Università degli Studi di Milano-Bicocca

Laurea Magistrale in Informatica(LM-18)

Relazione Progetto di Machine Learning

Autori:
Damiano Ficara: 919386
Luca Perfetti 919835
Alessandro Cassani 920015

Anno Accademico 2023/2024
Milano, Italia

Indice

Capitolo 1

Introduzione al progetto

1.1 Componenti del team

- Alessandro Cassani 920015
- Damiano Ficara 919386
- Luca Perfetti 919835

1.2 Il dataset

Si è scelto di utilizzare il dataset fornito al seguente link `https://www.kaggle.com/datasets/prosperchuks/health-dataset`, il quale contiene un modesto numero di samples relativi ad abitudini, stile di vita e pregressi eventi medici di una popolazione presa in considerazione. Si è deciso di prendere in esame il target relativo alla feature `Diabetes`, la quale consiste in un parametro binario avente le classi bilanciate. La variabile target rappresenta la presenza o l'assenza di questa condizione, il che la rende un esito estremamente oggettivo proprio per la sua natura diagnostica, oltre che per essere predetta a partire da condizioni non soggette e vincoli di interpretazione.

Il dataset affronta il tema del diabete, una malattia di notevole importanza clinica, con impatti significativi sulla salute pubblica. La possibilità di sviluppare modelli predittivi può contribuire all'ottimizzazione delle diagnosi e delle strategie di trattamento. La presenza di diverse caratteristiche nel dataset offre l'opportunità di esplorare la relazione tra variabili e la presenza o assenza del diabete, consentendo un'analisi dettagliata.

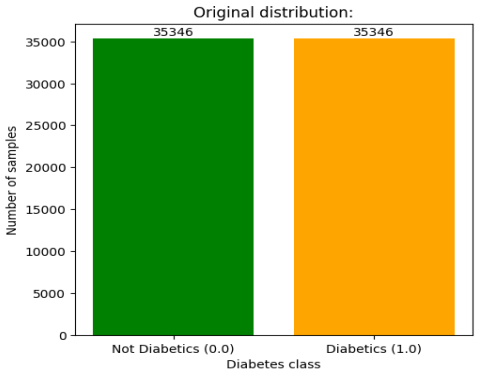
Capitolo 2

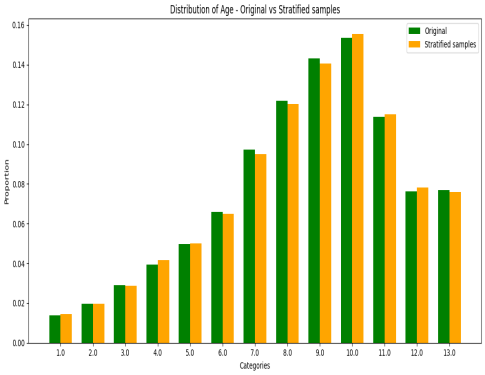
Analisi Esplorativa del Dataset

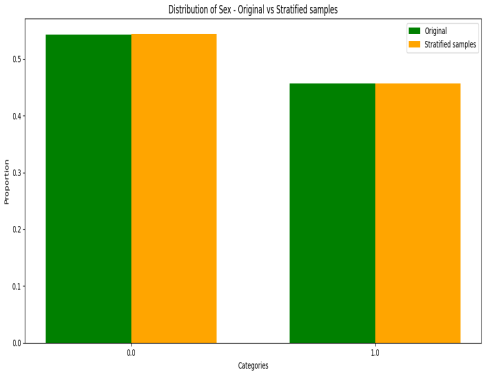
2.1 Struttura del progetto

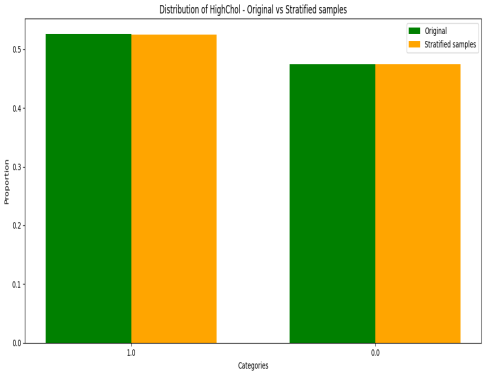
La prima fase del progetto è data da un'analisi esplorativa del dataset, dalla quale abbiamo tratto le informazioni principali riguardanti la distribuzione di ogni singola feature e il rapporto di correlazione tra di esse con la variabile target. È stata inoltre valutata l'integrità delle informazioni contenute nell'insieme di dati, valutando la presenza di eventuali ridondanze, valori mancanti e outliers. In seguito a questa fase si è passati all'addestramento di 3 modelli di apprendimento: Naive Bayes, Rete Neurale e Albero Decisionale. Dopo aver valutato le performance di ogni singolo modello su un insieme predefinito di metriche di seguito presentate, è succeduta una fase di conclusioni finali avente lo scopo di mettere a confronto le performance ottenute e stabilire delle relative considerazioni.

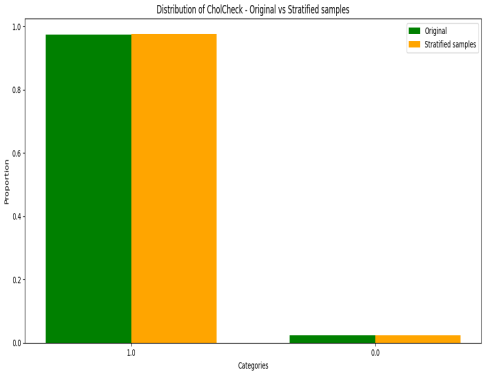
2.2 Riduzione del numero di campioni del dataset

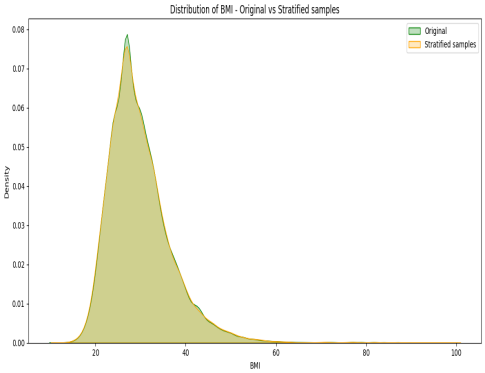


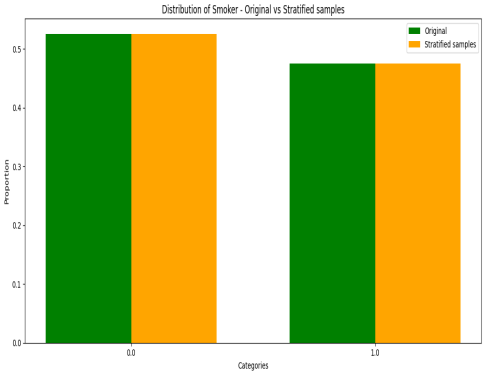

Distribution of Age - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0

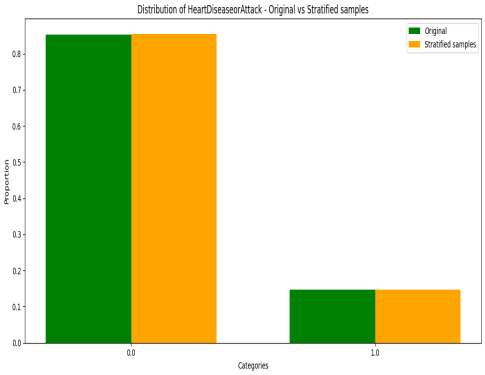

Distribution of Sex - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
0.0
1.0

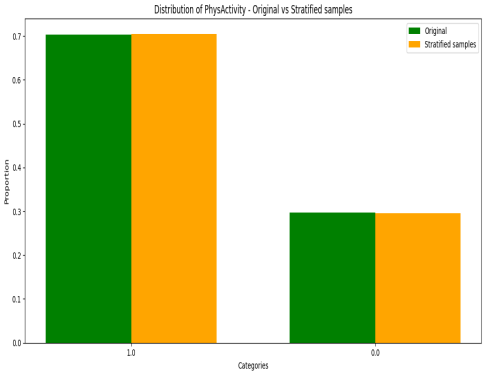

Distribution of HighChol - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
1.0
0.0

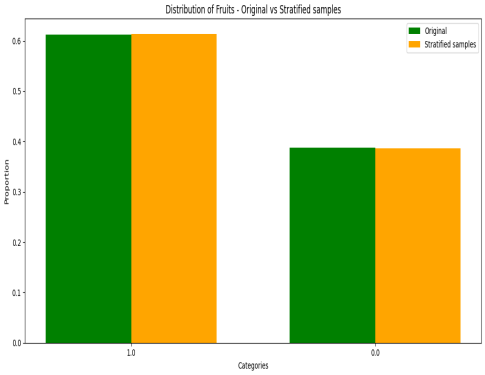

Distribution of CholCheck - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
1.0
0.0


Distribution of BMI - Original vs Stratified samples
Original
Stratified samples
Density
BMI
20
40
60
80
100


Distribution of Smoker - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
0.0
1.0


Distribution of HeartDiseaseorAttack - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
0.0
1.0


Distribution of PhysActivity - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
1.0
0.0


Distribution of Fruits - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
1.0
0.0

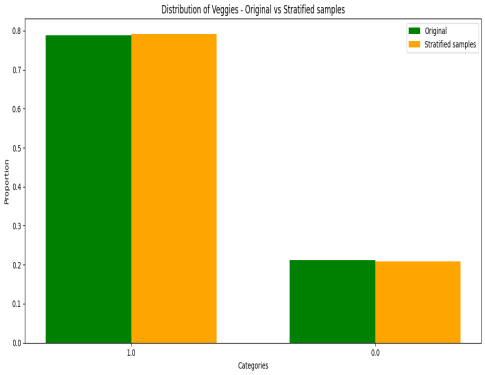

Distribution of Veggies - Original vs Stratified samples
Original
Stratified samples
Proportion
Categories
1.0
0.0

Figura 2.1: Analisi Features pre e post riduzione

La metodologia adottata per la riduzione del dataset è quella nota come "campionamento stratificato". Quest'ultima è stata scelta con l'intento di preservare le distribuzioni delle classi target del dataset originale nel dataset ridotto. Questa strategia si basa sulla divisione del dataset in gruppi omogenei in base alla variabile di

interesse, nel nostro caso la variabile 'Diabetes'. Successivamente, viene eseguita l'estrazione casuale di un campione da ciascun gruppo, assicurando che il numero di campioni estratti da ogni gruppo sia proporzionale alla sua numerosità nel dataset originale. Ciò consente di ottenere un nuovo dataset la cui distribuzione delle classi target è rappresentativa di quella del dataset originale. Ricreare un dataset ridotto che mantenga lo stesso numero di proporzioni degli elementi che lo compongono nella base di dati originale è molto importante, poiché permette in seguito di effettuare comparazioni con il dataset esteso. Inoltre, questa strategia si rivela efficiente nell'utilizzo delle risorse computazionali e dei tempi di addestramento del modello, poiché consente di ridurre la dimensione complessiva del dataset mantenendo una rappresentazione equilibrata delle classi. In seguito all'applicazione del metodo, si è passati da un dataset formato da **70.762** campioni ad uno da **20.000** campioni.

2.3 Analisi esplorativa del dataset

Age	Sex	HighChol	CholCheck	BMI	Smoker	HeartDiseaseorAttack	PhysActivity	Fruits	Veggies	HvyAlcoholConsump	GenHlth	MentHlth	PhysHlth	DiffWalk	Stroke	HighBP	Diabetes
9.0	0.0	1.0	1.0	53.0	0.0	0.0	1.0	1.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	1.0	0.0
5.0	0.0	0.0	1.0	18.0	0.0	0.0	1.0	1.0	1.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
2.0	0.0	0.0	1.0	18.0	1.0	0.0	1.0	1.0	1.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
12.0	1.0	0.0	1.0	28.0	0.0	0.0	1.0	0.0	1.0	0.0	3.0	0.0	1.0	0.0	0.0	1.0	0.0
5.0	1.0	0.0	1.0	24.0	1.0	0.0	1.0	1.0	1.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0

Figura 2.2: Features presenti nel dataset

Categorie di Età

- 1 = 18-24
- 2 = 25-29
- 3 = 30-34
- 4 = 35-39
- 5 = 40-44
- 6 = 45-49
- 7 = 50-54
- 8 = 55-59
- 9 = 60-64
- 10 = 65-69

- 11 = 70-74
- 12 = 75-79
- 13 = 80 o più

Genere

- 1 = Maschio
- 0 = Femmina

Colesterolo Alto (HighChol)

- 0 = Colesterolo normale
- 1 = Colesterolo alto

Controllo del Colesterolo (CholCheck)

- 0 = Nessun controllo del colesterolo negli ultimi 5 anni
- 1 = Controllo del colesterolo negli ultimi 5 anni

BMI (Indice di Massa Corporea)

Fumatore (Smoker)

- 0 = No (non ha mai fumato almeno 100 sigarette)
- 1 = Sì (ha fumato almeno 100 sigarette in tutta la vita)

Malattia Coronarica o Infarto (HeartDiseaseorAttack)

- 0 = No
- 1 = Sì

Attività Fisica (PhysActivity)

- 0 = No (escludendo l'attività fisica legata al lavoro)
- 1 = Sì

Consumo di Frutta (Fruits)

- 0 = No (non consuma frutta 1 o più volte al giorno)
- 1 = Sì (consuma frutta 1 o più volte al giorno)

Consumo di Verdura (Veggies)

- 0 = No (non consuma verdura 1 o più volte al giorno)
- 1 = Sì (consuma verdura 1 o più volte al giorno)

Consumo Elevato di Alcol (HvyAlcoholConsump)

- 0 = No (non consuma alcol in modo eccessivo)
- 1 = Sì (consumo elevato di alcol definito come $\geq$ 14 bevande a settimana per uomini adulti e $\geq$ 7 bevande a settimana per donne adulte)

Stato di Salute Generale (GenHlth)

- Scala 1-5
- 1 = Eccellente
- 2 = Molto Buono
- 3 = Buono
- 4 = Discreto
- 5 = Scarso

Salute Mentale (MentHlth) Giorni di cattiva salute mentale su una scala da 1 a 30 giorni

Salute Fisica (PhysHlth) Giorni di malattia fisica o infortunio negli ultimi 30 giorni su una scala da 1 a 30 giorni

Difficoltà nella Camminata (DiffWalk)

- 0 = No (nessuna seria difficoltà a camminare o salire le scale)
- 1 = Sì (difficoltà seria nella camminata o salita delle scale)

Ictus (Stroke)

- 0 = No
- 1 = Sì

Pressione Sanguigna Alta (HighBP)

- 0 = Non alta
- 1 = Pressione sanguigna alta

Diabete

- 0 = No diabete
- 1 = Diabete

L'insieme di dati comprende informazioni sullo stile di vita, la storia medica e i fattori di rischio per il diabete. La suddivisione delle caratteristiche può essere effettuata tra "categoriche" e "non categoriche". Le variabili categoriche includono le feature binarie, l'età e lo stato di salute generico (*GenHlth*). D'altra parte, le variabili non categoriche comprendono *BMI*, *MentHlth* e *PhysHlth*, poiché non possono essere suddivise in gruppi distinti.

