Studio su dataset per schiene in salute e non

Alessandro Carella – Matricola: 726006

Email: [a.carella49@studenti.uniba.it](mailto:a.carella49@studenti.uniba.it)

Link repository GitHub: <https://github.com/AlessandroCarella/ingegneria-della-conoscenza>

Requisiti funzionali:

il progetto è stato realizzato in python, in quanto esso risulta più idoneo per la trattazione e lanalisi dei dati e come ambiente di lavoro è stato utilizzato visual studio code con le estensioni relative.

Librerie utilizzate:

matplotlib: visualizzazione dei grafici

numpy: visualizzazione dei grafici

pandas: importazione dei dataset .csv

Scikit-learn: applicazione concetti di ML

Networkx: creazione di grafici

Installazione e avvio:

* Installare [visual studio code](https://code.visualstudio.com/) e l[estensione per compilare i file .py](https://marketplace.visualstudio.com/items?itemName=ms-python.python)
* installare [python 3.9](https://www.python.org/downloads/release/python-390/) tramite il sito ufficiale
* aprire leditor di testo
* aprire il file “main.py” e premere il tato F5

Dataset:

il dataset scelto, “Lower Back Pain Symptoms Dataset”, è una raccolta di dati di psicofisici sulla schiena delle persone analizzate. [Link sorgente](https://www.kaggle.com/datasets/sammy123/lower-back-pain-symptoms-dataset).

Preprocessing del dataset:

Immagine che contiene testo, targa

Descrizione generata automaticamenteIl dataset è stato modificato per poterlo utilizzare correttamente; non è presente una mancanza di dati ma le colonne non hanno dei nomi specifici ma vengono indicate come colX, colY, …, a parte per l’ultima che viene denominata come “Class\_att” e che rappresenta lo stato della schiena analizzata dai dati precedenti, dando come output 2 possibili stati “Abnormal” e “Normal”, ed è quindi una feature dicotomica; ho proceduto quindi con la sostituzione dei 2 valori, rispettivamente con 0 e 1.

Per rimodulare il dataset ho quindi rinominato le colonne con i nomi indicati dalla fonte.

Feature del dataset:

• Pelvic incidence: la somma dell'angolo di inclinazione del sacro e dell'angolo di orientamento dell'articolazione sacroiliaca.

• Pelvic tilt: l'angolo formato tra la linea orizzontale e la linea che passa attraverso le spine iliache anterosuperiori.

• Lumbar lordosis angle: l'angolo formato tra la linea orizzontale e la curva naturale della parte inferiore della colonna vertebrale.

• Sacral slope: l'angolo formato tra la linea orizzontale e la base del sacro.

• Pelvic radius: la distanza tra il centro dell'articolazione dell'anca e il centro dell'articolazione sacroiliaca.

• Degree spondylolisthesis: lo scorrimento in avanti di una vertebra rispetto alla vertebra sottostante.

• Pelvic slope: l'angolo formato tra la linea orizzontale e la linea che passa attraverso le spine iliache posterosuperiori.

• Direct tilt: l'angolo formato tra la linea orizzontale e la linea che passa attraverso la sommità del sacro e la verticale.

• Thoracic slope: l'angolo formato tra la linea orizzontale e la curva naturale della parte superiore della colonna vertebrale.

• Cervical tilt: l'angolo formato tra la linea orizzontale e la curva naturale della parte superiore della colonna vertebrale.

• Sacrum angle: l'angolo formato tra la linea orizzontale e la sommità del sacro.

• Scoliosis slope: l'angolo formato dalla curvatura laterale della colonna vertebrale rispetto alla linea verticale.

Panoramica dei dati:

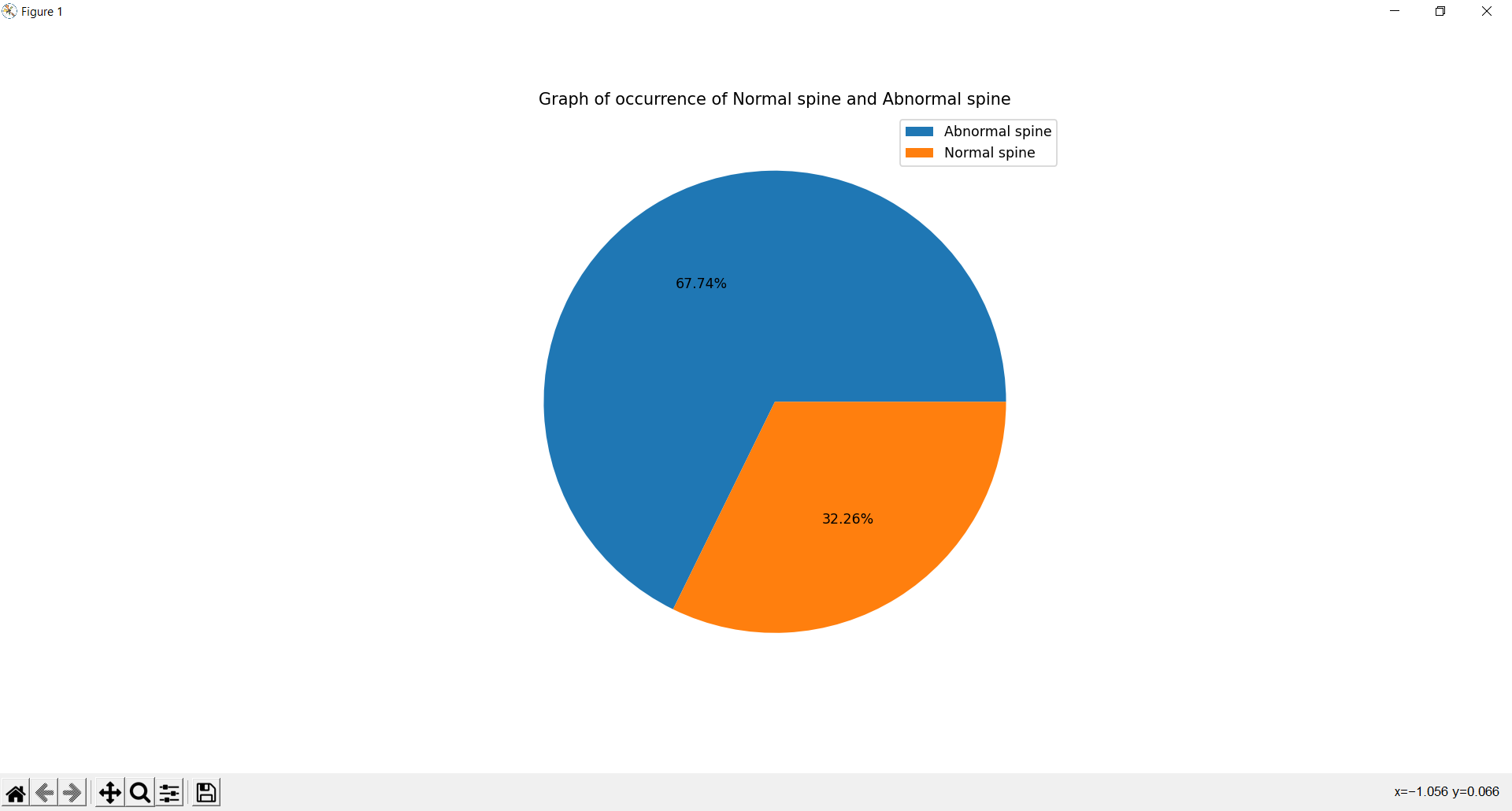
Attraverso la stampa di una tabella viene mostrata a schermo una panoramica delle informazioni contenute nel dataset:

Immagine che contiene testo

Descrizione generata automaticamente

Bilanciamento delle classi:

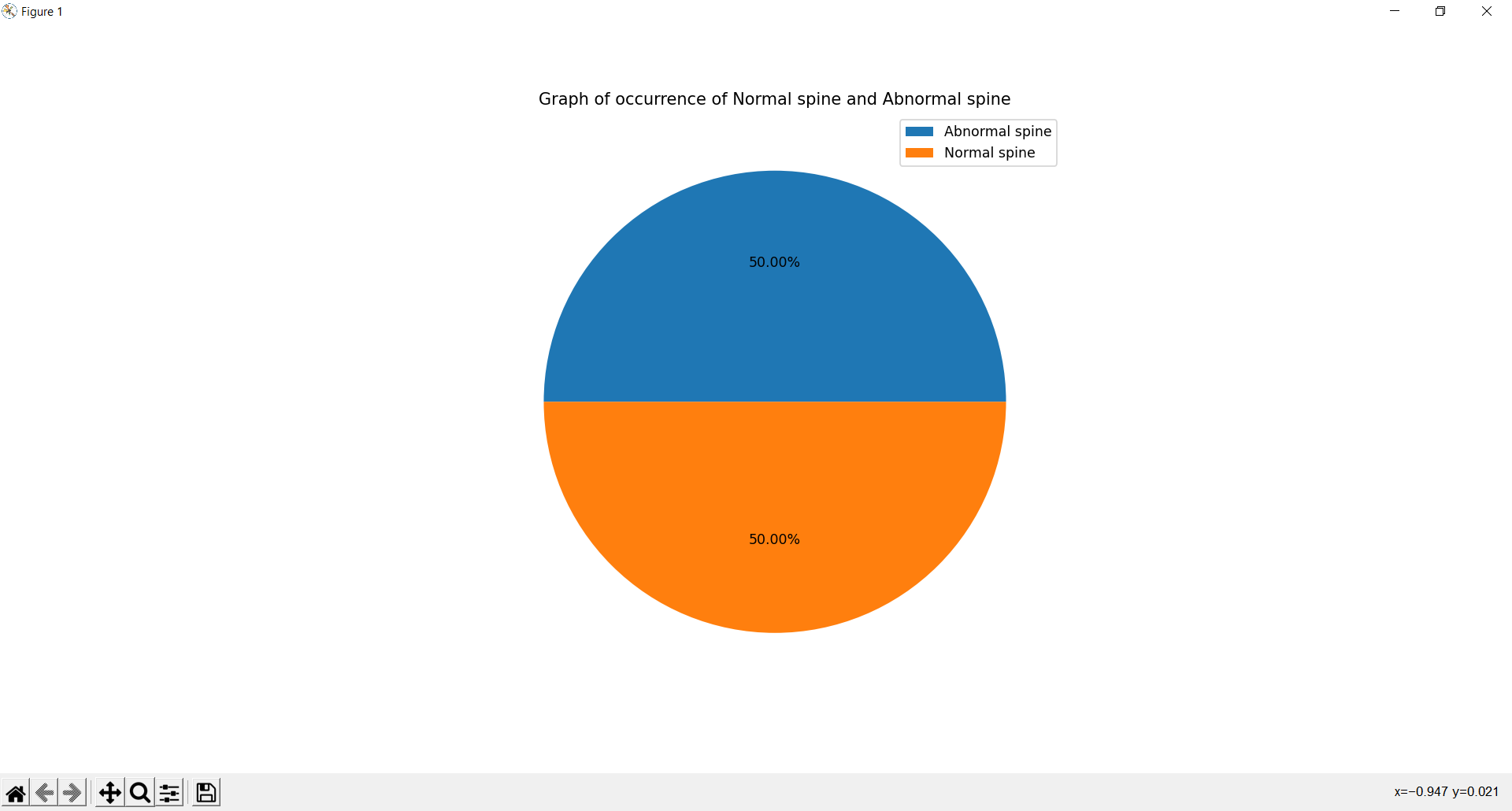
attraverso un grafico, realizzato con pyplot (matplotlib) vediamo che i dati del dataset non sono bilanciati:



Classi squilibrate inficiano sulla precisione. Questo è un problema molto comune nell’apprendimento automatico (in particolare nella classificazione), che si verifica in set di dati con un rapporto sproporzionato di osservazioni per le varie classi di dati coinvolte.

La precisione standard non misura più in modo affidabile le prestazioni, il che rende l’addestramento del modello molto più complicato.

Vi sono diversi modi per poter risolvere il problema dello sbilanciamento delle classi. La soluzione che ho deciso di applicare è l’oversampling. Per far ci`o abbiamo individuato la classe con più dati e la classe con meno dati ed ho effettuato un resampling facendo combaciare le occorrenze. Ottenendo così classi bilanciate:



Clustering per rimozione outliers:

Per quanto riguarda l’utilizzo dell’apprendimento non supervisionato ho deciso di implementare la ricerca di cluster per la rimozione degli outliers:

Ho inizialmente reso le feature del dataset indipendenti per poi standardizzare i dati in modo tale da non avere problemi con la varianza delle features nella loro comparazione.

Ho poi utilizzato la classe DBSCAN offerta dalla libreria sklearn per effettuare la ricerca dei cluster del dataset e la ricerca degli outliers.

