**MINI PROJECT REPORT**

**Prediction of Wine Quality**



NN

Submitted By

**Name-**

**Satwik**

**Varun**

**Sanyam**

**Roll No-**

**1803210134**

**1803210175**

**1803210132**

**SUBMITTED TO:**

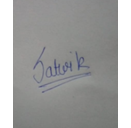
**Department of Computer Science & Engineering**

ABES ENGINEERING COLLEGE ,GHAZIABAD

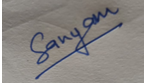
**DECLARATION**

I hereby declare that the Report entitled "Prediction Of Wine Quality" is an authentic record of my own work as requirements of Mini Project during the period from **15th May 2020** to

**30th June 2020** for the award of degree of B.Tech. (Computer Science & Engineering), ABES Engineering College, Ghaziabad, under the guidance of **Mr Gopal Gupta** and **Mr.Shashank Shekhar**







**Date: 4 December 2020**

**(Signature of student)**

***Name- Roll No-***

***Satwik 1803210134***

***Varun Rana 1803210175***

**Sanyam Jain 1803210132**

**ACKNOWLEDGEMENT**

*It gives us a great sense of pleasure to present the report of the Project Based Internship 2020 undertaken during Data analytics and Machine learning in year 2020. We owe special debt of gratitude to Gopal Gupta,* ***Data Ritz*** *Technologies for his constant support and guidance throughout the course of our work. His constant motivation have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.*

*We also take the opportunity to acknowledge the contribution of team members of* ***DataRitz*** *Technologies for their full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the motivation of Computer Science and Engineering(CSE) department in ABESEC to provide us the opportunity to undergo training at* ***DataRitz*** *Technologies.*

(iii)

**TABLE OF CONTENTS**

LIST OF TABLES (iv)

LIST OF FIGURES (v)

**CHAPTER-1INTRODUCTION** 1

1.1 Problem Definition 1

1.2 Motivation

1.3 Objective of the Project 2

1.4 Scope of the Project 2

1.5 Need of Work 3

**CHAPTER 2 RELATED WORK** 4

**CHAPTER 3 PROPOSED METHODOLOGY** 7

3.1 Dataset Description 7

3.2 Methods 11

3.3 Hardware / Software Requirements 19

3.4 Our Methodology 20

**CHAPTER 4 EXPERIMENT AND RESULT ANALYSIS** 35

**CHAPTER 5 CONCLUSION** 39

5.1 Discussion 39

5.2 Future Work 39

**REFERENCES** 40

(iv)

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **DESCRIPTION** | **PAGE NO.** |
| Table 3.1 | Dataset | 8 |
| Table 3.2 | Data description Table | 10 |
| Table 4.1 | Logistic Regression Result  Table | 36 |
| Table 4.2 | Decision Tree Model Result Table | 37 |
| Table 4.3 | Random Forest Model Result Table | 38 |

(V)

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **DESCRIPTION** | **PAGE NO.** |
| Fig 3.1 | Classification of Data | 24 |
| Fig 3.2 | Bar plot for Categorical Data | 25 |
| Fig 3.3 | Pi Chart for Categorical Data | 26 |
| Fig 3.4 | Count plot of quality attribute | 27 |
| Fig 3.5 | Citric Acid vs. Quality bar plot | 28 |
| Fig 3.6 | Chlorides vs. Quality bar plot | 29 |
| Fig 3.7 | Sulphates vs. Quality bar plot | 30 |
| Fig 3.8 | Dist plot histogram for univariate analysis | 31 |
| Fig 3.9 | Pair plot for bivariate analysis | 32 |
| Fig 3.10 | Box Plot for Outliers | 32 |
| Fig 3.11 | Correlation Matrix | 34 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Problem definition:**

About wine: • Wine is a beverage made from fermented grape and other fruit juices with lower amount of alcohol content.

• Quality of wine is graded based on the taste of wine and vintage. This process is time taking, costly and not efficient.

• A wine itself includes different parameters like fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide, density, pH, sulphates, alcohol and quality.

• Problem Statement : In industries, understanding the demands of wine safety testing can be a complex task for the laboratory with numerous analytes and residues to monitor. • But, our application’s prediction, provide ideal solutions for the analysis of wine, which will make this whole process efficient and cheaper with less human interaction.

* 1. **Motivation:**

In recent years there is a modest increase in the wine consumption as it has been found that wine consumption has a positive correlation to the heart rate variability [1]. With the increase in the consumption wine industries are looking for alternatives to produce good quality wine at less cost. Different wines have different purposes. Although most of the chemicals are same for different type of wine based on the chemical tests, the quantity of each chemicals have different level of concentration for different type of wine. These days it is really important to classify different wine for quality assurance [2]. In the past due to lack of technological resources it become difficult for most of the industries to classify the wines based on the chemical analysis as it takes lot of time and also need more money. These days with the advent of the machine learning techniques it is possible to classify the wines as well as it is possible to figure out the importance of each chemical analysis parameters in the wine and which one to ignore for reduction of cost. The performance comparison with different feature sets will also help to classify it in a more distinctive way.

* 1. **Objective of the project:**

Our main objective is to predict the wine quality using machine learning through Python programming language and to determine which features are the most indicative of a good quality wine.

• A large dataset is considered and wine quality is modelled to analyse the quality of wine through different parameters like fixed acidity, volatile acidity etc.

• All these parameters will be analysed through Machine Learning algorithms like random forest classifier algorithm which will helps to rate the wine on scale 1 - 10 or bad - good.

• Output obtained would further be checked for correctness and model will be optimized accordingly.

* 1. **Scope of Project :**

The scope of this analysis is to understand relationship of various parameters which

impact the quality ratings for both red and white wine. The dataset utilized for the analysis is downloaded from UCI repository.

We also propose of enhancing the results of project by studying more variables involved and their interrelationship in the near future. The project also gives certain insights on different inter-related factors which determine the levels of improvements that can be made for understanding the wine industry with focus on other aspects or physiochemical properties of wine. The future objective could be directly or indirectly related with improving the quality of wine for health benefits.

LIMITATIONS OF THE PROJECT :Our present work is limited to exploring the factors related to composition of red wine and white wine. Time and cost factors were also responsible for limiting the boundaries of the current research. Most of the data had been collected from open ended secondary sources which are freely available over the internet.

The current Project can have numerous implications mostly related to how global wine manufacturing and trade and can be conducted in more efficientlyand effectively in this current age of cut throat competition. Further, the studies of the project can be incorporated for social benefits limiting the possibility of gaining higher demand in the market, higher growth in terms of market share, higher productivity and improved overall performance.

**1.5 . Need of Work :**

The need of this project is that wine is used for preventing heart disease and stroke, preventing decline of thinking skills in later life,diabetes, preventing a digestive tract infection that can lead to ulcers, and many other conditions.So getting knowledge of different compositions of chemical used in a wine will be very helpful in medical fields to develop medicines accordingly for any cure of disease.

There are a few wineries, gin factories and rum factories around cities. They would employ lab technicians who would take samples of alcohol. However, I’d be astonished if they knew that prediction and inference of those measures were possible to determine quality of the product.

The idea that they could improve the quality of their product through predictive modelling I’d suggest would be entirely new to them. It’s really going a step beyond simple measures of quality control**.**

**CHAPTER 2**

**RELATED WORK**

Various researches and students have published related works in national and international research papers, theses to know the objectives, types of algorithms used and various techniques to be processed in advance.

The College of Science Intelligence and Engineering, China has written a paper on Model Evaluation and Analysis of Wine Quality Based on Mathematical Models. They used various mathematical tests to predict the quality of the wine. Tests Mann-Whitney U test is used to analyze the results of the evaluation of wine and two wine tasters, and found that a significant difference between the two. This paper then uses the Cronbach Alpha coefficient method to analyze the credibility of both groups of data.

