di vari modelli, comunemente chiamati "weak learners" o modelli deboli, allo scopo di creare un modello più robusto e ad alte prestazioni. Questa tecnica si concentra sull'apprendimento sequenziale, il che significa che costruisce una sequenza di modelli, ognuno dei quali cerca di correggere gli errori del modello precedente.

L'algoritmo prevede di scegliere una funzione di perdita differenziabile L(y,F), partendo da un primo modello semplice, per esempio la media, vengono calcolati i gradienti negativi per ogni esempio

$$g_j(x_j) = -\frac{\delta L(y_i, x_i)}{\delta F_{j-1}}$$

Viene poi appreso un albero di regressione $h_j(x)$ sui valori dei gradienti negativi e calcolato il nuovo modello, come segue

$$F_j(x) = F_{j-1}(x) + \gamma_j h_j(x)$$



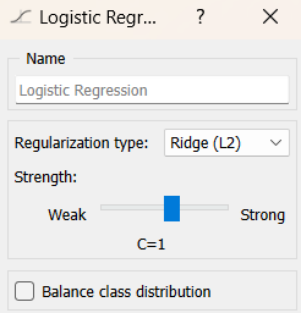
Gradient Boosting - Widget

La funzione di perdita utilizzata nell'algoritmo è una Log Loss

$$L = \frac{1}{N} \sum_{i=1}^{M} \left(\sum_{j=1}^{N_i} y_{ij} \cdot \log(p(y_{ij})) \right)$$

Il Gradient Boosting cerca di ridurre progressivamente gli errori commessi dal modello precedente, rendendolo molto potente, tuttavia bisogna prestare attenzione all'overfitting e scegliere con accuratezza i parametri di profondità massima degli alberi e il tasso di apprendimento.

3.1.3 Logistic Regression



Logistic Regression - Widget

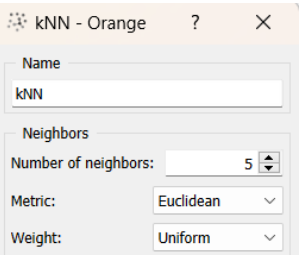
La regressione logistica è una tecnica di apprendimento supervisionato utilizzata per problemi di classificazione. Essa prevede di stimare direttamente la distribuzione a posteriori dei dai dati P(C|X), viene utilizzata una funzione logistica che dipende da set di parametri θ:

$$f_\theta(x) = \frac{e^{\theta \cdot x}}{1 + e^{\theta \cdot x}}$$

I paramteri θ vengono stimati con il metodo della massima verosomiglianza, la log-likelihood è calcolata con la funzione log-loss e viene penalizzata perchè soggetta ad overfitting. Utilizzando la penalità Ridge mi permette di avere una maggiore efficienza nella stima dei parametri in cambio di una quantità tollerabile di bias.

3.1.4 K-Nearest Neighbour

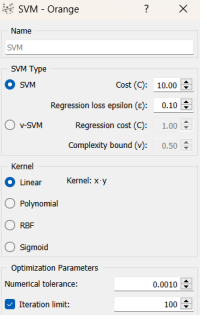
K-Nearest Neighbors (KNN) è un algoritmo utilizzato principalmente per risolvere problemi di classificazione e regressione. Si basa sul concetto di distanza tra gli esempi nello spazio delle caratteristiche e utilizza le informazioni dei vicini per fare previsioni o classificazioni. Dato un esempio x, vengono scelti i k nearest neighbours, ovvero i k punti che hanno la distanza minore dall'esempio x, la classe assegnata ad x è quella che è più comune tra i kNN. La distanza è stata calcolata utilizzando la distanza Euclidea e si è scelto parametro k = 5.



k-NN - Widget

3.1.5 SVM

L'obiettivo principale delle Support Vector Machines è trovare un iperpiano ottimale che separa i punti dei dati appartenenti alle classi diverse. Essendo in presenza di un problema multi-classe, verranno costruiti n classificatori a 2 classi che dividono la n-esima classe considerarta dalle altre. Si è scelto di utilizzare un kernel lineare. Nonostante l'algoritmo è più indicato per dataset con numerosità delle classi bilanciate, poichè potrebbe indurre a classificazioni errate, i risultati ottenuti sono soddisfacenti.