Ho ottenuto questi risulati:  
Immagine che contiene testo

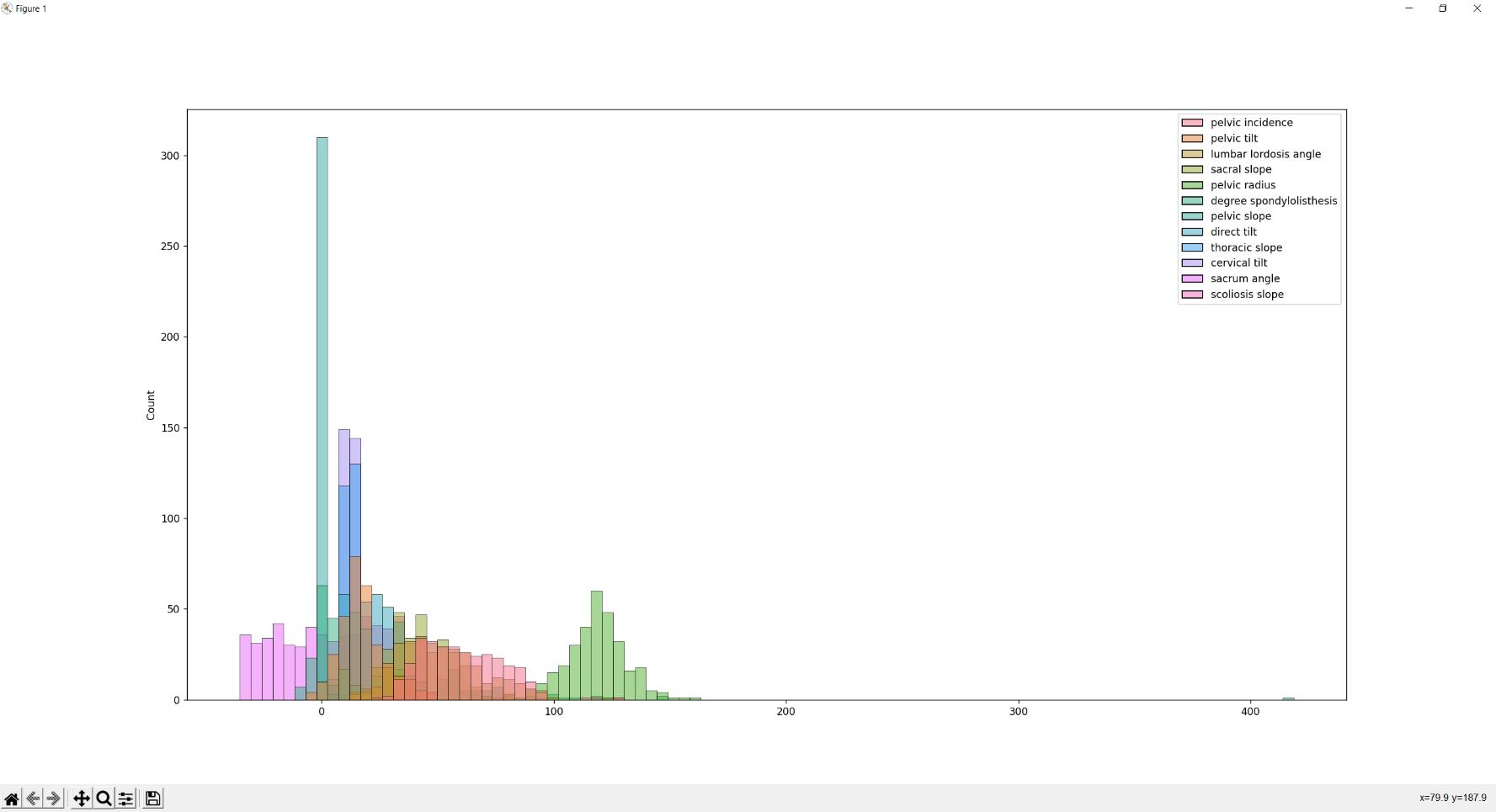
Descrizione generata automaticamente

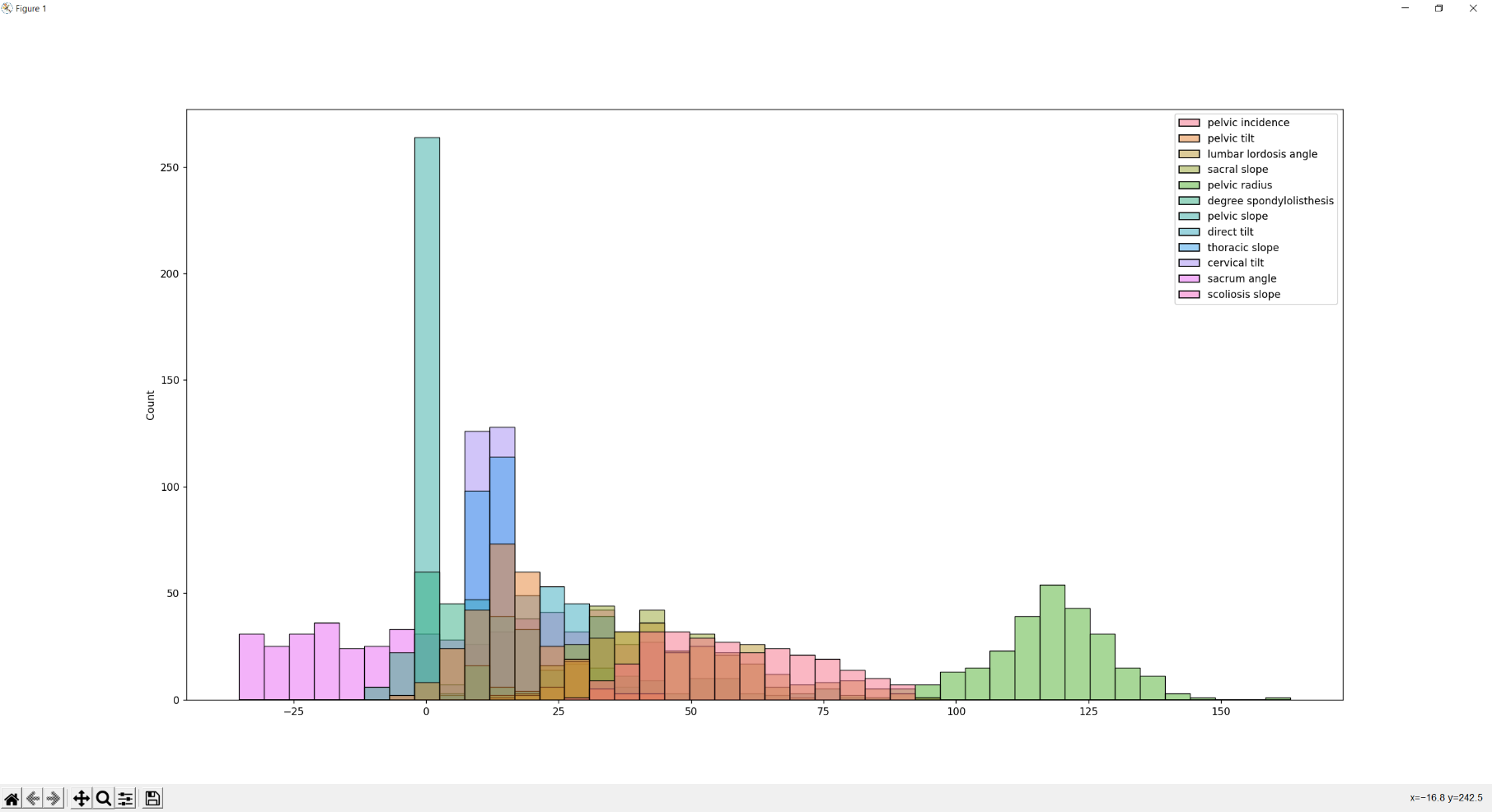
Non essendo presenti cluster i dati che risultano essere outliers sono l’intero dataset, ho quindi proceduto con il ritrovamento e la rimozione degli outliers attraverso il metodo dello z-score con threshold impostato a 3.

Il dataset resta però quasi del tutto invariato con questo valore di threshold e ho quindi deciso di abbassarlo a 1,8; modifica che ha portato il dataset ad avere una dimensione di 256 campioni (partendo da 310).