**Paulo Cortez, Juliana Teixeira, António CerdeiraFernando AlmeidaTelmo MatosJosé Reis** wrote a paper on wine quality Assesment using Data Mining techniques. In this paper, he proposes a data mining approach to predict wine selection based on analytical tests easily during the certification step. A large conference was presented with examples of white vinho verde from Portugal's Minho region. The quality of the wine is modeled according to the regression approach, which maintains the order of the titles. 95% accuracy was obtained using this data mining technique.

This paper was conducted in the International Journal of Intelligent Systems and Applications in Engineering and this paper was published on September 3, 2016. The main purpose of this research paper was to predict the quality of the grapes based on physicochemical data. In this study, two separate sets of data were obtained from the UC Irvine Machine Learning Republic. They were successfully classified as red and white wines with an accuracy of 99.5229% using the Random Forests Algorithm.

**Y. Subba Reddy et al.**They introduced a user-centric framework of similarity in which product similarity it is evaluated by user preferences. A popular dataset called "Quality of red wine" is considered in this work to evaluate the quality of the wine grouping the individual products in groups and then rate groups based in preferences. The user-centric approach provided quite different and interesting results than the conventional approaches have, they don't consider the preferences that customers have expressed. It is observed that the types of queries introduced in this work they need higher execution times than the number of users and Preferences increased. This can lead to scalability problem of the proposed framework. This could be smoothing by introducing R-tree data structures for search and indexing purposes that can optimize The execution time of the proposed framework.The purpose of developing this type of system is support and advise wine users for a better selection and winemakers for providing better quality.It presents a critical review of research trends in wine.User-centric quality and similarity measures as well. A new user-centered similarity measure on the product Clustering is proposed to evaluate popular Wine data. dataset named Red Wine. The experimental results obtained in this work are able to provide better recommendations to product buyers than existing ones systems. The proposed approach is competent to group the red wine dataset in ordered groups of preferred wine variants and you can judge the quality of the wine based on these groups of user preferences. The proposed process groups the records from the wine dataset in groups based on priorities. Data grouped using classification forms a model to assign test data records with a recommended voting label. Most of Previous research on wine data limited to normal grouping and classification approaches depending on the taster detection data while the proposed novel hybrid approach can recommend the user a better wine combination without relying on taster detection data.

**P. Appalasamy et al.** discussed about modeling the Complex human taste is an important focus in wine industries. The main objective of this study was predict the quality of wine based on physicochemical data. This study was also conducted to identify outliers or anomaly in the set of sample wines to detect adulteration of wine. In this project, two large separate data sets are used, which contain 1,599 instances to red wine and 4, 989 instances for white wine with 11 Physicochemical data attributes such as alcohol, PH and sulfates. Two classification algorithms, Decision tree and Naïve Bayes are applied in the dataset and the The performance of these two algorithms is compared. The results showed that the decision tree (ID3) exceeded Naive Bayesian techniques particularly in red wine, what is the most common type the study also showed that two attributes, alcohol and volatile acidity Highly contribute to the quality of the wine. White wine is also more sensitive to changes in physicochemistry as opposed to red wine, therefore, higher level of handling Care is necessary. This investigation concludes that classification approach will accommodate corrective action to be taken in an effort to increase the quality of came during production.From the resulting classification precision, we find that precision rate for white wine is influenced by a greater number of physicochemical attributes, which are alcohol, density, free sulfur dioxide, chlorides, citrus acid and volatile acidity. Meanwhile, the quality of red wine is highly correlated with only four attributes, which are alcohol, sulfates, total sulfur dioxide and volatiles acidity. This shows that the quality of white wine is affected by physicochemical attributes that do not affect red came in general. Therefore, we suggest that white wine the manufacturer must carry out a wider test range particularly towards the density and the content of chloride from the quality of white wine is affected by such substances. Since then, white wine is more sensitive to changes in physicochemical properties compared to red wine, We suggest a higher level of separation between targets wine production line and red wine with particularly greater customization to the production of white wine. The attribute selection algorithm we performed was also classified alcohol as the highest in both data sets, hence the Alcohol level is the main attribute that determines the quality in red and white wine. One suggestion is the winemaker to focus on maintaining a adequate alcohol content, can be by longer fermentation period or higher yield by fermenting yeast.

**\**

**CHAPTER 3**

**PROPOSED METHODOLOGY**

**3.1 DataSetDescription:**

Our project is Prediction of wine quality and we our dataset is taken from official website of UCI Machine Learnig Repository from kaegel.Here is the provided link for our dataset from where you can download our dataset.

