**Machine Learning Classification Model**

# (Red Wine Quality Prediction Problem)

Software used: Jupyter Notebook

Language: Python

Author:Vipin Kumar



**Table of Contents:**

1. Problem Definition.

2. Data Analysis.

3. EDA Concluding Remark.

4. Pre-Processing Pipeline.

5. Building Machine Learning Models.

6. Concluding Remarks.

**Problem Definition**

Red wine quality prediction is all about to classify the quality of wine with the help of used ingredients in its preparation. Data set is related to red and white variants of the Portuguese "Vinho Verde" wine.

Here we have a dataset which include 11 features and on the basis of these parameters we have to find out quality of wine. Target label have score from 1 to 9, so we have to convert its value in binary classification. In this problem statement cutoff given for your dependent variable (wine quality) 7 or higher getting classified as 'good/1' and the remainder as 'not good/0'.

Objective: Here, with the help of given ingredients/features, we would be able to predict the quality of wine and classify it whether it’s good or not good.

Problem: The above problem statement clearly explains that the target variable is categorical & but labels given as score from 1 to 10 so have convert them in binary number according to given cutoff.

So, to solve this problem I will use mentioned below classification Machine learning algorithm.

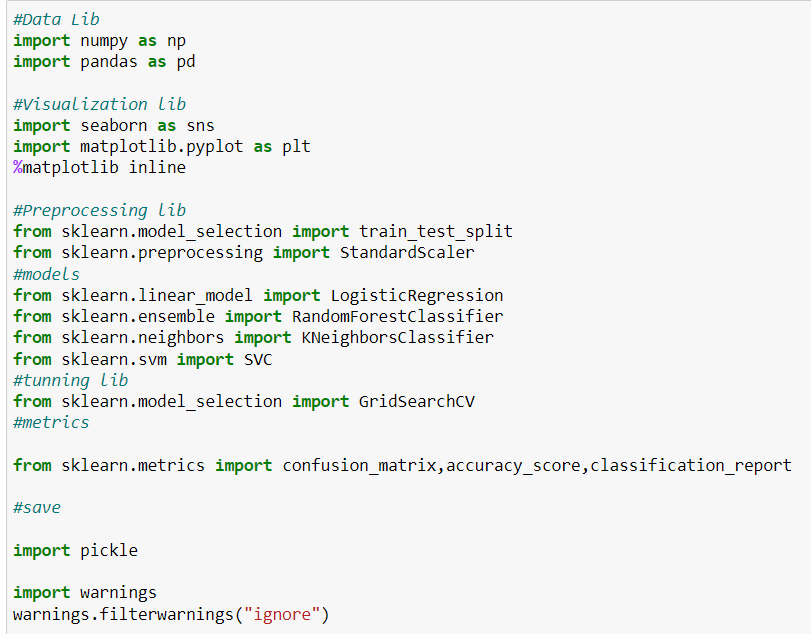
1-LogisticRegression

2-RandomForestClassifier

3-KNeighborsClassifier

4-SVC (Support vector Machine)

