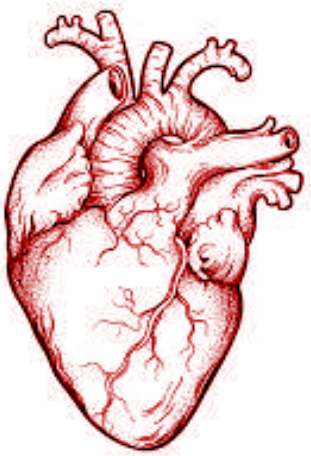


Heart Disease Classification

Addestramento di un modello di predizione con scikit-learn

Roberto Zanolli

Università di Bologna
Informatica per il management - L31
Corso di Statistica Numerica

Indice

1. Introduzione e scelta del dataset

Le malattie cardiovascolari (CVDs) rappresentano la principale causa di morte a livello mondiale. Tra le CVDs possiamo citare cardiopatie coronariche, malattie cerebrovascolari (come l'ictus), miocarditi e difetti congeniti.
Secondo l'Organizzazione Mondiale della Sanità, ogni anno muoiono circa 20 milioni[1] di persone a causa di queste malattie. Infarti e ictus costituiscono più di quattro quinti dei decessi per CVD, e un terzo di questi decessi avviene prima dei 70 anni di età.
Lo scopo principale di questo progetto è raccogliere le caratteristiche degli attacchi cardiaci o dei fattori che vi contribuiscono da un dataset e costruire un modello di classificazione in grado di prevedere situazioni di rischio.

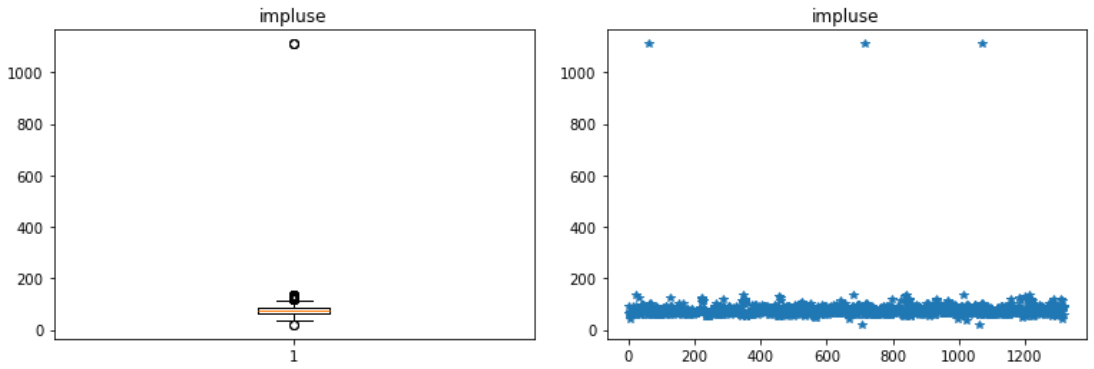
Il dataset utilizzato comprende 1319 rilevazioni, ciascuno dei quali presenta nove campi: otto di input e uno di output.
I campi di input contengono: età, genere (0=F, 1=M), frequenza cardiaca, pressione sistolica, pressione diastolica, glicemia, CK-MB (creatinchinasi MB, un enzima cardiaco) e Test-Troponina (un marker cardiaco). Questi campi sono identificati rispettivamente nel csv originale del dataset con i nomi: "*age*" (int), "*gender*" (int), "*impluse*" (int), "*pressurehight*" (int), "*pressurelow*" (int), "*glucose*" (float), "*kcm*"(float) e "*troponin*"(float).
Il campo di output indica la presenza di un attacco cardiaco, e puó assumere due valori: negativo (assenza di attacco cardiaco) e positivo (presenza di attacco cardiaco). Quest'ultimo appare nel file originale con l'identificativo "*class*".

Il dataset utilizzato è disponibile su kaggle seguendo il link: https://www.kaggle.com/datasets/bharath011/heart-disease-classification-dataset/data.

