Classificazione Binaria del Diabete

Chiodo Martina
Matricola: s343310

Danca Ema-Aldona
Matricola: s347940

Speca Leonardo
Matricola: s343483

Terenziani Andrea
Matricola: s349167

Vigè Sophie
Matricola: s339268

***Abstract*—Il report descrive il processo di selezione, addestramento e validazione di alcuni algoritmi di classificazione binaria implementati in Python facendo largo uso della libreria Scikit-Learn. In particolare, vengono analizzati l'approccio adottato, le caratteristiche del dataset utilizzato, il preprocessing e l'efficacia dell'algoritmo proposto nel distinguere correttamente tra pazienti diabetici e non diabetici.**

I. Introduzione

Una delle sfide del settore sanitario è la diagnosi precoce delle malattie croniche che richiede un continuo miglioramento degli strumenti diagnostici. Una di queste è la classificazione accurata dei pazienti affetti da diabete.

La diagnosi viene tradizionalmente effettuata grazie alla valutazione del personale medico degli esami clinici. Tuttavia, questo metodo può risultare lento, soggettivo e difficile da applicare su larga scala. Inoltre, l'analisi manuale di grandi volumi di dati clinici può essere onerosa sia in termini di tempo che di risorse economiche. In questa ricerca, abbiamo utilizzato diversi algoritmi di classificazione binaria basati su alcune caratteristiche dei pazienti.

II. Analisi dei dati

Il materiale fornito è comprensivo di due dataset, uno per la fase di training e uno per quella di test, questi presentano informazioni cliniche e demografiche raccolte su pazienti con e senza diabete. Il dataset di training è composto da 588 record, di cui solo un quarto di questi rappresenta pazienti con il diabete. Le variabili considerate includono fattori quali sesso, età, ipertensione, malattie cardiache, abitudini legate al fumo, indice di massa corporea (BMI), livello di emoglobina glicata (HbA1c), livello di glucosio nel sangue, sensibilità all'insulina, interazione tra BMI e glucosio e presenza di Troponina T.

I dati sono stati sottoposti a una fase preliminare di preelaborazione per garantirne la qualità e l'affidabilità.

Un'analisi esplorativa iniziale ha rilevato 59 record duplicati nel train set, successivamente rimossi. L'attributo *smoking_history* conteneva la voce "No Info", trattata come categoria a sé stante per evitare la perdita di dati, data la dimensione già ridotta del dataset. Poiché la variabile assume solo 6 valori categorici, a nostro avviso, ordinabili, si è optato per una codifica manuale invece del one-hot encoding, con il seguente ordine: "never", "ever", "No Info", "not current", "former", "current". Anche l'attributo *gender* è stato codificato, in questo caso tramite `LabelEncoding`.

Per quanto riguarda invece i valori mancanti dell'attributo *Insulin_Sensitivity_Est* si è deciso di utilizzare il valore medio in quanto (dai grafici esplorativi che seguono) non sembra che tale attributo abbia una forte correlazione con la variabile risposta. Questo è stato fatto usando la trasformazione `SimpleImputer(strategy='mean')`, la quale è stata fittata sui dati di train e poi usata per trasformare sia il train che il test set.