2.4 Prime operazioni sul dataset

Dato che i valori contenuti nelle feature non sono decimali, si è deciso di portare i dati da `float64` ad `int64`, per una migliore rappresentazione visiva e per ridurre il quantitativo di memoria utilizzato.

Column	Data Type	Column	Data Type
Age	float64	Age	int64
Sex	float64	Sex	int64
HighChol	float64	HighChol	int64
CholCheck	float64	CholCheck	int64
BMI	float64	BMI	int64
Smoker	float64	Smoker	int64
HeartDiseaseorAttack	float64	HeartDiseaseorAttack	int64
PhysActivity	float64	PhysActivity	int64
Fruits	float64	Fruits	int64
Veggies	float64	Veggies	int64
HvyAlcoholConsump	float64	HvyAlcoholConsump	int64
GenHlth	float64	GenHlth	int64
MentHlth	float64	MentHlth	int64
PhysHlth	float64	PhysHlth	int64
DiffWalk	float64	DiffWalk	int64
Stroke	float64	Stroke	int64
HighBP	float64	HighBP	int64
Diabetes	float64	Diabetes	int64

Tabella 2.1: Conversione Features da float64 ad int64

Successivamente, si è evidenziato come il dataset non contenga valori nulli, ed in seguito si è passati alla ricerca di eventuali duplicati ed all'eliminazione di essi.

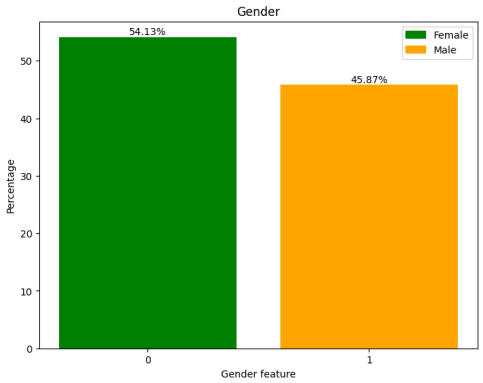
```
sampled_df.isnull().sum().any()
sampled_df.duplicated().sum()
sampled_df = sampled_df.drop_duplicates()
```

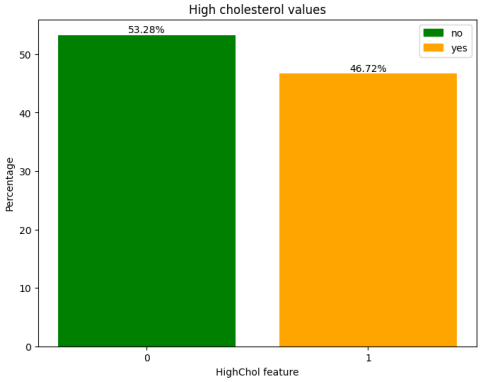
2.5 Analisi statistica univariata

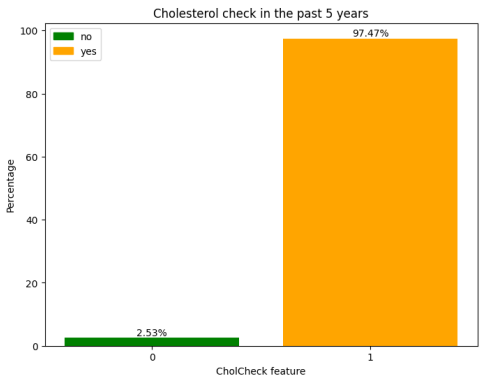
Dai precedenti grafici è possibile effettuare le seguenti osservazioni:

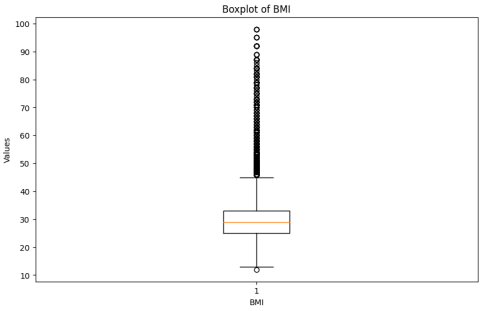
- Il 50% dei campioni presenti nel dataset è composto da pazienti con età compresa tra i 60 ed i 79 anni.
- Tra le variabili binarie presentate, unicamente le feature relative a `HighChol`, `Gender`, `Smoker` e `HighBP` si presentano con le rispettive sottoclassi bilanciate con un'incertezza massima del $\pm$ 7,25%.

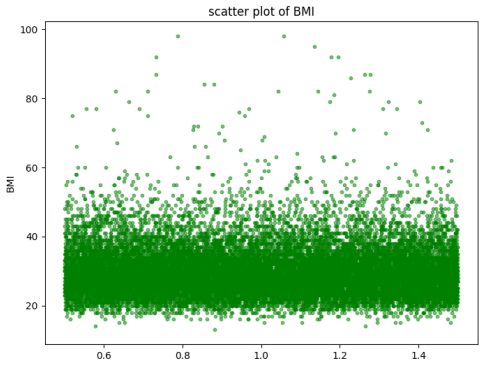

Bar Plot of Age
Frequency
Age

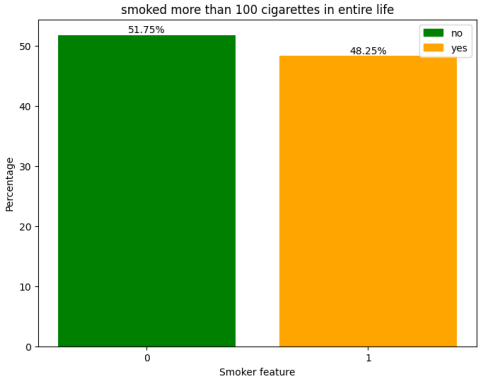

Gender
54.13%
45.87%
Female
Male
Percentage
Gender feature

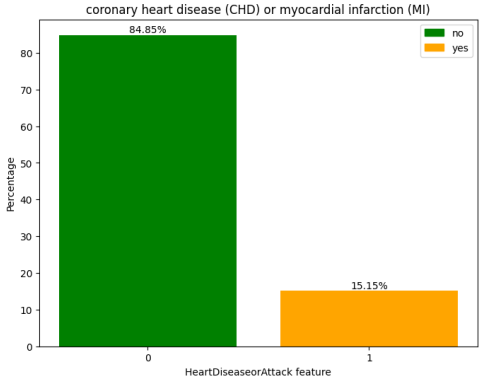

High cholesterol values
53.28%
46.72%
no
yes
Percentage
HighChol feature

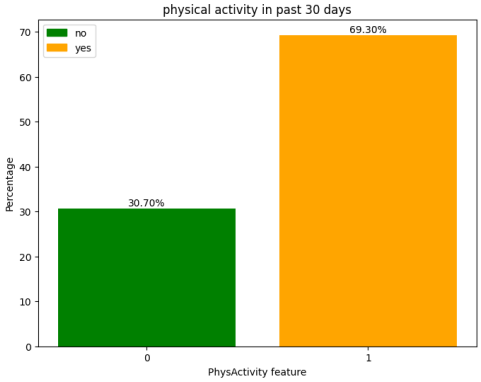

Cholesterol check in the past 5 years
97.47%
2.53%
no
yes
Percentage
CholCheck feature


Boxplot of BMI
Values
BMI

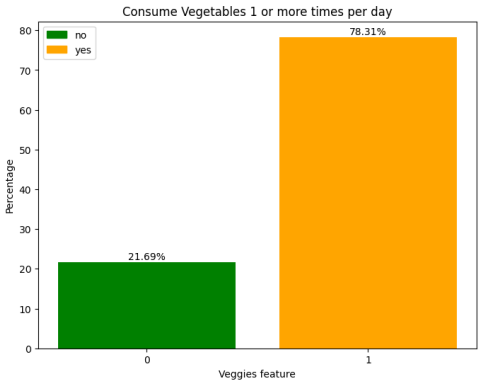

scatter plot of BMI
BMI

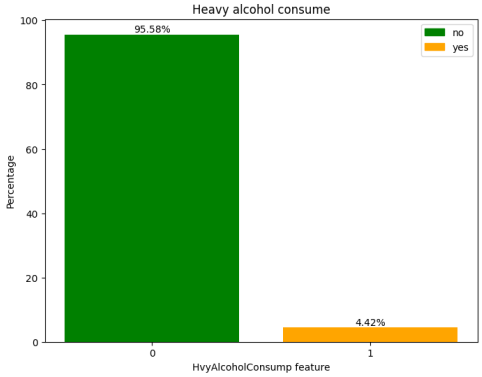

smoked more than 100 cigarettes in entire life
51.75%
48.25%
no
yes
Percentage
Smoker feature

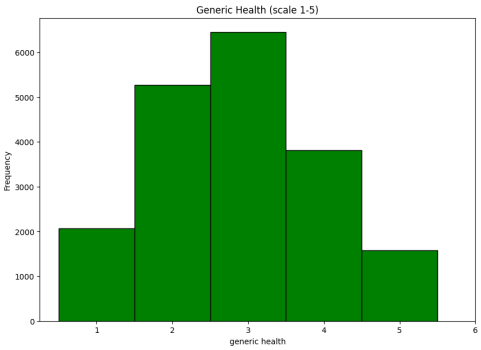

coronary heart disease (CHD) or myocardial infarction (MI)
84.85%
15.15%
no
yes
Percentage
HeartDiseaseorAttack feature

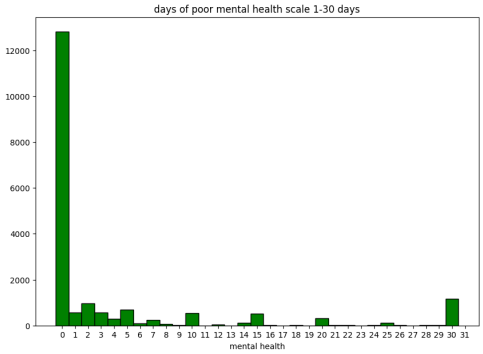

physical activity in past 30 days
30.70%
69.30%
no
yes
Percentage
PhysActivity feature

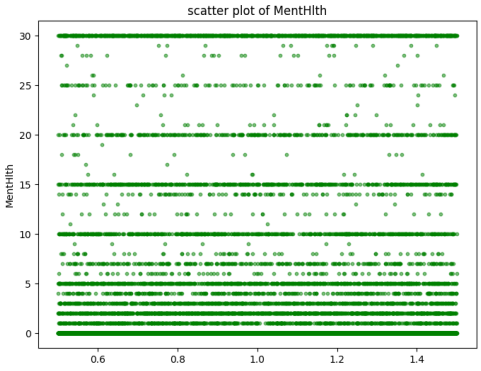

Consume Fruit 1 or more times per day
39.74%
60.26%
no
yes
Percentage
Fruits feature

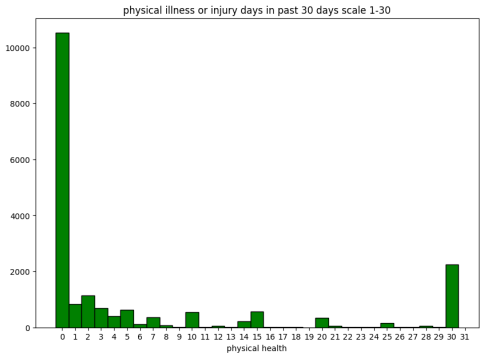

Consume Vegetables 1 or more times per day
no
yes
78.31%
21.69%
Percentage
Veggies feature