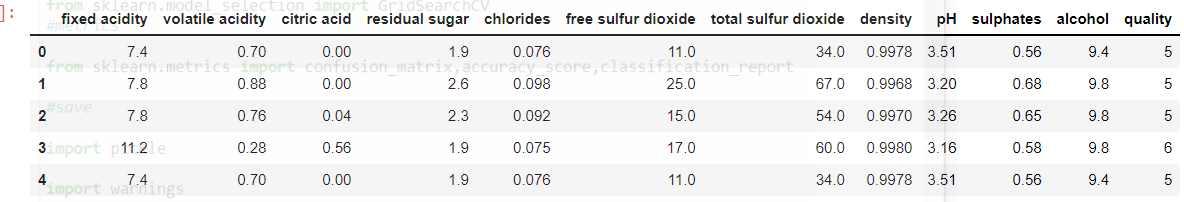
**Importing Libraries:**



**Data Analysis**

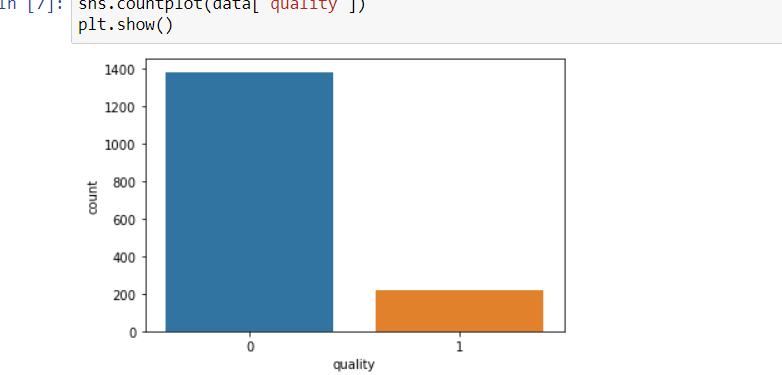
In this dataset we have 11 features on basis of that we have to prepare our classification model. Here we should have basic knowledge about these features so here I have mentioned all features and high level information.

* **volatile acidity :**   Volatile acidity*is the*gaseous acids present in wine.
* **fixed acidity :**Primary fixed acids found in wine are tartaric, succinic, citric, and malic
* **residual sugar :**Amount of sugar left after fermentation.
* **citric acid :** It is weak organic acid, found in citrus fruits naturally.
* **chlorides :**Amount of salt present in wine.
* **free sulfur dioxide :**   So2 is used for prevention of wine by oxidation and microbial spoilage.
* **total sulfur dioxide:** total sulfur dioxide
* **pH :**In wine pH is used for checking acidity
* **density:** density
* **sulphates**:    Added sulfites preserve freshness and protect wine from oxidation, and bacteria.
* **alcohol :**   Percent of alcohol present in wine.



* Dataset have 1599 rows & 12 columns.

**Data Imbalance**: In dataset we have convert quality labels in binary (0-1), so its important part to check in any classification that class should not be imbalance between classes.

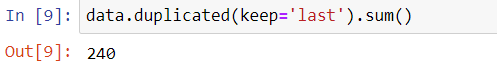


It’s clearly showing class imbalance problem **0 category %: 14 & 1 category %: 86.**

So in this scenario we have to have to up sample to balance both classes.

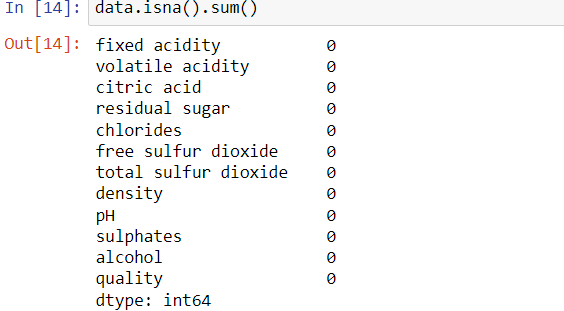
Here I will use resample technique for up sampling.

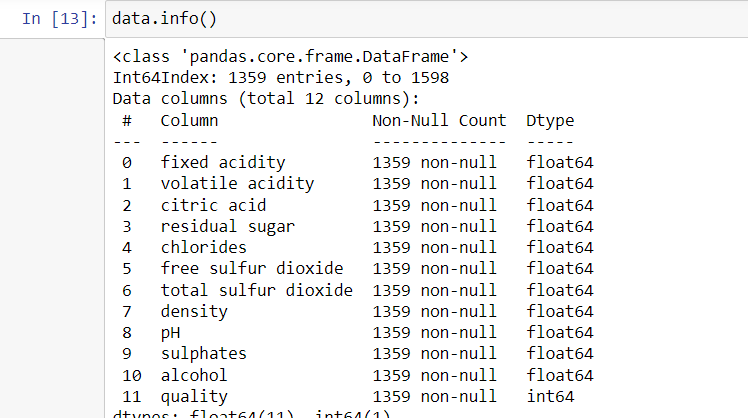
**Duplicate data Test:** we must check duplicate observations in any machine learning model because model have to iterate twice at trading, Machine learning algorithm would lead to consume more learning time.