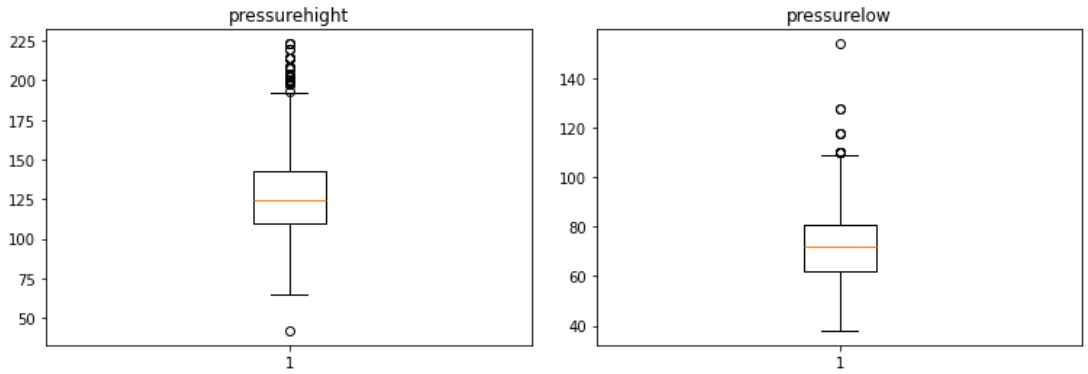
[1] fonte: www.ansa.it

2. Pre-Processing

In questa prima fase è stato innanzitutto caricato il file del nostro dataset "HeartAttack.csv" in un dataframe a cui è stato assegnato il nome "data" e subito dopo sono state stampate le informazioni di base dello stesso per confermare la correttezza di quelle presenti su Kaggle.
Da questa stampa è stato possibile avere una conferma sui dtype dei campi ed è stato possibile osservare l'assenza di campi non assegnati in quanto **nessuna delle 1319 righe presenta valori null**.
In seguito sono stati stampati per ogni campo di input (memorizzati in una lista dal nome "features") sia un boxplot che uno scatterplot per osservare possibili simmetrie ed **outliers** particolarmente evidenti (anche considerando valori di riferimento).
In particolare è stato osservato che alcuni valori rilevati per la frequenza cardiaca erano **frutto di un errore di misurazione** in quanto addirittura superiori a 1000 bpm.



Inoltre a causa di valori insolitamente bassi tra le rilevazioni per la pressione sistolica(massima) e insolitamente alti tra quelle per la diastolica(minima) si è ipotizzato che in alcune righe fossero state invertite in fase di inserimento le misurazioni per questi campi.



Dopo queste osservazioni sono state rimosse le rilevazioni contenenti valori fuori quota, eliminando dal dataset le righe con valori per il campo frequenza cardiaca superiori a 250 e quelle dove il valore della pressione minima superava quello della pressione massima.
Infine sono stati stampati per ogni feature un box plot e un istogramma per osservare come la rimozione degli outliers abbia influenzato le distribuzioni.

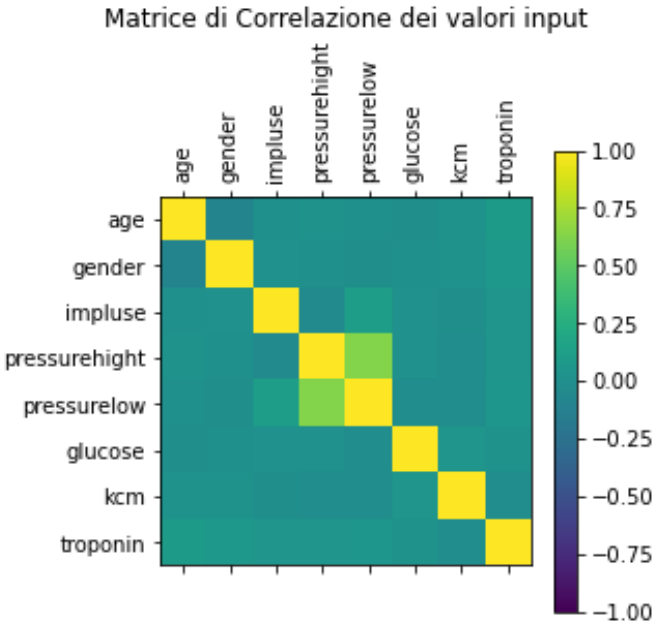
3. Exploratory Data Analysis

In questa fase innanzitutto é stato creato un sub-dataframe di data (num_data) contenente solo le feature di tipo numerico (coincidente con l'insieme dei dati di input) che è stato utilizzato per ottenere le principali statistiche come media, deviazione standard, min e max. In particolare si nota che i valori delle medie per i singoli campi rientrano negli intervalli ritenuti sani[2].

