Vashu Patel

December 11, 2022

1. First approach was importing all the libraries and packages required such as plotting, model libraries, pandas to read files, splitting libraries and various others.
2. Task 1:
    a. I renamed the 1,2,3 as Normal, Suspect and Pathological respectively.
    b. I created a labels list with the respective classes of health states.
    c. Used .count() function to give the bar graph with exact values.
    d. I also plotted a pie chart with respective percentages.
    e. Printed the values to the console.
3. Task 2:
    a. To print the 10 best features, I used the .corr() function to develop a correlation matrix.
    b. I created two empty sets of featureGT90 and featureGT95 to store any feature that is greater than 90% and 95% respectively.
    c. Since the matrix is a 2 dimensional-list, I used a for loop to access and iterate every correlation and checked with an if and elif statement if there are correlations that are significantly correlated with 90% and 95%.
    d. I created a dictionary that will hold the 10 best features.
    e. I created the same for loop, but this time to store both the features and their correlations.

f. I sorted the dictionary in a descending order to get the most correlated features and sliced the dictionary into 10.

g. Then printed them out.

4. Task 3:

    a. I split the data using train_test_split with a sample size of 30% as per the requirement. I also stratified to have a balance.

    b. I decided to use Gaussian Naïve Bayes and Decision Tree Classifier as those were one of the earliest models we learned in class.

    c. I fitted and predicted the data using X_test.

    d. Printed the report of each model.

5. Task 4:

    a. To print the confusion matrix, I used the confusion_matrix function and had different y_pred variables for each model.

    b. I also had a function definition created named conf_matrix that created the heatmap and visually present the confusion matrices.

6. Task 5:

    a. To print the F1 Score, I imported f1_score from sklearn and used weighted average type to print it.

    b. To print the ROC Curves and Precision vs Recalls, I simply used the predict_proba to to predict the X_test first.

    c. Then used metric.plot_roc_curve and plot_precision_recall_curve to plot the respective graphs.

7. Task 6:

a. For K means Clustering, I used KMeans() functions with the number of clusters, init and random state passed as the parameters. I sliced the x values according to the number of clusters.