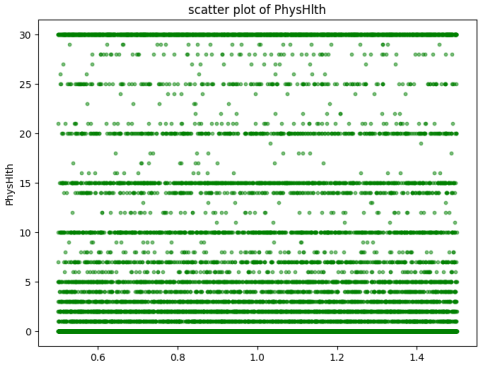

Heavy alcohol consume
no
yes
95.58%
4.42%
Percentage
HvyAlcoholConsump feature

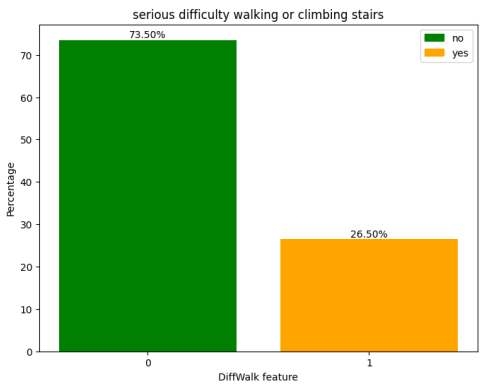

Generic Health (scale 1-5)
Frequency
generic health

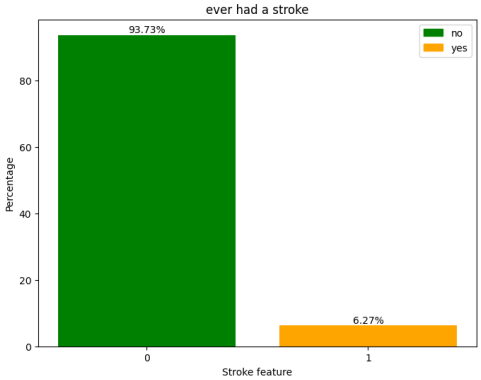

days of poor mental health scale 1-30 days
mental health

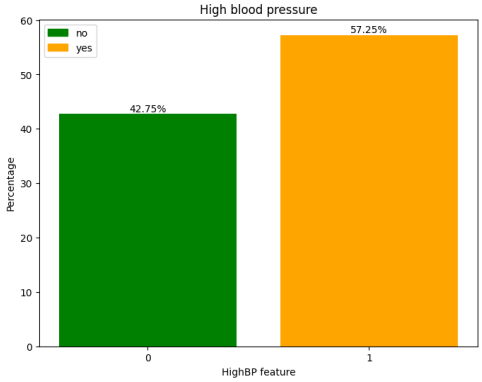

scatter plot of MentHlth
MentHlth


physical illness or injury days in past 30 days scale 1-30
physical health


scatter plot of PhysHlth
PhysHlth


serious difficulty walking or climbing stairs
no
yes
73.50%
26.50%
Percentage
DiffWalk feature


ever had a stroke
no
yes
93.73%
6.27%
Percentage
Stroke feature


High blood pressure
no
yes
42.75%
57.25%
Percentage
HighBP feature

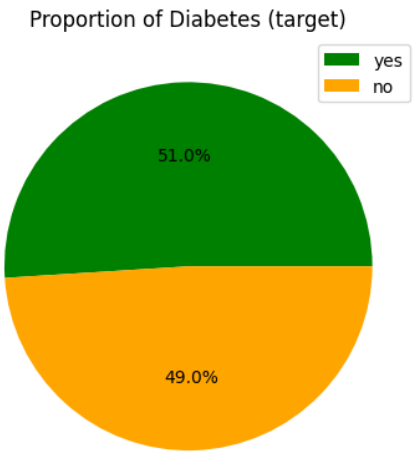
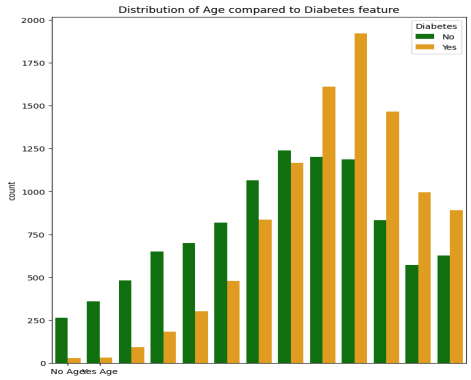


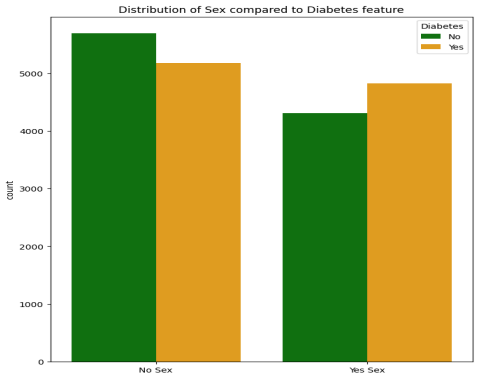
Figura 2.3: Rappresentazione delle features nell'analisi univariata

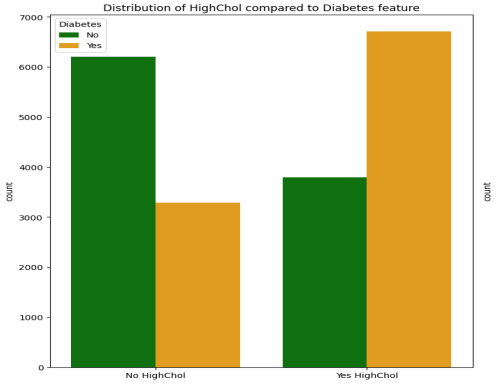
- La classe Target `Diabetes` risulta bilanciata.
- La feature continua `BMI` presenta degli outliers per valori superiori a 60.
- Nella feature `GenHlth`, la maggior parte dei rilevamenti (58%) considera la sua salute generica con un valore compreso tra 2 e 3.
- Nella feature `MentHlth`, il 64% dei pazienti comunica di non aver mai avuto problemi di salute mentale negli ultimi 30 giorni. Il secondo valore con frequenza più alta è il valore 30, ovvero chi ha definito che negli ultimi 30 giorni ha avuto problemi di salute mentale (5%). Dal grafico è possibile evidenziare come nel dataset per certi valori nella scala si ha una bassa frequenza di inserimento.
- Nella feature `PhysHlth`, il 52% comunica di non aver avuto problemi fisici negli ultimi 30 giorni. Il secondo valore più alto (11%) è fornito da chi comunica che, negli ultimi giorni, ha sempre avuto problemi fisici inserendo il valore 30. La frequenza dei restanti valori inseriti si attesta al di sotto del 5%. Dal grafico è possibile evidenziare come nel dataset per certi valori nella scala si ha una bassa frequenza di inserimento.

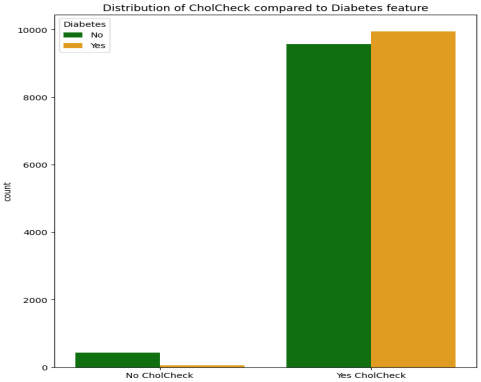
2.6 Analisi Statistica Multivariata

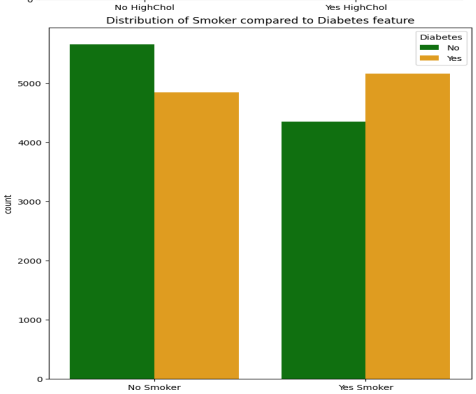
In questa fase si è deciso di mostrare la correlazione tra features categoriche e la variabile target. In primo luogo, si osserva che la probabilità di sviluppare il diabete aumenta con l'avanzare dell'età. Dai 55 ai 59 anni, questa probabilità è pressoché del 50%, mentre diventa nettamente più alta per le fasce di età più avanzate, toccando un picco tra i 65 ed i 75 anni. É importante notare che per le fasce di età più giovani è

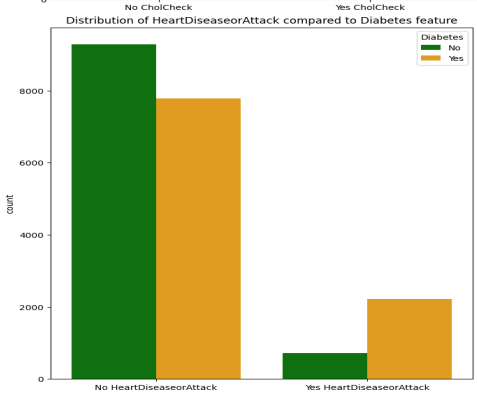

Distribution of Age compared to Diabetes feature
Diabetes
No
Yes
count
No Age
Yes Age

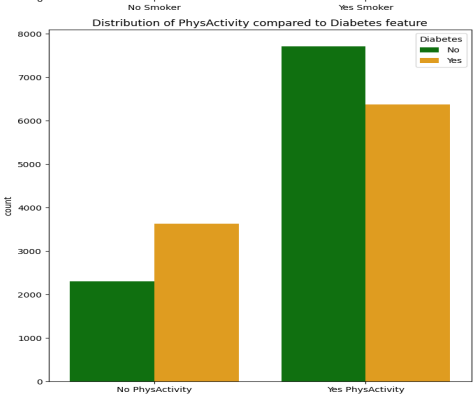

Distribution of Sex compared to Diabetes feature
Diabetes
No
Yes
count
No Sex
Yes Sex

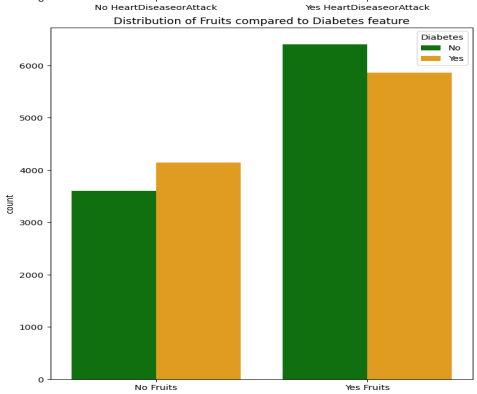

Distribution of HighChol compared to Diabetes feature
Diabetes
No
Yes
count
No HighChol
Yes HighChol

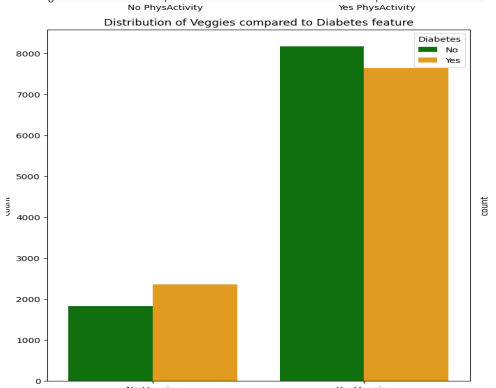

Distribution of CholCheck compared to Diabetes feature
Diabetes
No
Yes
count
No CholCheck
Yes CholCheck

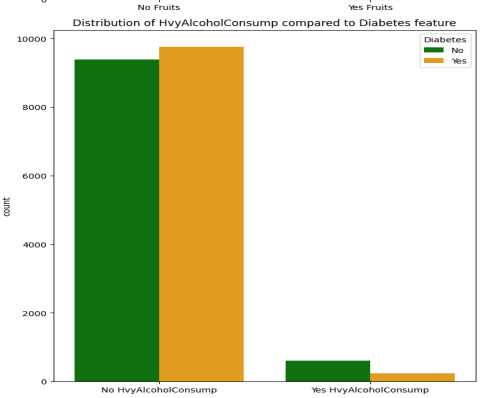

Distribution of Smoker compared to Diabetes feature
Diabetes
No
Yes
count
No Smoker
Yes Smoker


Distribution of HeartDiseaseorAttack compared to Diabetes feature
Diabetes
No
Yes
count
No HeartDiseaseorAttack
Yes HeartDiseaseorAttack


Distribution of PhysActivity compared to Diabetes feature
Diabetes
No
Yes
count
No PhysActivity
Yes PhysActivity


Distribution of Fruits compared to Diabetes feature
Diabetes
No
Yes
count
No Fruits
Yes Fruits


Distribution of Veggies compared to Diabetes feature
Diabetes
No
Yes
count
No Veggies
Yes Veggies


Distribution of HvyAlcoholConsump compared to Diabetes feature
Diabetes
No
Yes
count
No HvyAlcoholConsump
Yes HvyAlcoholConsump