Here we can see dataset have 240 duplicate values, so we have to remove all these duplicates.

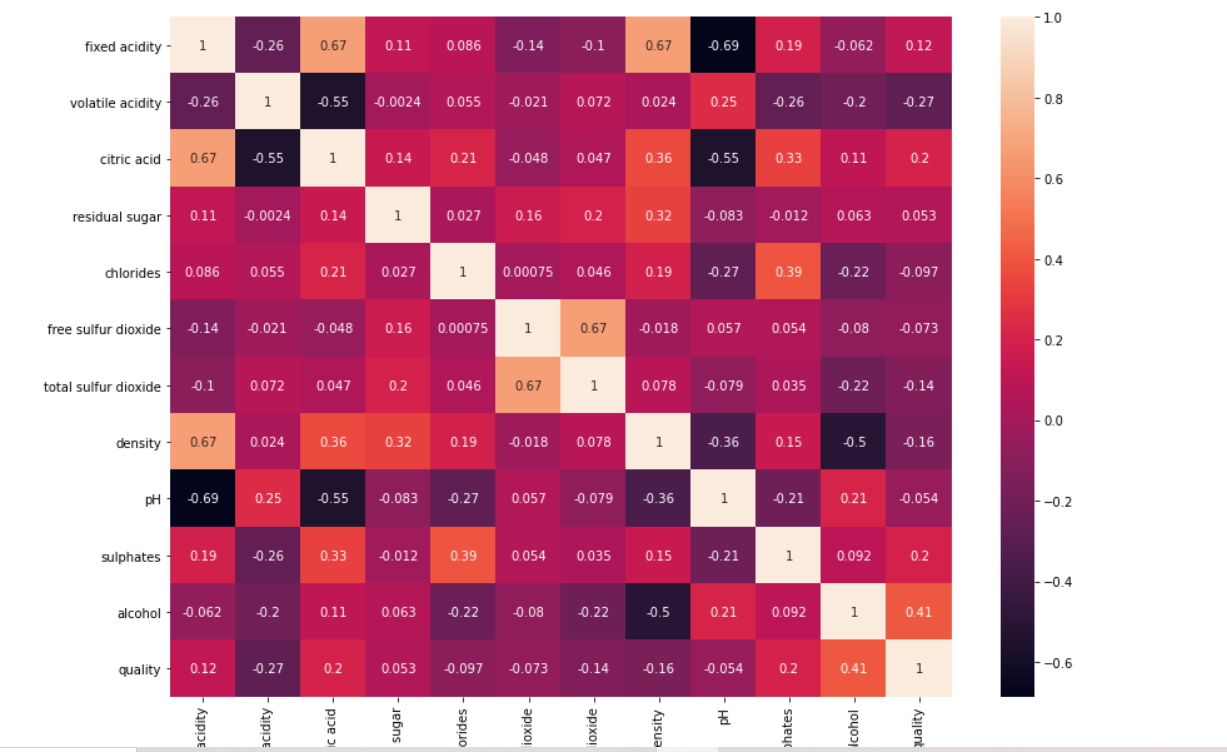
**Categorical & null Testing**





In this dataset we can see that all features are numeric and no null (blank) in dataset.

**Multicollinearity & Correlation using heat map**



With the help of above heat map we can see no multicollinearity found amongst all features. target quality have positive & negative correlation with all features but in few features its not very strong. We will apply feature selection to pick best features.

**EDA Concluding Remark**

* The dataset doesn’t have any **null values and all features are numeric**.
* Target quality have positive & negative correlation with all features but in few features its not very strong.
* The strongest **positive correlation** with the target features are: alcohol,

citric acid and sulphates.

* The strongest **negative correlation** with the target features are: volatile acidity and

density.