<https://www.kaggle.com/rajyellow46/wine-quality>

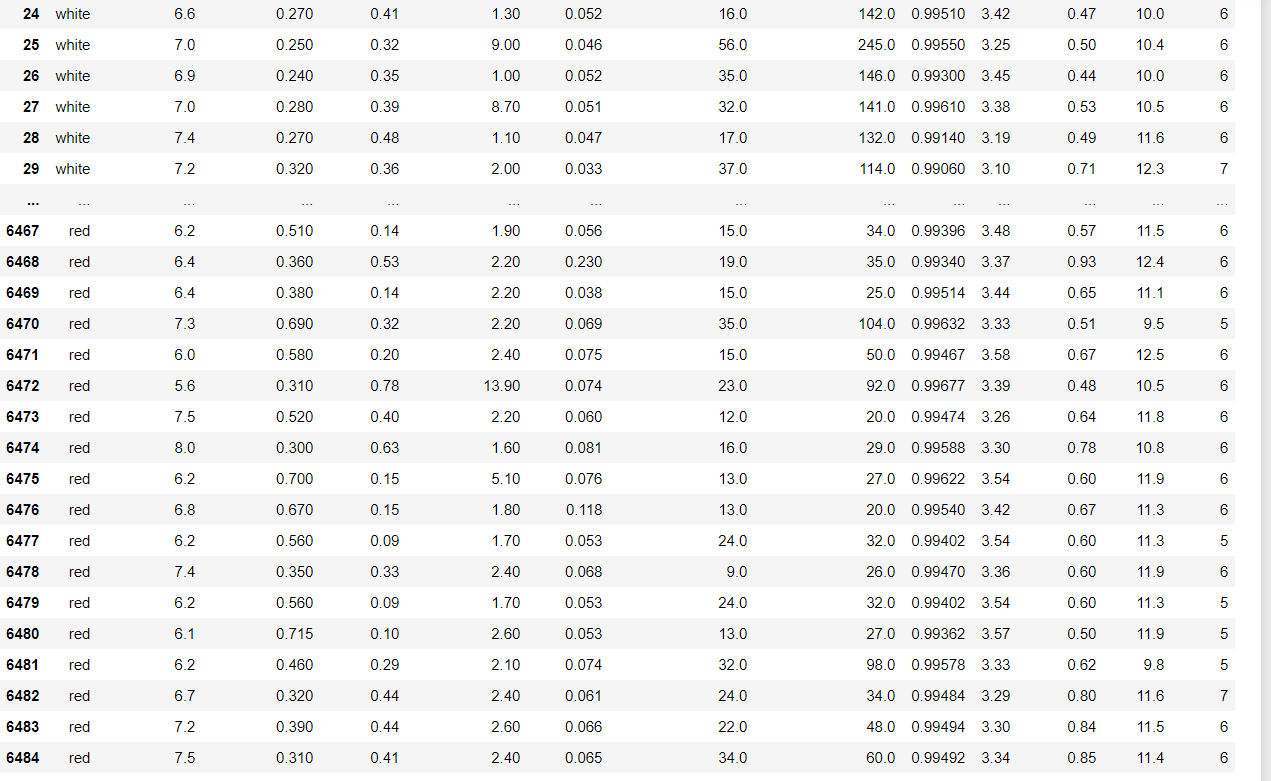
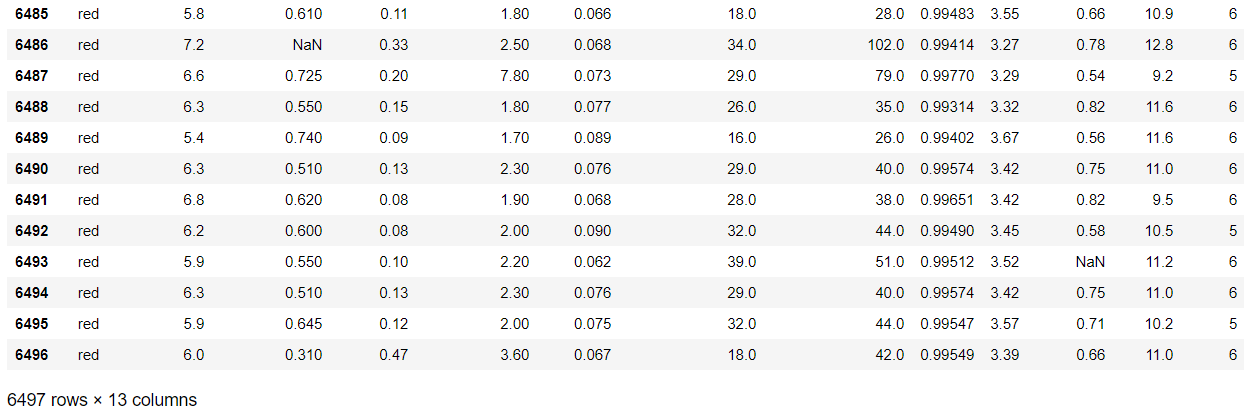
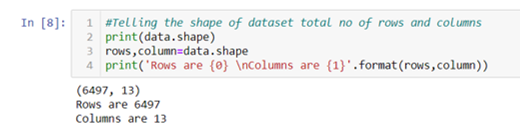
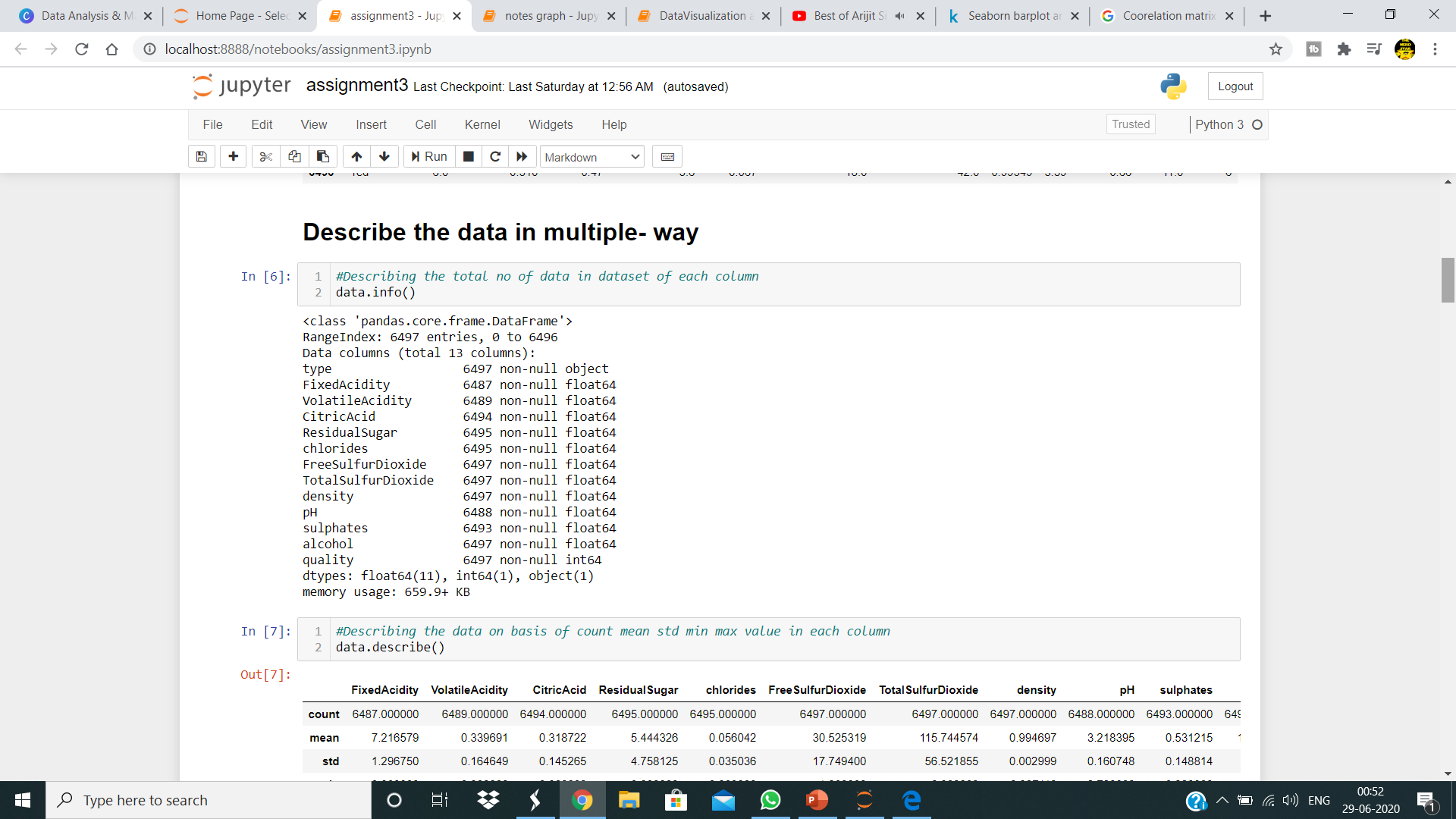
Dataset  

Table 3.1 Dataset

1) Our data contains 6497 rows and 13 columns and its shape is (6497,13)



2) Describing total no of data in dataset of each column



3) Atributes-

1 - fixed acidity: most acids involved with wine or fixed or nonvolatile (do not evaporate readily)

2 - volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste

3 - citric acid: found in small quantities, citric acid can add ‘freshness’ and flavor to wines

4 - residual sugar: the amount of sugar remaining after fermentation stops, it’s rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet

5 - chlorides: the amount of salt in the wine

6 - free sulfur dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine

7 - total sulfur dioxide: amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine

8 - density: the density of water is close to that of water depending on the percent alcohol and sugar content

9 - pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale

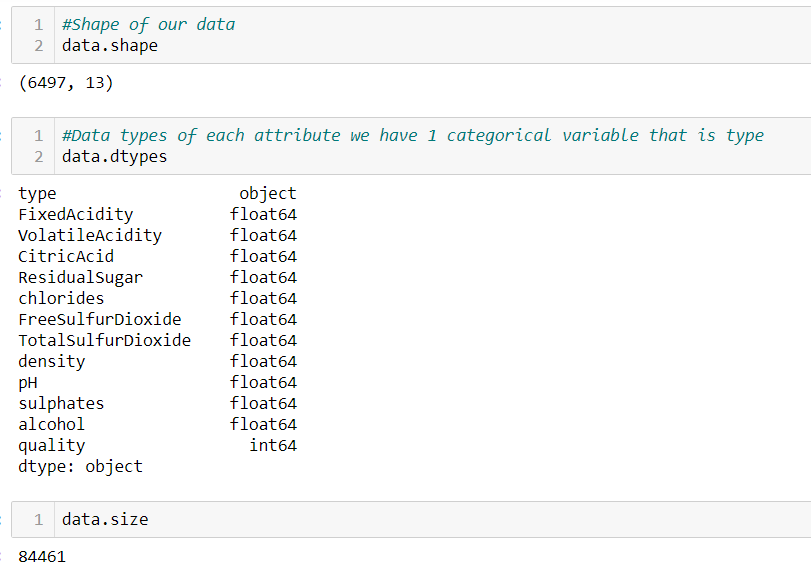
10 - sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant

11 - alcohol: the percent alcohol content of the wine

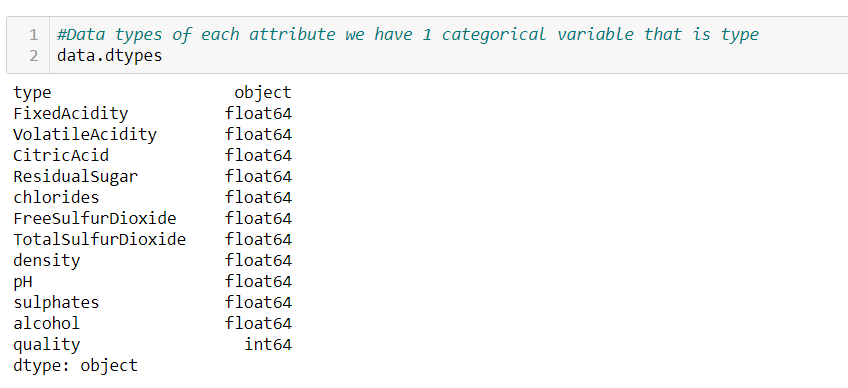
12- type: whether it is red wine or white wine

Output variable (based on sensory data): 13 - quality (score between 0 and 10)

4) Our total data size is 84461



5) Data type of each attribute of our dataset



6) data .describe() function is used to find out min ,max ,mean, quantile values of each attributes in our data.

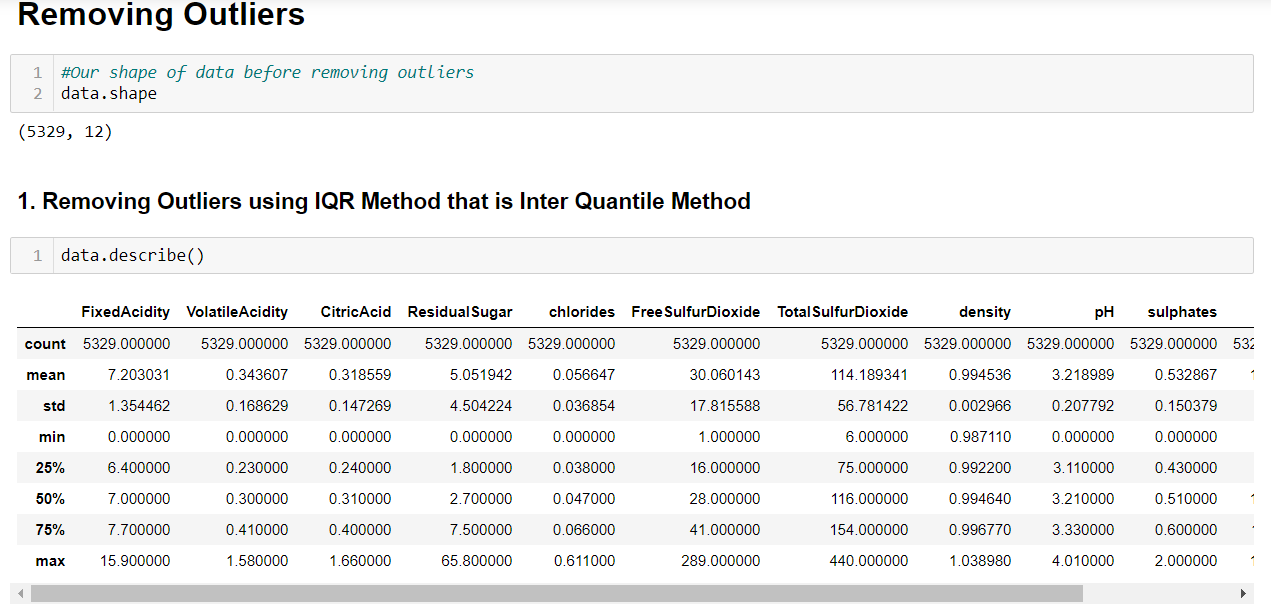


Table 3.2 Dataset Description

* 1. **Methods :**

**Libraries:**

**1. Numpy** : NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

A powerful N-dimensional array object

Sophisticated (broadcasting) functions

Tools for integrating C/C++ and Fortran code

Useful linear algebra, Fourier transform, and random number capabilities

**2. Pandas** : Pandas is open source, BSD-licensed library written in Python Language. Pandas provides high performance, fast, easy to use data structures and data analysis tools for manipulating numeric data and time series. Pandas is built on the Numpy library and written in languages like Python, Cython and C. In 2008, Wes McKinney developed Pandas library. In pandas, we can import data from various file formats like JSON, SQL, Microsoft Excel etc. Pandas also provides additional features like data cleaning, data wrangling, merging etc.

**3. Matplotlib** : Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. Pyplot is a Matplotlib module which provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

**4. Seaborn** : Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. This article deals with the distribution plots in seaborn which is used for examining univariate and bivariate distributions. In this article we will be discussing 4 types of distribution plots namely:

joinplot

distplot

pairplot

rugplot

**Models:**

**1) LogisticRegression Model-**

Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

For example,

* To predict whether an email is spam (1) or (0)
* Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.

From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

**Simple Logistic Regression**

(Full Source code: <https://github.com/SSaishruthi/LogisticRegression_Vectorized_Implementation/blob/master/Logistic_Regression.ipynb>)

**Model**

Output = 0 or 1

Hypothesis => Z = WX + B

hΘ(x) = sigmoid (Z)

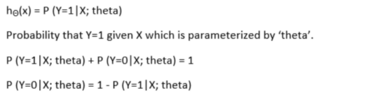
If ‘Z’ goes to infinity, Y(predicted) will become 1 and if ‘Z’ goes to negative infinity, Y(predicted) will become 0.

**Analysis of the hypothesis**

The output from the hypothesis is the estimated probability. This is used to infer how confident can predicted value be actual value when given an input X. Consider the below example,

X = [x0 x1] = [1 IP-Address]

Based on the x1 value, let’s say we obtained the estimated probability to be 0.8. This tells that there is 80% chance that an email will be spam.



This justifies the name ‘logistic regression’. Data is fit into linear regression model, which then be acted upon by a logistic function predicting the target categorical dependent variable.

**Types of Logistic Regression**

1. Binary Logistic Regression

The categorical response has only two 2 possible outcomes. Example: Spam or Not

2. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5

**2) Decision tree-**

The ability to restructure a decision tree efficiently enables a variety of approaches to decision tree induction that would otherwise be prohibitively expensive. Two such approaches are described here, one being incremental tree induction (ITI), and the other being non-incremental tree induction using a measure of tree quality instead of test quality (DMTI). These approaches and several variants offer new computational and classifier characteristics that lend themselves to particular applications.

A decision tree is a [flowchart](https://en.wikipedia.org/wiki/Flowchart)-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

In [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis), a decision tree and the closely related [influence diagram](https://en.wikipedia.org/wiki/Influence_diagram) are used as a visual and analytical decision support tool, where the [expected values](https://en.wikipedia.org/wiki/Expected_value) (or [expected utility](https://en.wikipedia.org/wiki/Expected_utility)) of competing alternatives are calculated.