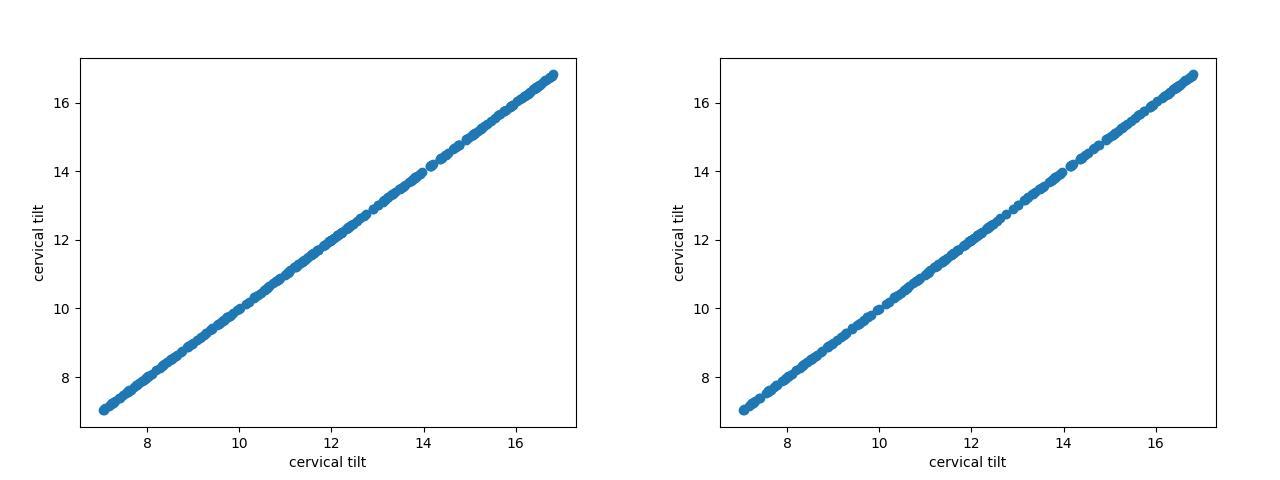
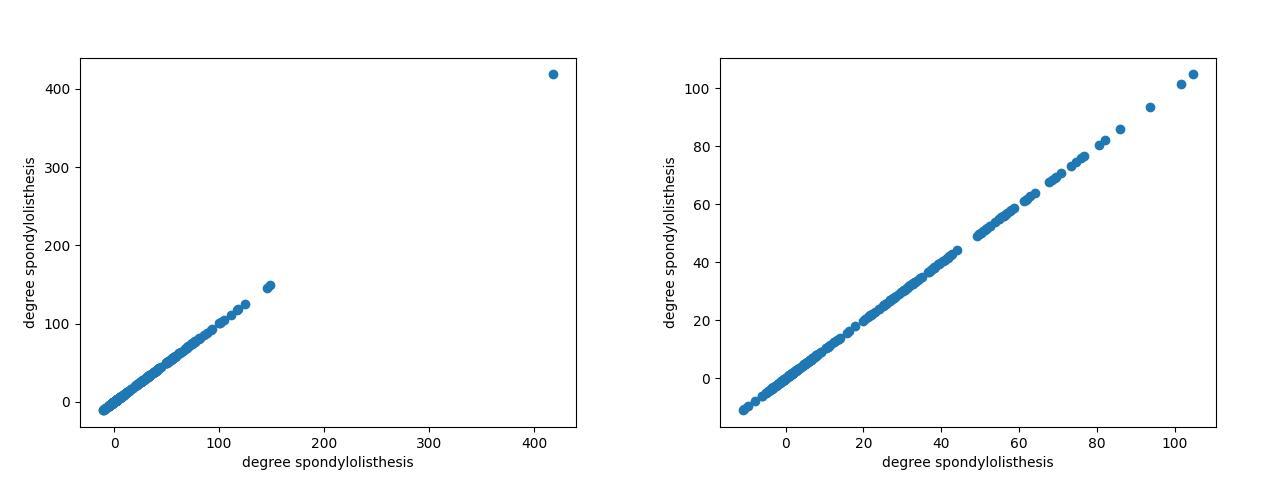
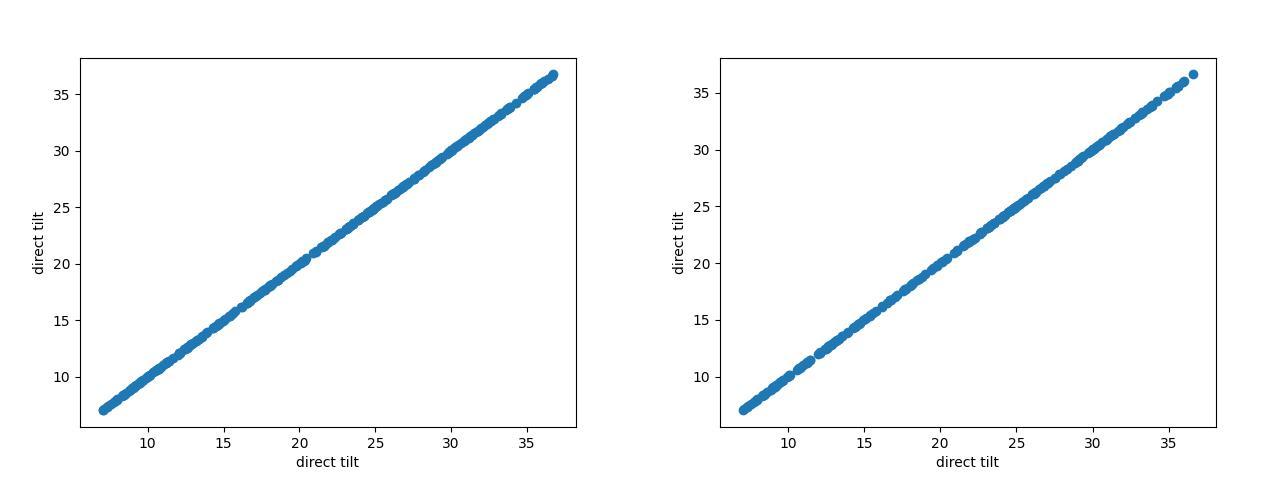
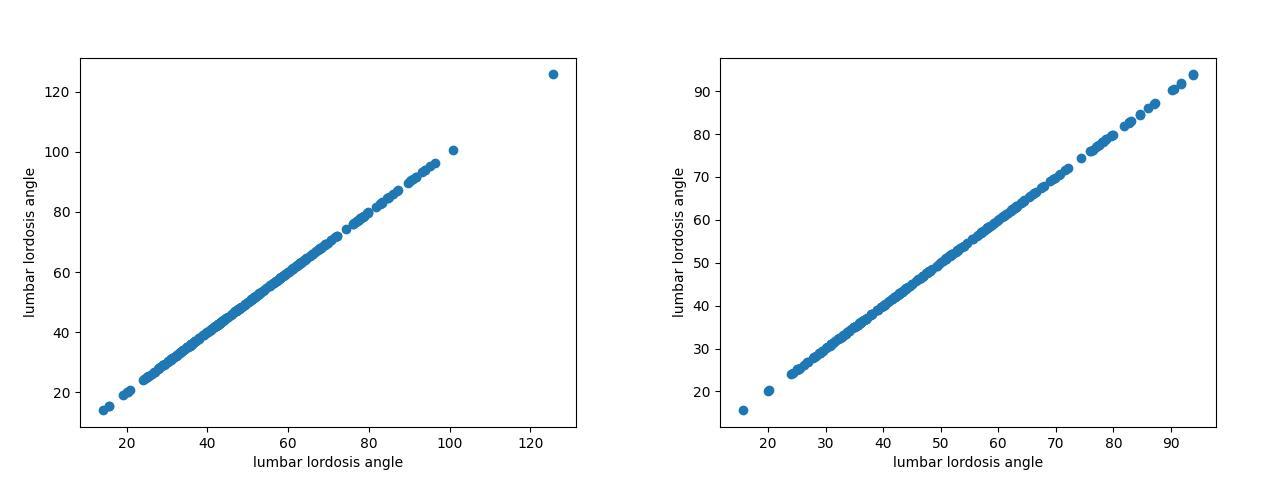
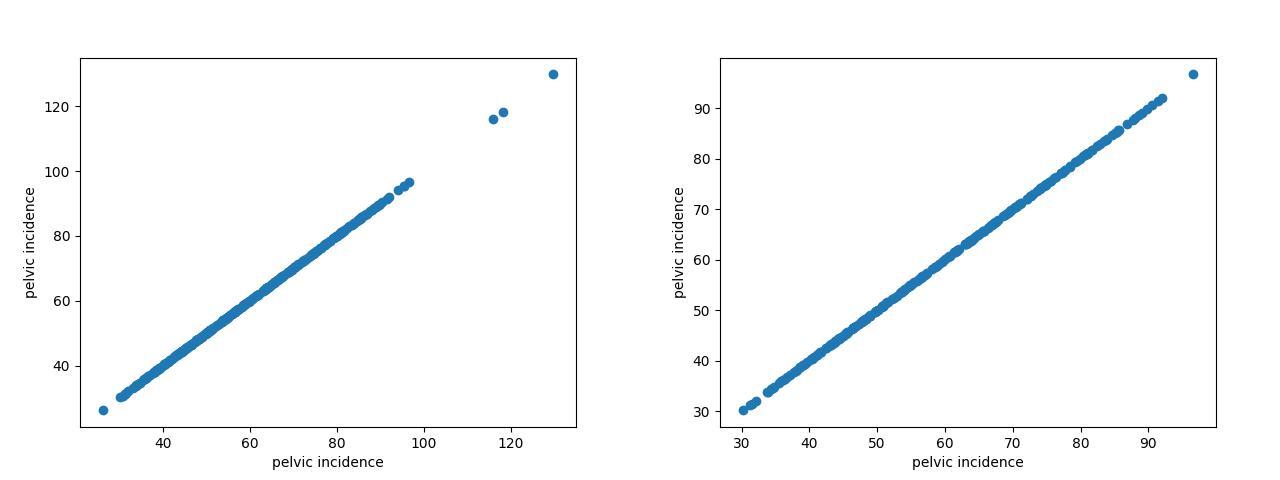
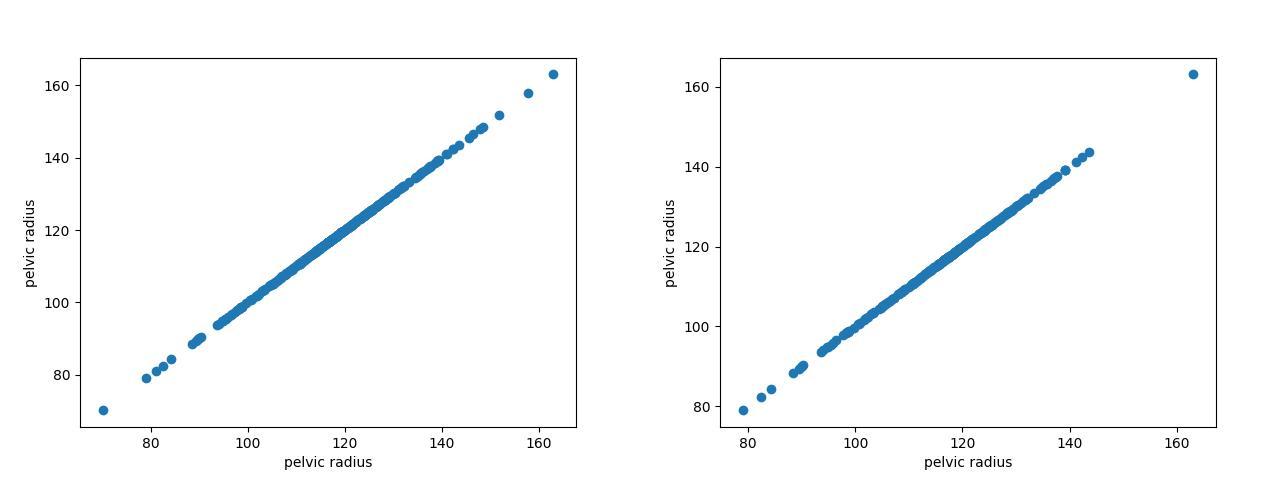
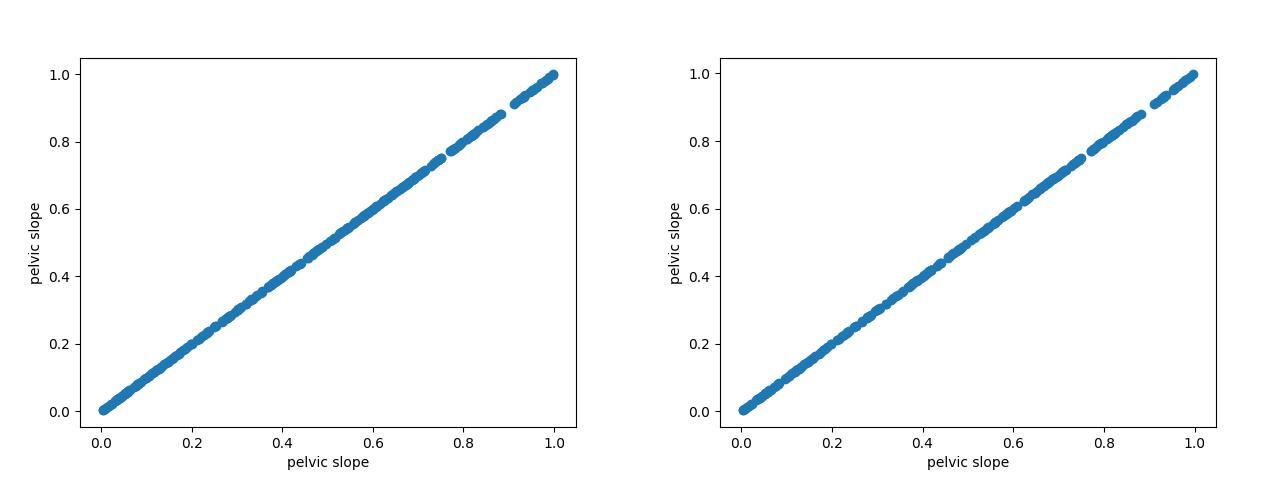
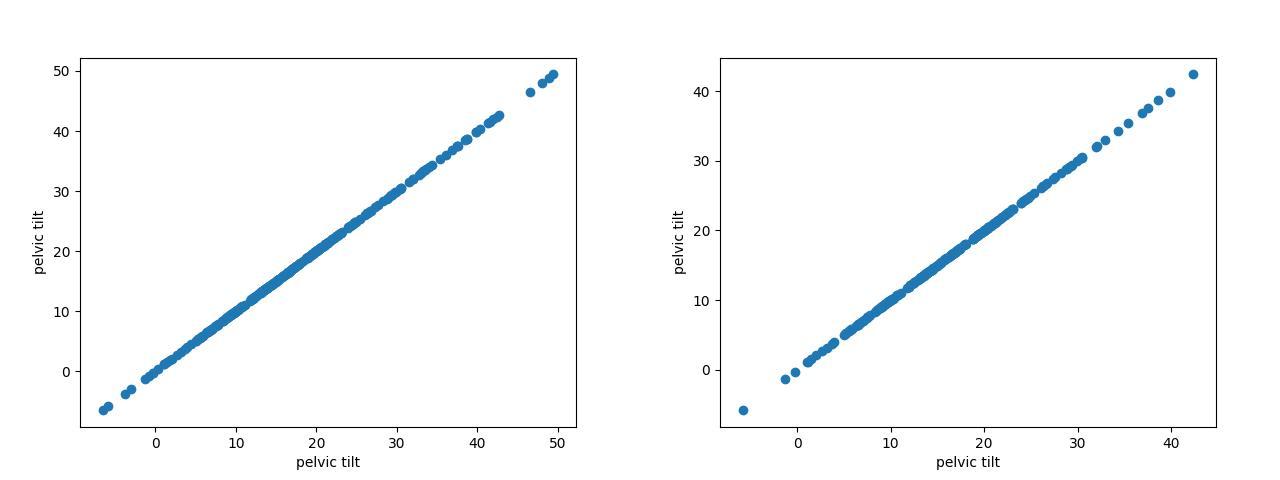
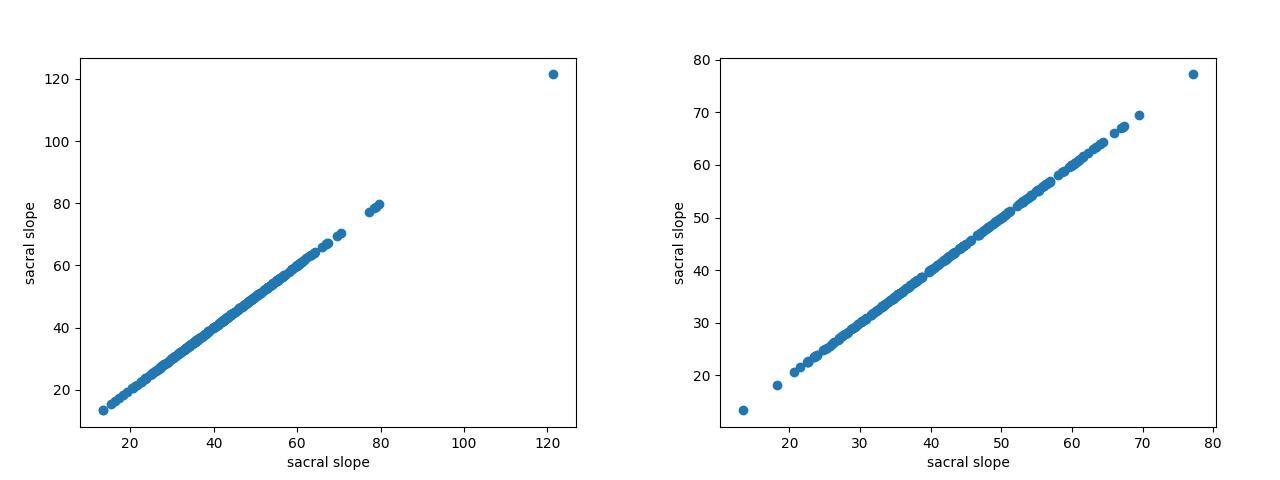
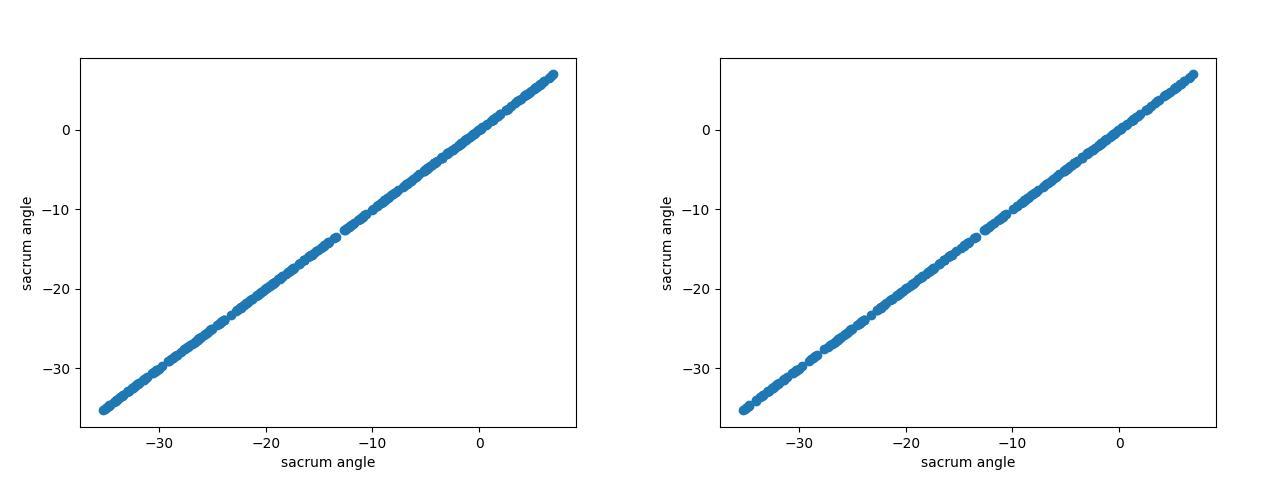
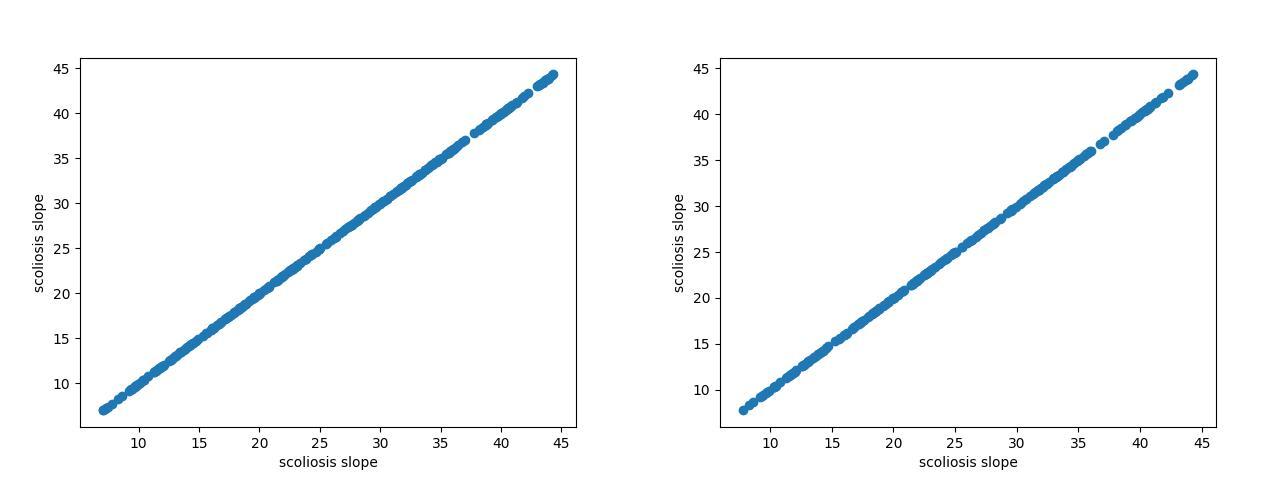