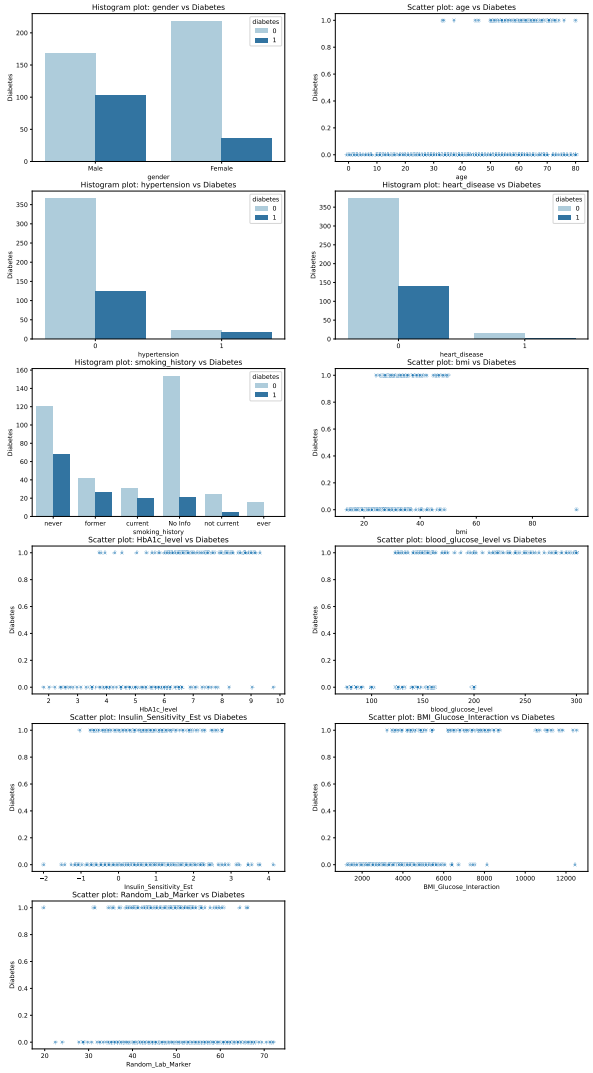

Fig. 1: Scatter plot di ogni feature e della variabile risposta

Successivamente, si è voluto indagare sulla presenza di outlier, ovvero valori anomali che si discostano significativamente dalla distribuzione dei dati, per cui si è proceduto con tecniche di pulizia dei dati. A tale scopo sono stati imposti alcuni vincoli di dominio sulle feature, i quali hanno portato

all'eliminazione dei record con età negativa o BMI negativo o superiore a 90 ad esempio (outlier che erano presenti nei dataset come mostrano i grafici 1 e 2).

Infine, per garantire un'elaborazione adeguata dei dati e l'utilizzo corretto degli algoritmi, senza dare pesi diversi agli attributi numerici sulla base del loro range di valori, i valori di tutti gli attributi (meno l'attributo *Insulinity_Sensitivity_Est* poiché è già frutto di una normalizzazione) sono stati riscalati tramite `MinMAxScaler()` nell'intervallo $[-3; 3]$. Si è scelto questo intervallo poiché è il range di dominio dell'attributo *Insulinity_Sensitivity_Est*.

La Figura 1 offre un'analisi qualitativa delle feature più influenti sulla classificazione, mostrando la relazione tra ciascuna e la variabile target. Ad esempio, valori elevati di *Blood_Glucose_Level* e *HbA1c_level* sono frequentemente associati a casi positivi di diabete, mentre *Insulin_Sensitivity_Est* mostra una correlazione meno evidente. Tra le variabili categoriche, si osserva che i pazienti di sesso maschile e quelli con una storia di fumo "former" o "current" risultano più inclini a sviluppare il diabete rispetto alle altre classi.

Si è poi fatto un lavoro di feature selection basata sulla matrice di correlazione (Figura 3). Questa evidenzia una correlazione molto alta tra l'attributo *BMI_Glucose_Interaction* e gli attributi *bmi* e *blood_glucose_level*, come era prevedibile dal nome. Si è quindi deciso di eliminare la feature di interazione perché, sebbene abbia una correlazione alta con la risposta binaria, questa dipendenza viene spiegata dagli altri due attributi. Le feature con bassa correlazione rispetto alla variabile target sono state mantenute, sia per la presenza di meccanismi interni di selezione nei modelli utilizzati, sia perché la ridotta dimensionalità del dataset non giustificava ulteriori eliminazioni.

III. Metodologia

Verificata la qualità dei dati, il dataset è stato partizionato separando le feature dalla variabile target. Si è quindi passati alla selezione degli algoritmi di classificazione, concentrandosi in particolare sui seguenti modelli:

- **Decision Tree**, che permettere di ottenere risultati di facile interpretabilità a basso costo computazionale.
- **Random Forest** un insieme di alberi decisionali costruiti su diversi sottoinsiemi del dataset, che offre maggiore robustezza rispetto al singolo albero, riduce l'overfitting e migliora la precisione della classificazione.
- **Support Vector Machine** un algoritmo efficace per problemi di classificazione binaria, che cerca di trovare l'iperpiano ottimale che separa le due classi massimizzando il margine tra i dati.
- **Multi-layer Perceptron** una rete neurale artificiale in grado di modellare relazioni complesse non lineari tra le variabili. Richiede maggiore potenza computazionale ma può offrire alte prestazioni.

