GlucoHealth : Documentazione

A.A. 2024/2025

Progetto di Ingegneria della Conoscenza

A cura di:
Giuseppe Giampietro, 774968
g.giampietro3@studenti.uniba.it

Professore del corso :
Fanizzi Nicola

GitHub del progetto : https://github.com/giuseppegmp/ICON24-25

Indice dei contenuti della documentazione:

Capitolo 0) Introduzione

Il **diabete mellito**, comunemente conosciuto solo come **diabete**, è una patologia di tipo cronico dovuta a una disfunzione a carico dell'**insulina**, l'ormone secreto del pancreas che regola il livello di glucosio nel sangue, trasportando gli zuccheri (il glucosio) dal sangue alle cellule affinché vengano assorbite e convertiti in fonte di energia per l'organismo. Nel momento in cui questo meccanismo subisce un'alterazione, come accade nei pazienti diabetici, il glucosio si accumula e i suoi livelli nel sangue aumentano, portando il paziente in uno stato di **iperglicemia**.
Esistono principalmente diverse tipologie di diabete:

- Diabete di tipo I : una condizione autoimmune che porta l'organismo a distruggere le cellule del pancreas preposte alla produzione di insulina. La principale conseguenza della distruzione delle cellule beta (il bersaglio del sistema immunitario) è un aumento della glicemia, detto in altri termini, l'iperglicemia. Va specificato che questo tipo di diabete viene definito insulino-dipendente o giovanile. Queste due denominazioni sono dovute al fatto che il diabete di tipo I insorge solitamente durante l'età adolescenziale. Secondo i dati del Ministero della Salute, in Italia, a soffrire di questo tipo di diabete sono circa trecentomila persone. Richiede la somministrazione quotidiana di insulina per eguagliare i livelli di glucosio nel sangue.

- Diabete di tipo II : una patologia di natura cronica dovuta al cattivo funzionamento dell'insulina che determina un eccesso di zuccheri nel sangue. Questo eccesso di glucosio prende il nome di iperglicemia e può essere la conseguenza di due condizioni, ossia l'insufficiente produzione di insulina e l'azione dell'insulina non adeguata. Questa è la tipologia di diabete più comune, diffusa a causa delle cattive abitudini legate al movimento e all'alimentazione. È diffuso maggiormente tra gli adulti.

- Diabete gestazionale : una condizione che si verifica quando una donna sviluppa alti livelli di zucchero nel sangue per la prima volta durante la gravidanza. La sua comparsa è legata al fatto che, durante il periodo della gravidanza, la placenta produce diversi tipi di ormoni che antagonizzano l'azione dell'insulina, causando così un aumento dei valori della glicemia nel sangue. Nella maggior parte dei casi, l'organismo femminile reagisce aumentando la produzione di insulina, ma nel caso in cui il pancreas non sia in grado di produrre una quantità maggiore di questo ormone, la glicemia nel sangue aumenta e si manifesta dunque trasportarsi in questo stato di diabete. Va contrastato con una corretta alimentazione, adeguato esercizio fisico e nei casi estremi con un controllo farmacologico.

Nella maggior parte dei casi possono trascorrere anni prima di accorgersi di essere diabetici, questo perché la condizione di eccesso di glucosio nel sangue si sviluppa in maniera graduale

e i sintomi non sono facilmente riconoscibili. Tra i più comuni ci sono una sete intensa, il bisogno di urinare frequentemente e in quantità elevate, visione offuscata, calo di peso, affaticamento e stanchezza. Tuttavia, la sintomatologia varia a seconda del tipo di diabete cui si soffre. Se il diabete non viene curato adeguatamente, può portare a gravi problemi cardiovascolari, ai reni e alla retina, oltre alle varie infezioni.
Ad oggi, il diabete non è ancora una malattia da cui si può guarire, ma è trattabile sia con una terapia farmacologica, sia adottando stili di vita corretti che prevedono una sana alimentazione e un esercizio fisico regolare.
Nel caso dei pazienti diabetici di tipo I, l'esercizio fisico e le abitudini alimentari vanno abbinate a una più rigorosa terapia insulinica. Quelli diabetici di tipo II, oltre a seguire un corretto stile di vita, in alcuni casi sono costretti ad assumere farmaci ipoglicemizzanti.
Attraverso la realizzazione di questo progetto, si potrà essere in grado di **predire** attraverso delle misurazioni diagnostiche, la possibilità che un paziente sia diabetico o meno, diagnosticandone la malattia.

Capitolo 1) Requisiti Fondamentali

Il progetto è stato realizzato utilizzando il linguaggio Python, scelto per la sua efficienza e flessibilità nella manipolazione e analisi dei dati. Come ambiente di sviluppo è stato utilizzato Visual Studio Code (versione 1.99.3), data la sua leggerezza e ampia disponibilità di estensioni dedicate alla programmazione in Python.
Sono state impiegate diverse librerie, ognuna con uno scopo specifico :

- **Pandas** : per la gestione, la pulizia e l'analisi dei dati in formato tabellare;
- **NumPy** : per l'esecuzione di operazioni numeriche avanzate, in particolare su array e strutture dati complesse;
- **Scikit-learn** : per l'implementazione degli algoritmi di classificazione e la valutazione delle prestazioni del modello tramite metriche dedicate;
- **Matplotlib** e **Seaborn** : utilizzate per la visualizzazione dei dati e dei risultati attraverso grafici chiari e informativi
- **Os** : per operazioni legate al sistema operativo, tra cui la gestione del colore dell'output testuale;
- **Warnings** : per la gestione e il filtraggio dei messaggi di avviso durante l'esecuzione del codice;
- **Pickle** : per il salvataggio e il caricamento dei modelli addestrati, rendendo possibile il riutilizzo senza dover ripetere la fase di training.