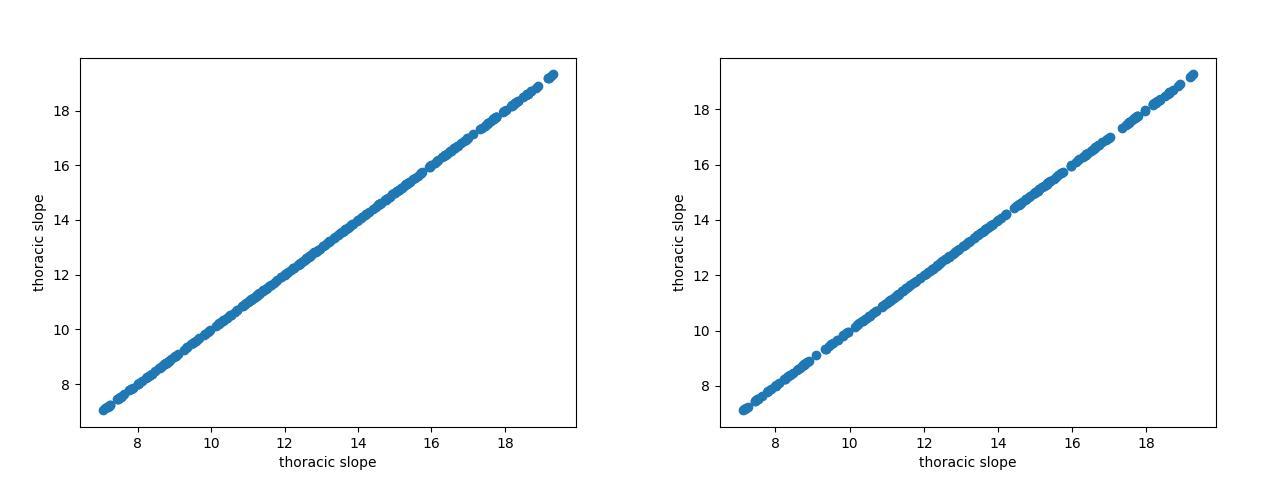
Ho poi implementato il metodo dell’isolation forest per la rimozione degli outliers ed ho ottenuto simili risultati (258 campioni restanti) ed ho quindi deciso di mantenere quest’ultimo dataset; l’ho preferito in quanto il valore standard del threshold per il metodo dello z-score è 3, che non ha portato a risultati ottimali, mentre per l’isolation forest non ho dovuto apportare nessuna modifica.

