

Esame Analisi e Gestione Dati per la Biomedica

Analisi dati per previsione di malattie cardiovascolari

Panoramica problema

Le malattie cardiovascolari rappresentano la principale causa di morte a livello globale, solo in Italia si contano circa 200 mila decessi l'anno (dati Istat/Iss).

Tra i principali fattori di rischio determinanti delle malattie cardiovascolari troviamo il fumo, i livelli elevati di colesterolo nel sangue e uno stile di vita sedentario.

Si dimostra quindi essenziale velocizzare i processi diagnostici supportando i medici con strumenti tecnologici avanzati. I modelli di apprendimento automatico, tra cui le reti Deep Learning, sono sicuramente uno strumento valido per la diagnosi di malattie cardiovascolari.

Uno strumento sicuramente valido per lo scopo sono i modelli di apprendimento automatico, tra cui le reti neurali profonde (DL).

L'efficacia di tali modelli dipende in larga misura dalla qualità e dalla rappresentatività dei dati utilizzati per l'addestramento. Più il dataset di input raccoglie un numero ampio e diversificato di caratteristiche e parametri clinici relativi ai soggetti, maggiore sarà la capacità del modello di apprendere pattern predittivi affidabili.

Parti del problema

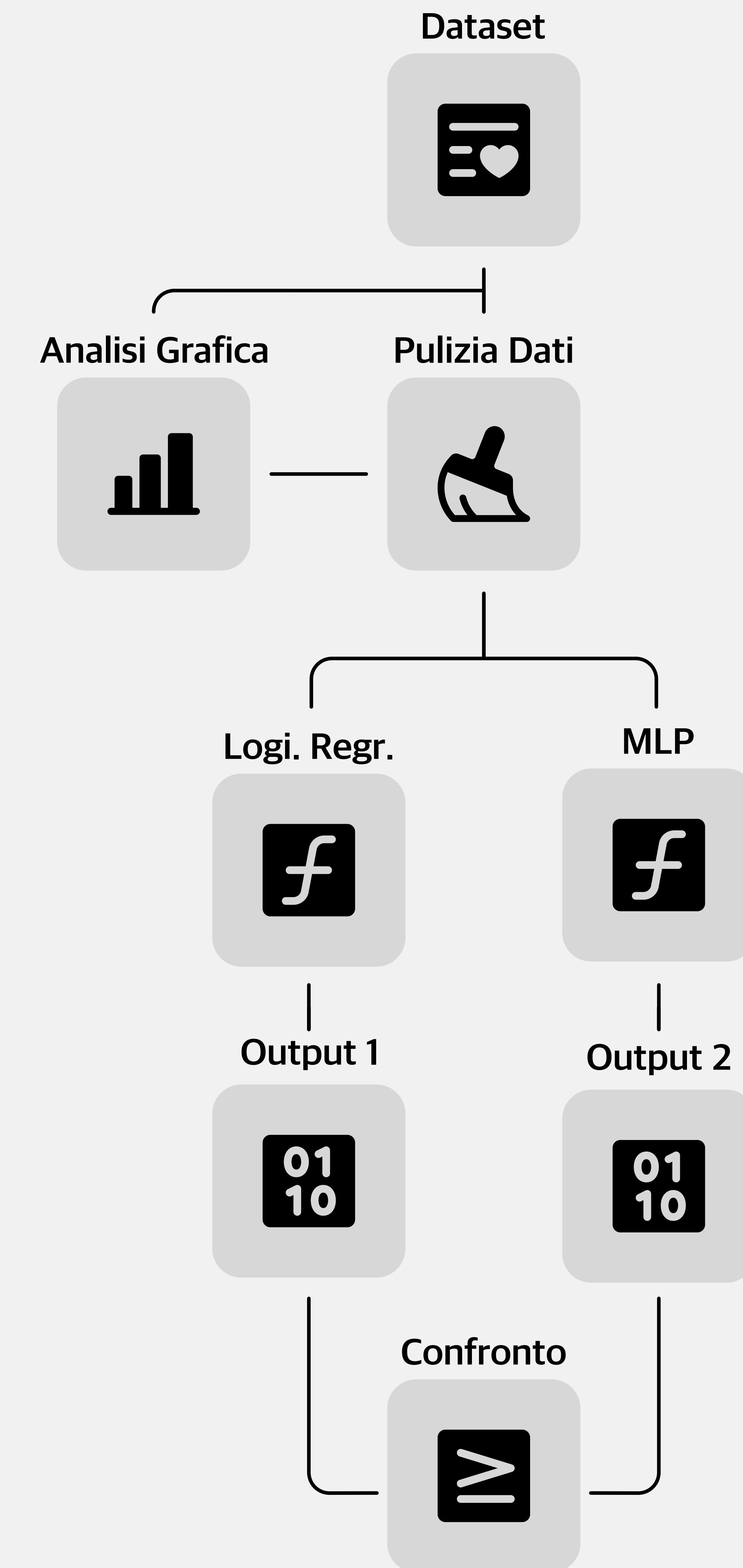
Analisi dei dati e Previsione

Analisi dei dati

- Descrizione delle caratteristiche del dataset
- Valutazione grafica dei dati
- Descrizione statistica
- Pulizia dei dati

Previsione

- Preparazione dei dati in ingresso ai modelli.
- Scelta ed implementazione dei modelli di previsione.
- Fase di addestramento
- Fase di valutazione
- Confronto risultati ottenuti dai modelli



Strumento per la risoluzione

Regressione Logistica e MLP

Il dataset è stato recuperato da Kaggle (<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>) e il progetto è stato sviluppato in python con pytorch, pandas e matplotlib.

Per risolvere un problema di classificazione binaria, ovvero prevedere la presenza o assenza di una malattia cardiovascolare, si è scelto di utilizzare due modelli diversi per poi confrontarne le prestazioni.

A tal fine sono stati scelti i seguenti modelli: il primo, di tipo lineare, è un modello di regressione logisitica mentre il secondo è una configurazione di MLP Multi Layer Perceptron ad un singolo strato nascosto e funzione di attivazione non lineare ReLU. In entrambi i casi, l'output finale è una stima della probabilità di presenza della malattia cardiovascolare.

La differenza principale risiede nel fatto che la regressione logistica è un modello lineare, mentre la MLP con ReLU è un modello non lineare, potenzialmente in grado di apprendere relazioni più complesse tra le caratteristiche e la variabile target.