Capitolo 2) Dataset

Questo Dataset è stato scaricato online dal sito www.kaggle.com e appartiene originariamente al "National Institute of Diabetes and Digestive and Kidney Diseases". Al suo interno sono stati inseriti diversi vincoli alla selezione di questi casi da un database più ampio. In particolare, tutti i pazienti analizzati sono donne di almeno 21 anni. Le variabili predittive includono il numero di gravidanze avute dal paziente, il loro BMI (in italiano IMC, Indice di Massa Corporea, ossia un parametro che mette in relazione il peso di una persona con la sua altezza, fornendo una stima del grado di massa corporea che il soggetto dovrebbe avere), livello di insulina, età e così via.
Nel Dataset sono presenti otto features continue e una dicotomica, elenchiamole:

- **Pregnancies** : numero di gravidanze avute dalla paziente.
- **Glucose** : concentrazione di glucosio nel plasma, misurata due ore dopo l'ingestione di glucosio in un test da carico orale.
- **BloodPressure** : pressione arteriosa diastolica, espressa in *mmHg*.
- **SkinThickness** : spessore della plica cutanea del tricipite, espresso in millimetri.
- **Insulin** : livello di insulina nel sangue misurato due ore dopo la somministrazione orale di glucosio (*mu U/ml*).
- **BMI** : indice di massa corporea, calcolato come peso (kg) diviso altezza al quadrato (m^2).
- **DiabetesPedigreeFunction** : valore che stima la predisposizione genetica al diabete sulla base della storia familiare.
- **Age** : età del paziente.
- **Outcome** : variabile target che indica la presenza (1) o l'assenza (0) di diabete.

Approfondiamo nel dettaglio le variabili:

- **Glucose** : valori inferiori a 140 mg/dl sono considerati normali; valori compresi tra 140 e 199 mg/dl indicano una tolleranza ridotta al glucosio, mentre i valori superiori a 200 mg/dl sono associati a una diagnosi di diabete.
- **BloodPressure** : la pressione arteriosa diastolica rappresenta il valore della pressione nel momento di rilassamento del cuore. Secondo l'American Heart Association, una pressione diastolica inferiore a 89 mmHg e una sistolica inferiore a 120 mmHg sono considerati normali. Tuttavia, nei pazienti diabetici, anche la pressione troppo bassa può essere rischiosa.
- **SkinThickness** : le persone affette da diabete possono presentare alterazioni cutanee dovute a un metabolismo del glucosio compromesso. È stato osservato che tendono ad avere uno spessore della pelle maggiore rispetto ai soggetti sani.
- **Insulin** : l'insulina è un ormone fondamentale per la regolazione dei livelli di glucosio nel sangue, permettendo alle cellule di assorbirlo e utilizzarlo come fonte di energia. I valori normali di insulina, dopo l'assunzione di 75 g di glucosio, sono :
    - A digiuno : 5-25 µU/ml
    - Dopo 30 minuti : 41-125 µU/ml

    - Dopo 60 minuti : 20-120 µU/ml
    - Dopo 90 minuti : 20-90 µU/ml
    - Dopo 120 minuti : 18-56 µU/ml

- **BMI** : questo indice è utile per valutare la condizione ponderale di un individuo, ossia comprendere se si è in sottopeso, normopeso, sovrappeso o obeso. In termini di salute metabolica, i valori ideali di BMI si aggirano intorno ai 22,5 kg/m^2 per l'uomo e 21 kg/m^2 per la donna.
- **DiabetesPedigreeFunction** : è una funzione che quantifica la predisposizione ereditaria al diabete, in base alla presenza della patologia nei familiari. I valori possono variare da 0 a 1, dove valori più alti indicano una maggiore probabilità di ereditarietà.

Successivamente, si effettua un ragionamento basato su precisi step logici del *preprocessing* e *analisi dati*:

1. Pulizia dei dati (gestione dei valori nulli e anomali);
2. Trattamento dei dati mancanti;
3. Controllo e bilanciamento delle classi (*oversampling*);
4. Analisi esplorativa per identificare le variabili più informative.

Il Dataset viene innanzitutto analizzato e preprocessato per garantirne un utilizzo ottimale nelle fasi successive.
Il primo passo ha riguardato la verifica dell'eventuale presenza di valori nulli o anomali.
Nel Dataset sono effettivamente presenti valori nulli, che devono essere opportunamente gestiti per evitare errori o distorsioni durante l'addestramento del modello. In particolare, alcuni zeri presenti in colonne dove non hanno significato clinico (es. Glucose, BloodPressure, BMI, SkinThickness, Insulin) sono stati trattati come valori mancanti e sostituiti con valori più appropriati (es. media o mediana della colonna).
In seguito alla sostituzione di questi valori, il risultato del comando

```
print(df.isnull().sum()[1:6])
```

ha restituito il seguente output:

```
Values with zeroes.
Glucose             5
BloodPressure      35
SkinThickness     227
Insulin           374
BMI                11
dtype: int64
```