Come è possibile vedere dall’istogramma i dati agli estremi del dataset sono stati rimossi:





Di seguito gli scatter plot per ognuna delle feature del database, prima e dopo la rimozione degli outliers:



Apprendimento supervisionato:

Questi sono i modelli valutati:

* KNN (K-Nearest Neighbors)
  + K-Nearest Neighbors è un algoritmo utilizzato per il riconoscimento di pattern e la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini al campione in esame.
* Decision Tree
  + Decision Tree è un classificatore strutturato ad albero in cui ogni nodo può essere una foglia o un nodo interno. Se è una foglia, indica il valore della classe assegnata all'istanza. Se è un nodo interno, specifica il test effettuato su un attributo. Per ogni valore dell'attributo in un test, l'algoritmo crea un ramo e il relativo sottoalbero.
* Random Forest
  + Random Forest è un classificatore ottenuto dall'aggregazione di alberi di decisione tramite bagging. Questo metodo minimizza l'overfitting del set di training rispetto agli alberi di decisione.
* SVC (Support-Vector Classification)
  + SVC è un modello di apprendimento per la regressione e la classificazione. Dato un insieme di esempi etichettati con una delle due possibili classi, un algoritmo di addestramento per le SVC costruisce un modello che assegna i nuovi esempi a una delle due classi, ottenendo così un classificatore lineare binario non probabilistico.
* Classificatori Naive Bayes
  + BernoulliNB (Bernoulli Naive Bayes) è simile al multinomiale naive bayes, ma i predittori sono variabili booleane. I parametri utilizzati per prevedere la variabile di classe possono assumere solo i valori sì o no.
  + GaussianNB (Gaussian Naive Bayes) è una variante di Naive Bayes che segue la distribuzione normale gaussiana e supporta dati continui.

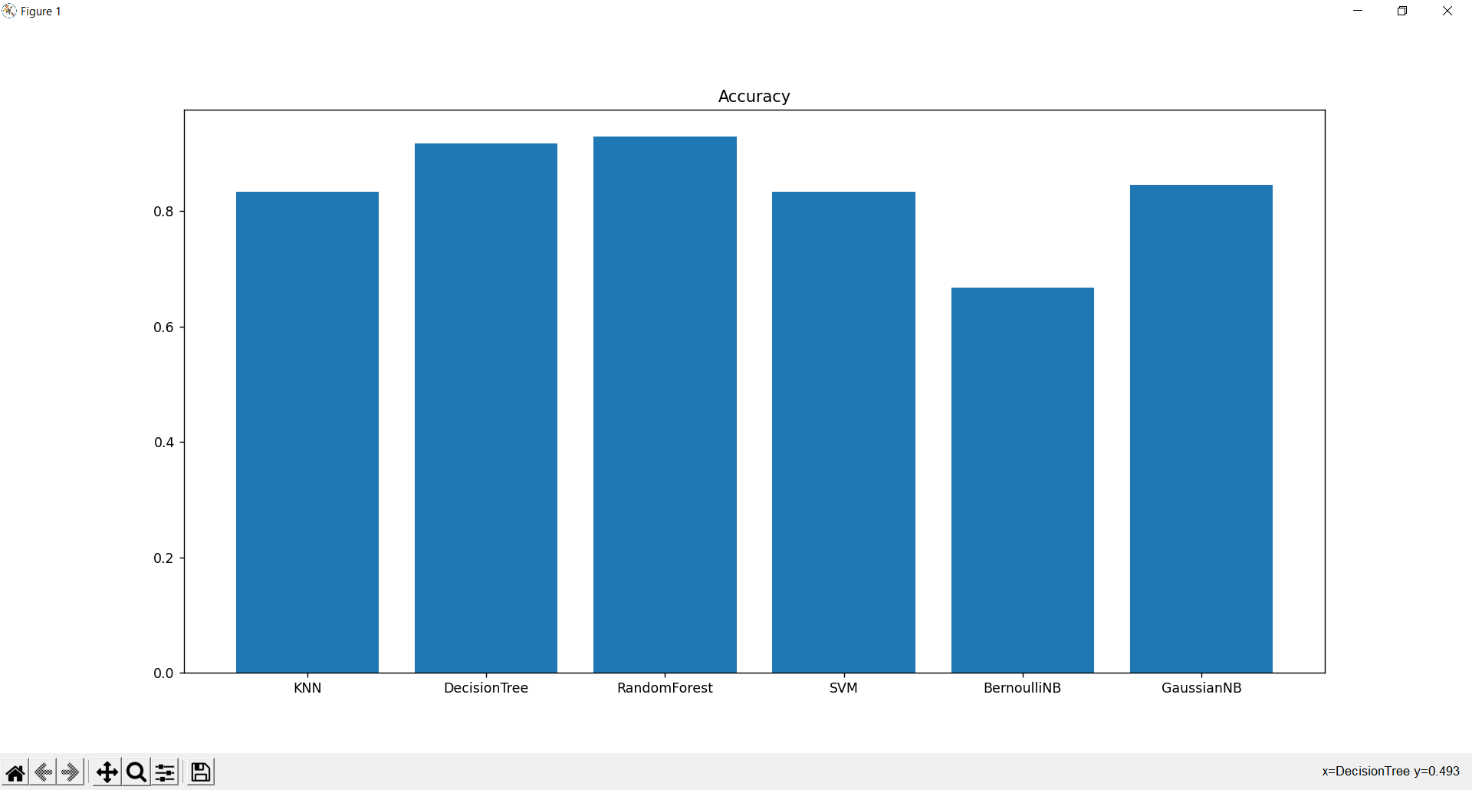
Su questi ho eseguito il K-Fold Cross Validation (in particolare il RepeatedKFold con 5 ripetizioni) per capire quale di questi sia il più attendibile.