Campo	Media (mean)	Intervallo
frequenza	75.9678	60 < x < 100
pressione. max	127.615	90 < x <139
pressione .min	72.0789	60< x <89
glicemia	146.945	140 < x <199
CK-MB	15.3854	< 25
T-Troponina	0.363568	<0,5 (5 ng/L)

[2] fonte:www.fondazioneveronesi.it/

Dopo questa prima osservazione, i dati sono stati utilizzati per creare la matrice di correlazione.



Dallo studio della stessa si osserva che vi é **quasi una totale assenza di correlazione** tra i singoli campi (sia positivamente che negativamente) ad eccezione delle due coppie **pressione minima-pressione massima** e **pressione minima-frequenza cardiaca** che risultano lievemente piú in risalto rispetto alle altre.

4. Splitting

In questa fase si é suddiviso il dataset in **training set** (65% → data_train) necessario per l'addestramento dei modelli, **validation set** (20% → data_val) necessario per effettuare l'ottimizzazione degli iperparametri e **test set** (15% → data_test) necessario per effettuare la misurazione delle performance.
Prima di effettuare lo splitting è stato inizializzato un **seed** randomico e lo si è memorizzato (inizialmente seed=99) in modo da poter mantenere lo splitting costante anche in esecuzioni successive dello script.
Una volta eseguito lo splitting si é suddiviso ogni set (ognuno dei quali sub-dataframe di data) in x ed Y ovvero campi input ed output.

5. Regressione lineare semplice di variabili correlate

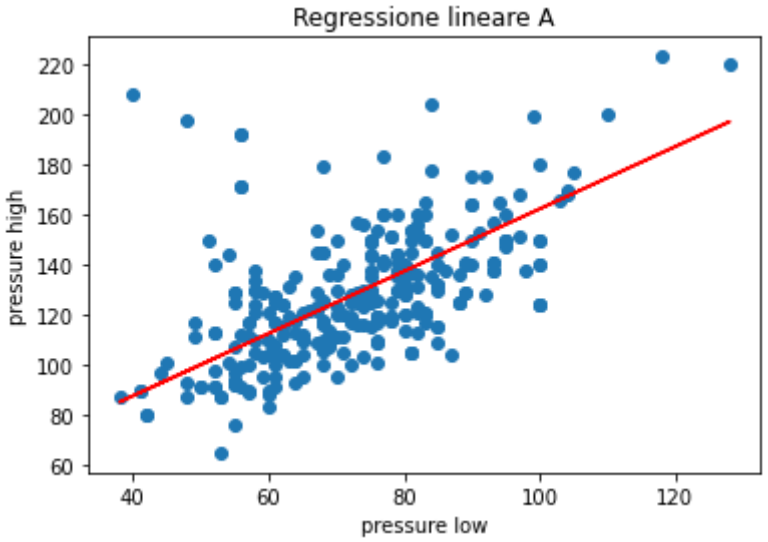
L'analisi della regressione lineare delle 2 coppie di variabili correlate (osservate nel punto 3) è stata suddivisa in due casi: caso A per la coppia pressione minima-pressione massima e caso B pressione minima-frequenza cardiaca.
In entrambi i casi sono state svolte le stesse operazioni e sono state calcolate le stesse metriche, qui di seguito vengono elencati i passaggi solo per un caso (l'altro è stato svolto analogamente).

5.1 CASO A

Come prima cosa sono state estratte le variabili di riferimento (pressurelow-pressurehight) dal set di training e da quello di validation (xA_train, yA_train, xA_val, yA_val).
Una volta separati i dati è stato creato un modello di regressione lineare allenandolo con i dati di training:

```
>>modelA = linear_model.LinearRegression()
>>modelA.fit(xA_train,yA_train)
```

Subito dopo sono stati utilizzati i dati di input del validation set per effettuare delle predizioni (è stato scelto di effettuare una predizione **fuori dal training set,** ma di non utilizzare il test set). Una volta ottenuti i valori della predizione è stato stampato uno scatterplot dei valori nel validation set nel quale é stata inserita la retta di regressione (usando come ordinate yA_pred).



