**CORIZO JANUARY COHORT 2024 PROJECT- 1**

**Name- Sarvesh Tikekar**

**Title- Wine Quality analysis using Data Science and Machine Learning Models**

**Introduction**

In this project, we have built a wine quality prediction system using machine learning with Python, using a Random Forest Classifier.

The parameters used for building the system are:

1. fixed acidity of wine
2. volatile acidity of win
3. citric acid present in wine
4. residual sugar in the wine
5. Amount of chlorides, present in wine
6. Amount of free sulphur dioxide
7. Amount of total sulphur dioxide
8. Density
9. pH of wine
10. Amount of Sulphates
11. Amount of alcohol present.

**GETTING THE DATA/ DATASET FOR THE SYSTEM**

The dataset used for the analysis is given below:



**DATA ANALYSIS AND VISUALISATION**

1. First, we found out the total no. of rows and columns in the dataset using the **shape()** function. The output was: (1599, 12)
2. Next step, was to find the total number of rows, having missing values, which was found out using the **isna().sum()** function.
3. The number of such rows, found were 2 out of the total of 1599 rows, so instead of filling in the missing values, we decided to drop those rows, as they would not affect the working of our model later, so no need to fill in the missing values.
4. After dropping the values, we tried to find out the co-relation between the wine quality and rest of the parameters, using the **corr()** function.
5. Also using the **describe()** function, so it gives an overview about the dataset, i.e. mean, mode, median, etc.

**QUALITY ANALYSIS**

1. Firstly, using a catplot from the **seaborn** library we found out from the dataset, what is the range of values that the column ‘quality’ has, which was between 3 and 8 (inclusive).
2. To further understand the relation, between the wine quality and the rest of the parameter, we made use of a **barplot** and was tested on 4 different relations:
3. Quality vs pH: Directly Proportional
4. Quality vs volatile acidity: Inversely Proportional
5. Quality vs Citric acid amount: Directly Proportional
6. Quality vs Alcohol content: Directly Proportional
7. To get a, more in-depth idea of the correlation, we used a heatmap.

**PREPROCESSING THE DATA**

1. As we were concerned about the wine quality, so we split the dataset first into 2 parts:
2. The data in the ‘Quality’ column.
3. Rest of the data.
4. Using a special **lambda** function, we segregated the wine quality into two parts, referencing the range of wine quality we found out during the analysis:
5. Quality 1: If quality of wine is greater than equal to 6.
6. Quality 0: If quality of wine is less than 6.

The values for the quality before, were replaced by these new values.

**PREPARING THE MACHINE-LEARNING (ML) MODEL**

1. To prepare our model, we split the data first into two parts:
2. The Training data, used to train our ML model.
3. The Testing data, used to test our ML model.
4. For this splitting, we used the **train\_test\_split()** method from **sklearn** module. The inputs given are the 2 dataset parts we formed while pre-processing the data.
5. For the test-size, we kept is as 0.15 or 15% so to increase the accuracy of prediction of the model, with a random state as 2 so whenever we train the model the result would be the same i.e. same values in train and test datasets.

**TRAINING THE MACHINE LEARNING MODEL**

1. The Machine Learning model was built using a **Random Forest Classifier.**
2. Random forest is a type of machine learning model that combines multiple decision trees to improve accuracy and reduce overfitting.
3. Each decision tree in the random forest is constructed using a subset of the training data and a random subset of features introducing diversity among the trees, making the model more robust.
4. Firstly, we made a **Random Forest Classifier** “wineModel” using the **RandomForestClassifier()** and then fitted the two training datasets made while preparing the model, in the classifier.

**EVALUATING THE MODEL**

1. Now, for the evaluation, we passed the test dataset into our classifier, which returned us the predicted values of quality for the dataset.
2. Then, we passed these predicted values and the actual values of the wine quality, into an **accuracy\_score()** function, and the score was in the range of 80-85 inclusive.(i.e. 80-85% accuracy)

**MAKING PREDICTIONS**

1. In order to make a prediction, whether the quality is good or bad, we passed an array of the values, of each of the 12 parameters in the dataset, we used for the quality analysis.
2. Here, we ran 2 test cases in which one got a quality value of 1- i.e. actual quality is either a 7 or 8, and the other one got a quality value of 0, i.e. actual quality is less than 6.
3. In both the test cases, the array passed contained values that were already in the dataset.
4. In addition, to that another test case was ran, where the user had to input the parameter values, which gave us the correct answer.

**GITHUB LINK FOR THE PROJECT**