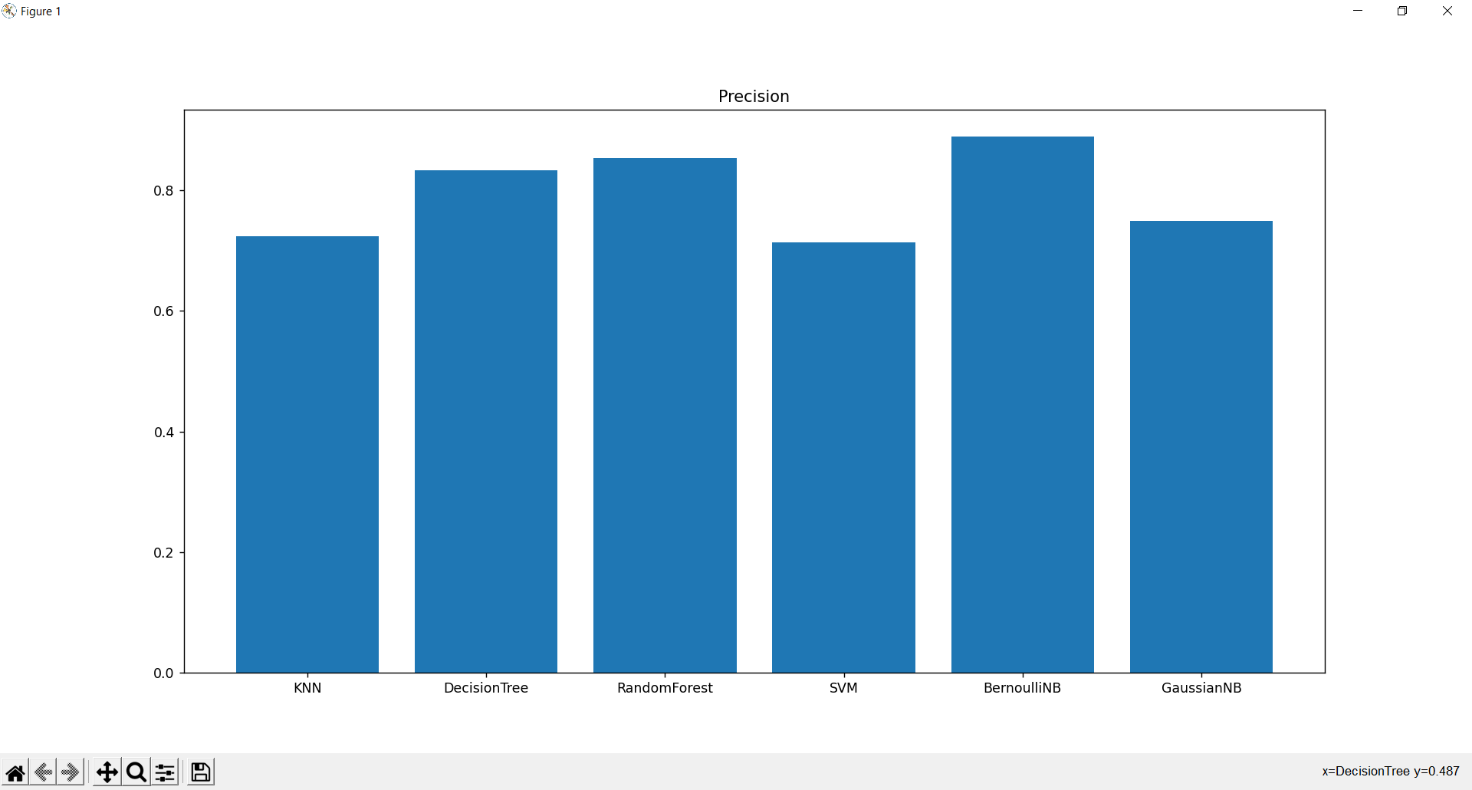
Le metriche utilizzate per la valutazione e i loro relativi risultati sono:

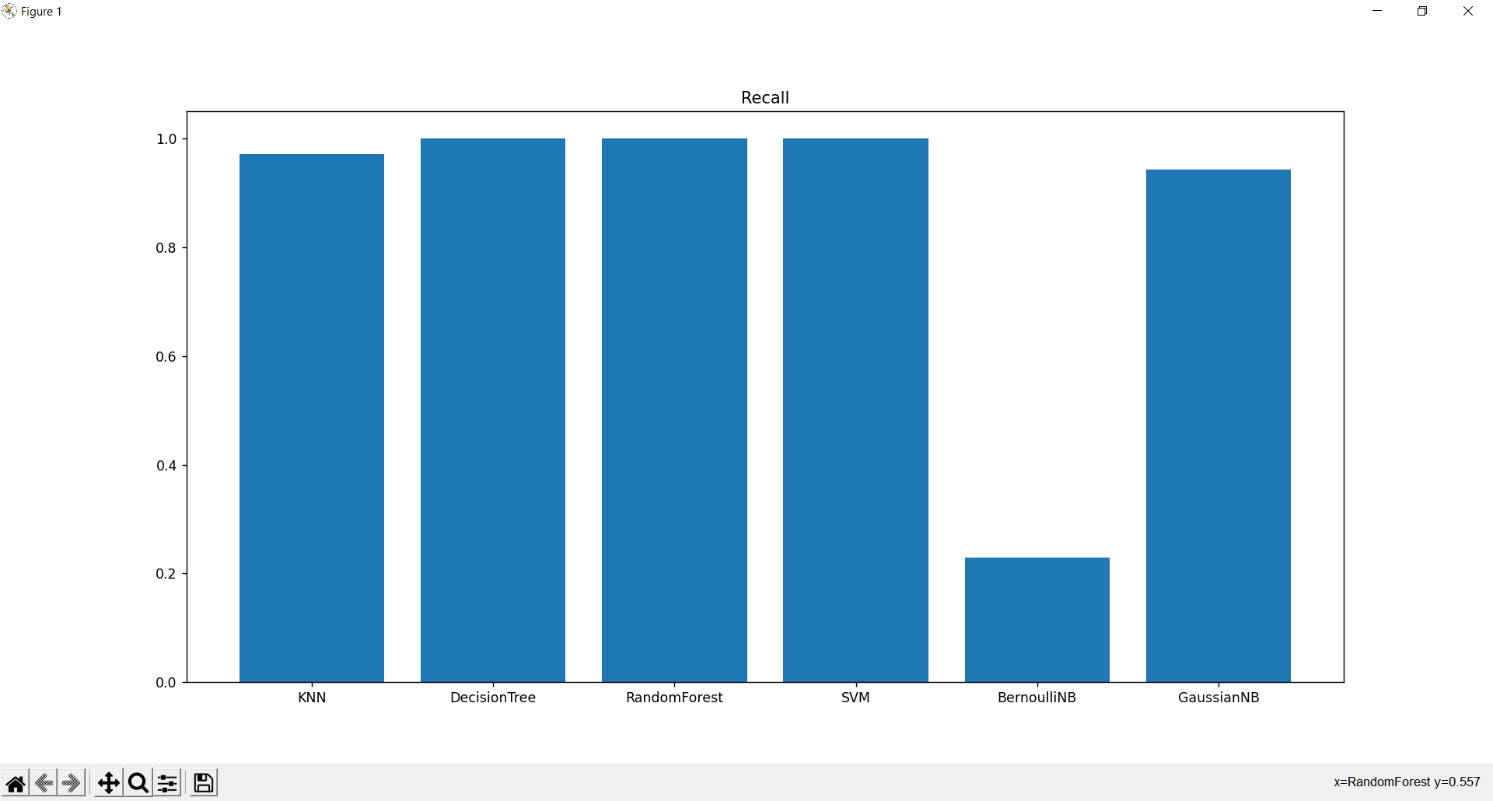
Immagine che contiene testo

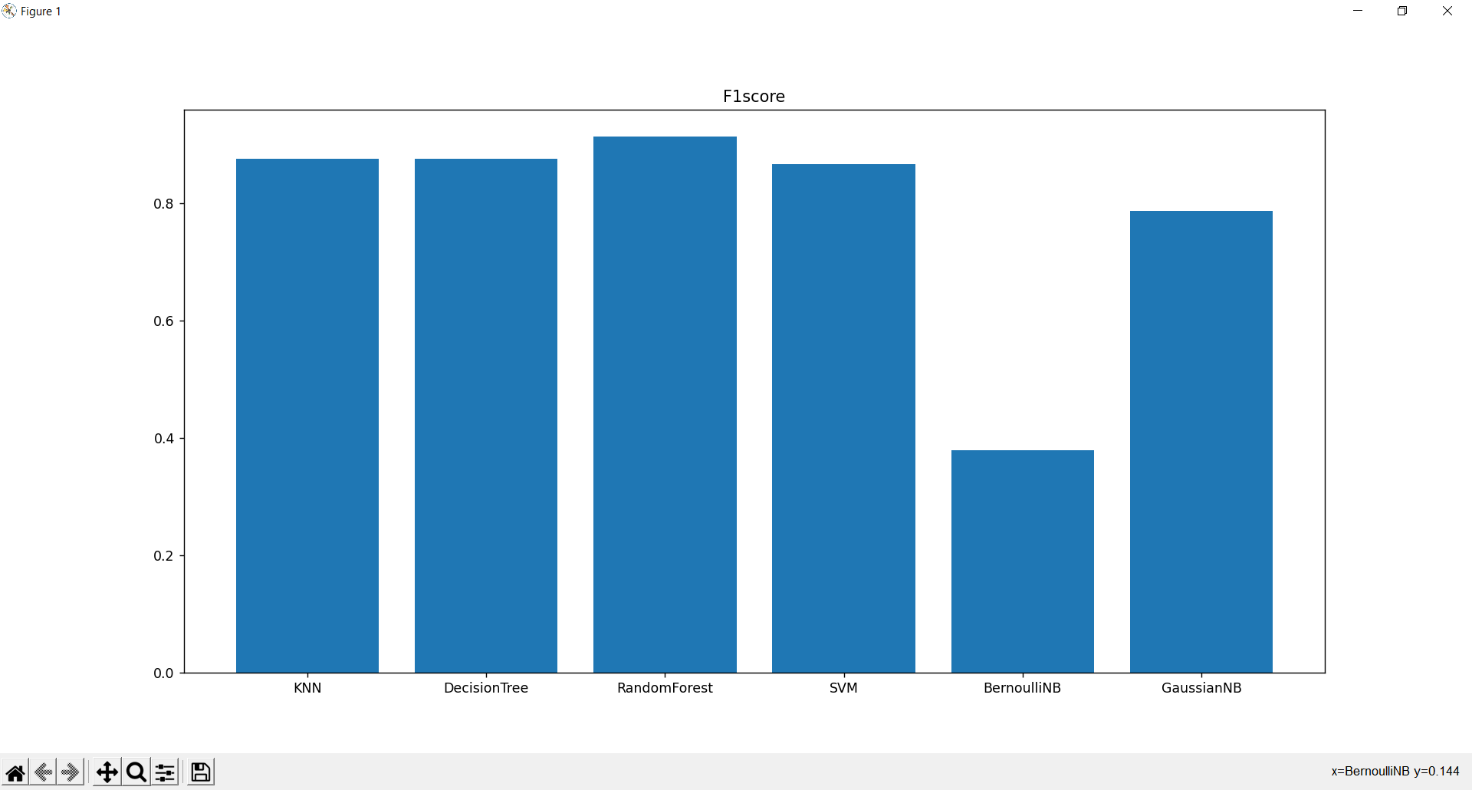
Descrizione generata automaticamente

In particolare:









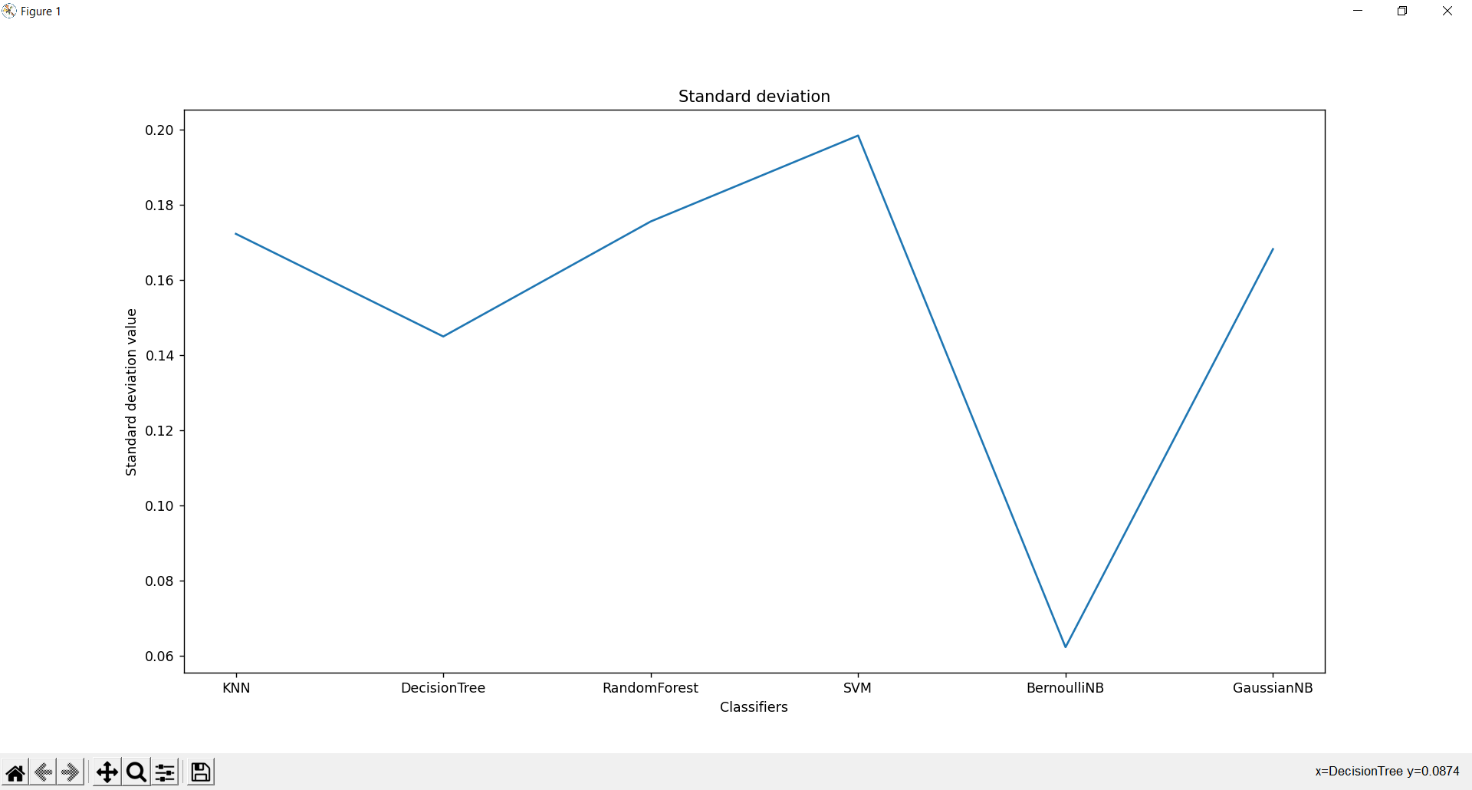


Immagine che contiene testo

Descrizione generata automaticamente

Il classificatore migliore risulta essere il random forest.

Verifica dell’importanza delle features:

Ho successivamente generato un grafico, basato sulle analisi effettuate in precedenza, che estrae le features più importanti basandosi sul random forest:

