**Red Wine Quality Prediction Project**

1. **Problem Definition:**

The dataset is related to red and white variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

We have used machine learning to determine which physiochemical properties make a wine 'good'.

This dataset can be viewed as classification task. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

* 1. **Aim:** The aim of the project is to build a classification model.
  2. **Attribute Information:** The dataset consists of the following input variables

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. Output variable (based on sensory data): quality (score between 0 and 10)
13. **Data Analysis**
    1. **Calling the libraries and loading the dataset:**

* Pandas is a useful library in data handling.
* Numpy library used for working with arrays.
* Seaborn/ Matplotlib are used for data visualisation purpose.
  1. **Data shape:** The dataset consists of 1599 rows and 12 columns
  2. **Datatype:** All the columns consists of same datatype i.e float. And the output variable is integer. Since all the columns are of same datatype so we need not do any transformation.
  3. **Null values:** The dataset showed no null values in any of the column.
  4. **Unique values:** Most of the columns have a varied number of unique values. The dependent variable i.e the quality column consists of 6 unique values ranging from 3 to 8.

Unique Values Frequency

5 681

6 638

7 199

4 53

8 18

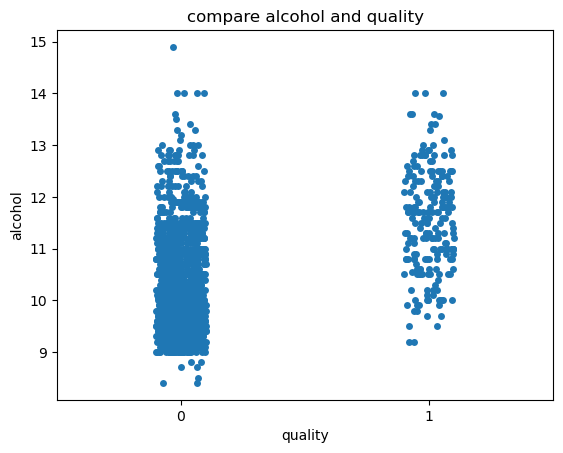
1. 10
   1. **Setting an arbitrary cutoff for the dependent variable**: We have set an arbitrary cutoff for the dependent variable (wine quality) at e.g. 7 or higher getting classified as 'good/1' and the remainder as 'not good/0'. This allows us to practice with hyper parameter tuning on decision tree algorithms looking at the ROC curve and the AUC value.

Quality Frequency

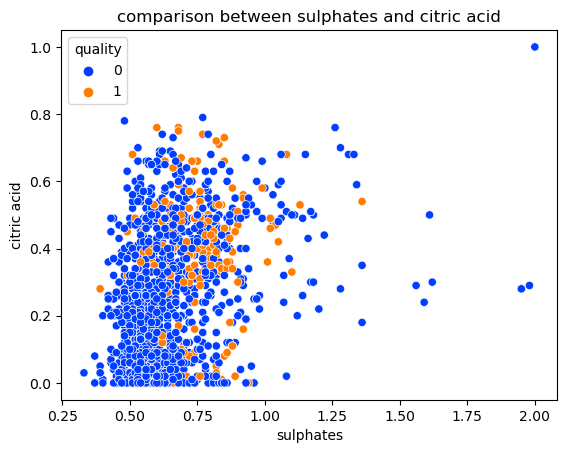
0 (not good) 1382

1 (good) 217

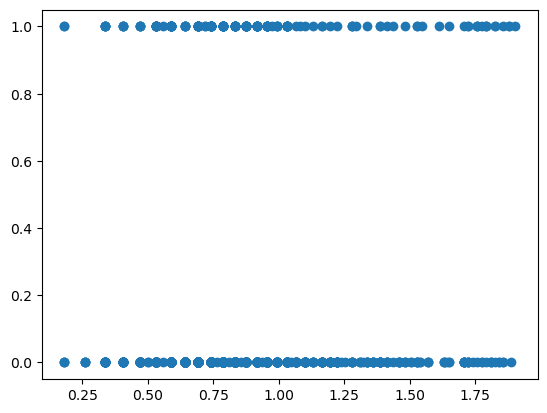
1. **Exploratory Data Analysis(EDA):** EDAis an approach to analysing the data using visual techniques. It is used to discover trends, and patterns, or to check assumptions with the help of statistical summaries and graphical representations.  Now let’s check the number of null values in the dataset columns wise.
   1. **Comparison between presence of alcohol and quality of wine**: Presence of alcohol shows a positive effect on quality of wine. That is more the amount of alcohol in the wine, more the quality of wine improves.



* 1. **Comparison between sulphates and citric acid in the quality of wine:** Presence sulphates and citric acid in wine is not having much impact on quality of wine.



* 1. **Impact of residual sugar on quality of wine**: Scatterplot shows that the residual sugar has least impact on the quality of wine.