Il tuning degli iperparametri è stato effettuato tramite `GridSearchCV`, che applica una cross-validation interna per ottimizzare una metrica specifica. La scelta è ricaduta

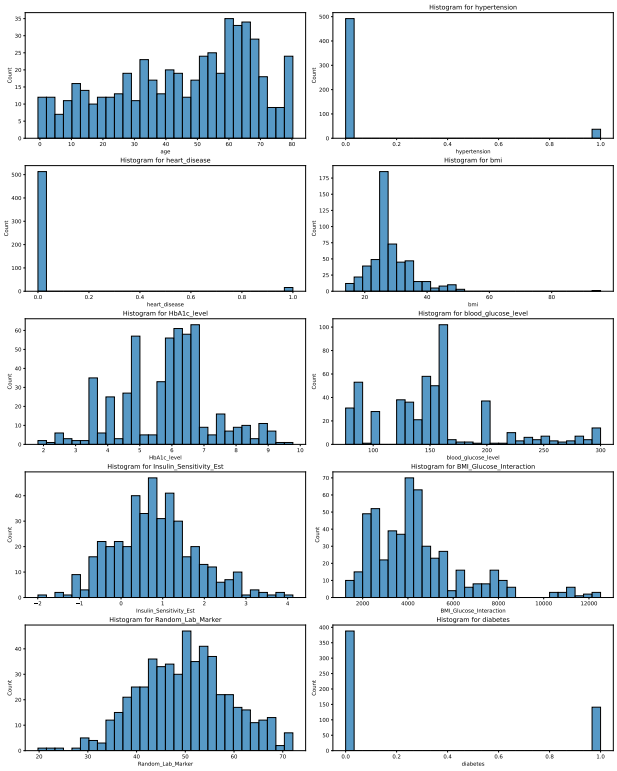

Fig. 2: Istogrammi raffiguranti la distribuzione dei valori di ogni feature.

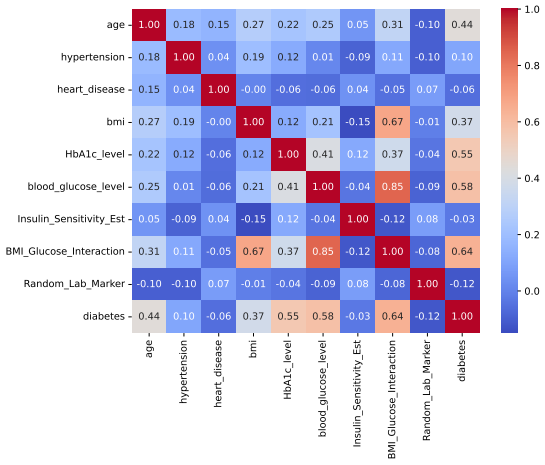

Fig. 3: Matrice di correlazione tra le features

sull'utilizzo di 5 fold e come metrica la F1-Score [1] scelta per gestire lo sbilanciamento tra le classi, dove altre metriche, come l'accuratezza, risulterebbero poco affidabili e fuorvianti. Il numero di fold è stato determinato sperimentalmente, confrontando le performance su train e test set. Una volta individuati gli iparametri migliori, il modello è stato allenato nuovamente su tutto il dataset di train.

[1]Questa metrica è implementata da SkLearn come descritto nella loro documentazione.

A. *Decision Tree*

Sono stati testati diversi iperparametri per il Decision Tree: come misura d'impurità *Gini index* e *Entropy*; profondità massima (10, 20, 50); numero minimo di campioni per split (2, 5, 10); e soglia minima di riduzione dell'impurità (0.001, 0.01, 0.1). La configurazione ottimale è risultata {Gini, 20, 2, 0.001}.Dall'albero risultante, riportato nella Figura 4, è possibile identificare visualmente le feature che maggiormente influenzano la classificazione, essendo utilizzate come attributi di splitting nei nodi più alti. Come ipotizzato precedentemente, gli attributi *HbA1c_level* e *blood_glucose_level* risultano particolarmente rilevanti, insieme a *age* e *bmi*, presenti anche loro tra i primi nodi di splitting.

