**WINE DATA ANALYSIS**

**Explanation**:

The jupyter notebook – **Wine\_Data\_Analysis.ipynb** consist of the complete analysis components and all the initial model building activities.

This notebook will give a through idea about the analysis done with the dataset and my approach to the problem.

**In this notebook,**

* I have handled the missing data and used median measure to deal with the missing data
* Performed outlier analysis to find the outliers in the dataset and removed the outliers
* Extracted new feature from the existing feature
* Performed exploratory data analysis to understand the dataset better, finding the relation between the features and the target variable and also between the features
* Performed feature selection to identify the features that holds primary importance and will help in maximizing the model performance.
  + I have used RFE (Recursive Feature Elimination) and RFECV to perform feature selection
* Divided the data into train and test set to build classifiers.
* I have used three machine learning techniques to build the classifier
  + Random Forest
  + Logistic regression
  + Support Vector Machine
* I have also categorized the data into three bins to classify the data better
  + Bad
  + Average
  + Best
* After categorizing the data into bins, built new classifiers around the new data and improved the classification accuracy

2.

The **wine.py** file contains the modulized version of the above analysis.

I have created different functions to perform the analysis done above to come up with a cleaner code.

The functions in the file includes:

**Preprocessing**:

* + This functions takes care of
    - Missing Values
    - Feature Extraction
    - Outliers
  + This functions returns a dataframe that is free of missing values & outliers which can be used for further analysis.

**FeatureSelection**:

* + This function accepts the preprocessed data frame as input and performs the following operations.
  + Split the data into dependant and independent features
  + Splits the fetaures into train and test set
  + Performs normalization (MinMax) on the train and test set
  + Used RFE to identify the top 7 (k=7) features from the dataset which will be used for model building
  + Returns the train and test data with the features selected

**CategorizeBin:**

This function takes the preprocessed data as input and creates a three bins and adds a new column for the bins

**RandomForestModel:**

This function trains a Random Forest classifier based on the train and test data

**LogisticRegressionModel:**

This function trains a Logistic Regression classifier based on the train and test data

**SupportVectorMachineModel:**

This function trains a Support Vector Machine classifier based on the train and test data

**PerformanceMetrics:**

* + This functions prints the performance metrics for the models trained above.
    - Accuracy
    - Confusion Matric
    - Classification Report

**main():**

* This function parses the user input from the command line
* Creates an object for the class and calls the above mentioned function to perform analysis as per the user input.

**Steps to run the wine.py file:**

**python wine.py “Path to the csv” “Method for classification” “Algorithm to use”**

Path to the csv – Path to the csv file in your computer

Method for classification – This argument takes value either 1 or 2

1 – No categorization using bins

2 – Categorize the data into bins

Algorithm to use – This argument takes value of 1,2 or 3

1. Random forest
2. Logistic Regression
3. Support Vector Machine

Example : **python wine.py “D:/Dataset/wine-dataset.csv" 2 1**

This implies load the data and perform Random Forest classification after categorizing the data into bins.