SVM - Widget

3.2 Valutazione dei classificatori

Le metriche di valutazione scelte sono le seguenti:

- **Accuracy**: proporzione di esempi classificati correttamente, $acc = \frac{TP+TN}{N}$
- **Recall**: proporzione di esempi positivi correttamente classificata tra tutti i casi positivi, $rec = \frac{TP}{TP+FN}$
- **Precision**: proporzione di esempi realmente positivi tra i classificati positivi, $prec = \frac{TP}{TP+FP}$
- **F1-Score**: media armonica tra Recall e Precision (misura sintetica), $F1 = \frac{2 \cdot recall \cdot precision}{recall+precision}$

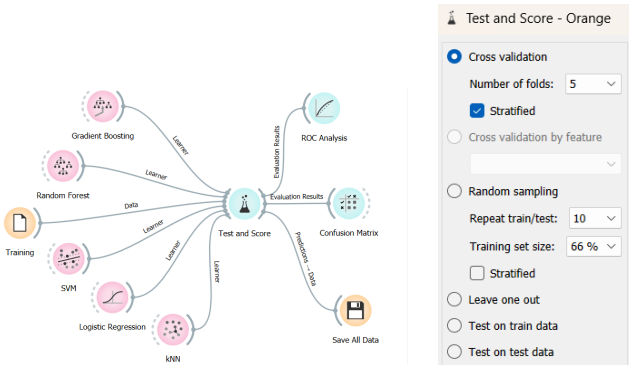
Le misure di recall, precision e F1-score sono indicatori di Information Retrival, introdotti in supporto all'accuratezza, la quale è molto informativa quando si hanno classi bilanciate, tuttavia i classificatori possono comportarsi in modo diverso nella predizione della classe più numerosa rispetto alla predizione delle classi meno frequenti.

4 Scelta e Raffinamento del Modello

Le performance dei classificatori sono state valutate utilizzando una tecnica di **Cross Validation a 5-fold**, come illustrato nella Figura 4.1. Questa tecnica suddivide il dataset in cinque insiemi di dimensioni simili, mantenendo una distribuzione equa delle classi grazie alla stratificazione dei campioni. In ogni iterazione, il modello viene addestrato su quattro dei cinque insiemi (k-1 fold) e il quinto insieme viene utilizzato per il test.

Il parametro k è stato scelto in modo da garantire un numero sufficiente di dati per il test.

Sono stati quindi ottenuti cinque classificatori con le relative valutazioni. Nella Tabella 4.1 sono riportate le performance medie su tutte e tre le classi.



(a) Training (b) CV

Figura 4.1: *Training dei modelli e 5-fold CV*

Per enfatizzare la minimizzazione degli errori nella classificazione dei pazienti malati, sono stati calcolati l'accuratezza media e il recall medio con i relativi intervalli di confidenza al 95%. Gli intervalli di confidenza sono stati calcolati utilizzando la deviazione standard campionaria e il t-test con la formula: $p = \hat{a} \pm t_{k-1,1-\frac{\alpha}{2}} \frac{s}{\sqrt{k}}$. I risultati sono riportati nella Tabella 4.2.

Model	AUC	CA	F1	PREC	REC
FOLD 1					
kNN	1.000	1.000	1.000	1.000	1.000
RF	1.000	1.000	1.000	1.000	1.000
GB	1.000	0.974	0.974	0.970	0.980
SVM	1.000	1.000	1.000	1.000	1.000
LR	1.000	0.974	0.973	0.981	0.967
FOLD 2					
kNN	1.000	1.000	1.000	1.000	1.000
RF	1.000	0.946	0.947	0.939	0.961
GB	0.979	0.919	0.922	0.914	0.941
SVM	1.000	1.000	1.000	1.000	1.000
LR	1.000	0.919	0.914	0.950	0.900
FOLD 3					
kNN	0.998	0.973	0.974	0.970	0.980
RF	0.996	0.946	0.947	0.947	0.947
GB	1.000	1.000	1.000	1.000	1.000
SVM	0.997	0.973	0.973	0.981	0.967
LR	0.994	0.892	0.889	0.937	0.867