A decision tree consists of three types of nodes:[[1]](https://en.wikipedia.org/wiki/Decision_tree" \l "cite_note-1)

1. Decision nodes – typically represented by squares
2. Chance nodes – typically represented by circles
3. End nodes – typically represented by triangles

Decision trees are commonly used in [operations research](https://en.wikipedia.org/wiki/Operations_research) and [operations management](https://en.wikipedia.org/wiki/Operations_management). If, in practice, decisions have to be taken online with no recall under incomplete knowledge, a decision tree should be paralleled by a [probability](https://en.wikipedia.org/wiki/Probability) model as a best choice model or online selection model [algorithm](https://en.wikipedia.org/wiki/Algorithm). Another use of decision trees is as a descriptive means for calculating [conditional probabilities](https://en.wikipedia.org/wiki/Conditional_probability).

Decision trees, [influence diagrams](https://en.wikipedia.org/wiki/Influence_diagrams), [utility functions](https://en.wikipedia.org/wiki/Utility_function), and other [decision analysis](https://en.wikipedia.org/wiki/Decision_analysis) tools and methods are taught to undergraduate students in schools of business, health economics, and public health, and are examples of operations research or [management science](https://en.wikipedia.org/wiki/Management_science) methods.

Drawn from left to right, a decision tree has only burst nodes (splitting paths) but no sink nodes (converging paths). Therefore, used manually, they can grow very big and are then often hard to draw fully by hand. Traditionally, decision trees have been created manually – as the aside example shows – although increasingly, specialized software is employed.

### Decision rules

The decision tree can be [linearized](https://en.wikipedia.org/wiki/Linearization) into **decision rules**,[[2]](https://en.wikipedia.org/wiki/Decision_tree#cite_note-2) where the outcome is the contents of the leaf node, and the conditions along the path form a conjunction in the if clause. In general, the rules have the form:

*if* condition1 *and* condition2 *and* condition3 *then* outcome.

Decision rules can be generated by constructing [association rules](https://en.wikipedia.org/wiki/Association_rule_learning) with the target variable on the right. They can also denote [temporal](https://en.wikipedia.org/wiki/Time) or [causal](https://en.wikipedia.org/wiki/Causal) relations.[[3]](https://en.wikipedia.org/wiki/Decision_tree#cite_note-3)

Analysis can take into account the decision maker's (e.g., the company's) [preference](https://en.wikipedia.org/wiki/Preference) or [utility function](https://en.wikipedia.org/wiki/Utility_function), for example:

The basic interpretation in this situation is that the company prefers B's risk and payoffs under realistic risk preference coefficients (greater than $400K—in that range of risk aversion, the company would need to model a third strategy, "Neither A nor B").

Another example, commonly used in [operations research](https://en.wikipedia.org/wiki/Operations_research) courses, is the distribution of lifeguards on beaches (a.k.a. the "Life's a Beach" example).[[4]](https://en.wikipedia.org/wiki/Decision_tree#cite_note-4) The example describes two beaches with lifeguards to be distributed on each beach. There is maximum budget *B* that can be distributed among the two beaches (in total), and using a marginal returns table, analysts can decide how many lifeguards to allocate to each beach.

**3)Random Forest Model-**

**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 operate by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees.[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1)[[2]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1998-2) Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting" \o "Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):587–588

The first algorithm for random decision forests was created by [Tin Kam Ho](https://en.wikipedia.org/wiki/Tin_Kam_Ho)[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1) using the [random subspace method](https://en.wikipedia.org/wiki/Random_subspace_method),[[2]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1998-2) which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.[[4]](https://en.wikipedia.org/wiki/Random_forest#cite_note-kleinberg1990-4)[[5]](https://en.wikipedia.org/wiki/Random_forest#cite_note-kleinberg1996-5)[[6]](https://en.wikipedia.org/wiki/Random_forest#cite_note-kleinberg2000-6)

An extension of the algorithm was developed by [Leo Breiman](https://en.wikipedia.org/wiki/Leo_Breiman)[[7]](https://en.wikipedia.org/wiki/Random_forest#cite_note-breiman2001-7) and [Adele Cutler](https://en.wikipedia.org/wiki/Adele_Cutler),[[8]](https://en.wikipedia.org/wiki/Random_forest#cite_note-rpackage-8) who registered[[9]](https://en.wikipedia.org/wiki/Random_forest#cite_note-9) "Random Forests" as a [trademark](https://en.wikipedia.org/wiki/Trademark) (as of 2019, owned by [Minitab, Inc.](https://en.wikipedia.org/wiki/Minitab)).[[10]](https://en.wikipedia.org/wiki/Random_forest#cite_note-10) The extension combines Breiman's "[bagging](https://en.wikipedia.org/wiki/Bootstrap_aggregating)" idea and random selection of features, introduced first by Ho[[1]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho1995-1) and later independently by Amit and [Geman](https://en.wikipedia.org/wiki/Donald_Geman" \o "Donald Geman)[[11]](https://en.wikipedia.org/wiki/Random_forest#cite_note-amitgeman1997-11) in order to construct a collection of decision trees with controlled variance.

## Algorithm

### Preliminaries: decision tree learning

*Main article:*[*Decision tree learning*](https://en.wikipedia.org/wiki/Decision_tree_learning)

Decision trees are a popular method for various machine learning tasks. Tree learning "come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining", say [Hastie](https://en.wikipedia.org/wiki/Trevor_Hastie) *et al.*, "because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate".[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):352

In particular, trees that are grown very deep tend to learn highly irregular patterns: they [overfit](https://en.wikipedia.org/wiki/Overfitting" \o "Overfitting) their training sets, i.e. have [low bias, but very high variance](https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff). Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance.[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):587–588 This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

Forests are like the pulling together of decision tree algorithm efforts. Taking the teamwork of many trees thus improving the performance of a single random tree. Though not quite similar, forests give the effects of a K-fold cross validation.

### Bagging

*Main article:*[*Bootstrap aggregating*](https://en.wikipedia.org/wiki/Bootstrap_aggregating)

The training algorithm for random forests applies the general technique of [bootstrap aggregating](https://en.wikipedia.org/wiki/Bootstrap_aggregating), or bagging, to tree learners. Given a training set *X* = *x1*, ..., *xn* with responses *Y* = *y1*, ..., *yn*, bagging repeatedly (*B* times) selects a [random sample with replacement](https://en.wikipedia.org/wiki/Sampling_(statistics)#Replacement_of_selected_units) of the training set and fits trees to these samples:

For *b* = 1, ..., *B*:

1. Sample, with replacement, *n* training examples from *X*, *Y*; call these *Xb*, *Yb*.
2. Train a classification or regression tree *fb* on *Xb*, *Yb*.

After training, predictions for unseen samples *x'* can be made by averaging the predictions from all the individual regression trees on *x'*:

{\displaystyle {\hat {f}}={\frac {1}{B}}\sum \_{b=1}^{B}f\_{b}(x')}

or by taking the majority vote in the case of classification trees.

This bootstrapping procedure leads to better model performance because it decreases the [variance](https://en.wikipedia.org/wiki/Bias%E2%80%93variance_dilemma) of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.

Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on *x'*:

{\displaystyle \sigma ={\sqrt {\frac {\sum \_{b=1}^{B}(f\_{b}(x')-{\hat {f}})^{2}}{B-1}}}.}

The number of samples/trees, *B*, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimal number of trees *B* can be found using [cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)), or by observing the [*out-of-bag error*](https://en.wikipedia.org/wiki/Out-of-bag_error): the mean prediction error on each training sample *xᵢ*, using only the trees that did not have *xᵢ* in their bootstrap sample.[[13]](https://en.wikipedia.org/wiki/Random_forest#cite_note-islr-13) The training and test error tend to level off after some number of trees have been fit.

