

Student Name: Lohith karri

Student ID: 23040177

Git Hub: <https://github.com/LohithKarri/lohit.git>

Introduction

In Predicting Parkinson's Disease, using the concept of machine learning algorithms helps analyze patient data for early findings regarding the presence of the disease. The prediction is built with features of motor symptoms, speech, and even general health to get early indicators. Libraries in Python that provide this capacity are powerful and can do many things-science in their core use-cases with Scikit-learn and TensorFlow.

Histogram of MDVP:Fo(Hz)

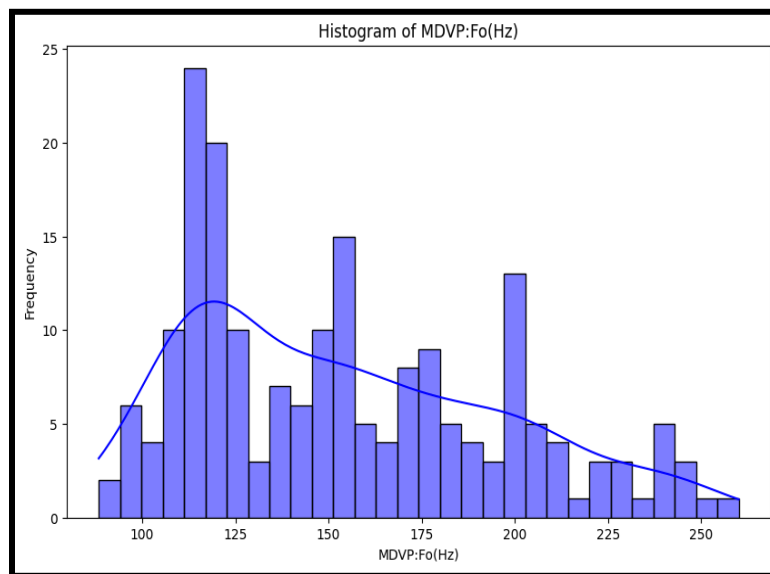


Figure 1: Histogram

This figure makes a histogram of the 'MDVP:Fo(Hz)' feature from the dataset, utilizing Seaborn's histplot function. The histogram draws a representation of the data's distribution with an added curve of a Kernel Density Estimate. The plot appears with customized labels for the title and for both axes, yielding an understanding of the frequency distribution of this feature.

Scatter Plot of MDVP:Fo(Hz) vs MDVP:Fhi(Hz)

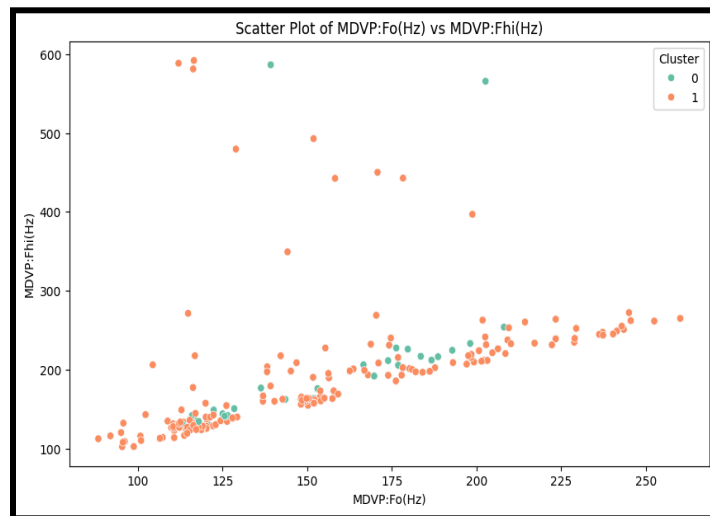


Figure 2: Scatter plot

It draws a scatter plot to demonstrate how 'MDVP:Fo(Hz)' and 'MDVP:Fhi(Hz)' are correlated according to the data. Each data point is colored according to its 'Cluster' label from the 'Set2' color palette. The titles and axis labels are all personalized in a way that it allows comparison of the two features while allowing observation of any patterns or clustering present.