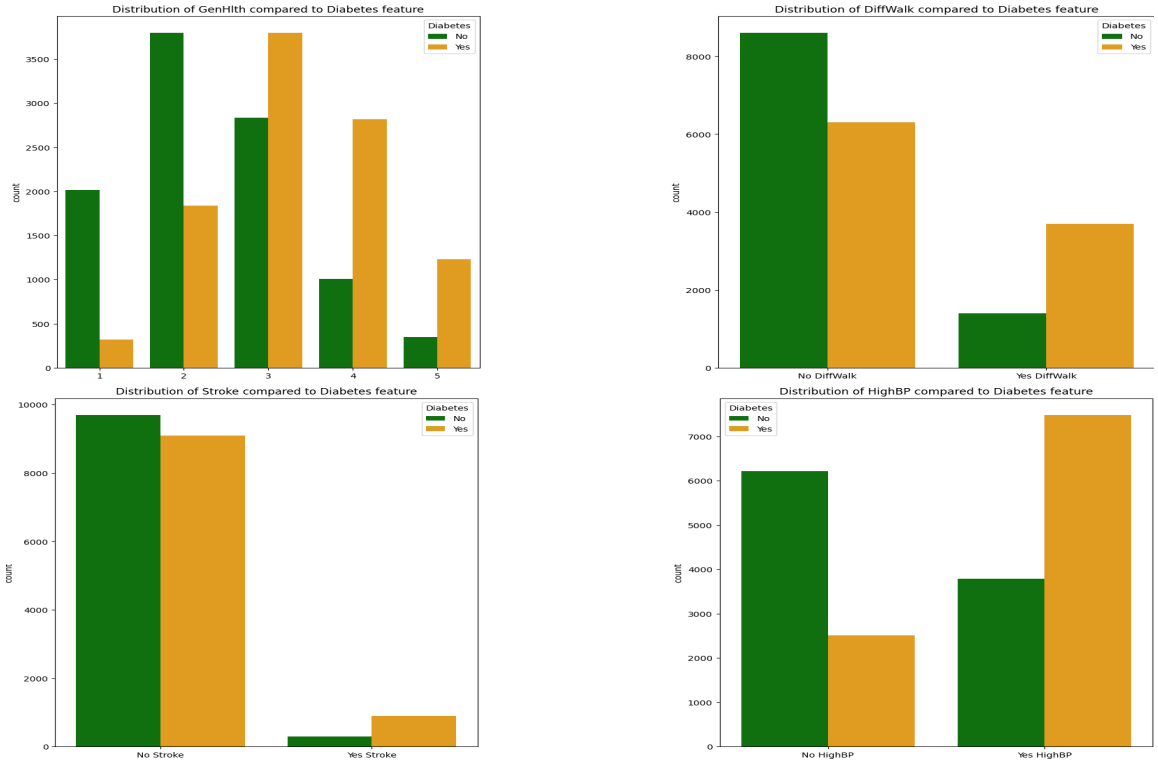


Figura 2.4: Rappresentazione delle features categoriche nell'analisi multivariata

più probabile non avere il diabete. Inoltre, si nota una leggera prevalenza del diabete negli uomini rispetto alle donne, con un'incidenza del 53% contro un 49%, essendo comunque una differenza minima la feature non fornisce un apporto significativo alla previsione del target. La presenza del colesterolo alto `HighChol` sembra essere correlata all'incidenza del diabete, con una probabilità del 64% per chi ha il colesterolo alto rispetto al 36% per chi non lo ha. Chi ha effettuato un controllo del colesterolo negli ultimi 5 anni (`CholCheck`) ha un'incidenza di diabete del 52%, mentre chi non lo ha effettuato ha un incidenza dell'11

2.7 Analisi multivariate non categoriche

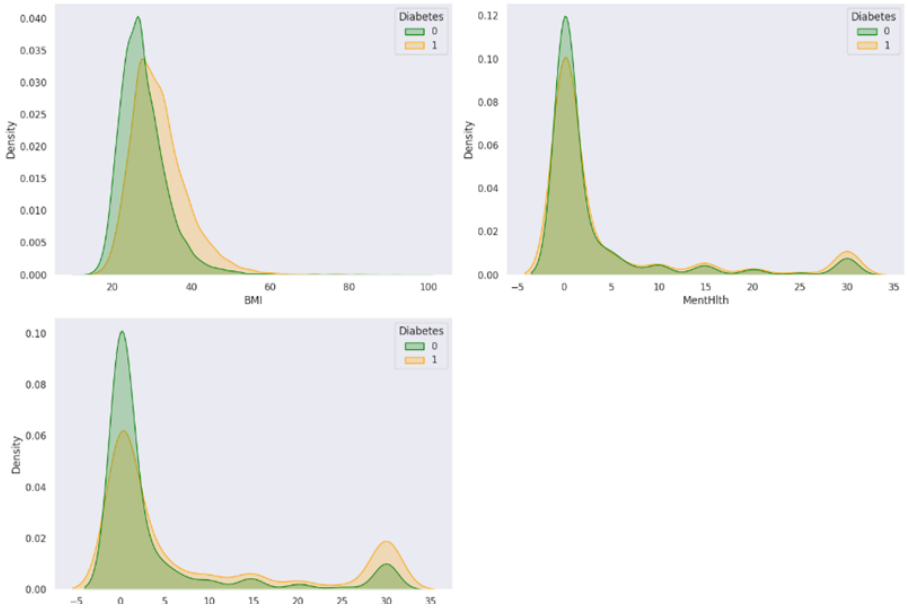


Figura 2.5: Rappresentazione delle features non categoriche nell'analisi multivariata

- BMI: Il grafico mostra la distribuzione di densità per i valori contenuti nella feature BMI rispetto ai pazienti diabetici e non. La densità per gli individui non diabetici raggiunge il picco intorno a un BMI di 25, valore fino al quale la densità dei pazienti diabetici è strettamente inferiore rispetto ai pazienti "sani". Dal valore di BMI = 25 in poi, la densità dei pazienti diabetici cresce notevolmente mantenendosi stabile per tutti i valori successivi, ricoprendo interamente l'area rappresentante la densità dei pazienti non diabetici. Per valori successivi ad un BMI di 60, le due distribuzioni si sovrappongono con una densità pari a 0, non apportando contenuto informativo per la previsione del diabete.

- MentHlth: Il grafico mostra la distribuzione di densità per i valori contenuti nella feature MentHlth rispetto ai pazienti diabetici e non. Gli individui non diabetici hanno un picco netto intorno a zero, ovvero coloro che non soffrono di bassa salute mentale, mentre quelli diabetici hanno una distribuzione più ampia ma raggiungono anch'essi il picco a zero. Per i valori successivi, le distribuzioni dei pazienti diabetici e non sono simili nella forma e contemplano i medesimi picchi locali. La densità dei pazienti diabetici supera la densità dei pazienti non diabetici in modo distinto unicamente per il valore 30, corrispondente a chi ha comunicato di soffrire di bassa salute mentale negli ultimi 30 giorni.

- PhysHlth: Il grafico mostra la distribuzione di densità per i valori contenuti nella feature PhysHlth rispetto ai pazienti diabetici e non. La linea verde (rappresentante i non diabetici) ha un picco intorno al segno 0 sull'asse PhysHlth,

indicando una alta densità di individui senza diabete e senza problematiche fisiche negli ultimi 30 giorni. La linea gialla (diabetici) mostra una distribuzione più diffusa, con il suo picco intorno al segno 30 sull'asse PhysHlth, indicando che gli individui con diabete tendono ad avere elevati giorni con problematiche fisiche.

2.8 Correlazione complessiva di ogni feature rispetto al diabete

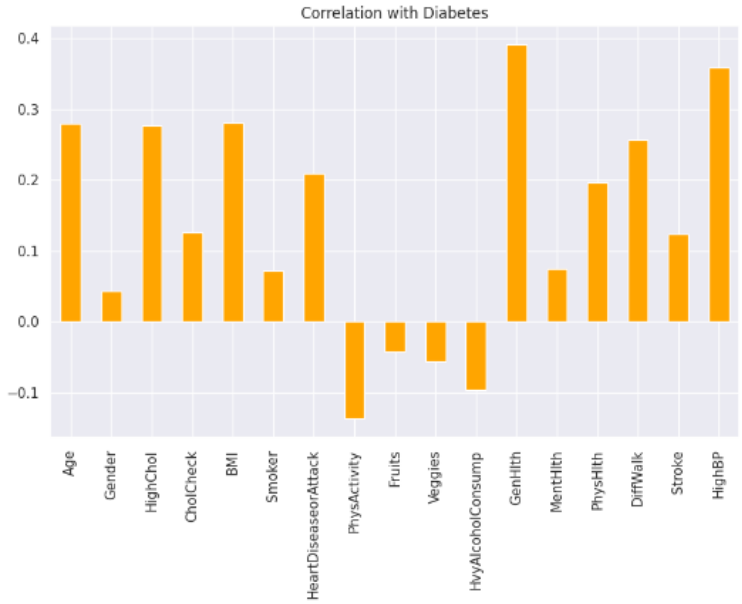


Figura 2.6: Correlazione feature rispetto al diabete

Da questo grafico è possibile visualizzare il valore di correlazione di ogni singola feature rispetto al target 'Diabetes'. In esso si possono notare come i valori di correlazione oscillino tra *-0,1 e + 0,4*, con i valori positivi che rappresentano un valore di correlazione diretta mentre quelli negativo un valore di correlazione inversa. Dal grafico è possibile notare come le feature Gender, Smoke, Fruits, Veggies e HvyAlcoholConsump abbiano un rapporto di correlazione inferiore al *+-0.1*, non influendo particolarmente sulla feature target. La feature più significativa è la GenHlth *(0.39)*, seguita da HighBP *(0.36)*. Tutte le restanti feature sono compreso in un intervallo di correlazione tra *+0.1 e +0.3*, fuorché PhysActivity che si attesta al *-0.12* circa.

2.9 Heatmap di correlazione tra le feature

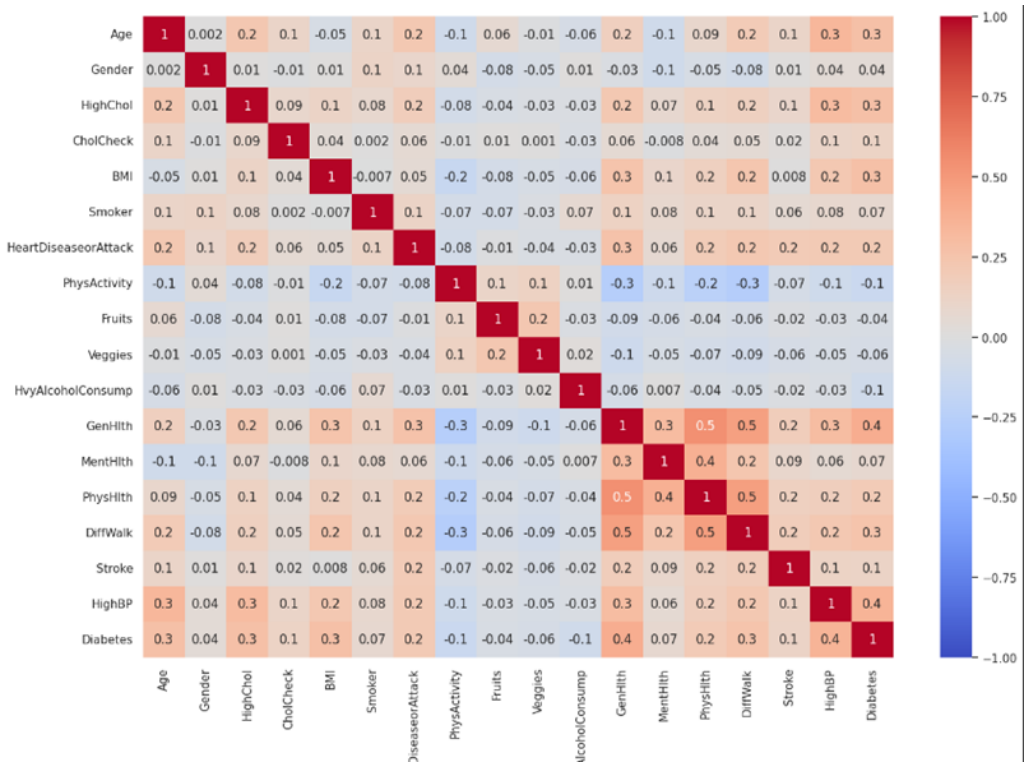


Figura 2.7: Heatmap di correlazione tra le feature

Il grafico soprastante rappresenta la matrice di correlazione, rappresentata come **heatmap**, delle feature del dataset. Esso rappresenta non solo la correlazione presente tra ogni feature e la variabile target , come nell'immagine 2.6 bensì anche la correlazione diretta ed inversa tra le feature stesse.

2.10 Selezione delle feature ed eliminazione degli outliers

Come è stato analizzato all'interno della sezione 2.5 dai plot generati in fase di analisi univariata delle features, si è scelto di eliminare gli outliers relativi al parametro BMI con valori superiori a 60. Questo perché la feature di BMI utilizzata ricopre valori fino a 65, le restanti rilevazioni con valori superiori a questa soglia sono state considerate come fuorvianti ed errate per l'addestramento di un modello su di esse.

```
sampled_df = sampled_df.drop(sampled_df[sampled_df['BMI'] > 60].index)
```

Il medesimo procedimento è stato eseguito eliminando per le feature *PhysHlth* e *MentHlth*

```
df_count = sampled_df['MentHlth'].value_counts()
```

```
values_to_drop = df_count[df_count < 80].index
sampled_df = sampled_df.drop(sampled_df[sampled_df['MentHlth'].isin(values_to_drop)].
    index)
```

```
sampled_df_count = sampled_df['PhysHlth'].value_counts()
values_to_drop = sampled_df_count[sampled_df_count < 80].index
sampled_df = sampled_df.drop(sampled_df[sampled_df['PhysHlth'].isin(values_to_drop)].
    index)
```

Si nota che i valori che non sono stati mai inseriti dai pazienti sulla scala da 1 a 30, oltre che per i valori inseriti con una frequenza minore di 80, in modo da permettere al modello di apprendimento di generalizzare in maniera ottimale su un insieme di dati inseriti con maggior frequenza, oltre che ridurre il rumore nei dati e concentrare l'attenzione sulle informazioni più rilevanti per la previsione del target.