### From bagging to random forests

*Main article:*[*Random subspace method*](https://en.wikipedia.org/wiki/Random_subspace_method)

The above procedure describes the original bagging algorithm for trees. Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a [random subset of the features](https://en.wikipedia.org/wiki/Random_subspace_method). This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few [features](https://en.wikipedia.org/wiki/Feature_(machine_learning)) are very strong predictors for the response variable (target output), these features will be selected in many of the *B* trees, causing them to become correlated. An analysis of how bagging and random subspace projection contribute to accuracy gains under different conditions is given by Ho.[[14]](https://en.wikipedia.org/wiki/Random_forest#cite_note-ho2002-14)

Typically, for a classification problem with *p* features, √*p* (rounded down) features are used in each split.[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):592 For regression problems the inventors recommend *p/3* (rounded down) with a minimum node size of 5 as the default.[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):592 In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.[[3]](https://en.wikipedia.org/wiki/Random_forest#cite_note-elemstatlearn-3):592

### ExtraTrees

Adding one further step of randomization yields *extremely randomized trees*, or ExtraTrees. While similar to ordinary random forests in that they are an ensemble of individual trees, there are two main differences: first, each tree is trained using the whole learning sample (rather than a bootstrap sample), and second, the top-down splitting in the tree learner is randomized. Instead of computing the locally *optimal* cut-point for each feature under consideration (based on, e.g., [information gain](https://en.wikipedia.org/wiki/Information_gain) or the [Gini impurity](https://en.wikipedia.org/wiki/Gini_impurity" \o "Gini impurity)), a *random* cut-point is selected. This value is selected from a uniform distribution within the feature's empirical range (in the tree's training set). Then, of all the randomly generated splits, the split that yields the highest score is chosen to split the node. Similar to ordinary random forests, the number of randomly selected features to be considered at each node can be specified

### Variable importance

Random forests can be used to rank the importance of variables in a regression or classification problem in a natural way. The following technique was described in Breiman's original paper[[7]](https://en.wikipedia.org/wiki/Random_forest" \l "cite_note-breiman2001-7) and is implemented in the [R](https://en.wikipedia.org/wiki/R_(programming_language)) package randomForest.[[8]](https://en.wikipedia.org/wiki/Random_forest#cite_note-rpackage-8)

The first step in measuring the variable importance in a data set {\displaystyle {\mathcal {D}}\_{n}=\{(X\_{i},Y\_{i})\}\_{i=1}^{n}} is to fit a random forest to the data. During the fitting process the [out-of-bag error](https://en.wikipedia.org/wiki/Out-of-bag_error) for each data point is recorded and averaged over the forest (errors on an independent test set can be substituted if bagging is not used during training).

To measure the importance of the {\displaystyle j}-th feature after training, the values of the {\displaystyle j}-th feature are permuted among the training data and the out-of-bag error is again computed on this perturbed data set. The importance score for the {\displaystyle j}-th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences.

Features which produce large values for this score are ranked as more important than features which produce small values. The statistical definition of the variable importance measure was given and analyzed by Zhu *et al.*[

This method of determining variable importance has some drawbacks. For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Methods such as [partial permutations](https://en.wikipedia.org/wiki/Partial_permutation) and growing unbiased trees can be used to solve the problem. If the data contain groups of correlated features of similar relevance for the output, then smaller groups are favored over larger groups.

### Relationship to nearest neighbors

A relationship between random forests and the [*k*-nearest neighbor algorithm](https://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm) (*k*-NN) was pointed out by Lin and Jeon in 2002.[[22]](https://en.wikipedia.org/wiki/Random_forest#cite_note-linjeon02-22) It turns out that both can be viewed as so-called *weighted neighborhoods schemes*. These are models built from a training set {\displaystyle \{(x\_{i},y\_{i})\}\_{i=1}^{n}} that make predictions {\displaystyle {\hat {y}}} for new points *x'* by looking at the "neighborhood" of the point, formalized by a weight function *W*:{\displaystyle {\hat {y}}=\sum \_{i=1}^{n}W(x\_{i},x')\,y\_{i}.}

Here, {\displaystyle W(x\_{i},x')} is the non-negative weight of the *i*'th training point relative to the new point *x'* in the same tree. For any particular *x'*, the weights for points {\displaystyle x\_{i}} must sum to one. Weight functions are given as follows:

* In *k*-NN, the weights are {\displaystyle W(x\_{i},x')={\frac {1}{k}}} if *xi* is one of the *k* points closest to *x'*, and zero otherwise.
* In a tree, {\displaystyle W(x\_{i},x')={\frac {1}{k'}}} if *xi* is one of the *k'* points in the same leaf as *x'*, and zero otherwise.

Since a forest averages the predictions of a set of *m* trees with individual weight functions {\displaystyle W\_{j}}, its predictions are

{\displaystyle {\hat {y}}={\frac {1}{m}}\sum \_{j=1}^{m}\sum \_{i=1}^{n}W\_{j}(x\_{i},x')\,y\_{i}=\sum \_{i=1}^{n}\left({\frac {1}{m}}\sum \_{j=1}^{m}W\_{j}(x\_{i},x')\right)\,y\_{i}.}

**3.3 Hardware and Software Requirements :**

**3.3.1 Hardware Minimum system requirements are :**

Processors: Intel Core i3 processor or above

Disk space:1GB or above

Operating systems: Windows7 or later, macOS, and linux.

**3.3.2 Software Requirements :**

Any version of Jupyter Notebook in which :

Pycharm

Pandas

Matplotlib

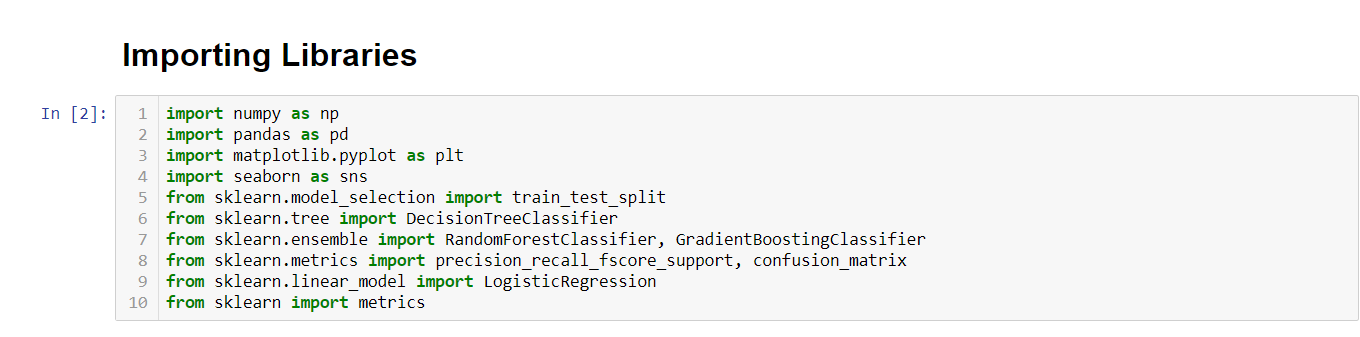
Seaborn

Is already installed.