1. **Preprocessing pipeline:** 
   1. **Outliers:** Boxplot showed almost all columns had outliers.
   2. **Skewness:** Histograms of each column to showed the distribution of data. Data of columns namely chloride, residual sugar, sulphur dioxide, alcohol, citric acid are negetively skewed. Whereas, the data in alcohol and sulphates are extremely skewed to the left.
   3. **Z-score:** To remove the outliers and skewness of the column we found out the Z-score of the rows and columns where the Z-score is more than +3 and less than -3. We deleted all such columns and rows. And got a new dataframe with 1458 rows and 12 columns.
   4. **Boxcox transformation**: Now to remove the remaining skewness in the data e did the boxcox transformation.
   5. **Correlations:** we have to find the correlation of all input variable amongst each other and with the output variable. It was found that 'alcohol' has highest correlation with 'quality'. 'Citric acid' and 'sulphates' are the next highest positively impacting factor on the quality after 'alcohol’.

'volatile acidity', 'total Sulphur dioxide' and ‘acidity’ are effecting negatively to the 'quality’. The table below shows the correlation of quality of wine with other variables.

**Variable Correlation with quality of wine**

quality 1.000000

alcohol 0.426751

sulphates 0.312522

citric acid 0.226809

fixed acidity 0.119643

residual sugar 0.096110

pH -0.079528

free sulfur dioxide -0.095752

chlorides -0.135898

density -0.154221

total sulfur dioxide -0.183569

volatile acidity -0.268236

1. **Machine learning model:**
   1. **Spliting the data into features and label**: Independent variables will regarded as X( features) as dependent variable will be regarded as y(label). Here all the columns other than the ‘quality’ column are regarded as X. Whereas, the ‘quality’ column is regarded as Y. As the aim of the project is to predict the quality of wine, so the ‘quality’ column is regarded as label or target.
   2. **Removing multicollinearity**: We found out the VIF (variance inflation factor) of X to remove multicollinearity.  VIF measures how much the variance of the estimated coefficients is inflated as compared to when the predictor variables are not linearly related. Columns with higher VIF and less correlation with the target are dropped. ‘Density’ and ‘pH’ columns were showing high VIF so we dropped them.
   3. **Balancing the target variable**: Value count of target variable Y showed that the classes were not balanced. So we have used the over sampling method to balance it. After the target classes are balance we can go further with modelling.

In an **imbalanced** dataset, the classes are not evenly distributed but rather highly skewed. Since one of the classes is overly frequent, it will affect the learning mechanism of the Machine Learning algorithm. That is why we balanced the dataset before going for Machine Learning.

* 1. **Finding the best Accuracy and maximum random state:** It was found that the model is showing 95% accuracy at a maximum random state of 17.
  2. **Train test split:** The train\_test\_split() method is used to split our data into train and test sets. First, we need to divide our data into features (X) and labels (y). The dataframe gets divided into X\_train,X\_test , y\_train and y\_test. X\_train and y\_train sets are used for training and fitting the model. The X\_test and y\_test sets are used for testing the model if it’s predicting the right outputs/labels. We can explicitly test the size of the train and test sets. Here we have split the dataset into 70% training dataset and the remaining 30% into test dataset.

x\_train,x\_test,y\_train,y\_test= train\_test\_split(x,y,test\_size=0.30,random\_state=17)

* 1. **Classification algorithms:** After splitting the dataset we have called all the following classification algorithms
* from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
* from sklearn.linear\_model import LogisticRegression
* from sklearn.svm import SVC
* from sklearn.ensemble import GradientBoostingClassifier, AdaBoostClassifier, BaggingClassifier
* from sklearn.metrics import classification\_report, confusion\_matrix, roc\_curve, accuracy\_score
* from sklearn.model\_selection import cross\_val\_score
  1. **Random Forest Classifier:** RandomForestClassifier showed 91% accuracy

0.9155251141552512

[[368 8]

[ 29 33]]

precision recall f1-score support

0 0.93 0.98 0.95 376

1 0.80 0.53 0.64 62

accuracy 0.92 438

macro avg 0.87 0.76 0.80 438

weighted avg 0.91 0.92 0.91 438

* 1. **Logistic regression:** Logistic regression showed 87% accuracy

0.8744292237442922

[[371 5]

[ 50 12]]

precision recall f1-score support

0 0.88 0.99 0.93 376

1 0.71 0.19 0.30 62

accuracy 0.87 438

macro avg 0.79 0.59 0.62 438

weighted avg 0.86 0.87 0.84 438

* 1. **Support vector classifier:** Support vector classifier showed 85% accuracy

0.8584474885844748

[[376 0]