Grazie a questo grafico è possibile osservare quanto i valori originali siano distanti rispetto alla retta di regressione, segnalando così la presenza di **residui** che dovranno essere analizzati piú nel dettaglio.
In seguito, sempre utilizzando i valori predetti si é continuata l'analisi statistica di questa regressione lineare effettuando la stima dei coefficienti ed ottenendo $\beta_0 = 37.9615$ (rappresenta l'intercetta verticale della retta) e $\beta_1 = 1.2431$ (rappresenta il coefficiente angolare).

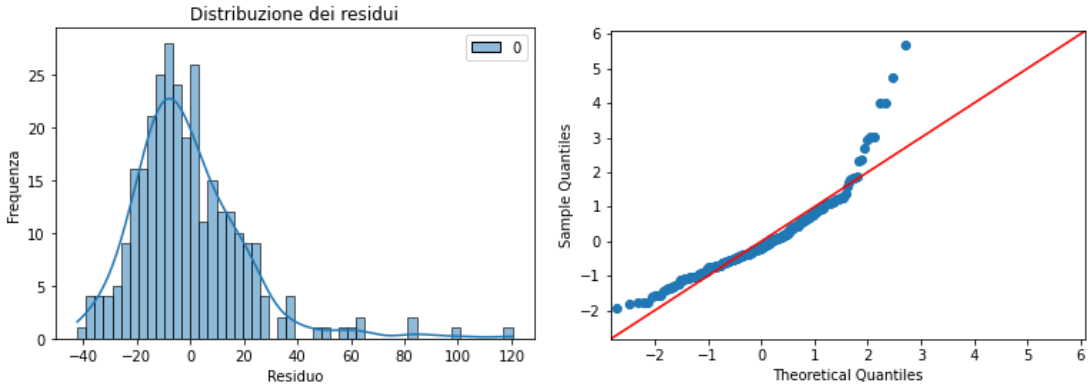
Da questi parametri si ricava che la retta di regressione ha equazione:

$$yA\_pred = \beta_0 + \beta_1 x = 37.9615 + 1.2431x$$

Una volta stimati i coefficienti è stata effettuata un'analisi della "bontà" della regressione attraverso il calcolo delle seguenti metriche:

- coefficiente di determinazione **R^2** = 0.3149
- mean squared error (**MSE**) = 452.3768
- mean absolute error (**MAE**) = 15.3025

Da queste metriche e in particolare dal coefficiente R^2 si può stabilire che questo modello non è considerabile rappresentativo per il campione di dati.
Una volta calcolate queste metriche è stato prima stampato un **istogramma** per osservare la distribuzione dei residui e poi è stato controllato se essi rispettassero l'ipotesi di normalità, prima con un **qq-plot** e in seguito attraverso il **test di Shapiro-Wilk**.



In particolare dal secondo è stato ottenuto un **p-value** = 5.4956*10^(-14).

Questo valore così basso conferma la **non normalità dei residui** già osservabile dal qq-plot in quanto molto distante dalla retta bisettrice.

5.2 CASO B

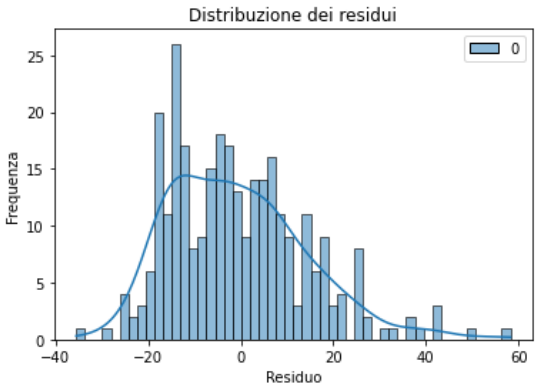
Come spiegato in precedenza, dopo aver estratto le variabili d'interesse (pressurelow-impluse), sono state svolte le stesse procedure utilizzate per il caso precedente, da cui sono stati tratti i seguenti risultati.
I coefficienti β_0 e β_1 sono rispettivamente uguali a 66.4827 e 0.1342, di conseguenza la retta assume l'equazione:

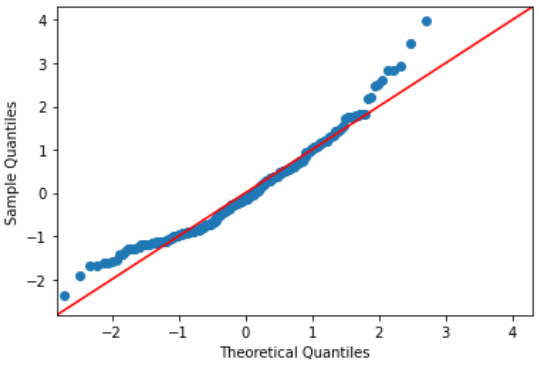
$$yB\_pred = \beta_0 + \beta_1 x = 66.4827 + 0.1342x$$

Come nel caso precedente sono state calcolate le metriche:

- coefficiente di determinazione R^2 = 0.0196
- mean squared error (MSE) = 219.8109
- mean absolute error (MAE) = 11.8957

E' stato osservato che il valore di R^2 risulta ancora più basso del precedente.
Infine è stata effettuata l' analisi dei residui attraverso l'osservazione dell' istogramma, del qq-plot e attraverso il test di Shapiro-Wilk che anche questa volta ha confermato la non normalitá dei residui a causa di un p-value = 4.0871*10^(-07) .





6. Scelta dei modelli ed hyperparameter tuning

Terminate le osservazioni in merito di correlazione tra variabili sono stati selezioni dei modelli da testare per poi selezionare quello con le performance migliori.
I modelli selezionati sono stati: **regressione logistica, SVM con kernel polinomiale e SVM con kernel radiale (rbf)**.
Per ognuno dei due modelli SVM, per evitare condizioni di overfitting/underfitting, sono state ripetute le fasi di addestramento con data_train e di predizione con data_val al variare degli iperparametri (in particolare il grado/**degree** per la SVM polinomiale e il parametro **gamma** per la SVM con kernel radiale) misurando **l'accuratezza** ad ogni iterazione per trovare il valore ottimale.
Da questa operazione si è osservato che nella SVM con kernel radiale l'accuratezza non variava al variare di gamma mentre nella SVM con kernel polinomiale si è raggiunta l'accuratezza massima con il degree=8.

7. Addestramento dei modelli ed analisi delle performance

Una volta ottenuti gli iperparametri è stato deciso, a causa della dimensione del dataset originale relativamente ridotta, di aggiungere al training set il validation set, in quanto quest'ultimo non sarebbe piú servito e avrebbe permesso al modello di essere addestrato su un quantitativo maggiore di dati.

Subito dopo sono stati creati ed addestrati i 3 modelli sfruttando gli iperparametri migliori e per ognuno di essi sono state calcolare le seguenti metriche: misclassification error (**ME**), misclassification rate (**MR**), percentuale di misclassification (MPer = MR*100) ed accuratezza(**ACC** = 1 - MR).

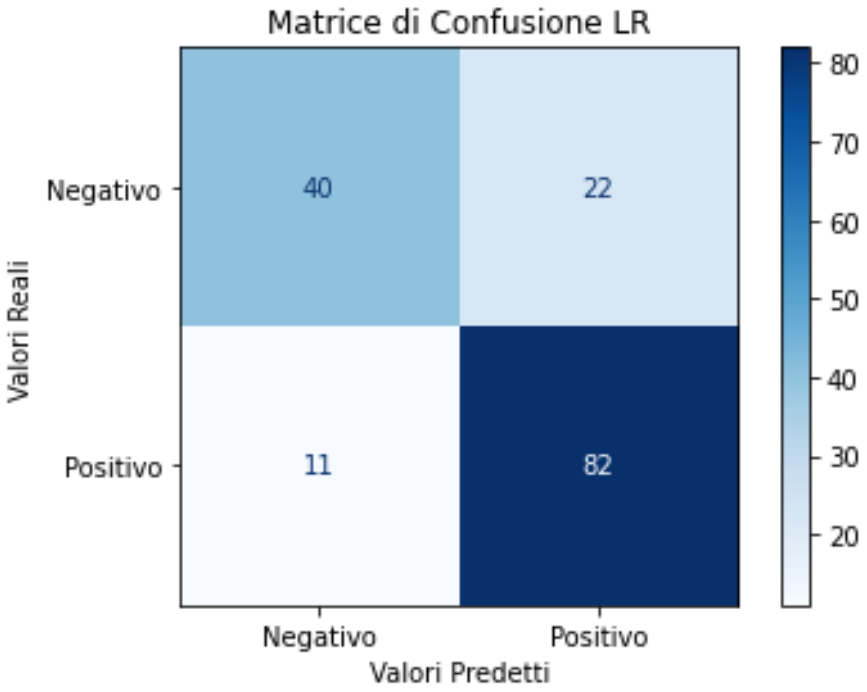
7.1 RISULTATI DELL'ANALISI DELLE PREDIZIONI