In seguito, si è deciso di eliminare le feature con correlazione con il target inferiore al +- *0.1*.

```
sampled_df.drop(columns=['Gender'], inplace=True)
sampled_df.drop(columns=['Smoker'], inplace=True)
sampled_df.drop(columns=['Fruits'], inplace=True)
sampled_df.drop(columns=['Veggies'], inplace=True)
sampled_df.drop(columns=['MentHlth'], inplace=True)
sampled_df.drop(columns=['HvyAlcoholConsump'], inplace=True)
```

Questo processo ha lo scopo di ridurre il quantitativo di risorse computazionali e temporali nella fase di addestramento del modello, oltre che permettere una migliore generalizzazione in fase predittiva conservando comunque i parametri con una correlazione maggiormente significativa con il target.

Capitolo 3

Albero Decisionale

Nel contesto di questo progetto, ci si è concentrati sull'applicazione del modello degli alberi decisionali. la prima operazione è stata quella di suddividere il dataset in set di addestramento e di test, una pratica fondamentale. Viene applicato un rapporto di divisione del 70-30, con il 70% dei dati utilizzato per addestrare il modello e il 30% riservato per valutarne le prestazioni. Il parametro random_state è impostato su 42 in modo che i dati siano sempre divisi nello stesso modo.

```
from sklearn.model_selection import train_test_split
X = df.drop('Diabetes', axis=1)
y = df['Diabetes']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
    =42)
```

3.1 Analisi del modello

Per prima cosa, si è addestrato un albero decisionale con parametri predefiniti per avere un'idea di base delle prestazioni, il modello è stato addestrato utilizzando il set di addestramento (`X_train e y_train`). Una volta addestrato, sono state generate previsioni sia per il set di addestramento che per il set di test utilizzando il metodo `predict`.

Per valutare le prestazioni del modello, sono stati utilizzati rapporti di classificazione `classification_report` stampati per entrambi i set. Questi rapporti forniscono dettagliate metriche di valutazione, come precision, recall e F1score, consentendo una valutazione completa delle capacità predittive del modello su dati già noti (set di addestramento) e su dati non visti durante l'addestramento (set di test).

	Precision	Recall	F1-Score	Support
Prestazioni sul Training Set				
Class 0	0.91	0.97	0.94	6413
Class 1	0.97	0.90	0.94	6577
Accuracy			0.94	12990
Macro Avg	0.94	0.94	0.94	12990
Weighted Avg	0.94	0.94	0.94	12990
Prestazioni sul Test Set				
Class 0	0.64	0.68	0.66	2736
Class 1	0.67	0.63	0.65	2832
Accuracy			0.65	5568
Macro Avg	0.66	0.66	0.65	5568
Weighted Avg	0.66	0.65	0.65	5568

Tabella 3.1: Prestazioni Set di Addestramento e Test per albero decisionale

Nel contesto della nostra analisi sono state valutate le performance del modello di classificazione utilizzando la curva ROC, la quale illustra le capacità del modello di discriminare tra la classe positiva e negativa. Di seguito si è ottenuto *un'area Under the Curve* (AUC) pari a 0.67. Nel dataset in esame il valore target è binario, dunque si trattandosi di un problema di classificazione binaria, si ritiene AUC un buon metodo di valutazione. Nello specifico si è ottenuto un AUC di 0.66, la quale suggerisce una performance moderata, indicando una discreta discriminazione tra le classi.

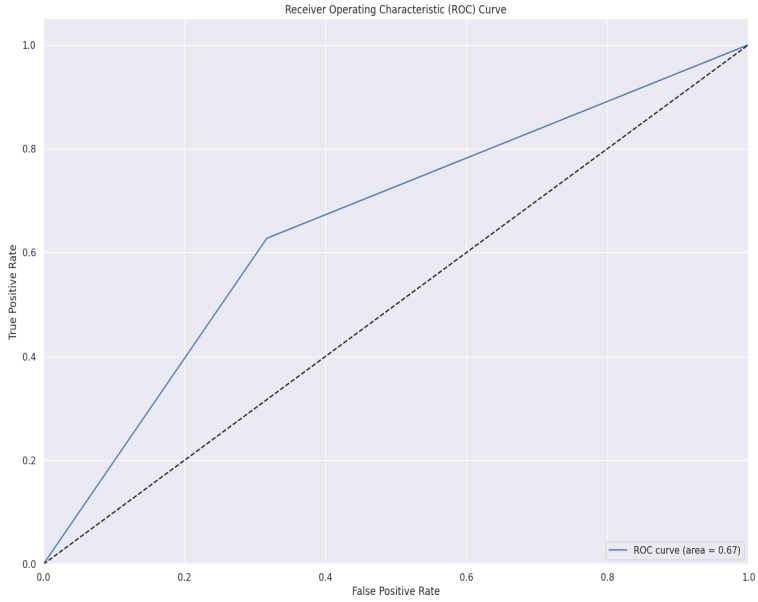


Figura 3.1: ROC iniziale

Nel grafico, la curva ROC si discosta dalla retta diagonale casuale, evidenziando il trade-off tra il tasso di falsi positivi e il tasso di veri positivi. Un'area maggiore sotto la curva ROC indica una migliore capacità discriminante del modello.

La matrice di confusione offre informazioni preziose per comprendere le prestazioni del modello e guidare l'ottimizzazione dei parametri, consentendo di apportare modifiche mirate al fine di migliorare le previsioni.

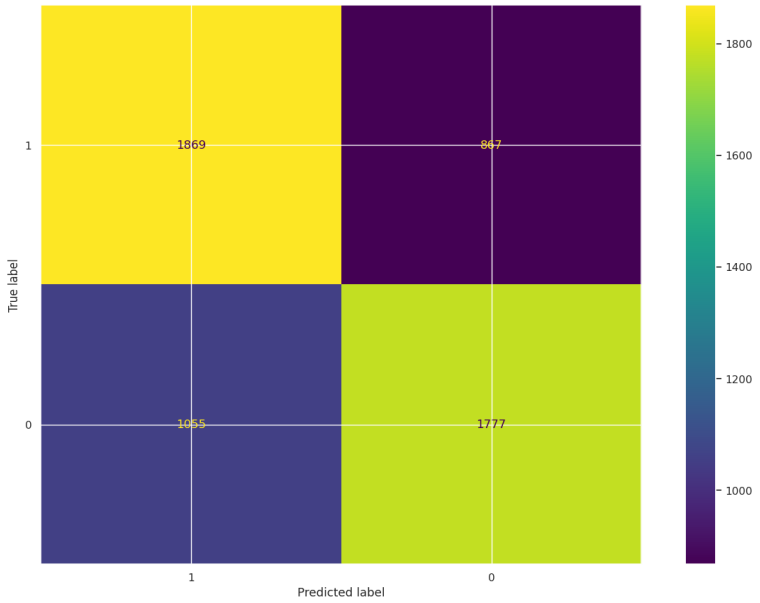


Figura 3.2: Matrice di confusione iniziale

3.2 Osservazioni

Esaminando la matrice di confusione e le metriche di valutazione associate, emergono importanti considerazioni sull'efficacia del modello, la quale fornisce una rappresentazione dettagliata delle prestazioni del modello di classificazione, permettendo di identificare specificamente i falsi positivi, falsi negativi e valutare il modo in cui il modello gestisce le diverse classi.

Il modello mostra una buona capacità di predire correttamente la classe positiva (Veri Positivi, TP: 1869) e la classe negativa (Veri Negativi, TN: 1777) nel set di test. Tuttavia, occorre porre attenzione ai Falsi Positivi (FP: 1055) e Falsi Negativi (FN: 867), poiché indicano rispettivamente errori di classificazione in cui il modello ha erroneamente previsto una classe positiva quando era negativa, e viceversa.

Come si nota dalla tabella 3.1 nel set di addestramento, il modello ha dimostrato una buona precisione per entrambe le classi. La recall, che indica la capacità del modello di individuare tutti i casi positivi, è elevata per la classe 0 e un po' più bassa per la classe 1. L'F1-score, una metrica che bilancia precisione e recall, è alto per

entrambe le classi. L'accuracy complessiva del modello sul set di addestramento è del 94%.

Nel set di test, come si nota dalla tabella 3.1 le performance del modello sono peggiorate.

In generale si può dire che il modello sembra avere buone prestazioni sul set di addestramento, ma mostra una certa perdita di performance quando applicato a dati non visti nel set di test. Si nota come la training set performance è praticamente perfetta e molto diversa da quella del test set, questo ci indica che il modello molto probabilmente è caratterizzato da **overfitting**. Successivamente a queste osservazioni, si è presa la decisione di esplorare miglioramenti attraverso la ricerca di iperparametri. L'obiettivo è ottimizzare la configurazione del modello per cercare una maggiore capacità di generalizzazione, riducendo eventuali fenomeni di overfitting e migliorando le prestazioni sul set di test.

In particolare, sono stati testati sia l'approccio di ***Grid Search***, che esamina in modo esaustivo tutte le combinazioni di iperparametri definite in una griglia, sia l'approccio di ***Randomized Search***, che campiona casualmente combinazioni di iperparametri. Dopo confronto delle prestazioni, è stata scelta la strategia di `Randomized Search` per la sua maggiore efficienza e la capacità di gestire una vasta griglia di iperparametri.

3.3 Ricerca degli iperparametri

Per cercare di migliore le prestazioni, si è deciso di concetrarsi sul miglioramento dei parametri di default dell'albero decisionale.

La dichiarazione presenta la definizione di una griglia degli iperparametri, un componente essenziale nel contesto dell'ottimizzazione di modelli di machine learning, con particolare riferimento a un algoritmo basato su alberi decisionali.

La griglia è definita attraverso il dizionario `param_dist`, il quale incorpora una serie di iperparametri fondamentali da considerare durante la fase di ricerca degli iperparametri ottimali per il modello. I parametri inclusi sono:

- `'criterion'`: Misura di impurità utilizzata nella costruzione dell'albero, con opzioni `'gini'` e `'entropy'`.
- `'max_features'`: Numero massimo di features considerate per la suddivisione di un nodo, con opzioni `'sqrt'`, `'log2'`, e `'None'`.
- `'min_samples_split'`: Numero minimo di campioni richiesti per suddividere un nodo interno, con valori da 2 a 50, passo 2.

- `'min_samples_leaf'`: Numero minimo di campioni richiesti in una foglia, con valori da 1 a 8.
- `'max_depth'`: Limite massimo di profondità dell'albero, con valori specifici (5, 10, 15, 20) e opzione `'None'`.

A seguito dell'operazione di ricerca degli iperparametri, sono stati ottenuti i seguenti risultati.

```
Migliori Parametri: {'min_samples_split': 14, 'min_samples_leaf': 1, 'max_features':
    None, 'max_depth': 5, 'criterion': 'entropy'}
```

3.4 Analisi del modello ottimizzato

Una volta trovati i migliori parametri da utilizzare , il modello è stato ri-addestrato utilizzando il set di addestramento (`X_train e y_train`), per verificare se i parametri trovati possano portare ad un miglioramento del modello in uso. Una volta addestrato, sono state generate previsioni sia per il set di addestramento che per il set di test utilizzando il metodo predict.

Per valutare le prestazioni del modello, come in precedenza sono stati utilizzati i rapporti di classificazione `classification_report` stampati per entrambi i set.

	Precision	**Recall**	**F1-Score**	**Support**
Prestazioni sul Training Set				
Class 0	0.76	0.67	0.71	4032
Class 1	0.70	0.78	0.74	3874
Accuracy			0.73	7906
Macro Avg	0.73	0.73	0.73	7906
Weighted Avg	0.73	0.73	0.73	7906
Prestazioni sul Test Set				
Class 0	0.77	0.66	0.71	1769
Class 1	0.68	0.78	0.73	1620
Accuracy			0.72	3389
Macro Avg	0.72	0.72	0.72	3389
Weighted Avg	0.73	0.72	0.72	3389

Tabella 3.2: Prestazioni Set di Addestramento e Test del modello ottimizzato

Inoltre, come in precedenza si è deciso di visualizzare la curva ROC, al fine di confrontare le due curve ROC ottenute.

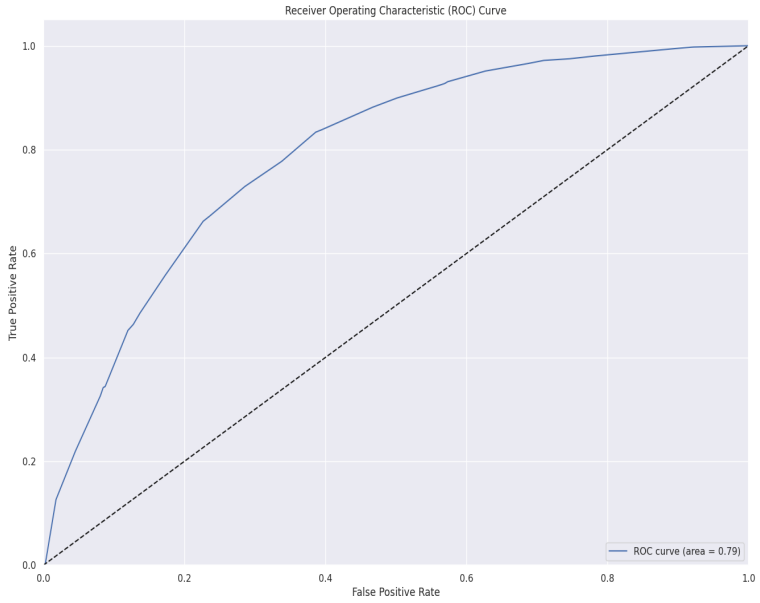


Figura 3.3: Curva ROC a seguito dell'ottimizzazione

la matrice di confusione si presenta nella seguente maniera.

