**Article / Blog.**

**On Red Wine Quality.**

The Dataset is from Portuguese red wine variant type of this wine is “Vinho verde”.

We have been provided with physicochemical inputs in a wine which is also an input variable of this dataset and Quality of the wine is provided with an score of the wine, where quality score equal to or more than 7 is considered as good quality of wine and quality score below 7 is considered as bad quality of wine.

Before we proceed further will need to import libraries which will be used in further analysis.

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore")

import sklearn

Will need to give variable in which dataset can be saved, also need to use pandas to read the CSV file.

df=pd.read\_csv("https://raw.githubusercontent.com/FlipRoboTechnologies/ML-Datasets/main/Red%20Wine/winequality-red.csv")

after dataset is reflecting against the variable will check shape of the dataset, df.shape.

This resulted as dataset have 1599 number of rows and 12 columns from which column Quality is our Output variable and rest of the columns are the input variables. On the basis of input, prediction model from this dataset will be built as the output is an categorical will be using classification models to get score and on the basis of score which model is giving us the better result same will be used in an prediction model.

Post that dataset is checked with df.head () which will show first 05 rows from the dataset, df.sample (5) will also give any 05 rows from the dataset and df.tail () will give us last 05 rows from the dataset.

Columns are:- fixed acidity, volatile acidity , citric acid , residual sugar , chlorides , free sulfur dioxide , total sulfur dioxide , density , pH, sulphates , alcohol.

1. First I have checked if the dataset contains any null values

Checked with df.isnull ().sum (). This resulted as there is No missing values present in this dataset.

1. To find the datatypes of columns I have used df.info () this gave information like datatype of each column and also showed if there are null values.
2. Checking the number of unique values from the columns with df.nunique (), as a result column quality have 6 unique values, same further checked with df["quality"].value\_counts() which shows how many times an number of value is repeating in the dataset.
3. Created an heatmap showing if there is null values present in the dataset. Using sns.heatmap(df.isnull())
4. Checking the numerical description of the dataset using df.describe() as datatype of columns is in float64 and int64 we can see the numerical description of each and every columns.
5. This shows count , mean , standard deviation , 25% percentile , 50% percentile , 75% percentile and max of each columns.
6. After this I have plotted columns as box to find out the outliers present in the dataset. Using df.plot(kind="box", subplots=True, layout=(2,6)).
7. This results column fixed acidity , residual sugar , chlorides and sulphates have outliers.
8. Plotting column Quality as count using sns.catplot(x="quality", data = df, kind = "count" ) gives result score 5 and 6 are mostly repeated in the dataset 5 is repeated for 681 times and 6 is repeated for 638 times as comparison with other scores 5 and 6 have highest number of count.
9. Plotted column Quality as X and fixed acidity as Y using kind as bar. sns.barplot(x="quality", y="fixed acidity", data = df) This resulted as score 3 and 8 have presence of fixed acidity is more as compare to other quality score.
10. Plotted column Quality as X and alcohol as Y using kind as bar. sns.barplot(x = "quality", y = "alcohol", data=df) This resulted as score 3 and 8 have presence of alcohol is more as compare to other quality score.
11. Same checked keeping citric acid as Y. sns.barplot(x="quality", y="citric acid", data = df) presence of citric acid is the highest in score 3.
12. Checking the correlation of the dataset using sns.heatmap(df.corr(),annot=True,fmt=".1f").

After this dataset is splitted into x and y for training and testing.

* In x column quality is only dropped rest of the column are saved as x which will be used as input.
* x = df.drop("quality", axis = 1)
* In y I have saved column quality. While saving also applied lambda to make the column quality as an categorical data where quality score equal to or more than 7 is considered as good quality of wine saved as 1 and quality score below 7 is considered as bad quality of wine saved as 0.
* Categorized the column using y = df["quality"].apply(lambda y\_value: 1 if y\_value>= 7 else 0).
* Importing train\_test\_split from sklearn
* from sklearn.model\_selection import train\_test\_split
* Then data splitted in to x\_train , x\_test , y\_train , y\_test where as test\_size is 0.20 which 20% of the data will go for testing and 80% of the data will go training.
* x\_train , x\_test , y\_train , y\_test = train\_test\_split(x,y,test\_size = 0.20, random\_state = 3)
* checking the shape of the dataset post splitting the data into x and y.
* x\_train.shape , x\_test.shape , y\_train.shape , y\_test.shape
* Importing classification models from sklearn and also importing accuracy score to get the result.
* from sklearn.ensemble import RandomForestClassifier
* from sklearn.ensemble import ExtraTreesClassifier
* from sklearn.linear\_model import LogisticRegression
* from sklearn.svm import SVC
* from sklearn.tree import DecisionTreeClassifier
* from sklearn.naive\_bayes import GaussianNB
* from sklearn.metrics import accuracy\_score
* fitting the splitted data into each model and finding out there predicted value.
* Once we get the predicted value same can send into testing y\_test
* For example:-
* First import model from sklearn.ensemble import RandomForestClassifier, then need to give and variable in which need to be saved. rfc = RandomForestClassifier().
* Second fit the x\_train and y\_train into the model , rfc.fit(x\_train,y\_train), then finding the predicted value using pred = rfc.predict(x\_test).
* Third need to find the accuracy score print(accuracy\_score(y\_test,pred)) this will give the result of the model used.
* As we have tried using many classification model to find out the better quality of wine,
* we have got highest and the best result of 0.934375 from Random Forest Classifier,
* Gaussian NB gives us accuracy score of 0.85625,
* Decision Tree Classifier gives us 0.90625 and the second best after Random Forest Classifier,
* SVC gives us 0.884375, Logistic Regression gives us 0.89375
* On this basis of the result I have created the Prediction model.

Pred\_model:-

data = (7.8,0.58,0.02,2.0,0.073,9.0,18.0,0.9968,3.36,0.57,9.5)

data\_as\_np\_array = np.asarray(data)

data\_reshaped = data\_as\_np\_array.reshape(1,-1)

prediction = rfc.predict(data\_reshaped)

print(prediction)

if (prediction[0]==1):

print("Good Qualty Wine")

else:

print("Bad Qaulity Wine")

* In the prediction model in data I have put the input data, the data which was saved as x while training and testing.
* As Random Forest Classifier has given the best result of 0.934375 the same is been used in the model as rfc.predict post this is giving the result in 0 or 1.
* If the result gives 0 as an prediction it will print as Bad Quality Wine and if the result gives 1 as an prediction it will print as Good Quality Wine.