		ME	**MR**	**MPer**	**ACC**
1	LOG. REG.	33	0.2129	21.29 %	0.7871
2	SVM - poly	37	0.2387	23.87 %	0.7613
3	SVM - rbf	51	0.3290	32.90 %	0.6710

Osservando le metriche di ogni modello e mettendole a confronto risulta evidente che:

$$ACC(1) > ACC(2) > ACC(3)$$

Una volta stabilito che il modello dall'accuratezza maggiore è quello di regressione logistica, a causa della natura medica del problema di classificazione trattato, é stata realizzata la matrice di confusione (perché un falso negativo è molto piú grave di un falso positivo) .



Dalla matrice di confusione emerge come le 33 previsioni errate (ME) sono suddivise in 22 falsi positivi (FP) e 11 falsi negativi (FN), mentre le previsioni esatte in 82 veri positivi (TP) e 40 veri negativi (TN).
Da questi valori sono stati calcolate le seguenti metriche:

- Sensibilitá (**TPR**) = 0.8817
- Specificitá (**TNR**) = 0.6451
- Precisione (**PPV**) = 0.7884

Grazie a queste metriche possiamo affermare che *quando viene predetto un esito positivo è molto probabile che si tratti di un vero positivo*, purtroppo *non si puó dire lo stesso per quanto riguarda le predizioni di esiti negativi* a causa di un TNR non troppo alto.
Il rapporto sbilanciato tra i falsi positivi e i falsi negativi puó essere una riflessione del fatto che il dataset non é equilibrato tra rilevazioni ad esito positivo (61% del dataset) e rilevazioni ad esito negativo (39%).

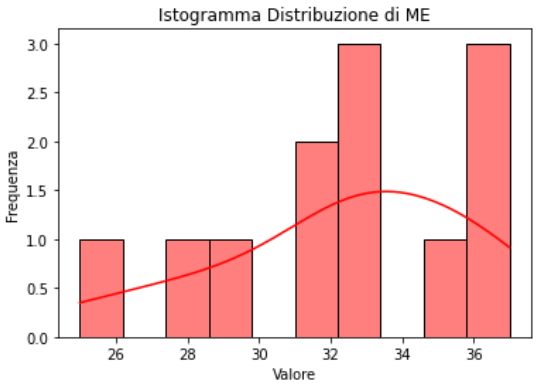
8. Indagine statistica sui risultati della valutazione

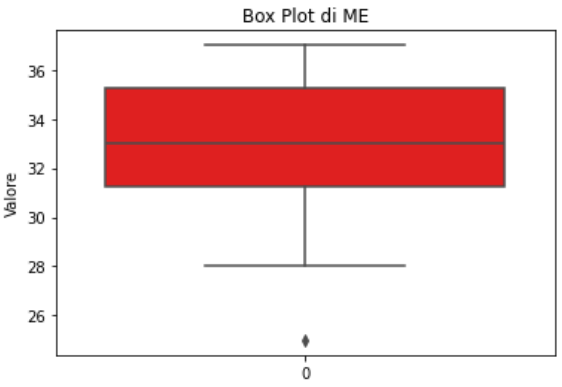
Una sola esecuzione non è sufficiente ad attestare le performance di un modello in modo preciso in quanto questo valore dell'accuratezza potrebbe essere stato solo un caso dovuto al seed scelto.
Per questo motivo è stato generato un vettore di k = 12 numeri interi casuali (nell'intervallo 1<rs<150, attraverso generatore_seed.py) in modo che la valutazione potesse essere effettuata k-volte e ottenere cosí k valori per ogni metrica di errore presa in esame.
Per questa parte è stato preso in considerazione solamente il modello di regressione logistica che nel punto precedente è stato ritenuto il migliore.
Per ogni seed (compreso quello iniziale s=99) è stato eseguito lo script precedente e sono state memorizzate le metriche di ME, MR, ACC in un file csv.