* Dataset have classes **imbalanced problem** with majority of observations describing

quality as not good.

* Dataset have **240 duplicates** observations which supposed to remove.
* No Multicollinearity found among features.
* Features-(residual sugar,free sulpher diaoxide,ph) have very less correlation with target.

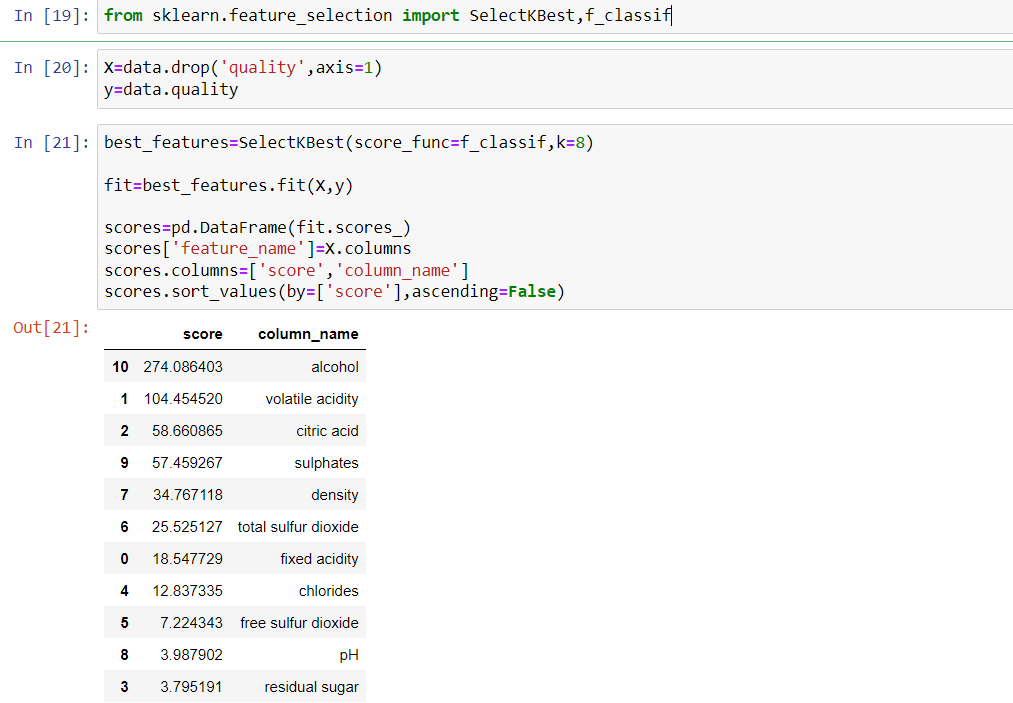
**Pre-Processing Pipeline**

**Feature Selection:** As we have seen in EDA part that few features having very less correlation with the target : quality.