In seguito, i valori nulli vengono rimpiazzati con la media della rispettiva colonna. Riempire i valori mancanti con la media permette di mantenere inalterata la dimensione del Dataset e non ottenere valori nulli.
Questo è il risultato mostrante il numero di valori nulli dopo la modifica:

```
Filling null values...
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Il passaggio seguente è stato controllare la distribuzione della variabile target (Outcome), che indica se un paziente è affetto dalla malattia o no. Inizialmente c'era uno squilibrio, ossia 268 diabetici e 500 non diabetici.

Ecco i valori prima del bilanciamento:

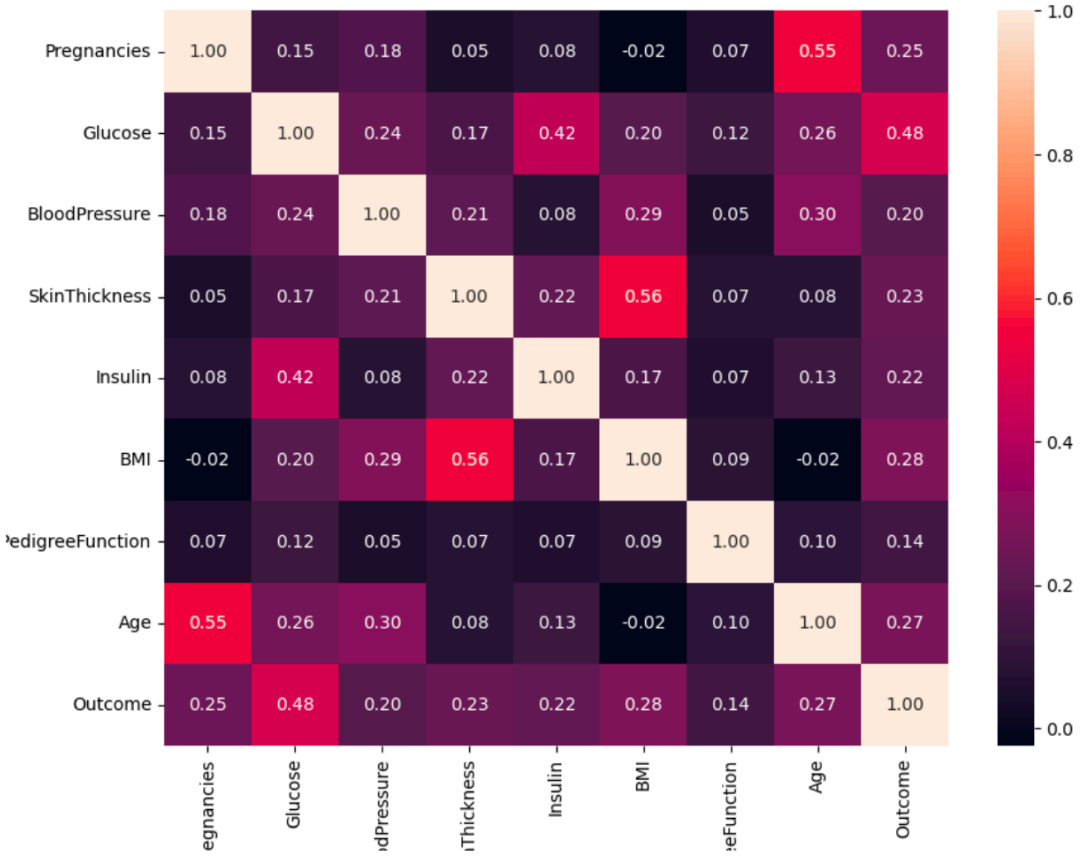
```
Controllo del bilanciamento delle classi
Non diabetici:  500 (% 65.10)
Diabetici:  268 (% 34.90)
```

Per risolverlo, si è applicato un **oversampling** della classe minoritaria, ossia un processo che consiste nel generare nuovi dati simili alla classe meno rappresentata, con lo scopo di colmare il divario tra le due classi e dunque portando entrambe le classi a 500 elementi.

Ecco i valori dopo il bilanciamento:

```
Valori dopo oversampling:
Non diabetici:  500 (% 50.00)
Diabetici:  500 (% 50.00)
```

Questo passaggio è fondamentale perché nei Dataset sbilanciati, i modelli tendono a "imparare" solo a riconoscere la classe più frequente, ottenendo un'altra accuratezza apparente ma scarsa capacità predittiva reale. Il **bilanciamento delle classi** garantisce che il modello impari in modo equo da entrambe.
Una volta aver bilanciato le classi, è stata creata una **heatmap** per visualizzare le correlazioni tra le varie feature del Dataset. Il fine è quello di comprendere quali caratteristiche sono più legate alla presenza del diabete.



Da qui si evince che le caratteristiche che possono far pensare ad una possibile causa di diabete sono principalmente i livelli del glucosio e dell'insulina; infatti, dalla rappresentazione grafica si può notare che *Glucose* è la feature più correlata con il diabete; perciò, sarà una delle variabili più utili per fare previsioni. Anche *BMI* e *Age* hanno una certa correlazione, evidenziando che peso e età sono importanti fattori di rischio.

Capitolo 3) Apprendimento Supervisionato

Prima di addestrare i modelli, è effettuata una **standardizzazione dei dati**, una tecnica fondamentale per migliorare le prestazioni degli algoritmi di machine learning. La standardizzazione trasforma ogni variabile in modo che abbia media 0 e deviazione standard 1, rendendo i dati confrontabili e prevenendo che variabili con scale differenti influenzino sproporzionatamente il modello. Questa operazione è particolarmente utile per algoritmi che si basano sulla distanza.
La standardizzazione consiste nel sottrarre la media (μ) da ogni valore (x) del vettore e dividere la differenza per la deviazione standard (σ) :

$$x = \frac{x - \mu}{\sigma}$$

X = vettore; μ = media; σ = deviazione standard.

Successivamente viene eseguita un'ottimizzazione degli iperparametri, fase essenziale per garantire il corretto funzionamento e l'efficacia dei modelli Machine Learning. La tecnica utilizzata è quella **Grid Search** , che esplora in modo esaustivo tutte le possibili combinazioni di iperparametri e seleziona quella che massimizza le prestazioni del modello (cioè che restituisce un punteggio di errore inferiore).
In **GridSearchCV**, insieme alla Grid Search, viene eseguita anche la cross-validation:

```
#Pipeline per il Decision Tree Classifier
pipe_dct = Pipeline([('scaler', StandardScaler()), ('dtc', DecisionTreeClassifier())])
param_grid = {'dtc__criterion': ['gini', 'entropy'],
              'dtc__max_depth': range(1, 100)}
opt_dct = GridSearchCV(estimator=pipe_dct, param_grid=param_grid, scoring='accuracy')
```

La cross-validation viene usata durante l'addestramento del modello. Quello più comunemente utilizzato è il K-Fold cross-validation, una tecnica che aumenta l'affidabilità della valutazione.
In pratica, i dati vengono divisi in *k* gruppi (*folds*). Il modello viene allenato *k* volte, ogni volta usando un fold come test e gli altri *k-1* per il training. Ad ogni iterazione verranno registrate le prestazioni del modello e alla fine darà la media di tutte. In particolare, viene usato il **Repeated K-Fold**, che ripete la procedura del K-Fold per ottenere una media delle metriche ancora più accurata., anche se richiede un tempo di esecuzione maggiore.
Come algoritmi di classificazione ho utilizzato i seguenti modelli:

- **Random Forest**
- **Decision Tree**
- **K-Nearest Neighbors (KNN)**
- **Logistic Regression**
- **Multilayer Perceptron (MLP)**
- **Gaussian Naive Bayes**

Mentre, per valutare le performance di ciascun classificatore, ho considerato le diverse metriche:

- **Accuracy** : percentuale di previsioni corrette.
- **Precision** : misura la precisione delle previsioni positive (quanti dei positivi predetti erano davvero positivi).

- **Recall** : indica la capacità del modello di rilevare correttamente i casi positivi (quanti dei positivi reali sono stati trovati).
- **F1-score** : fornisce un bilanciamento tra precision e recall(media armonica tra P e R), particolarmente utile in presenza di classi sbilanciate.

Analizziamo step-by-step ciascun algoritmo di classificazione.

Random Forest

L'algoritmo è un modello *ensemble,* cioè che al proprio interno mette insieme altri modelli più semplici.
È basato sull'addestramento di N alberi decisionali (decision tree), ognuno dei quali effettua una classificazione per ogni esempio. Quando tutti gli alberi (o più precisamente tutta la foresta) hanno classificato l'esempio, si effettua una conta su qual è stata la classe maggiormente stimata e la si assume come predizione della foresta.

	Non Ottimizzato	*Ottimizzato*
Accuracy	0.840	0.860
Precision	0.850	0.820
Recall	0.939	0.942
F1-score	0.892	0.875

Deviazione standard = 0.06

Decision Tree

Il funzionamento degli alberi di decisione prevede che il valore di una feature obiettivo venga classificato sulla base di una serie di regole di decisione basate sui dati di input a disposizione. Nello specifico, ogni nodo interno dell'albero indica una condizione e i valori derivati dagli esempi di input costituiscono dei sottoalberi. Le foglie dell'albero, invece, contengono il valore della feature obiettivo.

	Non Ottimizzato	*Ottimizzato*
Accuracy	0.835	0.805
Precision	0.847	0.760
Recall	0.869	0.903
F1-score	0.810	0.830

Deviazione standard = 0.05

K-Nearest Neighbors

Il funzionamento dell'algoritmo KNN si basa sulla semplice memorizzazione degli esempi del Dataset (comprendendo la/le feature obiettivo), senza che venga appreso un modello.