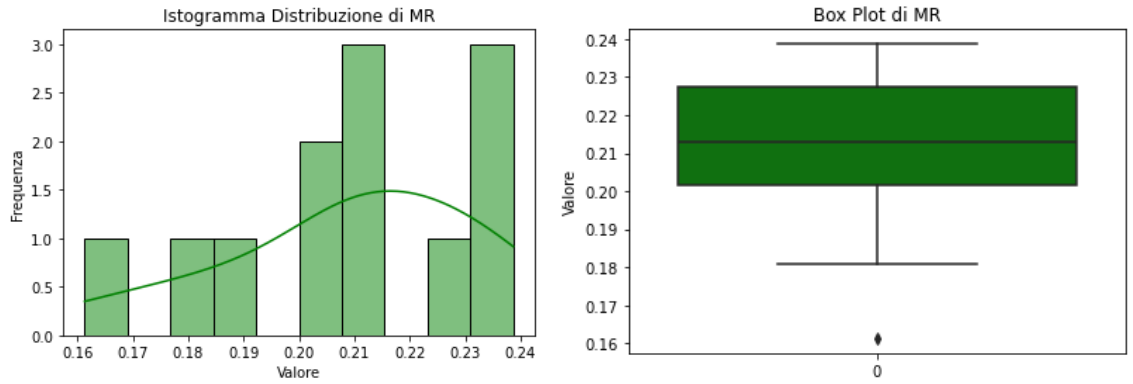
8.1 Statistica descrittiva

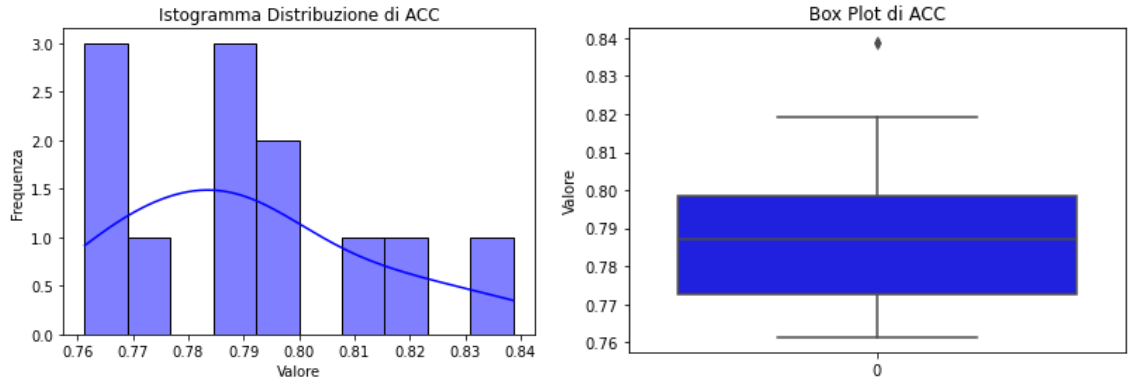
8.1.1 Misure del centro

Una volta in possesso dei nostri dati sono stati inizializzati 3 vettori rispettivamente coi nomi delle metriche da analizzare e come primo passo é stata calcolata la **media campionaria ($\overline{X}$)** per ognuno di essi. Da questa operazione risulta che $\overline{X}_{\mathbf{ME}}$ = 32.4167, $\overline{X}_{\mathbf{MR}}$= 0.2091 e $\overline{X}_{\mathbf{ACC}}$ = 0.7909. Successivamente sono stati visualizzati i grafici di ogni metrica (boxplot e istogramma) per controllare visivamente se i valori appena trovati fossero rappresentativi della realtá.









Osservando i grafici emerge evidente la relazione tra le metriche, infatti MR presenta un grafico molto simile a quello di ME perché indica semplicemente il rapporto tra ME e il numero delle predizioni, mentre quello di ACC è il simmetrico di quello di MR rispetto all'asse x = 1 (perché ACC = 1 - MR).

Dopo aver osservato i grafici sono state calcolate le misure della **mediana (Me)** per ogni metrica: $\boldsymbol{Me}_{\mathbf{ME}} = 33$, $\boldsymbol{Me}_{\mathbf{MR}} = 0.2129$, $\boldsymbol{Me}_{\mathbf{ACC}} = 0.7871$.