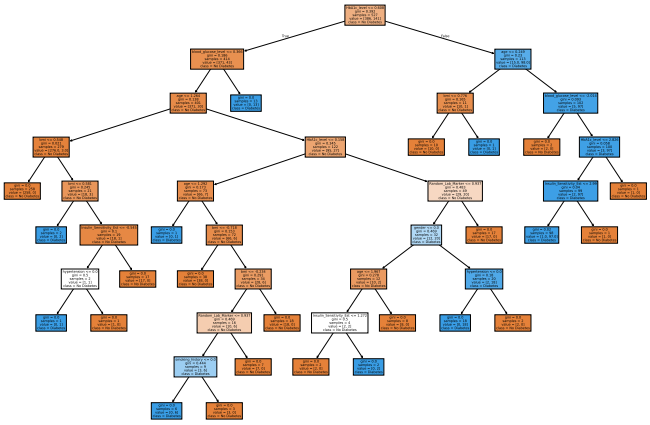

Fig. 4: Decision Tree

B. *Random Forest*

Per il Random Forest sono stati testati diversi iperparametri: numero di alberi stimatori (50, 100), profondità massima (10, 20, 50), minimo numero di campioni per split (2, 5, 10) e soglia minima di riduzione dell'impurità (0.01, 0.1, 0). L'ottimizzazione ha individuato come configurazione migliore {50, 50, 2, 0.0}.

C. *Support vector Machine*

I prinicipali iperparamenti del SVM testati sono stati i seguenti seguenti: il kernel, che definisce il tipo di confine decisionale scelto tra lineare, rbf e sigmoid; il parametro C, il coefficiente di regolarizzazione tra 0.01, 0.2, 2, 10; il gamma, che controlla l'influenza dei singoli punti tra i valori auto (che corrisponde a $1/\text{num\_feature}$), 0.01 e 1/30. La combinazione di parametri che è risultata migliore è {rbf, 2, auto}.

D. *Multi-layer Perceptron*

Per la grid-search del MLP sono state testate diverse configurazioni di rete, variando numero di layer e neuroni ((15,7,15), (11,15,11), (11,11,11,11,11)), funzioni di attivazione (ReLU, identità), learning rate (costante, adattivo) e solver (*Adam*, *sgd*). Il numero massimo di iterazioni è stato fissato a 3000. La combinazione migliore è risultata { (15, 7, 15), ReLu, learning rate adattativo, solver adam}.

Il fatto che tra le migliori performance del MLP ve ne sia una ottenuta con funzione di attivazione identitaria (ovvero con una rete lineare) ci ha suggerito che probabilmente una PCA [2] applicata sul dataset avrebbe avuto un'alta varianza spiegata anche con poche componenti.

IV. Risultati sperimentali

Utilizzando i parametri selezionati nella sezione precedente, si riportano di seguito le matrici di confusione dei vari algoritmi sia applicati al dataset di train sia a quello di test. Il confronto tra queste ci ha permesso di valutare se c'è stato overfitting in quanto i dati contenuti nel test non sono mai stati visti in fase di addestramento.

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	385	1
Actual Diabetes	0	141

TABLE I: Matrice di confusione del dataset di train con DT.

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	109	1
Actual Diabetes	3	34

TABLE II: Matrice di confusione del dataset di test con DT.

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	386	0
Actual Diabetes	0	141

TABLE III: Matrice di confusione per il train del RF

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	110	0
Actual Diabetes	3	34

TABLE IV: Matrice di confusione del test del RF.

Nelle tabelle I e II relative al Decision Tree si osserva un lieve overfitting, evidenziato da un numero inferiore di errori sul training set. Lo stesso comportamento emerge per il Random Forest, che, essendo un insieme di alberi decisionali, ottiene prestazioni leggermente migliori.

[2]PRINCIPAL COMPONENT ANALYSIS: metodo che utilizza la combinazione lineare delle feature e la matrice di covarianza per la ricerca delle direzioni di massima varianza spiegata. Nel seguente dataset, l'85% della varianza è spiegato da 6 principal component

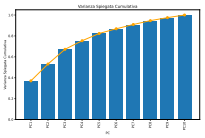

Fig. 5: Varianza Spiegata Cumulativa

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	382	4
Actual Diabetes	11	130

TABLE V: Matrice di confusione del train DEL SVM.

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	108	2
Actual Diabetes	5	32

TABLE VI: Matrice di confusione del test del SVM.

Le matrici di confusione dell'SVM (Tabelle V e VI) mostrano prestazioni inferiori rispetto ai modelli ad albero, probabilmente a causa del numero ridotto di dati e della difficoltà del modello nel catturare relazioni complesse, anche con kernel RBF (Radial Basis Function Kernel). Un'eventuale soluzione è proiettare le feature in uno spazio più adatto alla separazione lineare.

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	385	1
Actual Diabetes	2	139

TABLE VII: Matrice di confusione del train del MLP.