So its best practice to use best features only for prediction here I will use

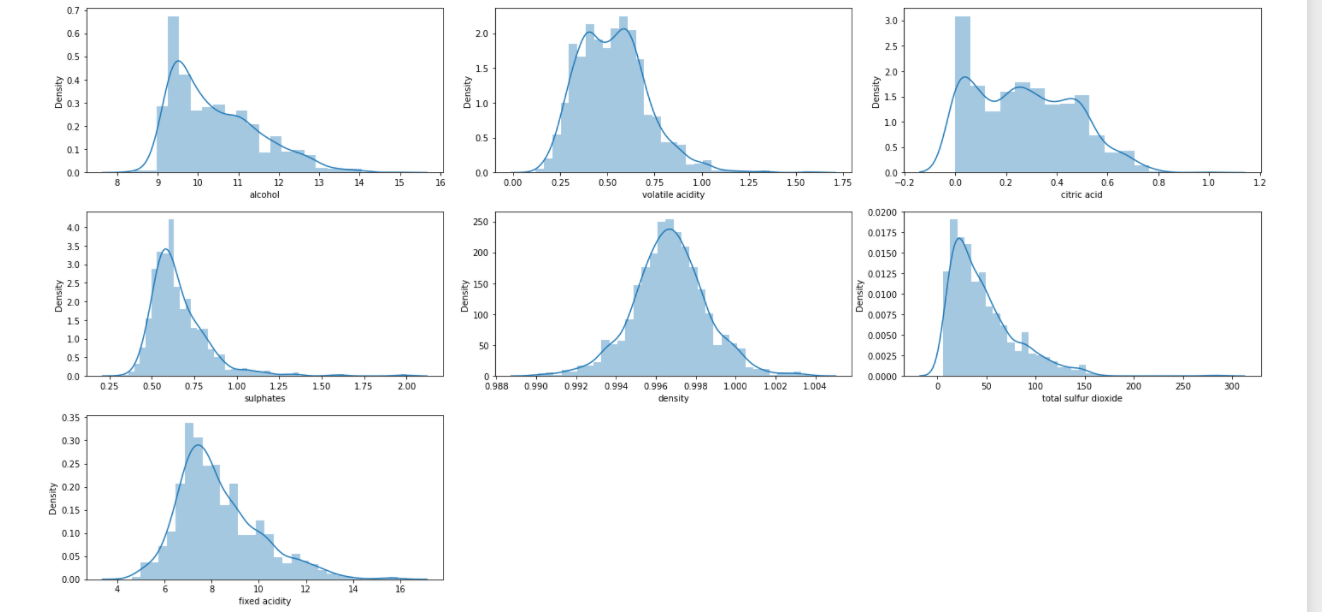
**SelectKBest** to select best features.

Scikit-learn API provides SelectKBest class for extracting best features of given dataset. The SelectKBest method selects the features according to the k highest score. Selecting best features is important process when we prepare a large dataset for training. It helps us to eliminate less important part of the data and reduce a training time.



Here we got the score against each feature and we can select n numbers of features with highest score, I will select top 7 features to build my model.

**Skewness/Outliers:**

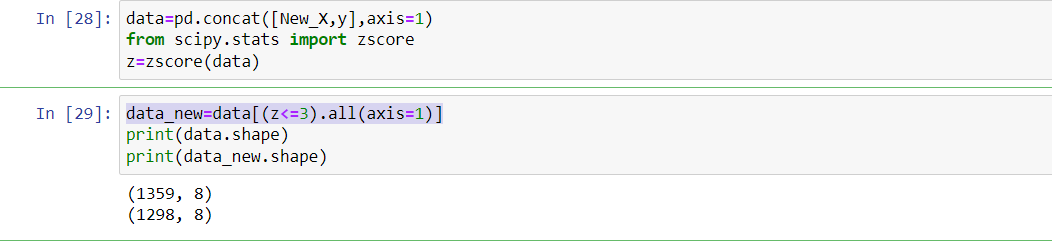


Here we observed some skewness mostly rightly skewed so I will apply **zscore** to remove it till some extent.

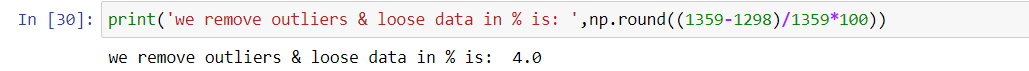
Z score is also called standard score. This score helps to understand if a data value is greater or smaller than mean and how far away it is from the mean. More specifically, Z score tells how many standard deviations away a data point is from the mean.

If the z score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be an outlier.

