**Red Wine Quality Prediction**

**Problem Definition :**

According to Wikipedia , Wine is an alcoholic drink typically made from fermented grapes. Today the five countries with the largest wine-producing regions are in Italy, Spain, France, the United States and China. Red wine gains its colour and flavour from the grape skin by allowing the grapes to soak in the extracted juice.

The dataset is related to red and white variants of the Portuguese “Vinho Verde Wines”. Only physicochemical (inputs) and sensory (the output) variables are available.

**Data Analysis :**

The dataset contains following variables :

Data Analysis is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusions and support decision making. Analysis refers to dividing a whole into its separate components for individual examination. Data analysis is a process for obtaining raw data and subsequently converting it into information useful for decision-making by users.

The dataset contains following variables :

The input variables (based on physicochemical tests) :

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

Output variable (based on sensory data):

12 - quality (score between 0 and 10)

**Type:** This is a classification type of problem.

**Shape:** The dataset contain 1599 rows and 12 columns.

**Datatype :** Except quality which is int type all other variables are float type.

**Missing Values :**

Dataset must present complete values in the features, without any missing data. Having an incomplete example makes connecting all the signals within and between features impossible. Missing values also make it difficult for the algorithm to learn during training. Too many missing values render more uncertain predictions because missing information could conceal any possible figure, consequently, the more missing values in the features , the more variable and imprecise the predictions.

Our dataset does not contain any missing values.

**Statistical Calculation :**

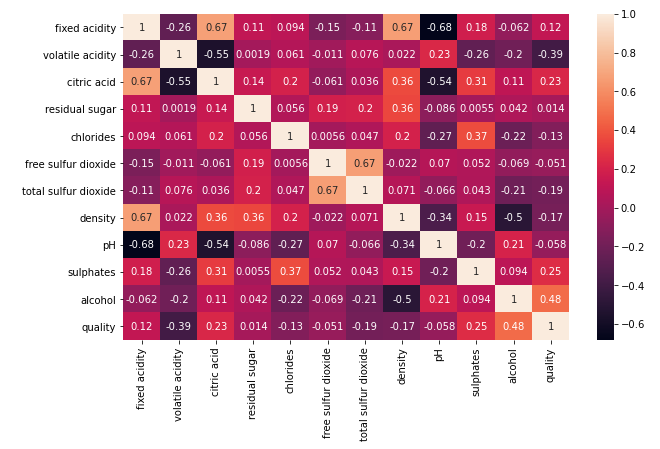
Outlier: Outliers are the values that are too high or too low or simply they are unusual and highly improbable. Outliers are a problem when learning from data . Because machines learn by observing examples and extracting rules and patterns from them, a weird case can prove difficult to understood and force the algorithm to reconsider what it has learned so far. A value that exists for outside the expected range will have a huge deviance from the expected value, leading the learning process to adapt to the anomaly abruptly by undervaluing the regular values (which instaed produce comparatively slight deviations).

In our dataset there is large difference in 75th percentile and max value in residual sugar, free sulfur dioxide, total sulfur dioxide. This suggests that there are extreme outliers present in these three columns.

**Checking Correlation :**

Correlation: Correlation in statistics is a measure ranging from +1 to -1 that tells how two variables relate linearly (that is, if you plot them together, they tell how lines resemble each other)

Alcohol, sulphates, citric acid, fixed acidity, residual sugar show positive correlation with column quality. Other features are negatively correlated with it.



**Exploratory Data Analysis [EDA] :**

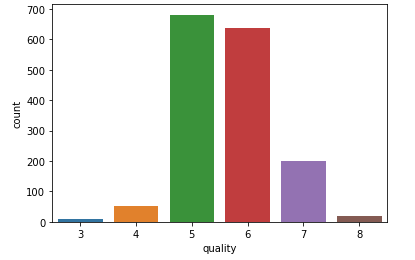
Exploratory data analysis is an approach of [analyzing](https://en.wikipedia.org/wiki/Data_analysis" \o "Data analysis) [data sets](https://en.wikipedia.org/wiki/Data_set) to summarize their main characteristics, often using [statistical graphics](https://en.wikipedia.org/wiki/Statistical_graphics) and other [data visualization](https://en.wikipedia.org/wiki/Data_visualization) methods.

1.The target variable contains numbers ranging from 3 to 8 which indicate quality of wine

Therefore we will check the count of each number.