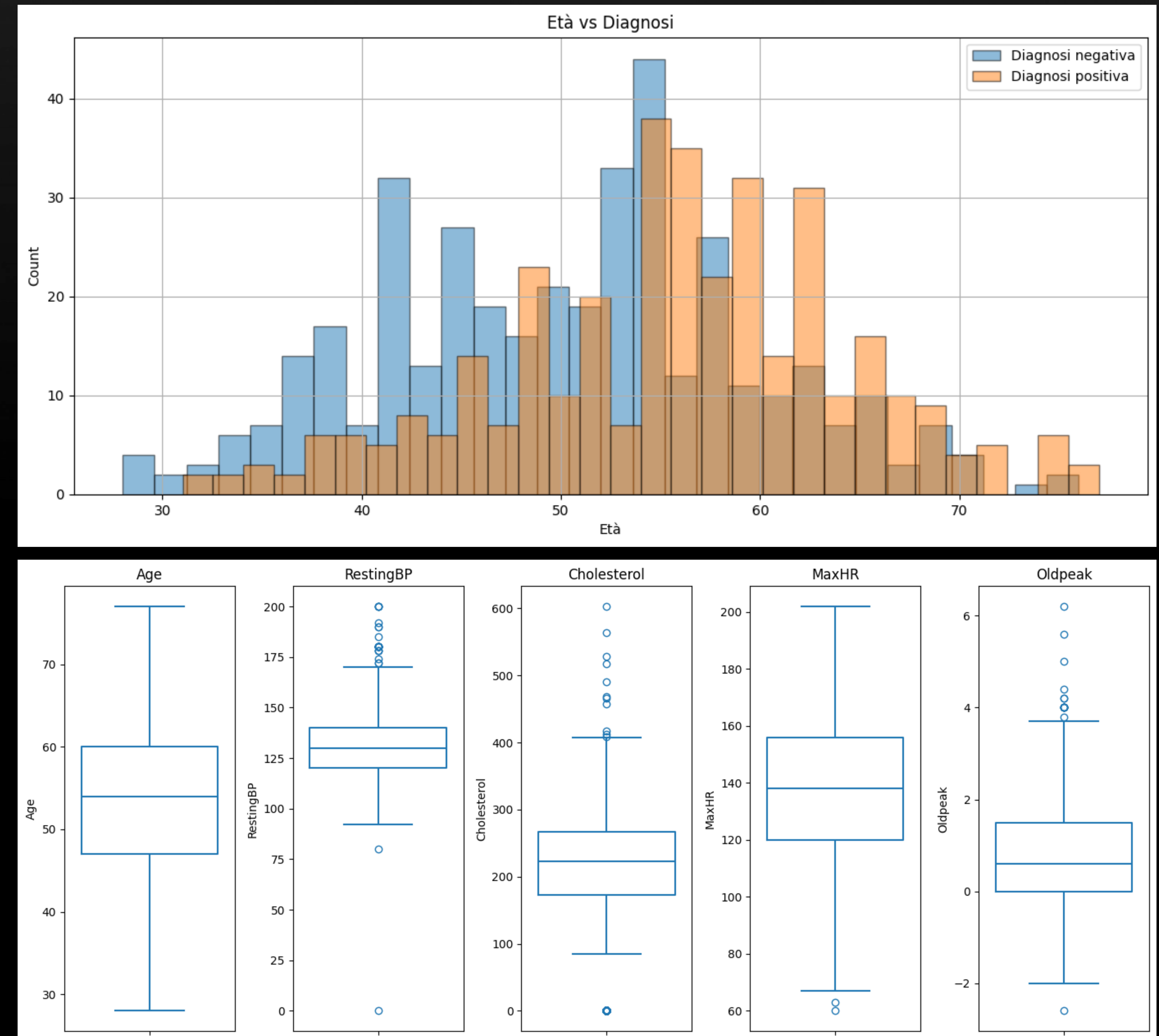
Analisi dei Dati

Distribuzione e Outliers

Una prima descrizione sintetica dei dati fornisce informazioni utili sul dataset come l'età media delle osservazioni di 55 anni e una deviazione standard di circa 10 anni.

Graficare i dati risalta relazioni importanti. L'istogramma in alto evidenzia, come da aspettativa, che sussiste un'aumento dei casi di cvd in età avanzata.

È fondamentale identificare e gestire gli outliers, poiché possono influenzare negativamente i modelli statistici. Il grafico sottostante mi permette di osservare la presenza di misurazioni errate e rimuoverle dal dataset.