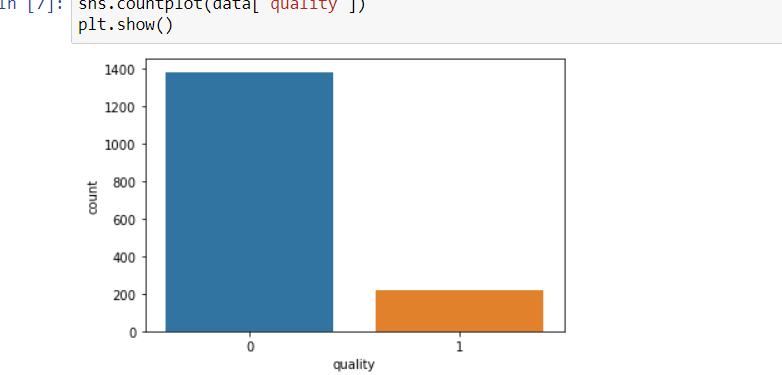
**Z score = (x -mean) / std. deviation**



Here we applied zscore to remove outliers and reduce skewness, we select only those observations in which zscore is less then 3. By apply this we have to loss some observations around 4%.



**Data Sampling:** During EDA part part we observed that we have class imbalance in out dataset so here I will apply sklearn up sampling technique resample to balance both classes.





In this dataset we have only 176 observation which classified as 1 and 1122 observations which classified as 0. So I upsample class 1 equal to class 0 i.e. 1122

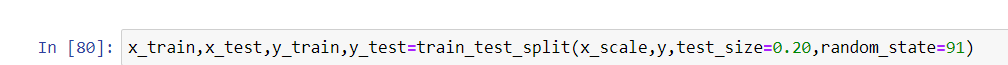
Now we have total 2244 observations and both class have balanced now.

**Feature Scaling:** Feature scaling is very important part before training a dataset. Because most of the time we got data in which different features have different numeric scale(for e.g. age range (18-60), salary(10000-90000). So feature scaling technique standardize all features in common scale. Here I will apply **Standard Scaler** for features scaling. As we do scaling part on features only so before applying scaler I have split dataset in X-independent features, y-Dependent Target.



## **Splitting data into training and testing sets**: Before applying machine learning machine algorithm we must have to split our master data in to 2 parts. First part is training data and second part is testing data. We train our model on train set and then evaluate model accuracy using testing set. We have select percentage of data for training and percentage of data for testing.