La classificazione avviene confrontando il nuovo esempio con un insieme formato da k vicini, che "voteranno" per decretare la classe di appartenenza del nuovo esempio. La votazione può avvenire come calcolo della moda, della media o a seguito dell'interpolazione dei k vicini.

	Non Ottimizzato	*Ottimizzato*
Accuracy	0.820	0.795
Precision	0.784	0.740
Recall	0.947	0.932
F1-score	0.858	0.825

Deviazione standard = 0.04

Logistic Regression

Il modello di classificazione della regressione logistica si basa su dei pesi di una funzione lineare appiattita dalle sigmoidee, minimizzando un errore su E. È importante specificare che parte come un modello di regressione, ma successivamente viene appiattito con la log loss.

	Non Ottimizzato	*Ottimizzato*
Accuracy	0.700	0.720
Precision	0.808	0.744
Recall	0.626	0.701
F1-score	0.705	0.722

Deviazione standard = 0.06

Multilayer Perceptron

Una **Rete Neurale** è un modello di apprendimento che si basa sul funzionamento dei neuroni cerebrali biologici. In questo caso viene utilizzata una Rete Neurale di tipo feed-forward che si basa su una gerarchia di funzioni lineari intervallate da funzioni di attivazione. In genere prende in input una serie di feature e le sottopone agli strati nascosti (detti feature non osservate) che le mappano secondo una funzione di attivazione e restituiscono in output una o più feature obiettivo. Formalmente è composta da tre strati o layer:

- Layer di input (in questo caso tutte le feature);
- Layer completo;
- Una funzione di attivazione f.

Per ottenere una classificazione accurata si fa uso della back-propagation che ricalcola e aggiorna i pesi.

	Non Ottimizzato	*Ottimizzato*
Accuracy	0.750	0.760
Precision	0.803	0.741
Recall	0.747	0.826
F1-score	0.774	0.781

Deviazione standard = 0.02

Gaussian Naive Bayes

È un algoritmo basato sul teorema di Bayes:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Laddove:

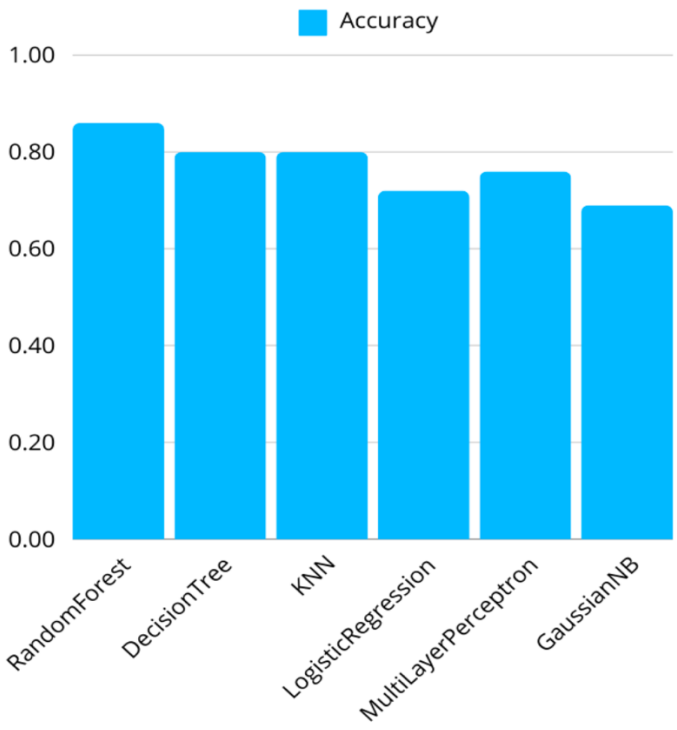
- **P(A|B)** è la probabilità condizionata dell'evento A dato l'evento B (ovvero la probabilità a posteriori della classe dati i predittori).
- **P(B|A)** è la probabilità condizionata di B dato l'evento A.
- **P(A)** è la probabilità a priori dell'evento A.
- **P(B)** è la probabilità a priori dell'evento B.

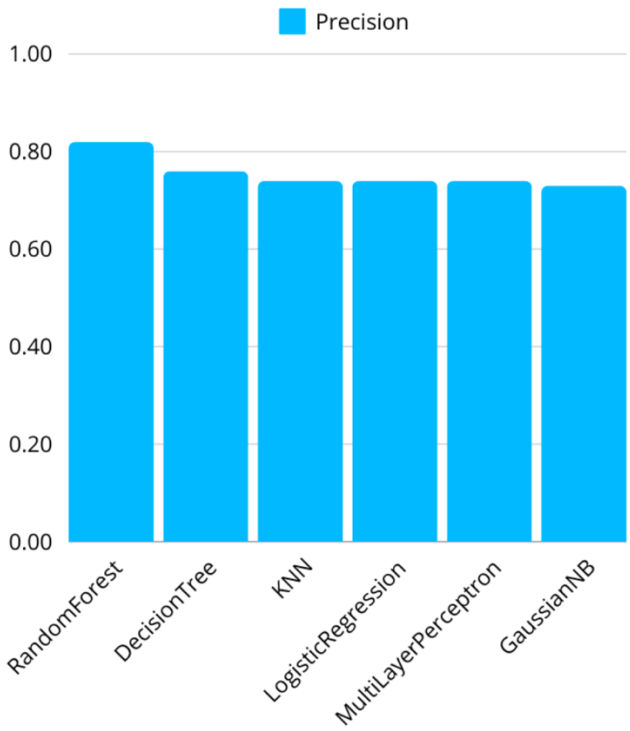
La versione "Gaussian" presuppone che i dati seguano una distribuzione normale gaussiana e che siano continui.

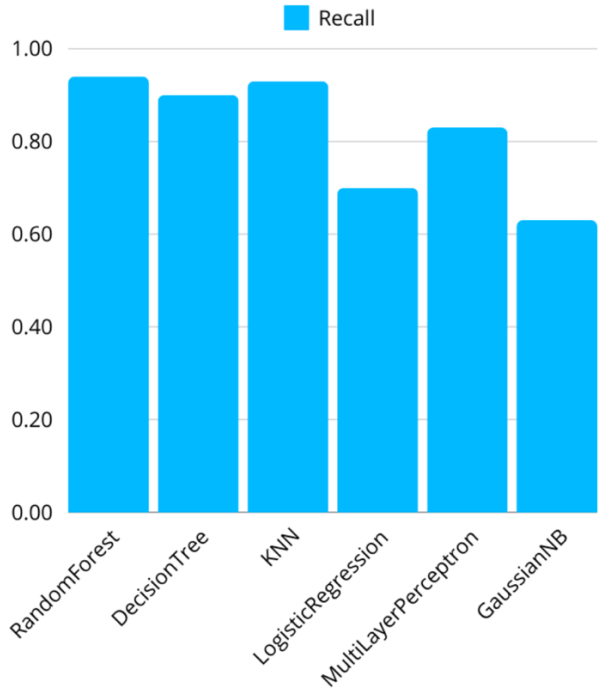
	Non Ottimizzato	*Ottimizzato*
Accuracy	0.690	0.685
Precision	0.791	0.725
Recall	0.626	0.634
F1-score	0.699	0.676

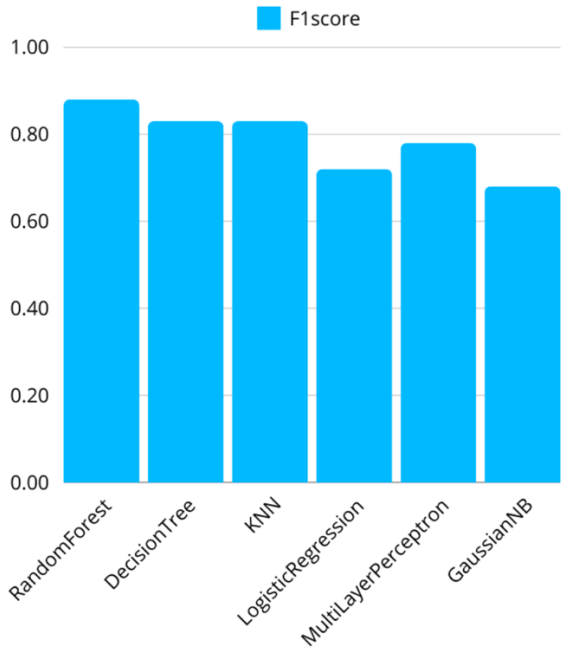
Deviazione standard = 0.07

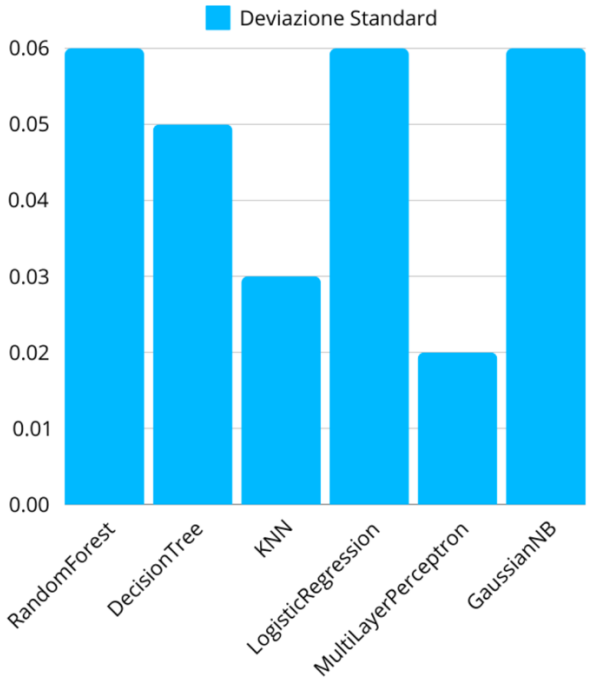
In seguito, vengono illustrati i seguenti grafici per ogni metrica.




Precision
1.00
0.80
0.60
0.40
0.20
0.00
RandomForest
DecisionTree
KNN
LogisticRegression
MultiLayerPerceptron
GaussianNB


Recall
1.00
0.80
0.60
0.40
0.20
0.00
RandomForest
DecisionTree
KNN
LogisticRegression
MultiLayerPerceptron
GaussianNB





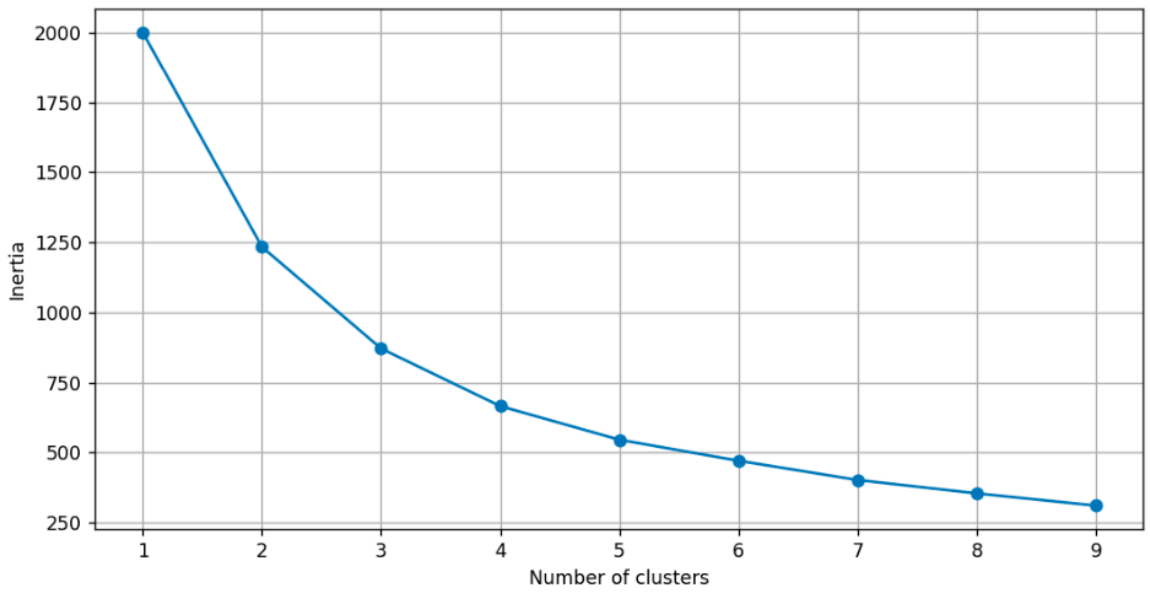
Dai dati analizzati, si evince che il Random Forest ha ottenuto in generale i risultati migliori, principalmente in termini di accuracy e F1-score.