[ 62 0]]

precision recall f1-score support

0 0.86 1.00 0.92 376

1 0.00 0.00 0.00 62

accuracy 0.86 438

macro avg 0.43 0.50 0.46 438

weighted avg 0.74 0.86 0.79 438

* 1. **Gradient Boosting Classifier:** Gradient boosting classifier showed 89% accuracy

0.8972602739726028

[[361 15]

[ 30 32]]

precision recall f1-score support

0 0.92 0.96 0.94 376

1 0.68 0.52 0.59 62

accuracy 0.90 438

macro avg 0.80 0.74 0.76 438

weighted avg 0.89 0.90 0.89 438

* 1. **Ada boost classifier:** Ada boost classifier showed 87% accuracy

0.8789954337899544

[[362 14]

[ 39 23]]

precision recall f1-score support

0 0.90 0.96 0.93 376

1 0.62 0.37 0.46 62

Accuracy 0.88 438

macro avg 0.76 0.67 0.70 438

weighted avg 0.86 0.88 0.87 438

* 1. **Bagging classifier:** Bagging classifier showed 90% accuracy

0.9018264840182648

[[364 12]

[ 31 31]]

precision recall f1-score support

0 0.92 0.97 0.94 376

1 0.72 0.50 0.59 62

Accuracy 0.90 438

macro avg 0.82 0.73 0.77 438

weighted avg 0.89 0.90 0.89 438

* 1. **Extra Trees Classifier:** Extra trees classifier showed 91% accuracy.

0.910958904109589

[[370 6]

[ 33 29]]

precision recall f1-score support

0 0.92 0.98 0.95 376

1 0.83 0.47 0.60 62

accuracy 0.91 438

macro avg 0.87 0.73 0.77 438

weighted avg 0.91 0.91 0.90 438

* 1. **Finding the CV (cross validation) score:** Cross-validation is a technique for evaluating a machine learning model and testing its performance. It helps to compare and select an appropriate model for the specific predictive modeling problem.

**Here Random forest classifier and ExtraTrees classifier both are showing almost equal performance.** In this case we tried to find out the CV score for both the models and find the difference between CV score and mean accuracy score of each model. The model showing the least difference between its CV score and mean accuracy score shows maximum accuracy.

In this case the Extra trees classifier showed the least difference. So we have selected ExtraTrees classifier as our predictive model for wine quality classification.

* 1. **Hyper parametric tuning:** Hyperparameter tuning is the process of selecting the optimal values for a machine learning model’s hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, random state, maximum depth, number of jobs, number of estimators etc. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

The best parameters of the model to get the best performance were:

Criterion: ‘gini’

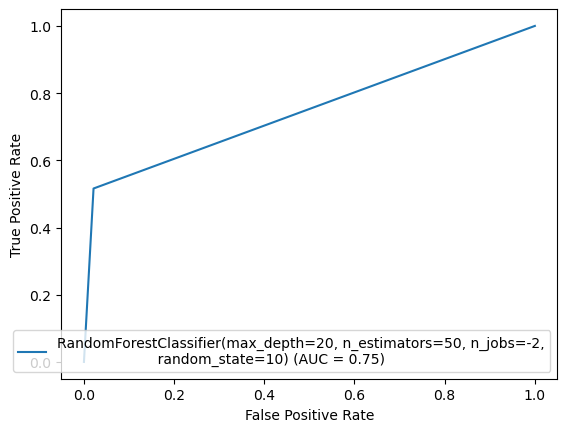
Maximum depth: 20

Number of estomators: 50

Number of jobs : -2

Random state: 10

* 1. **Final Model**: Now we have set the best parameters and will train the model to find the best accuracy in prediction. The Extra trees classifier with the above mentioned parameters showed 91% accuracy.
  2. **Testing the model:** We shall find the AUC-ROC curve to test the performance of the model. AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting.
  3. Saving the model and predicting: After the model showed a good performance and we had a satisfactory AUC-ROC graph, now it’s time to save the model in pickle library and set it for prediction.



* 1. **Prediction:** Now we can see the difference between original and predicted values.

1. **Concluding remarks:** With an aim to predict the quality of wine, we have used many machine learning algorithms namely Random forest classifier, Extratrees classifier, support vector classifier, adaboost classifier, bagging classifier. This allowed us to select the best suited algorithm to predict th equality of wine with the given variables with the maximum accuracy. In tuning the model parameters we have test the learning rate from low to high for the number of trees in the range 1 to 200. All the model’s performances mentioned above are evaluated using the confusion matrix, precision and f1 score.

The cross validation(CV) is used to select the best tuning parameter. Our work shows that among the various machine learning models, Extra trees classifier works best in predicting the quality of wine. It shows an accuracy of 91%. So we have used extra trees classifier in predicting the quality of wine in this dataset.

It can also be concluded that the presence of ‘Alcohol’ is the most important factor in determining the quality of wine. Higher the amount of alcohol present in the wine, better is the quality.