# Diabetics Classifier
## Documentazione Progetto

AA 2022-2023

**Progetto di**

Aldo Mangione, 742287, a.mangione12@studenti.uniba.it

**Repository GitHub:**

# INDICE

## Introduzione

Ricavare informazioni sulle possibili patologie di un paziente, a partire dalla sua cartella clinica è molto utile, specie se consideriamo che con il machine learning questo è reso particolarmente veloce e poco dispendioso. Un insieme di modelli in grado di identificare diverse patologie è dunque uno strumento di fondamentale importanza per la medicina. Con questo progetto lavorerò su modelli che si occuperanno nello specifico di rilevare il diabete, a partire dallo storico del paziente. Il progetto è stato anche utilizzato per studiare quale tipo di modello si adatta meglio al problema, verranno infatti raccolti dati sul testing e verranno confrontate le metriche che descrivono la performance dei classificatori. In aggiunta si proverà ad utilizzare come approccio anche il clustering, attraverso l'algoritmo K-Means.

## Sommario

Il compito di valutare la possibile presenza del diabete è chiaramente un compito di classificazione su target feature binaria. Per questo scopo sono stati utilizzati quattro modelli diversi: Gaussian Naive Bayes, Knn, Random Forest Classifier e un Neural network, che sono stati confrontati in base a diverse metriche (precisione, F1 score, accuratezza). Per poter addestrare questi modelli sono state necessarie alcune operazioni di preprocessing. Come già detto si è provato anche un approccio alternativo attraverso un algoritmo di clustering, chiamato K-Means. La precisione di quest'ultimo verrà valutata attraverso il punteggio silhouette, spiegato più avanti.

## Elenco argomenti di interesse

- Apprendimento Supervisionato: K Nearest neighbors, Random Forest Classifier
- Apprendimento Non supervisionato: K Means (Hard clustering)
- Multilayer Perceptron Classifier (Neural network)
- Apprendimento Probabilistico: Gaussian Naive Bayes

## Risorse strumentali

Il progetto è stato realizzato con il linguaggio Python in Visual Studio Code. Le librerie utilizzate sono state le seguenti:

- sklearn -- per gli algoritmi di apprendimento e la loro valutazione;
- pandas e numpy -- per la manipolazione dei dati;
- matplot e seaborn -- per la rappresentazione grafica dei dati;

# Informazioni dataset e Preprocessing

```
o-------------------------------------------o
| Progetto ICON 22-23: Diabetics classifier |
o-------------------------------------------o


Prime 5 righe del dataset:
    gender   age  hypertension  heart_disease smoking_history    bmi  HbA1c_level  blood_glucose_level  diabetes
0  Female  80.0             0              1           never  25.19          6.6                  140         0
1  Female  54.0             0              0         No Info  27.32          6.6                   80         0
2    Male  28.0             0              0           never  27.32          5.7                  158         0
3  Female  36.0             0              0         current  23.45          5.0                  155         0
4    Male  76.0             1              1         current  20.14          4.8                  155         0


Dimensioni del dataset:
 (100000, 9)


Colonne del dataset:
 Index(['gender', 'age', 'hypertension', 'heart_disease', 'smoking_history',
       'bmi', 'HbA1c_level', 'blood_glucose_level', 'diabetes'],
      dtype='object')
```

Il dataset prevede 100'000 record descritti dai seguenti campi: sesso, età, ipertensione, storia da fumatore, bmi, problemi al cuore, livello HbA1c, livello glucosio nel sangue, e diabete. Innanzitutto, son partito assicurandomi che non vi fossero campi nulli, e fortunatamente non ce ne sono.