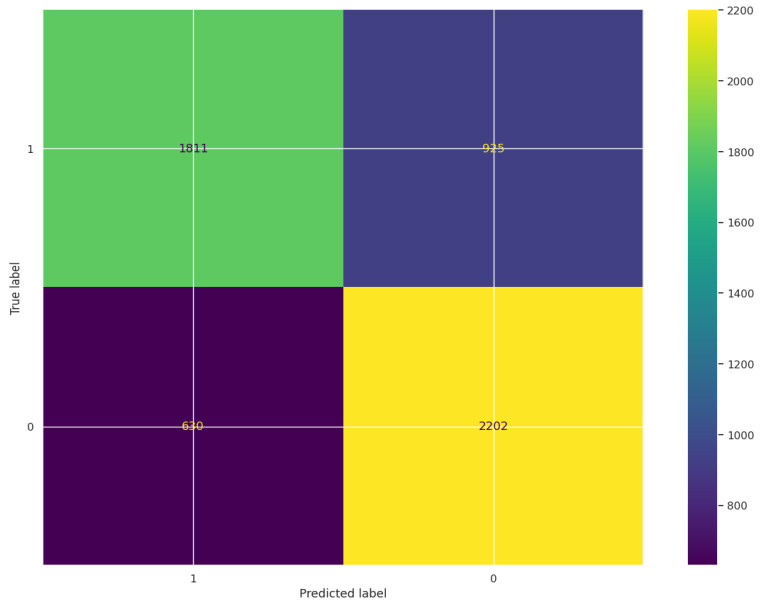


Figura 3.4: Matrice di confusione a seguito dell'ottimizzazione

3.5 Osservazioni sul modello ottimizzato

Le prestazioni del modello sui set di addestramento e test mostrano alcune considerazioni importanti. Nel set di addestramento, si osserva un bilanciamento relativamente equo tra le classi. Tuttavia, la precisione leggermente inferiore e la differenza tra recall delle due classi potrebbero indicare una maggiore sensibilità alla classe positiva.

Sul set di test, le metriche mostrano una performance simile, ma con un leggero calo rispetto al set di addestramento, che però si può considerare accettabile.

In generale, l'accuracy complessiva del 72% nel set di test riflette una capacità accettabile di generalizzazione del modello.

L'ottimizzazione degli iperparametri ha prodotto un discreto miglioramento nella performance del modello, evidenziato attraverso la matrice di confusione. Nella classe negativa, sebbene il numero di falsi negativi sia leggermente aumentato da 867 a 925, si è osservata una riduzione dei falsi positivi da 1055 a 630 nella classe positiva. Questo indica una maggiore precisione nella classificazione degli esempi positivi. Complessivamente, l'AUC è salito da 0.67 a 0.79, indicando una notevole miglioramento nella capacità del modello di discriminare tra le classi.

Questo risultato è promettente e suggerisce che la revisione degli iperparametri ha portato ad un modello più efficace nella separazione delle classi, dunque possiamo ritenerci soddisfatti.

3.6 10-fold cross validation

Come ulteriore metrica utilizzata abbiamo utilizzato la 10-fold cross validation , allenando i modelli con albero decisionale avente i nuovi iperparametri. L'utilizzo della validazione incrociata 10-Fold è essenziale in quanto fornisce una stima robusta delle prestazioni del modello su diverse suddivisioni dei dati di addestramento, riducendo il rischio di valutazioni distorte dovute a particolari configurazioni casuali dei dati.

```
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score

n_fold = 10
folds = KFold(n_splits=n_fold, shuffle=True)

accuracy_k_fold_dt = []

for n_fold, (train_idx, valid_idx) in enumerate(folds.split(X_train, y_train)):
    X_train_fold, X_valid_fold = X_train.iloc[train_idx], X_train.iloc[valid_idx]
    y_train_fold, y_valid_fold = y_train.iloc[train_idx], y_train.iloc[valid_idx]

    random_search.fit(X_train_fold, y_train_fold)
    # Fai previsioni sul set di validazione e calcola l'accuratezza
    y_valid_pred = random_search.predict(X_valid_fold)
    accuracy_k_fold_dt.append(accuracy_score(y_valid_fold, y_valid_pred))
```

In seguito è stato calcolato intervallo di confidenza al 95%, ottenendo

$$\boxed{(0.7191166379306725, 0.7336932158029686)}$$

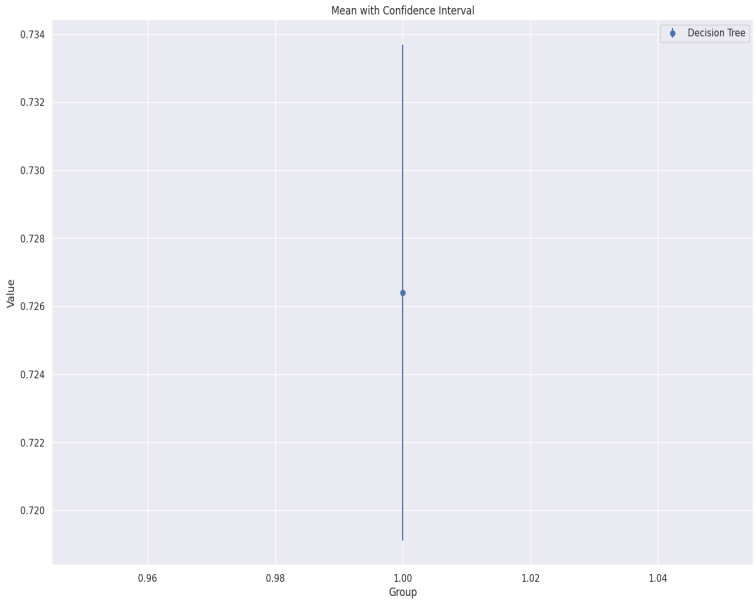


Figura 3.5: Intervallo di confidenza

L'accuratezza del modello allenato precedentemente rientra nell'intervallo calcolato con la 10-fold cross validation

3.7 Considerazioni Finali

La riduzione dell'overfitting rappresenta un avanzamento significativo nel miglioramento della capacità del modello di apprendere da dati specifici di addestramento e, al contempo, generalizzare in maniera più efficace a nuovi dati. Tale progresso è evidente nella minore disparità tra le performance registrate sui set di addestramento e di test, indicando una maggiore stabilità nel comportamento del modello.

L'applicazione della RandomSearch per la ricerca degli iperparametri ottimali ha giocato un ruolo cruciale in questo processo. Attraverso l'esplorazione efficiente dello spazio degli iperparametri, è stato possibile individuare una configurazione ottimale che ha massimizzato il valore dell'AUC. L'ottimizzazione indiretta della AUC ha contribuito a mitigare l'overfitting, migliorando la capacità del modello di generalizzare su dati precedentemente non osservati.

La visualizzazione dell'albero decisionale post-ottimizzazione riflette la semplificazione della struttura dell'albero stesso. Questa semplificazione è risultata in una maggiore capacità di generalizzazione.Pertanto, l'applicazione della RandomSearch non solo ha migliorato direttamente le prestazioni del modello, ma ha anche agito come un meccanismo di regolarizzazione, riducendo la complessità del Decision Tree e, di conseguenza, mitigando l'overfitting complessivo.

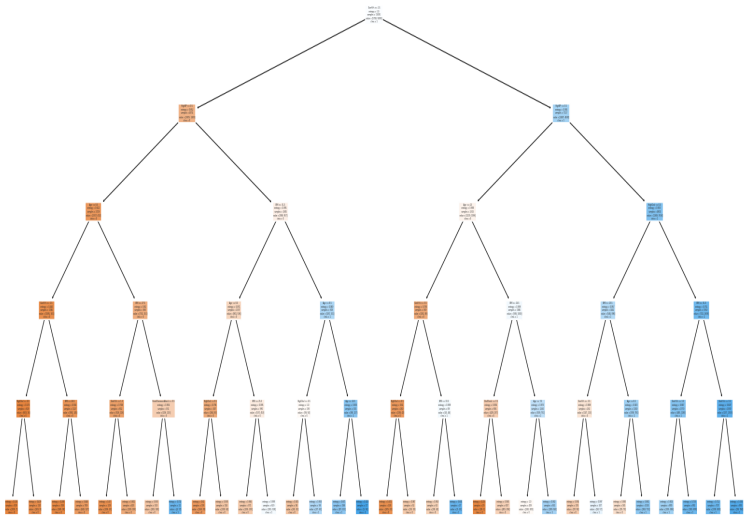


Figura 3.6: Rappresentazione Albero Decisionale

Capitolo 4

Reti Neurali

Il secondo modello utilizzato per lo sviluppo del progetto sono le **reti neurali**.
Nella fase iniziale, è stata effettuata l'estrazione delle features e della variabile target dal dataset. Questo processo è importante perché i valori appena recuperati verranno utilizzati nella fase successiva di divisione del dataset in due insiemi distinti: un set di addestramento e un set di test. In particolare, è stato adottato un rapporto di divisione del 70-30, con il 70% dei dati utilizzato per addestrare il modello e il 30% riservato per valutarne le prestazioni. Il parametro *random state* è impostato con un valore di 42, in modo da ottenere sempre la sempre la stessa divisione dei dati del dataset.

```
from sklearn.model_selection import train_test_split

y = sampled_df['Diabetes']
X = sampled_df.drop(columns=['Diabetes'])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
    =42)
```

4.1 Architettura della rete neurale

Per lo sviluppo del modello si è utilizzata la funzione *Sequential* della libreria *Keras*, che permette di implementare facilmente una rete neurale. L'architettura scelta per tale modello è composta da un layer di input a cui viene assegnato un numero di neuroni pari a 11, che rappresentano il numero di features in input. Al secondo layer è stato scelto di assegnare lo stesso numero di neuroni, a seguito di diversi esperimenti con risultati ottimali. La funzione di attivazione scelta per questo strato è la *ReLU (Rectified Linear Activation)*. Dato che il caso studio ha natura binaria, il layer di output ha dimensione uno, e pertanto è stata impiegata la funzione di attivazione *sigmoide*

(sigmoid). Quest'ultima trasforma i valori in ingresso in un intervallo compreso tra 0 e 1.

```
model = Sequential()
model.add(Dense(11, input_shape=(11,), activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Per la configurazione del processo di addestramento del modello è stato utilizzato il metodo *model.compile*. Essendo in un contesto di classificazione binaria, la funzione di perdita (*loss function*) adottata è la cross-entropy (*binary_crossentropy*). Come optimizer si è scelto di adottare l'algoritmo *Adam*, mentre la metrica di valutazione è l'accuratezza (*accuracy*).

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

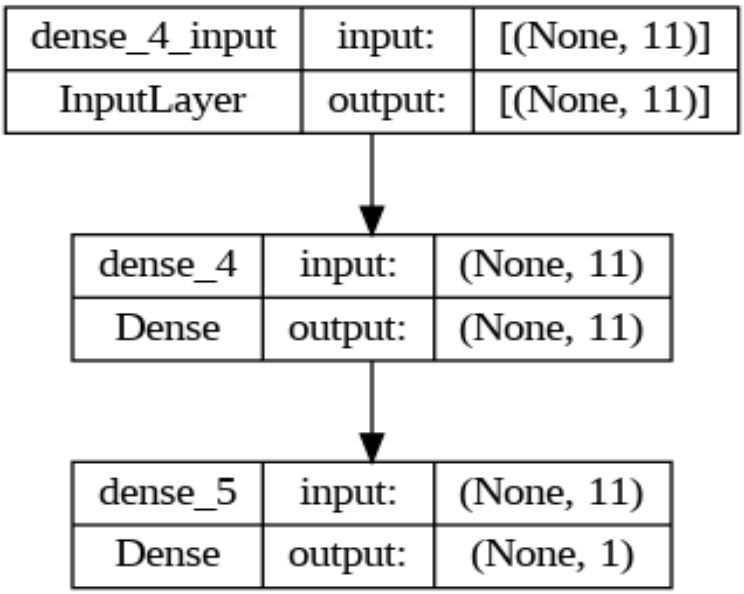


Figura 4.1: Rappresentazione architettura della rete neurale

4.2 Addestramento della rete neurale

Per l'addestramento della rete neurale è stato impiegato il metodo *model.fit*. Quest'ultimo addestra la rete neurale usando il set di training (*X train* e *y train*). Oltre a questi parametri, bisogna tenere in considerazione anche altri due aspetti:

- il numero di epoche (*epochs*), che denotano il numero di iterazioni in cui l'algoritmo opera sull'intero set di dati di addestramento,
- il parametro batch (*batch_size*), che specifica il numero di campioni da considerare per l'aggiornamento dei parametri del modello .

Per il caso studio in oggetto, il valore ottimale di epochs è risultato essere pari a 40, mentre per il parametro batch 10.

Da un'analisi dei risultati ottenuti, è possibile concludere che si è in assenza di underfitting e overfitting, come evincibile dai grafici (vedasi figura 4.2 e figura 4.3), grazie alla presenza di uno scarto minimo tra i valori finali. Infatti, dal primo grafico si denota come sia il training loss che la validation loss decrescono fino ad un valore di 53% di loss, pur mantenendo uno scarto minimo tra i valori finali di loss. Dal secondo grafico, invece, sono presenti un training accuracy e un validation accuracy che crescono fino a un determinato punto di stabilità attorno al 73% di accuracy e, analogamente al grafico precedente, viene mantenuto uno scarto minimo tra i valori finali di accuratezza.