Capitolo 4) Apprendimento Non Supervisionato

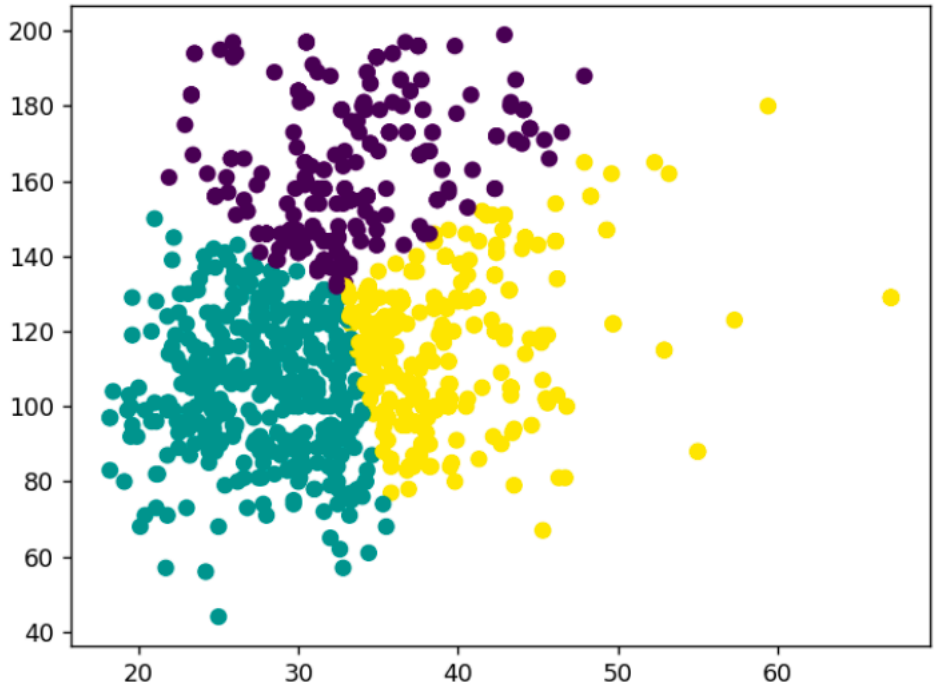
L'apprendimento non supervisionato è una tecnica di machine learning che analizza e modella i dati senza utilizzare etichette predefinite. A differenza dell'apprendimento supervisionato – in cui i dati di addestramento includono anche le risposte corrette – in questo caso l'obiettivo è scoprire strutture nascoste o pattern all'interno dei dati grezzi.

K-Means Clustering

Il K-Means è un algoritmo di clustering che suddivide i dati in gruppi (cluster) distinti non etichettati. Ogni punto può appartenere a un solo cluster. Il parametro **k** indica il numero di cluster desiderati : un valore di *k* maggiore comporta una suddivisione più fine e dettagliata dei dati.
Per questa analisi si è voluto rimuovere alcune feature dal dataset originale, in particole Pregnancies, BloodPressure, SkinThickness e Outcome.
Successivamente ho standardizzato i dati, riportando tutti. I valori in un intervallo tra -1 e 1. Questo passaggio è fondamentale per evitare che feature con scale diverse influenzino il calcolo delle distanze tra punti, criterio su cui si basa l'algoritmo.
Dopo la standardizzazione, è necessario selezionare il numero ottimale di cluster (k). L'algoritmo inizia scegliendo k centroidi iniziali casuali. Per ogni punto del Dataset si calcola la distanza da ciascun centroide e il punto viene assegnato al cluster più vicino.
Una volta assegnati tutti i punti, si ricalcolano i centroidi come media dei punti di ciascun cluster, e il processo si ripete finché i centroidi non si stabilizzano.
La scelta del valore k influisce in modo significativo sulla qualità del clustering. Per determinare il valore ottimale, si è scelto l'utilizzo dell'**Elbow Method**, che analizza la varianza intra-cluster in funzione di k. Il "gomito" del grafico indica il punto oltre il quale l'aumento di k produce miglioramenti trascurabili. Da questo metodo si è ottenuto questo risultato:



Dal grafico ottenuto, risulta che il numero ottimale di cluster è 3. Il modello finale è stato quindi costruito utilizzando principalmente le feature **Glucose** e **BMI,** che sono state suddivise in tre cluster distinti.



Ogni punto di questo *scatter plot* rappresenta un'osservazione del Dataset.
Il grafico rappresenta il risultato del clustering eseguito con il K-Means, utilizzando le variabili Glucosio (asse Y) e BMI (asse X). I dati sono stati suddivisi in tre cluster distinti, come suggerito dall'Elbow Method. Ogni colore indica un gruppo individuato in base alla somiglianza tra i valori delle due feature selezionate. Possiamo osservare come il clustering abbia identificato gruppi con caratteristiche simili, ad esempio un cluster con valori elevati di glucosio, uno con BMI più alti e uno con valori più moderati. Questo tipo di analisi permette di rilevare pattern nei dati anche in assenza di etichette, evidenziando possibili profili a rischio.

Capitolo 5) Modalità d'uso

Menù iniziale

All'avvio del programma, viene seguita automaticamente un'ottimizzazione preliminare del Dataset, che include l'eliminazione dei valori nulli e il bilanciamento delle classi:

```
Dataset caricato.
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000

Valori con zero.
Glucose            5
BloodPressure     35
SkinThickness    227
Insulin          374
BMI               11
dtype: int64
Filling null values...
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64


Controllo del bilanciamento delle classi
Non diabetici:  500 (% 65.10)
Diabetici:  268 (% 34.90)
```

Il Dataset caricato viene rappresentato con queste variabili e valori preliminari:

- *Count* => 768. Ci sono 768 righe per ogni feature.
- *Mean* => media aritmetica dei valori nella colonna di ogni feature.
- *Std* => Deviazione Standard, cioè quanto i valori sono "sparsi" rispetto alla media.
- *Min* => valore minimo trovato nella colonna (utile per individuare gli zeri).

- *25,50,75 %* => quartili.
- *Max* => valore massimo trovato nella colonna.

Completata questa fase, viene mostrato a schermo il menù principale dell'applicazione:

```
1 -- Apprendimento Supervisionato
2 -- Apprendimento non supervisionato
3 -- Ottimizza i parametri
4 -- Esegui Test
5 -- Predizione
6 -- Esci dal programma
scegli un numero per iniziare ad esplorare il programma:
```

Scelta 1

Inserendo 1 come opzione, il programma si occuperà dell'apprendimento supervisionato, allenando i classificatori precedentemente descritti. Per ogni modello esegue la Grid Search e la Repeated K-Fold Cross-Validation, restituendo le metriche di valutazione selezionate per ciascun classificatore.

```
                model  accuracy  precision    recall   f1score
0        DecisionTree  0.718954   0.659091  0.508772  0.574257
1        RandomForest  0.745098   0.704545  0.543860  0.613861
2                 KNN  0.699346   0.648649  0.421053  0.510638
3  LogisticRegression  0.732026   0.700000  0.491228  0.577320
4      MultiLayerPerc  0.732026   0.666667  0.561404  0.609524
5          GaussianNB  0.705882   0.630435  0.508772  0.563107

Calcolo la deviazione standard...

deviazione standard per il  DecisionTree: 0.03791660710945836

deviazione standard per il RandomForest: 0.03573776873607052

deviazione standard per il Knn: 0.039793663677317445

deviazione standard per il LogisticRegression: 0.046292039731866526

deviazione standard per il MultilayerPerceptron: 0.06355557334329197

deviazione standard per il GaussianNB: 0.06804880146097489
Models saved!
```

Scelta 2

Inserendo 2 come opzione, il programma si occuperà di addestrare l'algoritmo K-Means, in modo da suddividere sia i dati presenti nel Dataset sia quelli forniti successivamente in input nei rispettivi cluster, consentendo così da effettuare predizioni basate sulla loro appartenenza.

	Insulin_T	BMI_T	DiabetesPedigreeFunction_T	Age_T	labels
0	-3.345079e-16	0.166292	0.468492	1.425995	2
1	-3.345079e-16	-0.852531	-0.365061	-0.190672	1
2	-3.345079e-16	-1.332833	0.604397	-0.105584	2
3	-7.243887e-01	-0.634212	-0.920763	-1.041549	1
4	1.465506e-01	1.548980	5.484909	-0.020496	0
5	-3.345079e-16	-0.998077	-0.818079	-0.275760	1
6	-7.950054e-01	-0.212128	-0.676133	-0.616111	1
7	-3.345079e-16	0.413720	-1.020427	-0.360847	0
8	4.560094e+00	-0.284901	-0.947944	1.681259	2
9	-3.345079e-16	0.000000	-0.724455	1.766346	1
10	-3.345079e-16	0.748476	-0.848280	-0.275760	0
11	-3.345079e-16	0.806695	0.196681	0.064591	2
12	-3.345079e-16	-0.779758	2.926869	2.021610	1
13	8.126238e+00	-0.343120	-0.223115	2.191785	2
14	2.289367e-01	-0.968968	0.347687	1.511083	2
15	-3.345079e-16	-0.357674	0.036615	-0.105584	1
16	8.762565e-01	1.941955	0.238963	-0.190672	0
17	-3.345079e-16	-0.415893	-0.658012	-0.190672	1
18	-8.538527e-01	1.578089	-0.872441	-0.020496	0
19	-7.008498e-01	0.311838	0.172520	-0.105584	0

Ora ci sono le nuove colonne standardizzate (ossia trasformate in valore con media 0 e dev. Standard 1) delle colonne originali. Questo passaggio è importante per l'algoritmo K-Means perché si basa su distanze: se le scale sono diverse, influenzerebbero troppo il risultato. Inoltre, viene aggiunta la colonna **labels** che rappresenta a quale cluster appartiene ogni paziente.

Scelta 3

Inserendo 3 come opzione, attraverso la Grid Search verranno restituiti i parametri migliori per gli algoritmi di apprendimento supervisionato.

```
scegli un numero per iniziare ad esplorare il programma:
 3
 - Esecuzione del decision tree classifier con grid search
        Calcolo degli iperparametri ottimali...
- DTC Best Params: {'dtc__criterion': 'entropy', 'dtc__max_depth': 2}
 - Esegui RFC con grid View
        Calcolo degli iperparametri ottimali
- RFC Best Params: {'rfc__bootstrap': True, 'rfc__criterion': 'gini', 'rfc__max_depth': 6, 'rfc__max_features': 'log2', 'rfc__n_estimators': 10}
 - Esegui KNC con grid view
        Calcolo degli iperparametri ottimali ...
- KNN Best Params: {'knn__algorithm': 'kd_tree', 'knn__n_neighbors': 30, 'knn__p': 2, 'knn__weights': 'distance'}
 - Execute LR with Grid View
        Calcolo degli iperparametri ottimali
- LR Best Params: {'logr__C': 0.1, 'logr__penalty': 'l2'}
 - Esegui MLP con grid view
        Calcolo degli iperparametri ottimali
- MLP Best Params: {'mlp__activation': 'tanh', 'mlp__hidden_layer_sizes': (9,), 'mlp__solver': 'adam'}
 - Esegui NB con Grid View
        Calcolo degli iperparametri ottimali
- NB Best Params: {'nb__var_smoothing': 0.23101297000831597}
End of optimization!
```

Scelta 4

Inserendo 4 come opzione, vengono caricati i classificatori precedentemente addestrati durante la fase di training. Per ciascuna di essi viene eseguito un predict_proba su un input di test, che per semplicità viene caricato automaticamente. I test utilizzati corrispondono a due casi reali presenti nel dataset: uno positivo (caso di diabete) e uno negativo (assenza di diabete). Dopo aver ottenuto le predizioni da tutti i modelli, viene calcolata la media delle probabilità fornite da ciascun classificatore, e viene restituita la percentuale più alta tra le due classi.

```
TEST su un paziente non diabetico:  [1, 89, 66, 23, 94, 28.1, 0.167, 21]

predizione KMeans Cluster:  [2]
questo paziente non ha il diabete .
-Probabilità:  98.36 %

TEST su un paziente diabetico:  [2, 197, 70, 45, 543, 30.5, 0.158, 53]

predizione KMeans Cluster:  [0]
questo paziente ha il diabete .
-Probabilità:  97.11 %
```

Scelta 5

Inserendo 5 come opzione, il programma permette di effettuare una predizione inserendo valori da parte dell'utente. Alcuni valori sono opzionali : l'utente può digitare -1 se non conosce il dato. Le uniche caratteristiche obbligatorie sono : Glucose, Insuline e Age.

```
scegli un numero per iniziare ad esplorare il programma:
 5
Per favore inserisci i tuoi valori.
se un parametro è rosso è obbligatorio!
Inserisci il numero delle tue gravidanze:
5
Inserisci il tuo livello di glucosio nel sangue:
(>70 mg/dl)500
se non conosci la risposta inserisci '-1'.
Inserisci la tua pressione sanguigna diastolica:
(>40 mmHg)56
se non conosci la risposta premi '-1'.
Inserisci lo spessore della tua pelle:
(>15 mm)66
Inserisci il tuo livello di insulina:
(>16 µU/ml)34
se non conosci la risposta premi '-1'.
Inserisci la tua BMI:
(>19 kg/m²)34
se non conosci la risposta premi '-1'.
Inserisci la tua funzione di pedigree del diabete
(0<x<100 %)34
Inserisci la tua età:
55

-- modelli caricati.

valori del paziente:     [[5, 500.0, 56, 66.0, 34.0, 34.0, 34, 55]]
predizione KMeans Cluster:  [2]
questo paziente ha il diabete .
-Probability:  51.67 %
```

Scelta 6
Inserendo 6 come opzione, il programma terminerà.

Capitolo 6) Conclusioni e sviluppi futuri

Questo progetto ha permesso di sviluppare un modello predittivo efficace che, grazie all'inserimento di alcuni parametri essenziali, può supportare l'identificazione precoce del rischio diabetico.
In futuro, il progetto potrebbe essere arricchito da un'interfaccia grafica intuitiva che guidi l'utente nell'inserimento dei dati e nella comprensione del risultato. Inoltre, si potrebbero integrare funzionalità di monitoraggio nel tempo, permettendo all'utente di salvare i risultati e visualizzare l'evoluzione del proprio stato di salute.

Riferimenti bibliografici

Descrizione del Dataset:

https://www.kaggle.com/code/paultimothymooney/predict-diabetes-from-medical-records

Materiale scientifico informativo sul diabete:

https://www.santagostino.it/magazine/diabete/#cose-il-diabete

https://www.ospedalebambinogesu.it/diabete-gestazionale-111885/