FOLD 4					
kNN	0.975	0.892	0.880	0.933	0.867
RF	1.000	0.865	0.856	0.921	0.836
GB	0.947	0.865	0.856	0.921	0.836
SVM	1.000	0.919	0.913	0.947	0.900
LR	0.996	0.757	0.742	0.880	0.712
FOLD 5					
kNN	1.000	1.000	1.000	1.000	1.000
RF	0.995	0.973	0.974	0.980	0.970
GB	0.993	0.946	0.949	0.949	0.949
SVM	1.000	1.000	1.000	1.000	1.000
LR	1.000	0.892	0.893	0.933	0.876

Tabella 4.1: *Indicatori di performance dei 5-fold*

Proseguendo nell'analisi, è stata valutata la significatività delle differenze tra i classificatori utilizzando il Test non parametrico di **Kruskal-Wallis**. Questa scelta è stata motivata dal fatto che non è possibile fare ipotesi sulla distribuzione delle accuratezze. I risultati del test indicano che ci sono differenze significative tra alcuni dei classificatori. In particolare, la Regressione Logistica mostra una significativa differenza rispetto alla Random Forest, alla Support Vector Machine e Gradient Boosting, mentra tra gli altri classificatori non sono presenti differeze significative.

model	**mean**	**lower**	**upper**	**mean**	**lower**	**upper**
metric	Accuracy			Recall		
kNN	0.973	0.915	1.031	0.969	0.897	1.042
RF	0.946	0.883	1.009	0.943	0.865	1.020
GB	0.941	0.876	1.005	0.941	0.863	1.020
SVM	0.978	0.935	1.022	0.973	0.919	1.027
LR	0.887	0.787	0.986	0.864	0.748	0.980

Tabella 4.2: *Valutazione accuratezza*

La visualizzazione del risultato è presentata in Figura 4.2, grafico ottenuto utilizzando Matlab.

Successivamente, è stata eseguita un'ulteriore analisi esaminando le matrici di confusione di ciascun modello, mostrate in Figura 4.3 . Tra i vari classificatori, la Regressione Logistica commette il maggior numero di errori, classificando in modo errato gli elementi delle classi 2 ('Hyper') e 3 ('Hypo'). Al contrario, i due classificatori migliori, che commettono meno errori, sono Support Vector Machine e k-Nearest Neighbour. SVM presenta un'accuratezza maggiore rispetto agli altri classificatori, mentre kNN ha un'accuratezza leggermente inferiore, anche il recall medio è maggiore nella SVM. Inoltre entrambi i classificatori hanno un intervallo di confidenza sufficientemente piccolo.