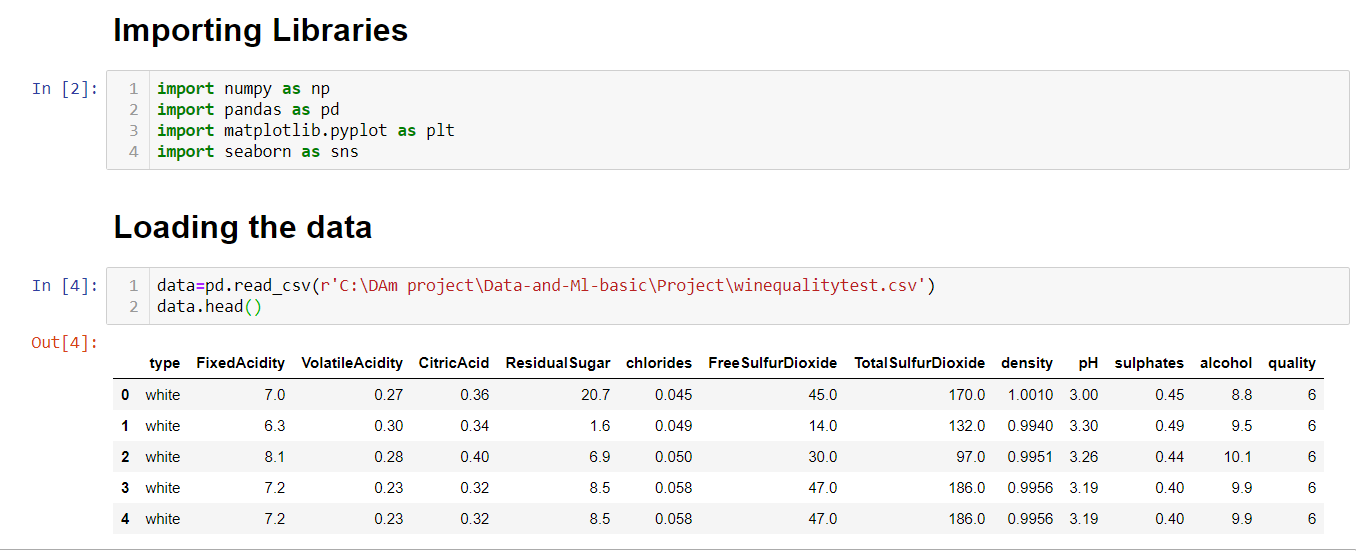
Latest version of Pycharm.

**3.4 Our Methodology:**

1. **Importing Libraries** in our jupyter notebook



1. **Loading dataset** in our jupyter notebook using pandas read.csv file command



1. **Dataset Description-** Here we are describing our whole data set by different methods so that we can get an insight of our data

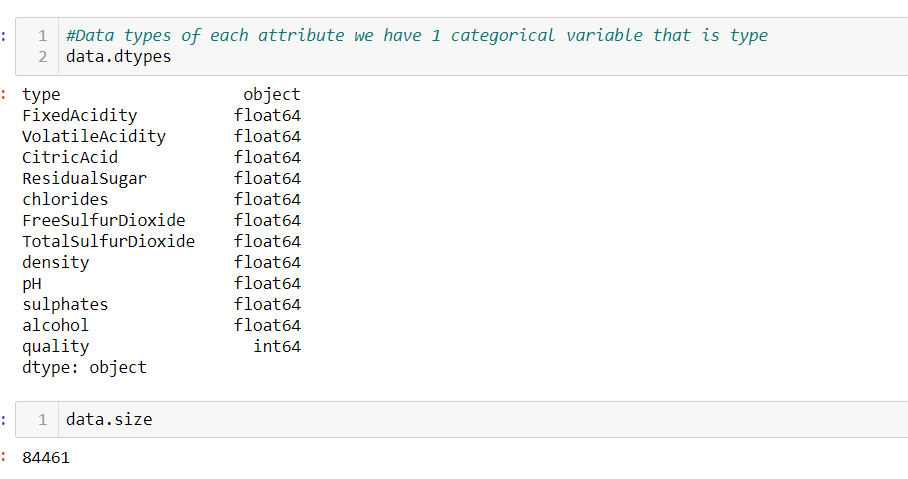
àdata.info() tells about all our attributes present in our dataset

àdata.shape tells abt the size of our data that we have 6497 rows and 13 collumns in our dataset.



àdata.dtypes tells about the data type of each attribute in our data set we have 1 of object type and 1 of int type rest all variables are of float type.

àdata.size tells about the size of our dataset.We have 84,461 data values in total in our dataset.



1. **Data Preprocessing-**

Data is in so many different forms: Structured Tables, Images, Audio files, Videos etc..

Machines don’t understand free text, image or video data as it is, they understand 1s and 0s.

Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models

3 Ways we used for Preprocessing

1)Checking Null values

2) Replacing them with different value

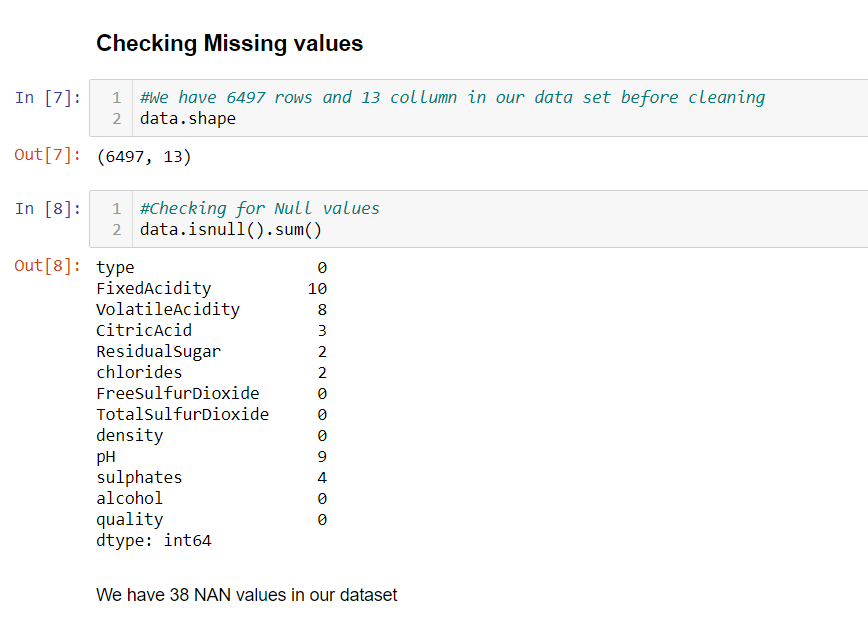
3)Removing duplicate values

* **Checking Missing Values :**

Currently we have 6497 rows in our data set and 13 collumns.

To check missing values we will use data.isnull().sum method.

We found out that there are 38 NAN or NULL values in our dataset.



* **Replacing NULL values :**

We have used fillna() method to replace NULL values with -1.

Now if you again check for missing values you will find that now there are no missing values in our dataset.