Unique values of “quality” are plotted using countplot.

Countplot clearly shows counts at 5 and 6 to be larger which indicate not so good wine.



2.Bivariate Analysis :

Scatterplot is used for the bivariate analysis , each input variable is plotted against the target variable which is “quality”. Following are the results obtained from the scatterplot:

fixed acidity : Composition of fixed acidity is at peak at quality score 7

volatile acidity : As the quality score increases volatile acidity composition decreases

citric acid : As the quality score increases citric acid composition increases

chlorides : As quality increases composition of chlorides decreases.

free sulfur dioxide : free sulfur dioxide increases till 5 and then decreses as the quality increases

total sulfur dioxide : total sulfur dioxide composition is at peak at 5 and then decreases as the quality increases

density : After quality score 5 density decreases as quality increases

pH : pH decreases as quality increases except at 6

sulphates : composition of sulphates increases as the quality increases

alcohol : composition of alcohol drastically increases after score 5

3.Box Plots :

Box plots showing the outliers are plotted. All the boxplots shows the presence of outliers

Distribution plots :

The distribution plot shows how is the data distributed , is it normally distributed or is it skewed . Following are the results of the distribution plots plotted of each variable.

fixed acidity is positively skewed.

volatile acidity is nearly normally distributed (Bimodal).

citric acid has Multimodal distribution.

residual sugar is highly positive skewed.

fixed acidity is slightly positively skewed.

chlorides is highly positively skewed

free sulfur dioxide is positively skewed.

total sulfur dioxide is positively skewed.

density is normally distributed

pH is normally distributed

sulphates is positively skewed

alcohol is slightly positively skewed

quality has multimodal normal distribution

**Pre-processing :-**

No matter the level of sophistication of the learning algorithm , if you don’t prepare foundation well i.e. your data –your algorithm won’t last long when tested in real data situations. According to the principle of garbage in, garbage out (GIGO for short) bad data can truly harm machine learning. Bad data consists of missing data, outliers, skewed value distribution, redundancy of information and features not well explicated.

**Categorical Division:-**

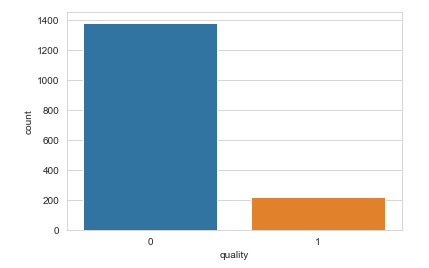
Dividing wine as good and bad by giving the limit for the quality from 2 to 6.5 it is considered as bad and 6.5 to 8 as good.

**Encoding:-**

‘quality’ variable is encoded as 0 and 1 for bad and good wine using label encoder.

**Class Imbalance:-**

There is a clear imbalance of class ‘quality’ which can be seen when the count plot is plotted. After the input and the target variables are separated the class is balanced by SMOT technique using oversampling .



**Outlier Removal :-**

Outliers are removed using zscore method . Total 141 rows were removed new shape of the dataset become (1458,12).

**Scaling :-**

MinMax scaler is used to scale the data which range from 0 to 1.

**Building machine learning model :-**

After the data is ordered and arranged properly now we can build our model. As this is a classification type of problem we will choose our algorithm accordingly.

1. **Finding best random state** :-

When tested best accuracy of 0.8518 is obtained for random state 13.

1. **Train-Test Split** :-

Here data is split into training data and testing data using the train-test split method. So here we have set the testing data to be 30% along with the random state to be 13.

1. **Model Selection** :-

For building the model five classification models used are as follows :-

**Logistic Regression:** Solves the problem by estimating a probability of being in a particular class.

**KNeighbors Classifier:** The algorithm finds the most similar observations to the one you have to predict and from which you derive a good intuition of the possible answer by averaging the neighboring values, or by picking the most frequent answer class among them.

The learning strategy in a KNN is more like memorization. It’s just like remembering

what the answer should be when the question has certain characteristics (based on circumstances or past examples) rather than really knowing the answer, because you understand the question by means of specific classification rules. In a sense, KNN is often defined as a lazy algorithm because no real learning is done at the training time, just data recording.