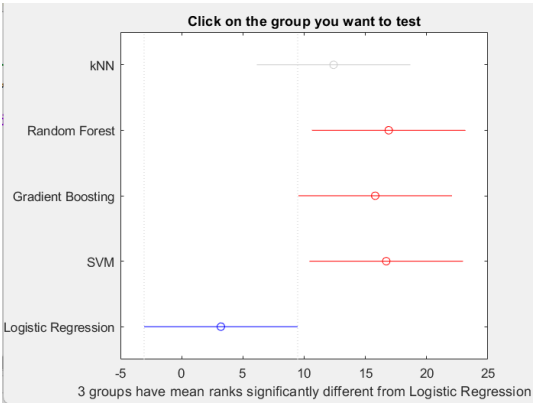


Figura 4.2: *Kruskal-Wallis test*

	1	2	3	Σ
1	82	1	0	83
2	0	52	0	52
3	4	0	47	51
Σ	86	53	47	186

(a) kNN

	1	2	3	Σ
1	80	2	1	83
2	3	49	0	52
3	4	0	47	51
Σ	87	51	48	186

(b) Random Forest

	1	2	3	Σ
1	78	4	1	83
2	2	50	0	52
3	4	0	47	51
Σ	84	54	48	186

(c) Gradient Boosting

	1	2	3	Σ
1	83	0	0	83
2	1	51	0	52
3	3	0	48	51
Σ	87	51	48	186

(d) SVM

	1	2	3	Σ
1	83	0	0	83
2	14	38	0	52
3	7	0	44	51
Σ	104	38	44	186

(e) Logistic Regression

Figura 4.3: *Matrici di confusione*

Dopo aver riconosciuto k-Nearest Neighbour e Support Vector Machine come i migliori classificatori, in base alle considerazioni precedentemente discusse, è stato eseguito un t-test di appaiamento tra questi due classificatori migliori per valutare la significatività delle differenze tra di loro, dato il differenziale di accuratezza $d_i = a_{i,SVM} - a_{i,kNN}$, ho calcolato:

$$t = \frac{d_m}{\frac{s_d}{\sqrt{k}}} = 1, s_d = \sqrt{\frac{\sum_{n=1}^{k}(d_i - d_m)^2}{k-1}} = 0.012$$

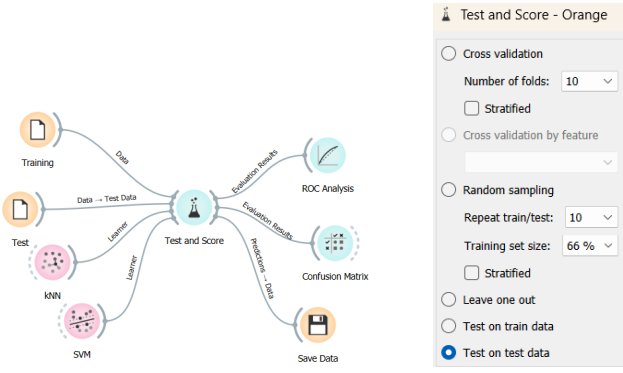
Il risultato del test indica che non ci sono differenze statisticamente significative tra di loro, quindi entrambi i classificatori saranno testati.

4.1 Testing

Dopo aver selezionato i modelli migliori, è giunto il momento di testarli sul test set, che è costituito da dati completamente nuovi. Questa fase è rappresentata nella Figura 4.4. Nella Tabella 4.3 sono riportate le metriche di valutazione per ciascun classificatore. L'obiettivo principale è di scegliere il modello che minimizza il numero di misclassificazioni nelle due classi di pazienti malati. Per valutare le performance, vengono utilizzate le curve ROC, come mostrato nella Figura 4.5, e le matrici di confusione, come illustrato nella Figura 4.6.

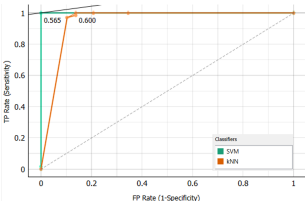
Le curve ROC consentono di visualizzare la relazione tra la percentuale di pazienti malati correttamente identificati (True Positives, TP) e la percentuale di individui sani erroneamente classificati come malati (False Positives, FP). Entrambi i metodi di valutazione indicano la Support Vector Machine come il modello migliore, il quale mostra un'eccellente capacità nel minimizzare gli errori in tutte le classi, l'altro classificatore commette più errori nei casi appartenenti alla classe 3.

Questo risultato è coerente con le osservazioni precedenti, in cui era stato notato che il problema poteva essere distinto con relativa facilità utilizzando le tecniche di apprendimento non supervisionato, come t-SNE e MDS. La Support Vector Machine presenta un'accuratezza media del **97.9%**, il che conferma la sua validità come modello per questo problema di classificazione.