	Predicted No Diabetes	Predicted Diabetes
Actual No Diabetes	109	1
Actual Diabetes	4	33

TABLE VIII: Matrice di confusione del test del MLP.

Nel contesto di questo studio, sono stati confrontati, inoltre, i quattro algoritmi usando le seguenti misure per la prestazione di ciascun modello : Precision (p), Recall (r) e F1-score (F1) per ciascuna delle due classi: "0" (nessuna malattia) e "1" (presenza di malattia).

Model	**Class 0**			**Class 1**		
	p	**r**	**F1**	**p**	**r**	**F1**
DT	0.97	0.99	0.98	0.97	0.92	0.94
RF	0.97	1.00	0.99	1.00	0.92	0.96
SVM	0.96	0.98	0.97	0.94	0.86	0.90
MLP	0.96	0.99	0.98	0.97	0.89	0.93

TABLE IX: Confronto tra RF, DT, MLP e SVM con le metriche di precision, recall e F1-score per le classi 0 e 1 sul test.

Il Random Forest risulta il modello migliore, probabilmente per la relativa semplicità del problema, offrendo anche una maggior robustezza rispetto al singolo Decision Tree. In tutti i modelli, la classe 1 presenta performance inferiori rispetto alla classe 0, a causa dello sbilanciamento nel dataset di training.

Nel Random Forest, per la classe dei pazienti malati si ottiene una precisione pari a 1 e un recall pari a 0.92 ciò significa che tutti i casi predetti come malati sono corretti, ma alcuni pazienti malati non vengono riconosciuti, un aspetto critico poiché sarebbe preferibile evitare di trascurare casi potenzialmente gravi.

L'SVM mostra performance più equilibrate, con precisione 0.94 e recall 0.86 sulla classe 1, risultando abbastanza affidabile nel rilevare il diabete, anche se tende a non identificare tutti i positivi. Dal punto di vista interpretativo, l'SVM è più trasparente rispetto al MLP, soprattutto con kernel lineare, ma meno rispetto ai modelli basati su alberi decisionali poiché l'SVM divide i dati secondo degli iperpiani, non necessariamente lineari, in dimensioni alte.

Il MLP mostra performance leggermente inferiori rispetto agli alberi decisionali, commettendo più errori soprattutto nel riconoscimento dei pazienti malati. Ciò può essere dovuto all'esiguo numero di record, dato che le reti neurali, pur essendo tra i modelli più accurati, richiedono generalmente un maggior quantitativo di dati. Inoltre, a differenza di modelli interpretabili come Decision Tree e Random Forest, il MLP è un modello "black-box", offrendo scarsa trasparenza nelle decisioni e rendendo più difficile l'interpretazione dei risultati.

Come già evidenziato, una criticità del dataset è il forte sbilanciamento tra le due classi di risposta. Per affrontarlo, si possono utilizzare tecniche di **upsampling** o **downsampling**, ovvero rispettivamente l'inserimento di record per la classe minoritaria (diabetici) o la riduzione di quelli per la classe maggioritaria (non diabetici), preservando la rappresentatività della popolazione originale. Esperimenti empirici[3] mostrano che i metodi di downsampling tendono a peggiorare le performance, poiché riducono eccessivamente il numero di record, favorendo l'overfitting nei modelli. L'utilizzo della tecnica di upsampling `SMOTE` (Synthetic Minority Over-sampling Technique)[4] ha migliorato le prestazioni del SVM, tuttavia ha avuto un effetto negativo sulla capacità predittiva dell'albero decisionale, che risulta più sensibile alla presenza di dati sintetici. Un'analisi più esaustiva andrebbe fatta comparando le performance di varie tecniche di bilanciamento di un dataset.

V. Conclusioni

Gli algoritmi di classificazione presentati possono essere molto preziosi non solo nel settore medico, per avere diagnosi più affidabili e possibilmente precoci contenendo le spese mediche, ma anche nel settore assicurativo. Infatti, grazie ad essi, le compagnie assicurative possono valutare con maggiore precisione il rischio di diabete nei clienti, definendo polizze assicurative personalizzate.

VI. Contributi

Il lavoro è stato svolto insieme da tutti i componenti del gruppo, quindi ognuno ha lavorato in egual modo a tutte le parti del progetto. ♡

[3]Gli esperimenti di upsampling/downsampling sono stati condotti tramite funzioni della libreria `Imbalanced-Learn` in Python.

[4]`SMOTE` genera nuovi campioni sintetici della classe minoritaria interpolando tra istanze vicine.