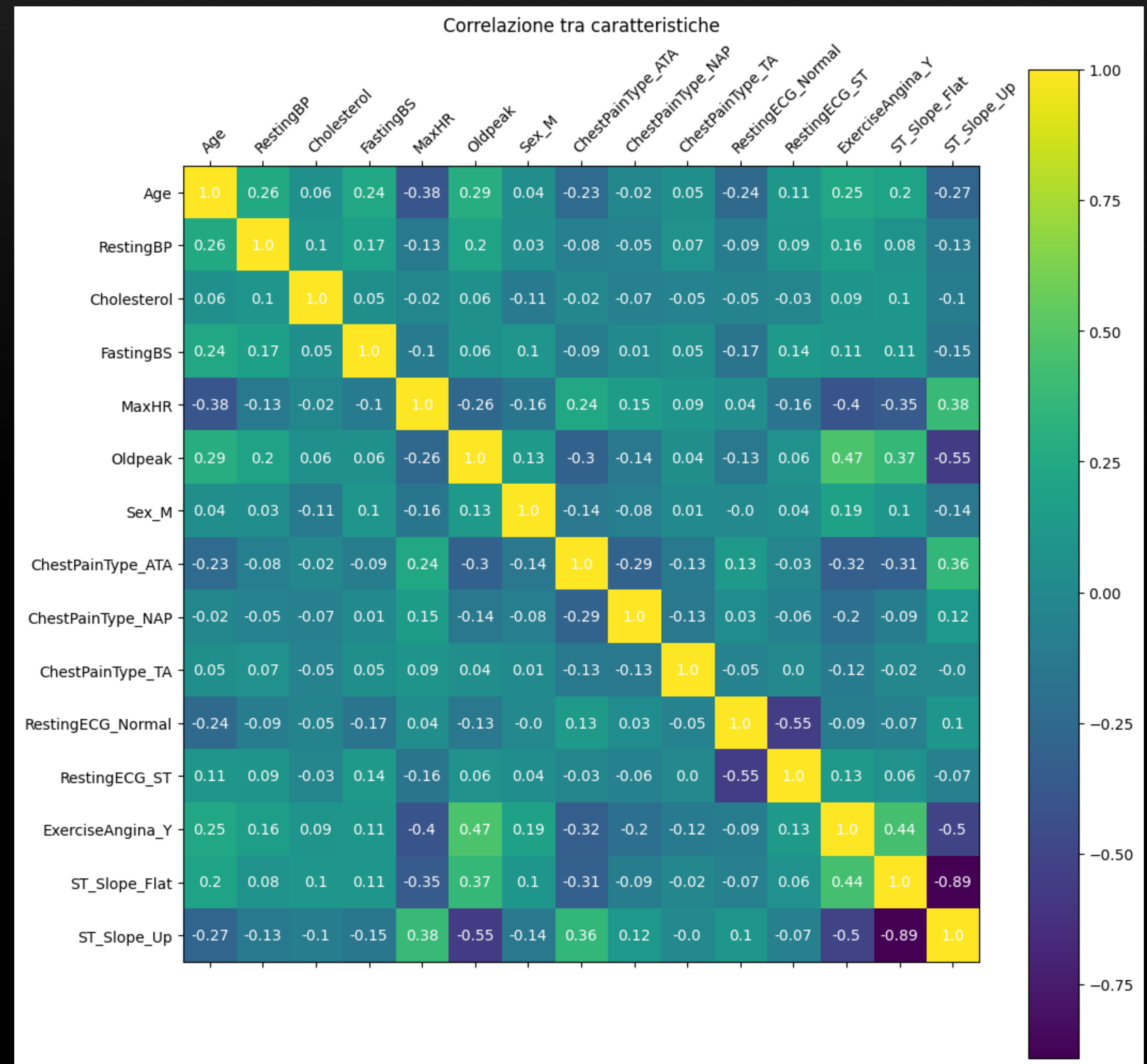
Analisi dei Dati

Correlazione

Il dataset scelto si compone di 11 caratteristiche/colonne e circa un migliaio di righe. Dopo aver codificato le colonne categoriche con la tecnica dell'hot-encoding, il numero di features passerà a 15.

Se delle variabili sono fortemente correlate vuol dire che danno la stessa informazione e il modello di regressione non riesce più ad attribuire un significato a ciascuna di esse.

Visualizzare la matrice di correlazione ci permette di notare che in questo caso c'è una forte correlazione tra le colonne ST_Slope_Flat e ST_Slope_Up.



Implementazione

I modelli LR e MLP

Regressione Logistica Binaria

```
1 class LogisticRegressionNet(nn.Module):
2     """
3     Regressione Logistica Binaria
4     Un modello lineare che, nonostante la sua semplicità,
5     offre spesso prestazioni soddisfacenti in problemi di classificazione binaria.
6     """
7
8     def __init__(self, input_size=input_data.shape[1]):
9         super(LogisticRegressionNet, self).__init__()
10        self.fc = nn.Linear(input_size, 1)
11        self.sigmoid = nn.Sigmoid()
12
13    def forward(self, x):
14        x = self.fc(x)
15        x = self.sigmoid(x)
16        return x
```

MLP

```
1 class MLPNet(nn.Module):
2     """
3     Rete Neurale Multistrato (MLP)
4     Questo modello è grado di apprendere rappresentazioni non lineari dai dati.