```
1 history = model.fit(X_train, y_train, epochs=40, batch_size=10, verbose=1,
    validation_data=(X_test, y_test))
```

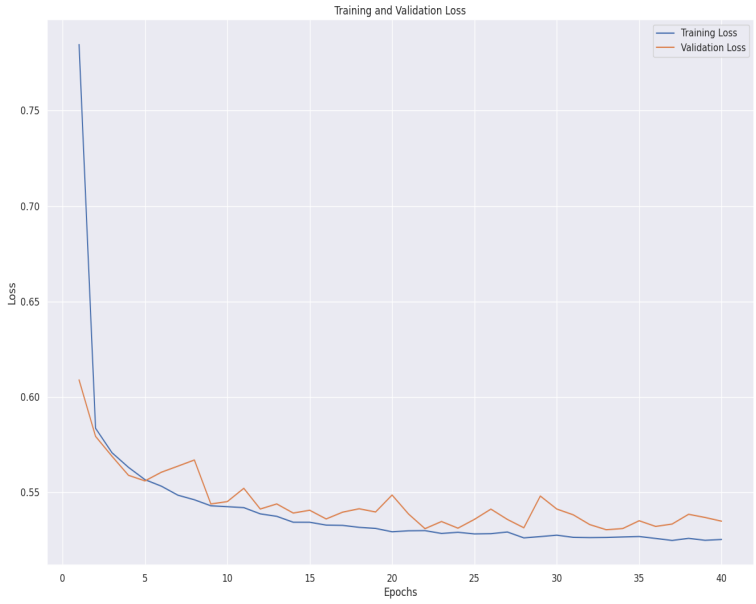


Figura 4.2: Training and Validation Loss

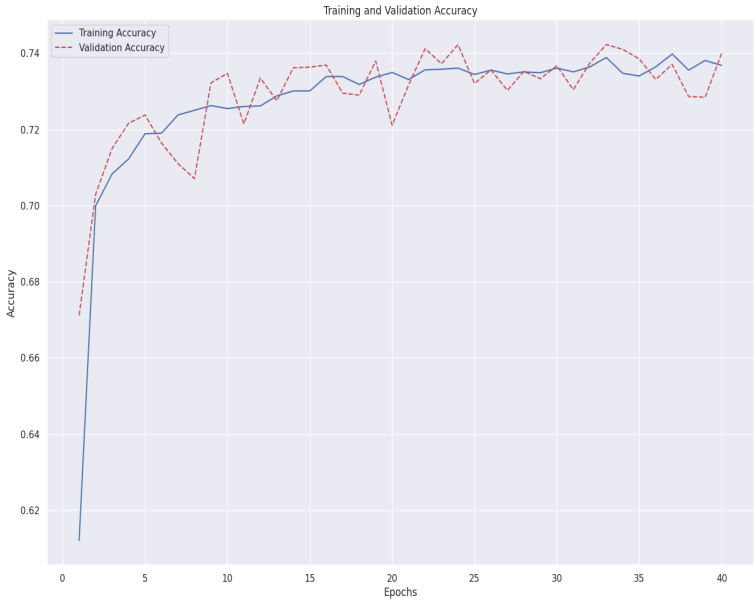


Figura 4.3: Training and Validation accuracy

In conclusione, dall'analisi dei grafici si è determinata l'assenza di overfitting, escludendo così il ricorso a tecniche atte per ridurre tale criticità come il dropout o la regolarizzazione.

4.3 Metodi di valutazione

4.3.1 Report di classificazione

Al fine di valutare le prestazioni del modello, sono stati impiegati i report di classificazione (*classification_report*) prodotti sia per il set di addestramento che per il set di test. Questi report forniscono metriche dettagliate di valutazione, come *precision*, *recall* e *F1-score*, permettendo una valutazione completa delle capacità predittive del modello.

	Precision	Recall	F1-Score	Support
Prestazioni sul Training Set				
Class 0	0.77	0.67	0.72	6413
Class 1	0.71	0.81	0.76	6577
Accuracy			0.74	12990
Macro Avg	0.74	0.74	0.74	12990
Weighted Avg	0.74	0.74	0.74	12990
Prestazioni sul Test Set				
Class 0	0.77	0.67	0.72	2736
Class 1	0.72	0.81	0.76	2832
Accuracy			0.74	5568
Macro Avg	0.74	0.74	0.74	5586
Weighted Avg	0.74	0.74	0.74	5568

Prestazioni Set di Addestramento e Test

Esaminando le metriche ottenute, emerge chiaramente un'accuratezza del 74% sia nel set di addestramento che in quello di test. Più precisamente, in entrambi i set, si evidenziano equilibri relativamente uniformi tra le classi, nonostante si riscontri un lieve aumento nella precision e le differenze nelle misurazioni di recall tra le due classi dei due set indicano una maggiore sensibilità verso la classe positiva.

In definitiva, l'accuratezza complessiva del 74% nel set di test suggerisce una buona capacità del modello di generalizzare i dati. Ciò sottolinea la robustezza della sua performance anche al di fuori del contesto di addestramento.

Infine, la media non ponderata (macro Avg) e la media ponderata (Weighted Avg) confermano che sussiste una coerenza tra il set di addestramento e quello di test, attestandosi tutte ad un valore uguale al 74%, coerente con le metriche di precision, recall e F1-score delle due classi.

4.3.2 Matrice di confusione

La matrice di confusione è uno strumento utilizzato per valutare le prestazioni del modello.

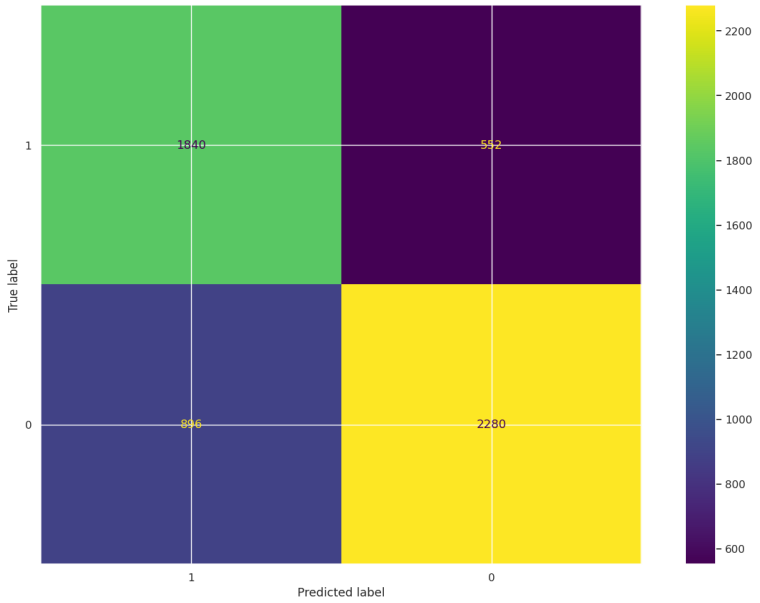


Figura 4.4: Matrice di confusione

Da un'analisi della matrice di confusione, emergono i seguenti risultati:

- *Veri positivi (TP)*: sono state correttamente identificate 1840 istanze come positive;
- *Falsi negativi (FN)*: sono stati identificati 552 casi in cui istanze positive sono state erroneamente classificato come negative;
- *Falsi positivi (FP)*: 896 istanze sono state erroneamente classificate come positive;
- *Veri negativi (TN)*: 2280 istanze sono state correttamente classificate come negative.

Dalla matrice di confusione creata per il modello, emerge una tendenza a classificare in modo più accurato i casi negativi rispetto a quelli positivi.

4.3.3 Curva ROC e AUC

Come fatto per il modello del Decision Tree, anche in questo caso studio si è deciso di valutare le performance della rete neurale tramite l'utilizzo della curva ROC, dove maggiore è l'area sottesa a tale curva, maggiore è la capacità del modello di discriminare.

Calcolando, quindi, l'Area Under the Curve (AUC) si ottiene un valore pari a 0.74, che indica una migliore capacità discriminatoria del modello.

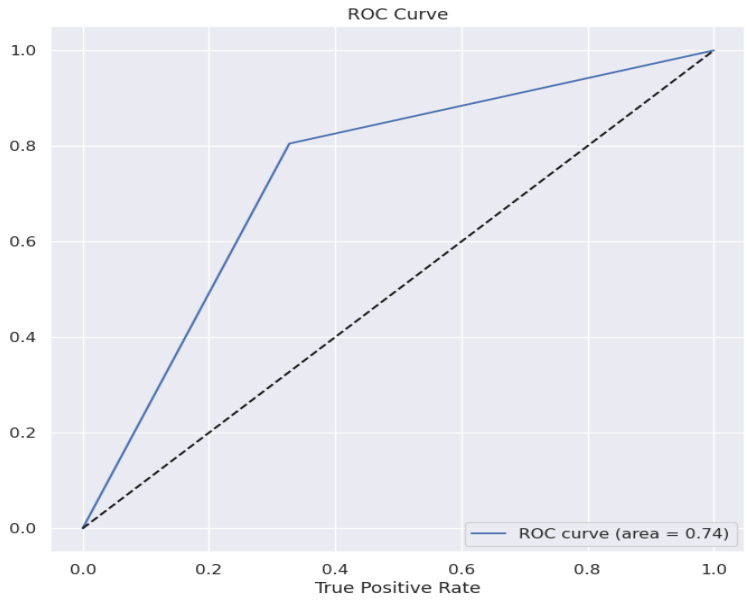


Figura 4.5: Curva ROC

4.4 10-fold cross validation

Per controprovare la robustezza del modello, è stato deciso di applicare anche in questo caso studio, la tecnica del k-fold cross validation utilizzata nel modello di Decision Tree, fissando a 10 il valore k. I risultati che derivano da questo approccio offrono una stima delle prestazioni del modello maggiormente accurata, in quanto considerano le variazioni nei dati di addestramento e di test.

In seguito è stato calcolato intervallo di confidenza al 95%, ottenendo

(0.727263583573923, 0.7441759853252303)

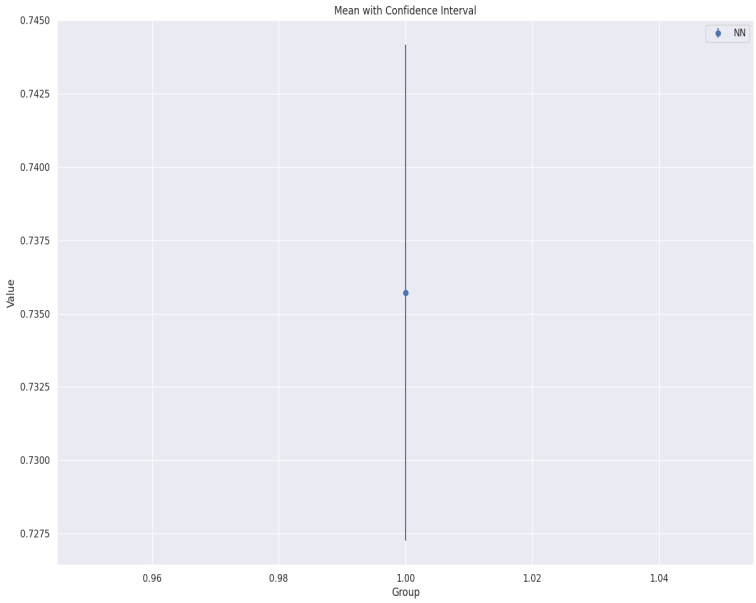


Figura 4.6: Intervallo di confidenza

L'accuratezza del modello allenato precedentemente rientra nell'intervallo calcolato con la 10-fold cross validation. Pertanto, il modello è robusto e i risultati sono attendibili e generalizzabili.

Capitolo 5

Naive Bayes

Un metodo utilizzato per la previsione della variabile target è *Naive Bayes*. In seguito all'eliminazione delle features meno correlate con il target il nostro dataset è composto da 11 feature, di cui 9 categoriche e 2 non. Si è scelto quindi di utilizzare la variante categorica di *Naive Bayes*, codificando le variabili non categoriche, ovvero BMI e PhysHlth, in categoriche attraverso l'uso di `LabelEncoder()`.

```
label_encoder = LabelEncoder()
sampled_df['PhysHlth'] = label_encoder.fit_transform(sampled_df['PhysHlth'])
sampled_df['BMI'] = label_encoder.fit_transform(sampled_df['BMI'])
```

5.1 Analisi del modello

Come nei modelli precedenti, si è deciso di utilizzare una divisione del 70-30, con il 70% dei dati utilizzato per addestrare il modello e il 30% riservato per valutarne le prestazioni. Successivamente, un classificatore Naive Bayes Categorical viene inizializzato, addestrato e valutato sia sui dati di allenamento che su quelli di test.