**Decision Tree** Classifier : Using a sample of observations as a starting point, the algorithm retraces the rules that generated the output classes (or the numeric values when working

through a regression problem) by dividing the input matrix into smaller and smaller partitions until the process triggers a rule for stopping. Such retracing from particular toward general rules is typical of human inverse deduction, as treated by logic and philosophy. In a machine learning context, such inverse reasoning is achieved by applying a search among all the possible ways to split the training in-sample and decide, in a greedy way, to use the split that maximizes statistical measurements on the resulting partitions.

**Ensemble Techique**

**Random Forest Classifier :** Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time. For classification tasks, the output of the random forest is the class selected by most trees. Random Forest is a classification (naturally multiclass) and regression algorithm that uses a large number of decision tree models built on different sets of bootstrapped examples and subsampled features.

**AdaBoost Classifier :** AdaBoost, short for Adaptive [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm)), is a [statistical classification](https://en.wikipedia.org/wiki/Statistical_classification) [meta-algorithm](https://en.wikipedia.org/wiki/Meta-algorithm). It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier.  AdaBoost training process selects only those features known to improve the predictive power of the model, reducing dimensionality and potentially improving execution time as irrelevant features don't need to be computed.

Accuracy Score and precision, recall f1 score are calculated for these models . Out of which accuracy score for respective models are :-

Logistic Regression : 0.8518

KNeighbors Classifier : 0.8903

Decision TreeClassifier : 0.8807

Random Forest Classifier : 0.9409

AdaBoost Classifier : 0.8855

**4. Cross Validation :**

A noticeable problem with the train/test set split is that you’re actually introducing bias into your testing because you’re reducing the size of your in-sample training data. When you split your data, you may be actually keeping some useful examples out of training. Moreover, sometimes your data is so complex that a test set, though apparently similar to the training set, is not really similar because combinations of values are different (which is typical of highly dimensional datasets). These issues add to the instability of sampling results when you don’t have many examples. There is a chance of overfitting in the above obtained accuracy . Therefore we need to check overfitting using cross validation score. The scores obtained are as follows :

1. Logistic Regression : 0.8674

2. KNeighbors Classifier : 0.8449

3. Decision TreeClassifier : 0.8042

4. Random Forest Classifier : 0.8680

5. AdaBoost Classifier : 0.8617

**5. Best Model :**

Observing the accuracy score and cross validation , we can choose our best model. As the least difference between accuracy and cross validation score is for logistic regression , we will choose it as our best model.

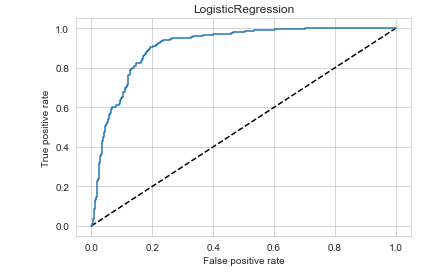
**6. Hyperparameter Tuning :**

Now we need to tune the parameters of our best model . The final source of performance derives from fine-tuning the algorithm’s hyperparameters, which are the parameters that you decide before learning happens and that aren’t learned from data. The Best parameters for hyperparameter tuning of logistic regression are obtained and the model is hyper tuned which gives final accuracy to be 84.33% .

**7. AUC-ROC Curve:**

The ROC curve is created by plotting the [true positive rate](https://en.wikipedia.org/wiki/True_positive_rate) (TPR) against the [false positive rate](https://en.wikipedia.org/wiki/False_positive_rate) (FPR) at various threshold settings. The true-positive rate is also known as [sensitivity](https://en.wikipedia.org/wiki/Sensitivity_(tests)), [recall](https://en.wikipedia.org/wiki/Precision_and_recall#Definition_(classification_context)) or probability of detection[[9]](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#cite_note-matlab-9) in [machine learning](https://en.wikipedia.org/wiki/Machine_learning). The false-positive rate is also known as probability of false alarm[[9]](https://en.wikipedia.org/wiki/Receiver_operating_characteristic" \l "cite_note-matlab-9) and can be calculated as (1 − [specificity](https://en.wikipedia.org/wiki/Specificity_(tests))).

The AUC score comes out to be 0.8516.



8. Finally the model is saved.

**Concluding Remarks:**

1. The dataset contains more amount of “5” and “6” quality i.e. not so good type of wine.
2. Logistic Regression is the best model for our prediction.
3. Final accuracy comes out to be 85%.

**References:**

1.Machine Learning for dummies by John Paul Muellerand and Luca Massaron

2. Wikipedia