Here I have done it using sklearn feature Train Test Split

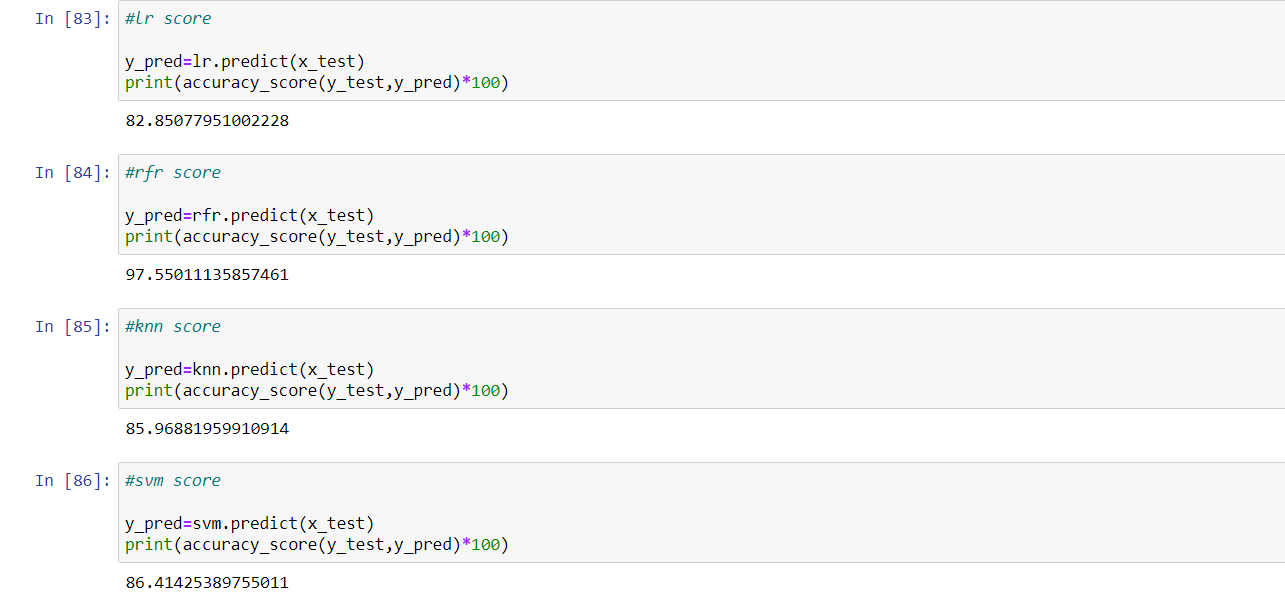
**Train Test Split**

# Building Machine Learning Models

# As our dataset ready now & here we finalize one suitable model which will give more accuracy, here we will use RandomForestClassifier, LogisticRegression, KNeighborsClassifier, SVC and then evaluate one by one.

# 

Here we have trained all 4 algorithms on our data set, now we can test accuracy score one by one.

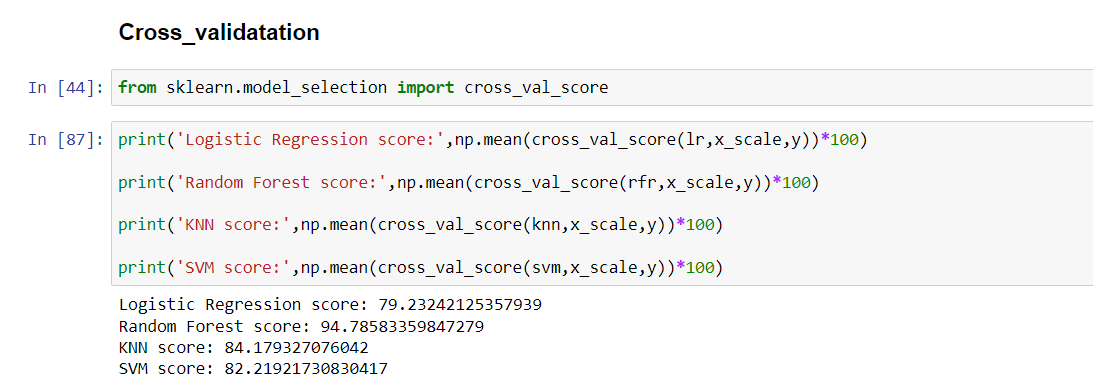


As you can see we have tested accuracy score for all 4 algorithms and find that Random Forest classifier predicting with highest accuracy that is 97.55%.

Before finalize our model we have to cross validate to all algorithms.

**Cross Validation**: In machine learning, we couldn’t fit the model on the training data and can’t say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. For this purpose, we use the cross-validation technique.

In cross validation we train our model on subsets of data as we can make n number of subsets and train all subset. I will apply cross\_val\_score here to cross validate all estimators. The best model is that which accuracy score highest and very close to cross val score.



As we havecross validate all estimators one by one using cross\_val\_score, we took default k value, k is number of subsets of data we want to create and do validation. cross\_val\_score by default take k value 5.

We can see in validation algorithm Random Forest classifier score is highest and its too close to accuracy which we got on testing data.

So we can finalize our model **random forest** which performing best on this problem statement.

However our model performing too well but a data scientist never satisfied and look for more accuracy if possible. So will try hyper parameter tuning to increase accuracy if possible.

### **Hyper parameter Tuning:** All machine learning algorithms have their default parameters but we can tune parameters to select best parameters which are best suitable for that particular problem statement.