8.1.2 Misure della diffusione

Come primo passo di questa fase é stata calcolata la **varianza (s^2)** delle metriche: $\mathbf{s^2_{ME}} = 12.8106$. $\mathbf{s^2_{MR}} = 0.0005$. $\mathbf{s^2_{ACC}} = 0.0005$.

Dopodiché é stata calcolata la **deviazione standard ($s \rightarrow \sqrt{s^2}$)** dei campioni: $\mathbf{s_{ME}} = 3.5792$, $\mathbf{s_{MR}} = 0.0231$, $\mathbf{s_{ACC}} = 0.0231$.

Infine sono stati osservati i **range interquartili (IQR)** dati dalla formula:

$$IQR = Q3 - Q1 = q(0.75) - q(0.25)$$

arrivando cosí ai seguenti risultati:
IQR$_{ME}$ = 4, **IQR$_{MR}$** = 0.0258, **IQR$_{ACC}$** = 0.0258.

8.1.3 Analisi della forma

In questa fase è stata analizzata la forma delle distribuzioni dei campioni.
Come prima cosa sono stati calcolati gli indici di **simmetria/skewness (g_1):**
g_1ME = -0.6792, **g_1MR** = -0.6792, **g_1ACC** =0.6792.
Questi valori rispecchiano la relazione che é stata osservata precedentemente sui grafici dei nostri campioni e indicano che ME ed MR sono asimmetrici a sinistra mentre ACC è asimmetrica a destra.
È giusto notare che questa asimmetria che è stata rilevata è considerabile trascurabile in quanto:

$$|g_1| < 2/\sqrt{6/n}$$

Successivamente é stata misurata la **curtosi (g_2)** che come ci si sarebbe potuto aspettare ha valore: **g_2ME** = **g_2MR** = **g_2ACC** = - 0.3661.
Da questi valori è stato confermato che non vi è eccesso di curtosi poiché:

$$|g_2| < 4/\sqrt{6/n}$$

Per concludere, a causa delle dimensioni ridotte del campione è stato effettuato il test di Shapiro-Wilk per verificare se il campione si distribuisse normalmente.
Da questo test sono emersi dei **p-value** che come nel caso della curtosi sono molto simili e tutti approssimabili a 0.3679. Grazie a questo valore di molto superiore alla soglia considerata significativa (p-value > 0.05) **non può essere rifiutata l'ipotesi di normalitá.**

8.2 Statistica inferenziale

In questa fase è stato calcolato l'intervallo di confidenza al 95%per la media in modo da rendere chiaro quanto questo modello sia effettivamente affidabile.

8.2.1 Intervallo di confidenza per la media

Come prima cosa è stato deciso che a causa della dimensione ridotta del campione verrá utilizzato il valore del **quantile della distribuzione t di student ($t_{\alpha/2}$)** e poiché non é nota la deviazione standard della popolazione, verrá utilizzata quella campionaria (s).
Utilizzando questi parametri si è ottenuto:

- I. di confidenza di ME: (30.1426, 34.6908)
- I. di confidenza di MR: (0.1945, 0.2238)
- I. di confidenza di ACC: (0.7762, 0.8055)

9. Interpretazione dei risultati e conclusioni

Osservando gli intervalli di confidenza di questi SRS(k) ed interpretando il significato di essi, é stata raggiunta la conclusione che **la <u>stima esatta</u> della media dell'accuratezza della popolazione (μACC) ha il 95% di probabilità di cadere nell'intervallo tra 77.62% e 80.55%**.

Per concludere è giusto osservare che **a causa della natura medica** del nostro problema di classificazione, data la percentuale relativamente alta di errore e data la percentuale alta di falsi negativi che la caratterizza (circa ⅓ di tutte le previsioni errate come visto dalla matrice di confusione) questo modello non è da considerarsi statisticamente affidabile.

Fonti consultate

- Slides del corso di statistica numerica
- Dispense del corso di statistica numerica
- Documentazione scikit-learn:https://scikit-learn.org/stable/
- Documentazione pandas:https://pandas.pydata.org/docs/
- Documentazione statsmodels:https://www.statsmodels.org/stable/index.html

Librerie utilizzate

Per la realizzazione degli script sono state utilizzate i seguenti package:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Scikit-learn
- Scipy
- Statsmodels