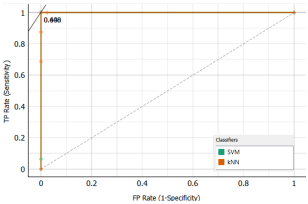


(a) Testing (b) Test and Score

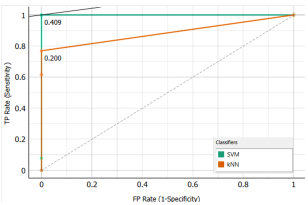
Figura 4.4: *Testing migliori modelli*



(a) Target 1



(b) Target 2



(c) Target 3

Figura 4.5: *Curve ROC*

	1	2	3	Σ
1	67	0	0	67
2	1	15	0	16
3	1	0	12	13
Σ	69	15	12	96

(a) SVM

	1	2	3	Σ
1	67	0	0	67
2	0	16	0	16
3	4	0	9	13
Σ	71	16	9	96

(b) kNN

Figura 4.6: *Matrici di confusione*

Model	AUC	CA	F1	PREC	REC
kNN	0.943	0.958	0.929	0.981	0.897
SVM	1.0	0.979	0.971	0.990	0.953

Tabella 4.3: *Indicatori di performance sul test set*

5 Utilizzo e Dissiminazione del Modello

Il risultato ottenuto, in relazione ai dati attualmente disponibili nel dataset, è complessivamente soddisfacente. Tuttavia, è importante sottolineare che questa analisi è limitata dalla quantità ridotta di dati a disposizione. Per garantire la robustezza e l'affidabilità del modello proposto, sarebbe auspicabile un'ulteriore raccolta di dati, preferibilmente su una scala più ampia.

In particolare, è fondamentale notare che il dataset attuale non include informazioni cruciali quali l'età e il sesso dei pazienti. Questi fattori sono di grande rilevanza nella diagnosi delle disfunzioni tiroidee, poiché i livelli degli ormoni tiroidei possono variare in base all'età e al genere del paziente. L'acquisizione di tali dati aggiuntivi consentirebbe una classificazione più dettagliata e precisa delle condizioni tiroidee.

Pertanto, mentre i risultati attuali sono promettenti, è chiaro che ulteriori dati e dettagli clinici potrebbero migliorare significativamente la qualità e l'applicabilità del modello diagnostico proposto. La ricerca futura dovrebbe concentrarsi sull'ampliamento del dataset e sulla raccolta accurata di informazioni pertinenti per una diagnosi completa e accurata delle disfunzioni tiroidee.

Dopo aver potenziato la affidabilità del modello, si potrebbe integrare efficacemente nei sistemi di telemedicina al fine di agevolare i pazienti

nell'esecuzione di esami di screening direttamente a casa, ottenendo così una valutazione preliminare basata sui dati raccolti. Questa soluzione consentirebbe di ridurre i tempi di attesa per le visite mediche e i risultati degli esami.

Bibliografia

Articoli

[1] M. P. J. Vanderpump. “The epidemiology of thyroid disease”. In: *British Medical Bulletin* (2011). DOI: 10.1093/bmb/ldr030.

[5] Khalid Salman e Emrullah Sonuç. “Thyroid Disease Classification Using Machine Learning Algorithms”. In: *Journal of Physics: Conference Series* (2021). DOI: 10.1088/1742-6596/1963/1/012140.

[6] Shaik Razia, P. Kumar e A. Rao. “Machine Learning Techniques for Thyroid Disease Diagnosis: A Systematic Review”. In: feb. 2020, pp. 203–212. ISBN: 978-3-030-38444-9. DOI: 10.1007/978-3-030-38445-6_15.

[7] Gyanendra Chaubey et al. “Thyroid Disease Prediction Using Machine Learning Approaches”. In: *National Academy Science Letters* (2020). DOI: 10.1007/s40009-020-00979-z.

Siti Web consultati

[2] *How does the thyroid gland work?* URL: https://www.ncbi.nlm.nih.gov/books/NBK279388/ (visitato il 09/04/2023).

[3] *Tiroide*. URL: https://it.wikipedia.org/wiki/Tiroide (visitato il 09/04/2023).

[4] *Sintomi*. URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/disorders-of-the-thyroid (visitato il 31/08/2023).