5     La seguente implementazione si compone di un solo strato nascosto,
6     con attivazione ReLU.
7     """
8
9     def __init__(self, input_size=input_data.shape[1], hidden_size=64):
10        super(MLPNet, self).__init__()
11        self.relu_stack = nn.Sequential(
12            nn.Linear(input_size, hidden_size),
13            nn.ReLU(),
14            nn.Linear(hidden_size, 1),
15        )
16        self.sigmoid = nn.Sigmoid()
17
18    def forward(self, x):
19        x = self.relu_stack(x)
20        x = self.sigmoid(x)
21        return x
```

Entrambi i modelli terminano con uno strato di attivazione sigmoidale e sono stati addestrati utilizzando l'ottimizzazione Adam con la funzione di perdita BCE (Binary Cross Entropy).

Conclusioni

Tenendo conto della ridotta dimensione del dataset analizzato, entrambi i modelli riescono ad ottenere risultati interessanti.

Massimizzare la percentuale di casi veri-positivi e minimizzare quella di falsi-positivi attraverso la distanza euclidea ci permette di calcolare i valori di soglia ottimali per entrambi i modelli.

Inoltre combinando questo strumento con le attuali soluzioni di modelli linguistici di grandi dimensioni (LLM) sarebbe possibile automatizzare la fase di screening lasciando che sia il modello a raccogliere alcuni dati essenziali per la diagnosi (e.g. tipo di dolore toracico, età).