```
# Initialize the Categorical Naive Bayes classifier
cnb = CategoricalNB()
# Measure the start time
start_time = time.time()
# Train the Categorical Naive Bayes classifier
cnb.fit(X_train, y_train)
# Measure the end time
end_time = time.time()
# Calculate the training time
nb_training_time = end_time - start_time
# Evaluate the model on training data
```

L'analisi delle metriche di performance del modello evidenzia una buona generalizzazione su entrambi i set di dati, training e test. La precisione e la recall bilanciate

per entrambe le classi indicano un modello capace di predire con accuratezza sia gli esempi positivi che quelli negativi. La coerenza tra le metriche sul training e sul test set suggerisce che il modello non sta subendo overfitting, mantenendo una robusta capacità predittiva su nuovi dati.

5.2 Osservazioni

L'accuracy complessiva del 73% sul test set è coerente con le metriche più dettagliate di precision, recall e F1-score. La media non ponderata (Macro Avg) e ponderata (Weighted Avg) delle metriche riflettono un equilibrio nelle performance tra le due classi, fornendo un'ulteriore conferma dell'efficacia del modello in contesti di classificazione bilanciata. La considerazione del supporto delle classi contribuisce a contestualizzare le metriche in relazione alle dimensioni delle stesse, fornendo una prospettiva più approfondita.

In conclusione, il modello presenta risultati promettenti in termini di prestazioni predittive, confermando la sua idoneità per il compito di classificazione in questione. Tuttavia, valutazioni più approfondite dei casi di falsi positivi e falsi negativi potrebbero offrire ulteriori spunti per ottimizzazioni future.

	Precision	**Recall**	**F1-Score**	**Support**
Prestazioni sul Training Set				
Class 0	0.73	0.73	0.73	6413
Class 1	0.73	0.74	0.74	6577
Accuracy			0.73	12990
Macro Avg	0.73	0.73	0.73	12990
Weighted Avg	0.73	0.73	0.73	12990
Prestazioni sul Test Set				
Class 0	0.72	0.73	0.73	2736
Class 1	0.74	0.73	0.73	2832
Accuracy			0.73	5568
Macro Avg	0.73	0.73	0.73	5568
Weighted Avg	0.73	0.73	0.73	5568

Tabella 5.1: Prestazioni sul Training Set e sul Test Set

I risultati di accuratezza sono comunque soddisfacenti, dalla matrice di confusione creata per il modello si nota come quest'ultimo tenda a classificare correttamente

maggiormente i casi negativi rispetto ai positivi.

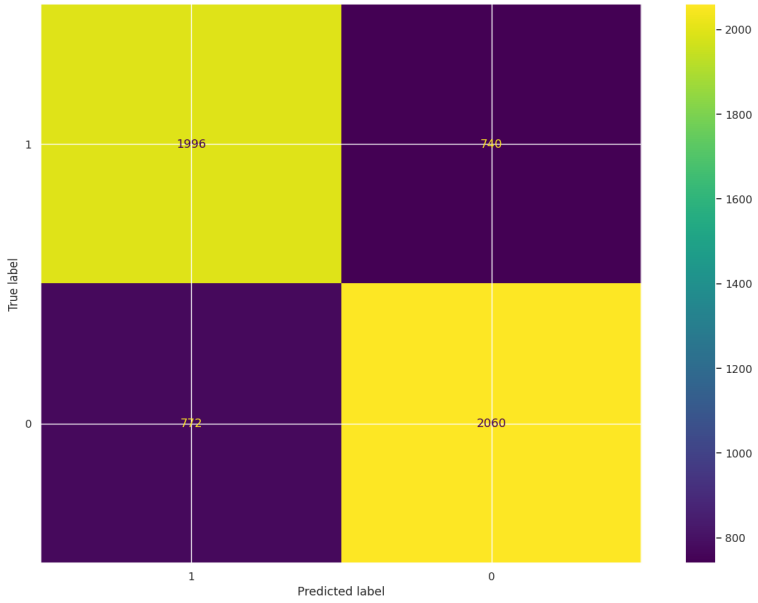


Figura 5.1: Matrice di confusione

Nel corso dell'analisi del nostro modello, è stata condotta un'approfondita valutazione delle prestazioni attraverso l'utilizzo della Curva ROC . I risultati della valutazione della Curva ROC per il nostro modello indicano un'Area Under the Curve pari a 0.80, riflettendo un'ottima capacità discriminativa del nostro modello.

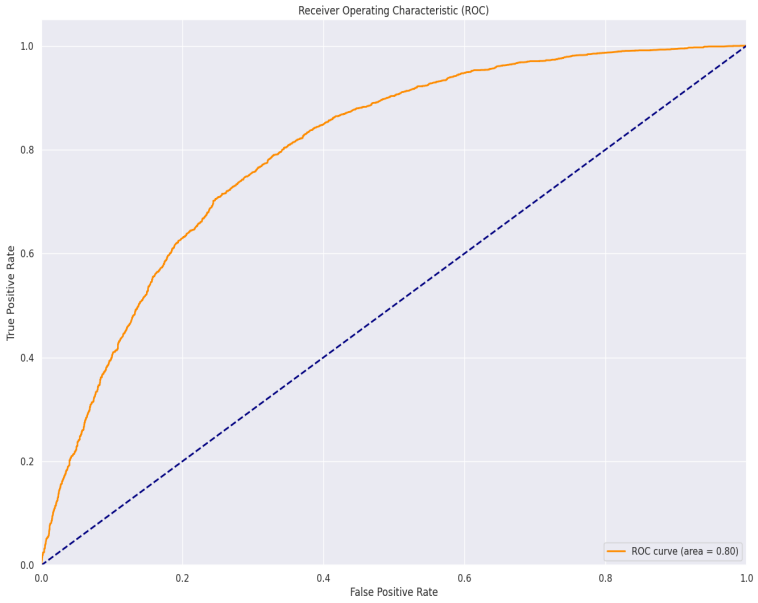


Figura 5.2: Curva ROC

5.3 10-fold Cross Validation

Nell'ultimo stadio della nostra analisi, si è utilizzata la 10-fold cross validation per valutare la robustezza e la generalizzazione del nostro modello basato su Naive Bayes. I risultati ottenuti da questa procedura forniscono una stima più accurata delle prestazioni del modello, in quanto tiene conto delle variazioni nei dati di allenamento e di test. La media delle metriche di performance su tutti i fold fornisce una valutazione più affidabile rispetto a una singola divisione del dataset.

In seguito è stato calcolato intervallo di confidenza al 95%, ottenendo

(0.7267134668533078, 0.7368739080504642)

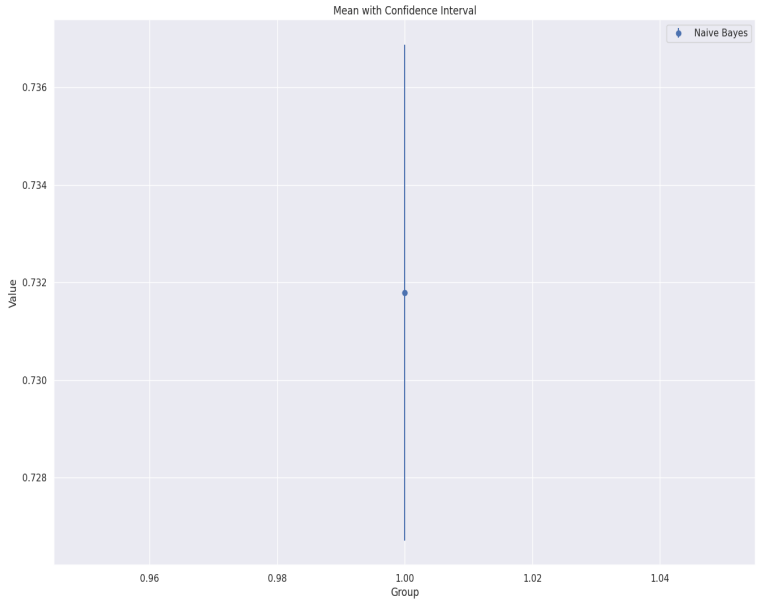


Figura 5.3: Intervallo di confidenza

L'accuratezza del modello allenato precedentemente rientra nell'intervallo calcolato con la 10-fold cross validation, dunque ci si può ritenere soddisfatti.

Capitolo 6

Considerazioni Finali

In questo progetto è stata affrontata la questione clinica del diabete cercando di sviluppare modelli predittivi che fossero in grado di contribuire all'ottimizzazione delle diagnosi e delle strategie di trattamento. Nello specifico, sono stati sviluppati e confrontati tre diversi modelli predittivi, nell'ordine uno inerente all'albero decisionale, uno alla rete neurale e infine uno al naive bayes. La comparazione dei modelli è avvenuta impiegando le curve ROC, gli intervalli di confidenza e i tempi di training. Le curve ROC sono state ampiamente utilizzate come indicatore diagnostico per valutare l'accuratezza dei modelli, mentre l'analisi dei tempi di elaborazione forniva informazioni cruciali sulla praticabilità e l'efficienza implementativa dei diversi approcci. Infine, l'inclusione degli intervalli di confidenza mirava a conferire una robustezza statistica alle valutazioni, fornendo una valutazione più approfondita delle differenze in termini di prestazioni tra i modelli.

6.1 Confronto Curve ROC

In questo contesto, Decision Tree e Naive Bayes hanno mostrato prestazioni simili, con valori di AUC soddisfacenti (pari a 0.79 e 0.80 rispettivamente) denotando una buona capacita di predire le classi positive e negative della variabile target.

La curva ROC in Naive Bayes è la più vicina all'angolo in alto a sinistra del grafico, il che indica che ha il miglior equilibrio tra il tasso di veri positivi e il tasso di falsi positivi. Questo suggerisce che il modello in considerazione ha una buona capacità di distinguere tra le due classi della feature 'Diabetes'.

La curva ROC per il modello relativo alle reti neurali è meno ottimale rispetto a quella di Naive Bayes. Questo suggerisce che il modello in considerazione potrebbe avere una maggiore probabilità di falsi positivi a determinate thresholds. L'AUC è 0.74, che è un risultato accettabile, ma non è ottimale. Questo valore indica che il

modello ha il 74% di probabilità di classificare correttamente un esempio positivo casuale rispetto a un esempio negativo casuale.

Anche se l'AUC per il modello Decision Tree è quasi simile a quello del Naive Bayes, la forma della curva mostra che il modello potrebbe avere un equilibrio meno favorevole tra il tasso di veri positivi e il tasso di falsi positivi a determinate thresholds. Questo potrebbe indicare che Decision tree potrebbe avere una maggiore probabilità di falsi positivi o una minore probabilità di veri positivi in determinate soglie rispetto al modello Naive Bayes. Un AUC di 0.79 è comunque valutabile come un ″buon″ risultato.

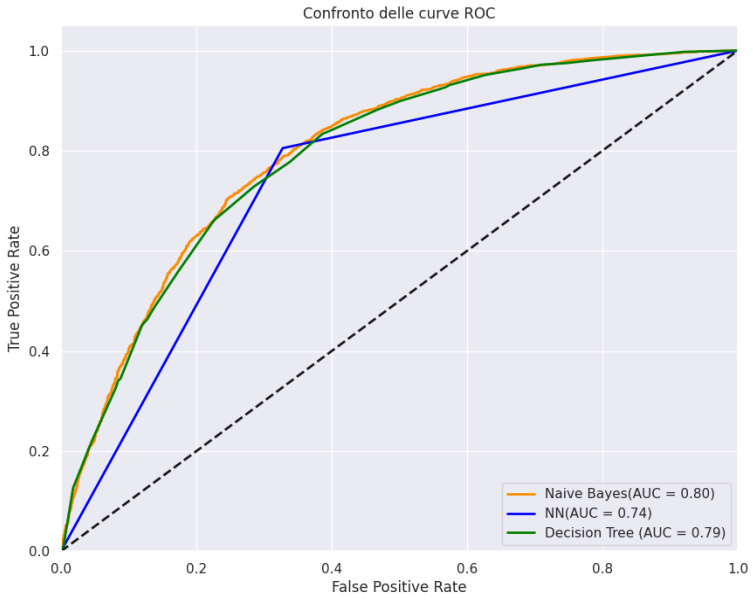


Figura 6.1: Confronto ROC modelli

6.2 Confronto Intervalli di confidenza

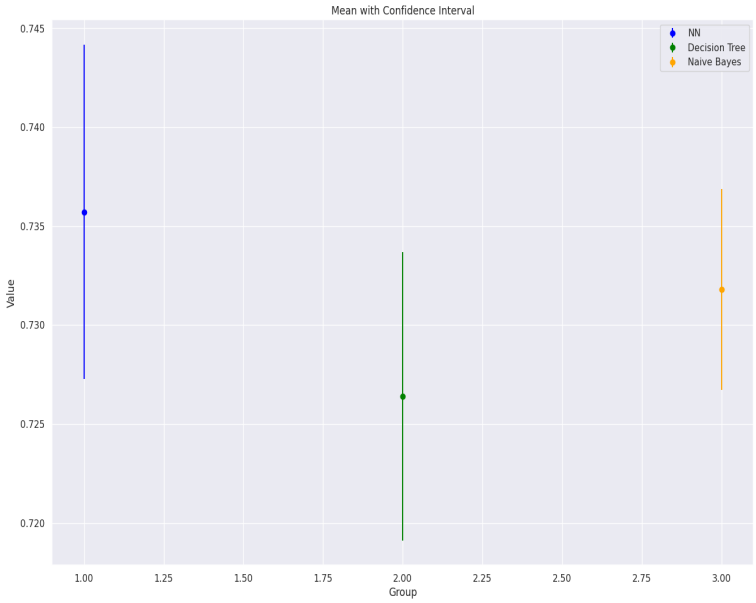


Figura 6.2: Confronto Intervalli di confidenza tra i modelli

Come ulteriore metodo di analisi è stato scelto di impiegare gli intervalli di confidenza in quanto rappresentano un valido strumento di valutazione della variabilità delle prestazioni del modello e forniscono una prospettiva sulla robustezza delle sue previsioni. Tutti e tre i modelli predittivi sviluppati mostrano risultati discreti e similari tra loro, ritenendosi così soddisfatti delle prestazioni ottenute.

6.3 Confronto Tempi di addestramento

```
Tempo di training Decision tree: 0.016 secondi
Tempo di training Neural Network: 148.755 secondi
Tempo di training Naive Bayes: 0.012 secondi
```

Per ultimo, si e deciso di comparare il tempo di addestramento tra i modelli. Tale parametro risulta di estrema importanza nello scegliere quale modello predittivo adottare. Confrontando i modelli in termini di tempistiche di training, l'addestramento di Decision Tree e ' Naive Bayes hanno riportato i risultati migliori, rispettivamente 0.016 secondi e 0.012 secondi. Purtroppo, data la sua complessità strutturale, la rete neurale ha richiesto un tempo di addestramento notevolmente maggiore, circa 148.755 secondi. Parametro da tenere in assoluta considerazione.