Confusion Matrix for KMeans Clustering

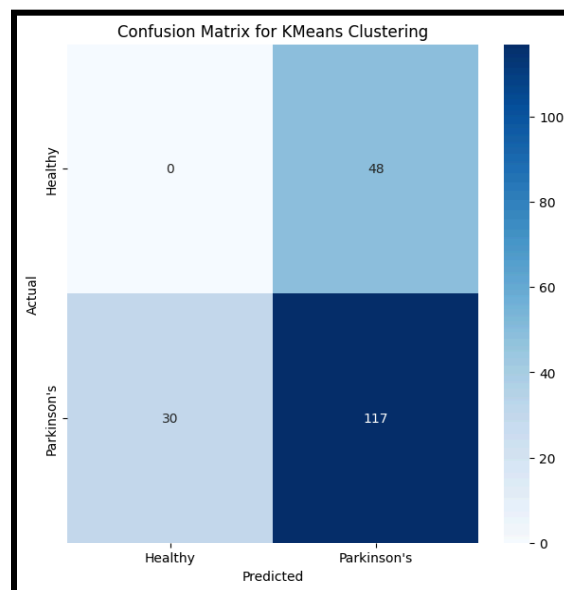


Figure 3: Confusion matrix

The following code produces a confusion matrix to evaluate the accuracy of KMeans clustering when compared with the actual class labels. The `confusion_matrix` function is applied to compare the predictions of cluster labels (`df['Cluster']`) against the labels (`y`). A heatmap is used for displaying the matrix, and annotations will be used in order to display counts with customized axis labels in an effort to compare 'Healthy' and 'Parkinson's'.

Elbow Plot for KMeans Clustering

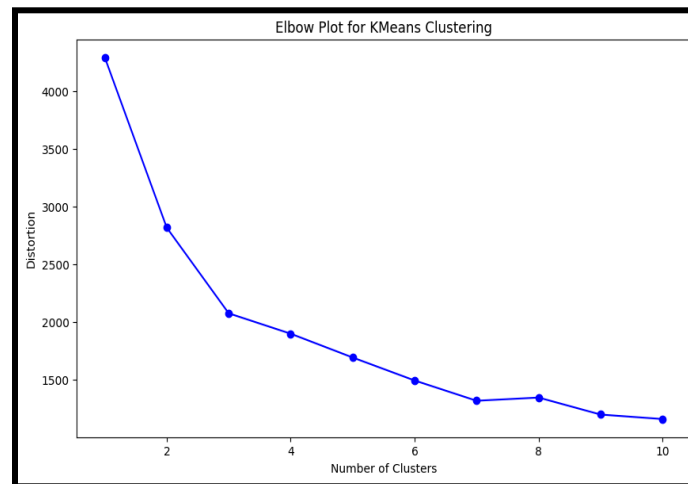


Figure 4: Elbow plot

The above graph produce an elbow plot for determining the number of clusters in KMeans. The code calculates the distortion or inertia at various k values between 1 and 10, then fits the KMeans model to the scaled data (X_{scaled}). The plot represents how distortion changes with respect to the number of clusters. By locating the "elbow," the distortion slows down and helps find a proper number of clusters.

Actual vs Predicted for Linear Regression (line-fitting)

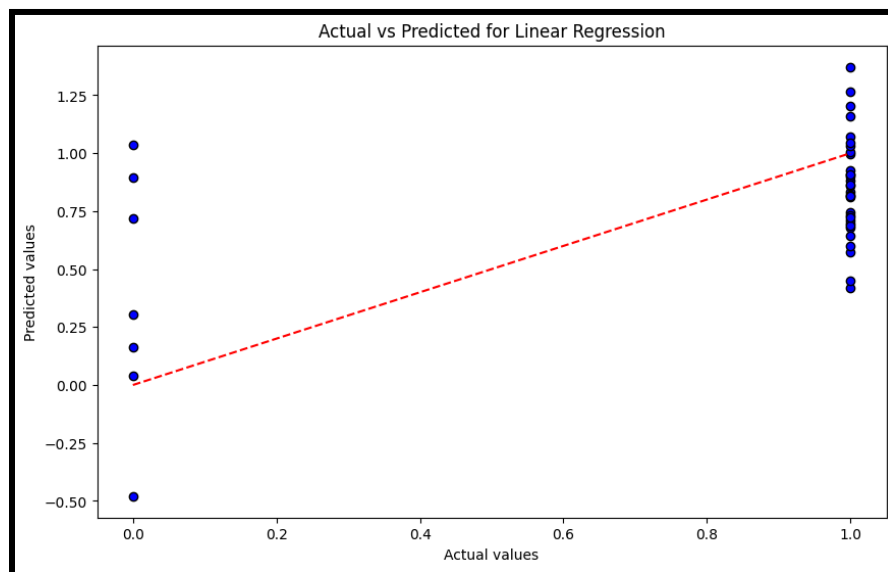


Figure 6: Linear regression

It breaks the data into training and testing. Using the training data, a linear regression model is fitted. The predicted values of test values are done, and it depicts the scatter plot of the actual and predicted values with a red dashed line depicting the perfect prediction ($y = x$). In this plot, the better it is to predict, it is possible to assess that the model is performing accurately.