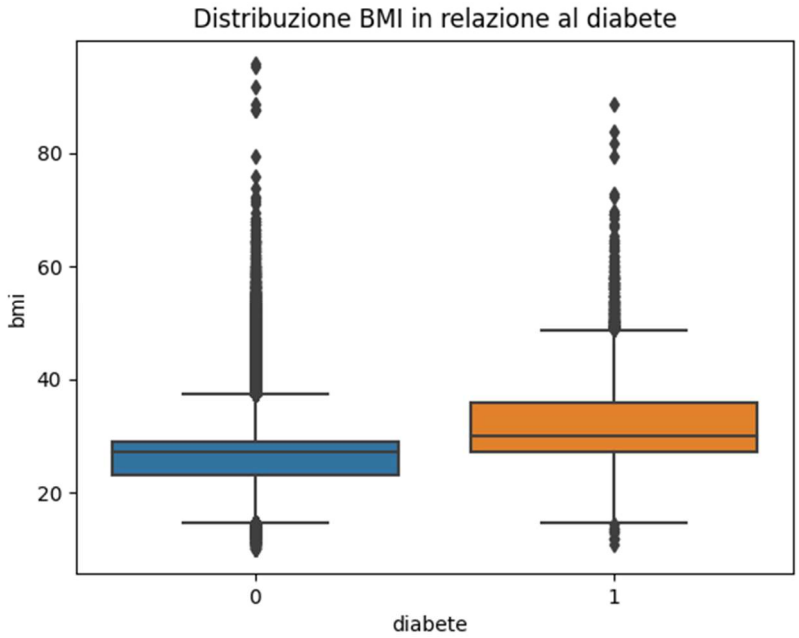
```
Statistiche dataset:
        gender            age  hypertension  heart_disease smoking_history            bmi    HbA1c_level  blood_glucose_level       diabetes
count   100000  100000.000000  100000.00000  100000.000000          100000  100000.000000  100000.000000        100000.000000  100000.000000
unique       3            NaN           NaN            NaN               6            NaN           NaN                  NaN           NaN
top     Female            NaN           NaN            NaN        No Info            NaN           NaN                  NaN           NaN
freq     58552            NaN           NaN            NaN           35816            NaN           NaN                  NaN           NaN
mean       NaN      41.885856       0.07485       0.039420             NaN      27.320767      5.527507           138.058060       0.085000
std        NaN      22.516840       0.26315       0.194593             NaN       6.636783      1.070672            40.708136       0.278883
min        NaN       0.080000       0.00000       0.000000             NaN      10.010000      3.500000            80.000000       0.000000
25%        NaN      24.000000       0.00000       0.000000             NaN      23.630000      4.800000           100.000000       0.000000
50%        NaN      43.000000       0.00000       0.000000             NaN      27.320000      5.800000           140.000000       0.000000
75%        NaN      60.000000       0.00000       0.000000             NaN      29.580000      6.200000           159.000000       0.000000
max        NaN      80.000000       1.00000       1.000000             NaN      95.690000      9.000000           300.000000       1.000000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100000 non-null  object
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64
 3   heart_disease        100000 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64
 8   diabetes             100000 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB


Altro:
 None
```

Ho poi dovuto standardizzare alcuni campi, infatti ‘smoking_history’ è descritta da cinque possibili testi diversi (‘never’, ‘ever’, ‘former’, ‘No info’, ‘current’), e gender è compilato in forma testuale. Detto questo, ho applicato le seguenti modifiche: ho aggiunto due campi al dataset, ‘smoker_bool’, che ci dice se l’individuo ha mai fumato, e ‘male_bool’ che ci dice in forma numerica se l’individuo è maschio o no. ‘smoker_bool’ è stato settato a 1 per tutti quegli individui che presentavano ‘former’ o ‘current’ in ‘smoking_history’.

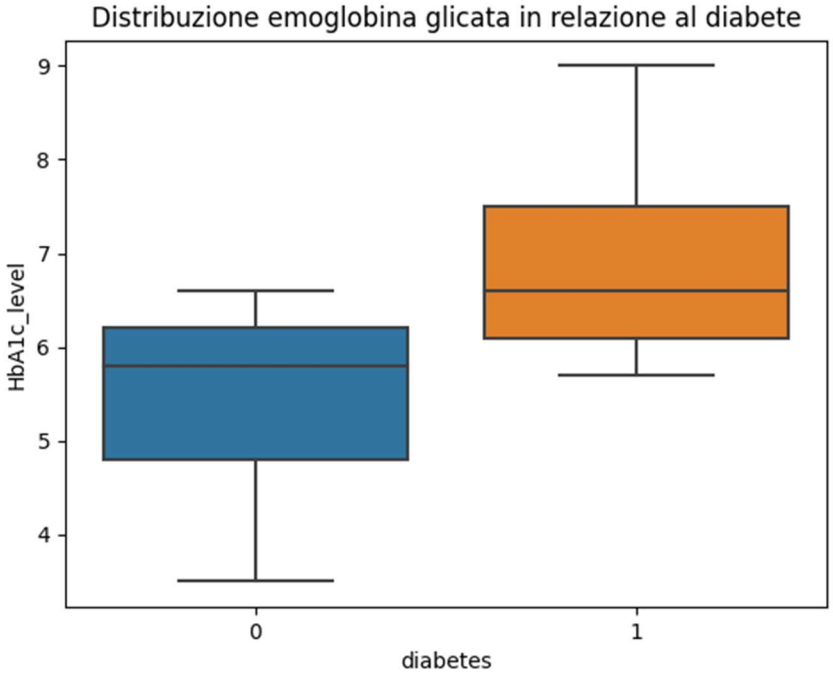
## Analisi dei dati

Ho efftuato delle osservazioni sui dati e le statistiche che mi permettessero di valutare la correlazione dei dati e quindi effettuare delle scelte più appropriate per l'addestramento dei modelli.

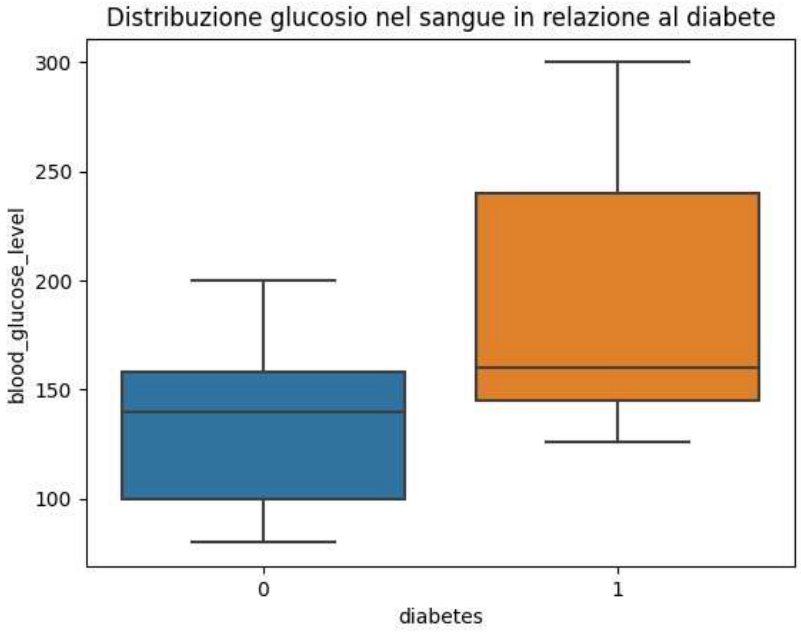

La prima cosa che ho notato è che il dataset è sproporzionato dal punto di vista del conteggio di diabetici. A causa di questo, è stato di fondamentale importanza assicurarsi di effettuare una giusta distribuzione dei casi diabetici nei casi per il training, e i casi per il test. Ciò è stato possibile attraverso il parametro 'stratify' in train_test_split().




C'è una chiara correlazione tra bmi alto e presenza della patologia. In particolare, la maggior parte dei diabetici ha un bmi vicino o superiore al 30, che è la soglia che indica l'obesità.




Anche la presenza di emoglobina glicata è fortemente correlata ai casi di diabete. Si nota facilmente infatti che dopo la quantità superiore alle sei unità inizia ad essere presente molto più frequentemente il diabete.




Infine, un altro indicatore molto forte del diabete è il glucosio nel sangue. Notiamo infatti che avvicinandoci a 150 iniziano ad esserci la maggior parte dei casi di diabete.

# Apprendimento supervisionato

L'apprendimento supervisionato è un tipo di approccio nell'ambito del machine learning in cui un algoritmo impara a fare previsioni basandosi su dati per l'addestramento che sono coppie di input e output corrispondenti. L'obiettivo dell'apprendimento supervisionato è quello di costruire un modello che possa generalizzare da questi esempi di addestramento, in modo da essere in grado di fare previsioni accurate su nuovi dati che non sono stati visti durante la fase di addestramento.

## K Nearest Neighbors

L'algoritmo "K nearest neighbors" (abbreviato in knn) è un algoritmo di machine learning che si basa sull'apprendimento supervisionato. L'idea alla base di questo algoritmo è che entità simili sono vicine tra loro nello spazio delle descrizioni. Quando bisogna classificare un nuovo dato, si utilizzano i k "vicini meno distanti" e le loro categorie di appartenenza per stabilire come debba essere classificato il nuovo dato.

### Strumenti utilizzati

Per l'implementazione del modello K Nearest Neighbors è stata utilizzata la libreria Scikit learn usando la classe KNeighborsClassifier.

### Decisioni di Progetto

Per costruire il modello di KNN migliore si è deciso di sperimentare con

1. Il numero di vicini
2. Algoritmi usati per computare il vicino più simile