### Most of the time we use these 2 tuning techniques in machine learning

### 1-GridSearchCV:Test model on all given parameters combinations and find the best.

### 2-RandomSearchCV :pick randomly combinations from given parameters and find the best.

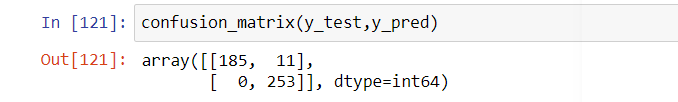
Here I will apply GridSearchCV for hypermeter tuning

### 

**I applied GridSearchCV to tune 4 parameters but getting almost same accuracy score. As accuracy score is already around 98%.**

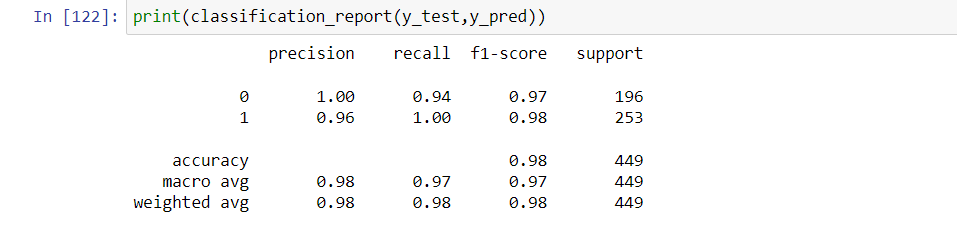
**Final Scoring:** Here we will check final scoring on our selected model.

**Confusion Matrix**



In confusion matrix we can see model predicting with 100% accuracy in True positive.

**Classification Report**



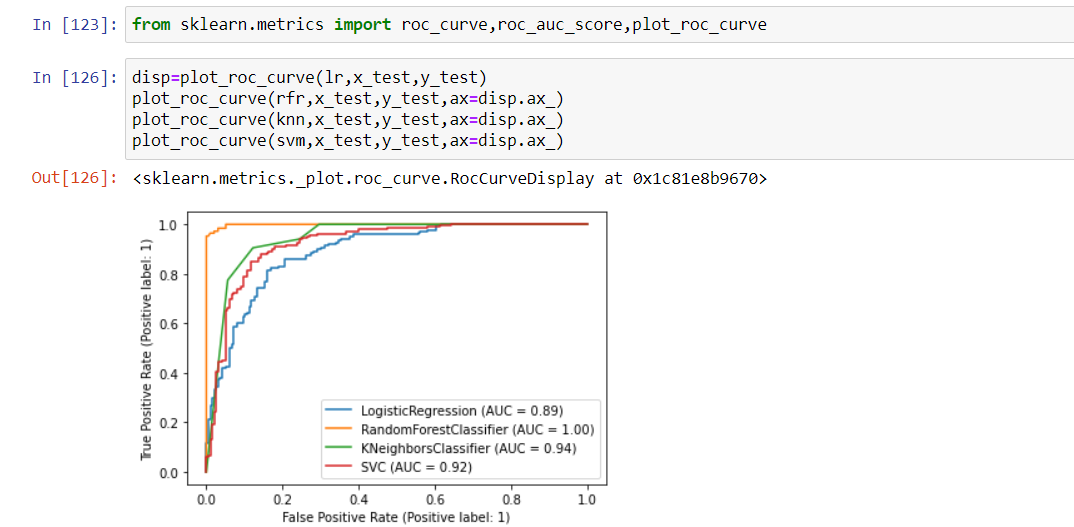
Precision for class 0 is 100% and class is 96%.

Recall for class 0 is 94% and class is 100%

F1-score for class 0 is 97% and class is 98%

Accuracy-98%

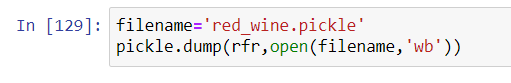
**AUC-ROC**



In roc plot we can see AUC score for random forest is best among all i.e. 100%

**Model Save**

Last part of any model but not least to save our finalize model.



**Concluding Remarks**

As we have seen, Random forest classification algorithm predicting best amongst all models we tried and test, cross validation score is around 95% of this algorithm which means the model predicted correctly & this could help in prediction of quality of wine. The governing body which do testing of red wines and provide approval to sell it in the market can use this model.

This model can be helpful for those Companies which produce & running wine factory can test quality of their wine before start producing it in bulk level. Hence by using Machine learning techniques we are able to solve this problem. Thanks.