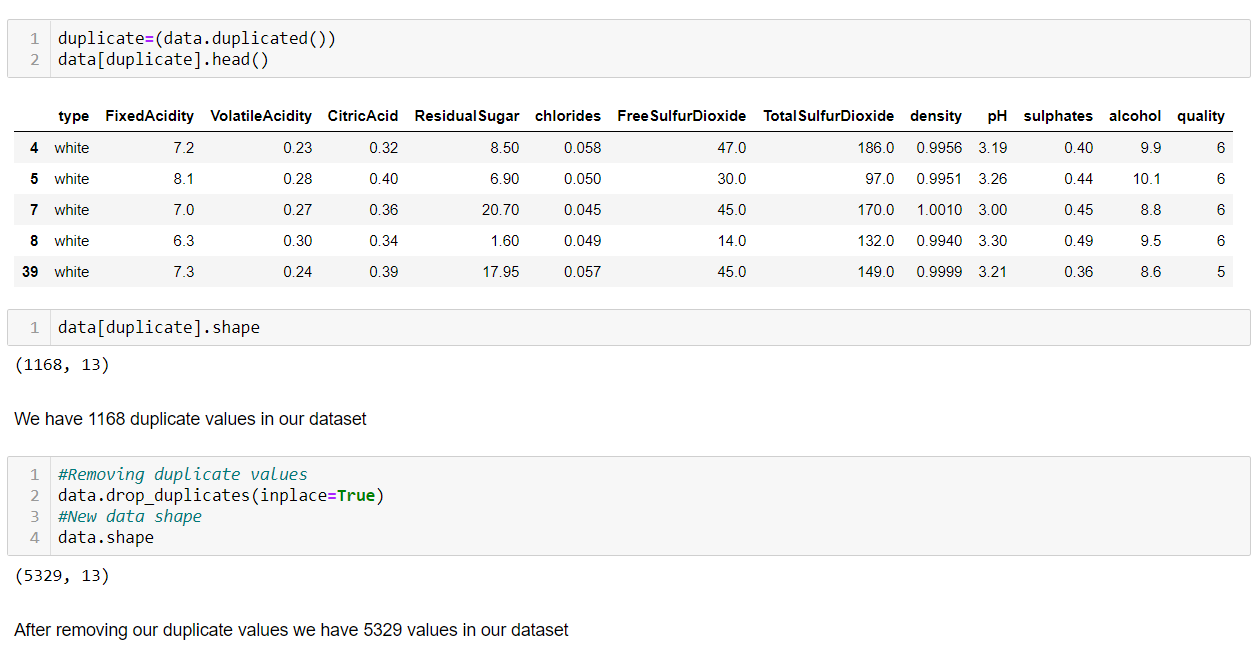


* **Removing duplicate values :**

-We check for duplicate values using data.duplicate function

-Using data[duplicate].shape we found out we have 1168 rows that are duplicated .

-To remove duplicate value we used data.drop.duplicate().Now after deleting values we have 5329 rows.



1. **Data Visualization :**

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Data in a dataset divided into 2 categories

Categorical Data

Quantative Data

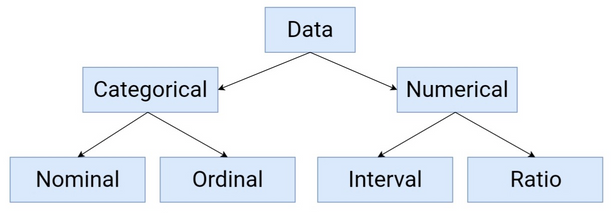
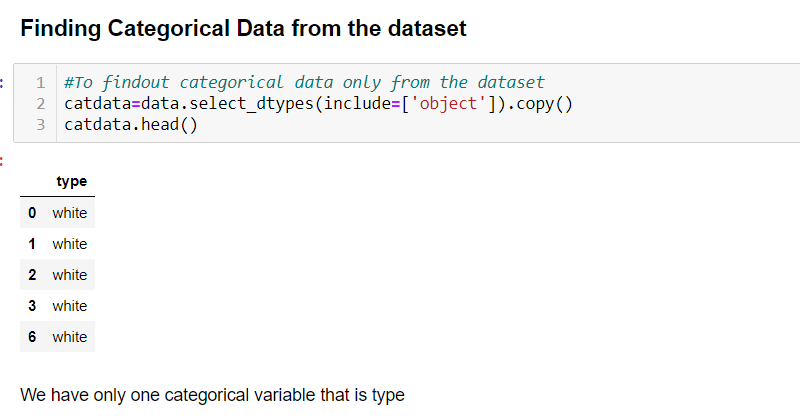


Fig-3.1 Classification of Data

**Categorical and Quantitative Variables :**

In our dataset we have 1 categorical variable that is type while tells whether data is for red wine or white wine and rest all our quantitative variables.

To find categorical variable we will find that variable whose data type is of object type.



**àPlotting a bar graph and a pi chart for our categorical data :**

**Bar Graph**-It tells about the no of data of red and white wine in total data set. Here you can see we have 4000 data values of white wine and nearly 1500 values of red wine in our data set

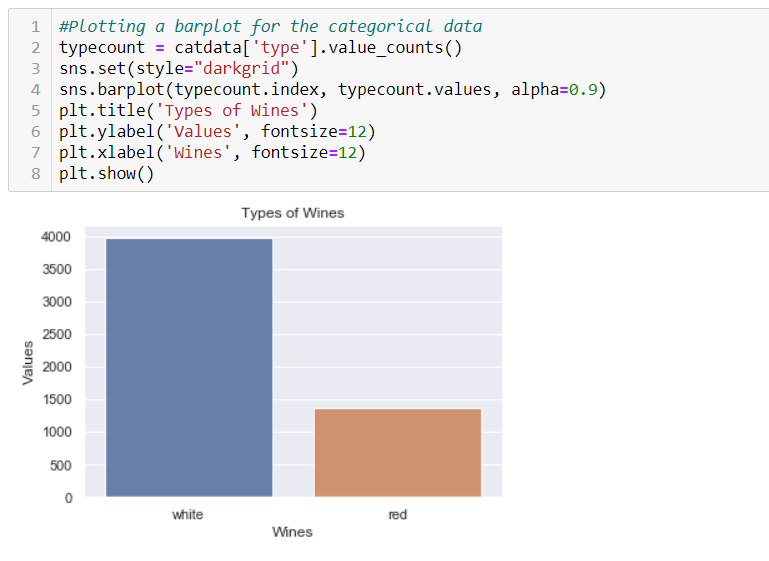


Fig-3.2 Bar plot for Categorical Data

**Pi Chart**- It shows the division of data in percentage with respect to total data set.Here you can see the white wine type is 75% and res wine type is 25%.

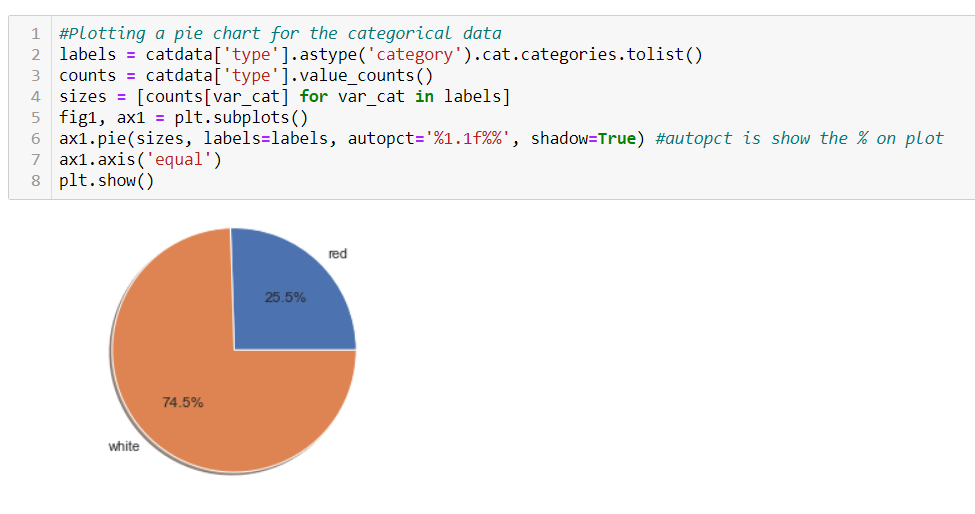


Fig-3.3 Pi Chart for Categorical data

à**Visualizing the quality of wine :** Here we have plot a countplot to analyze the count regarding each value of quality.

We can observe that the count of quality 6 is highest which is 2327 followed by 5 and 7 .



Fig-3.4 Count plot of quality

à**Visualizing other attributes vs quality :**

**Citric Acid vs Quality**- Composition of citric acid go higher as we go higher in the quality of wine.

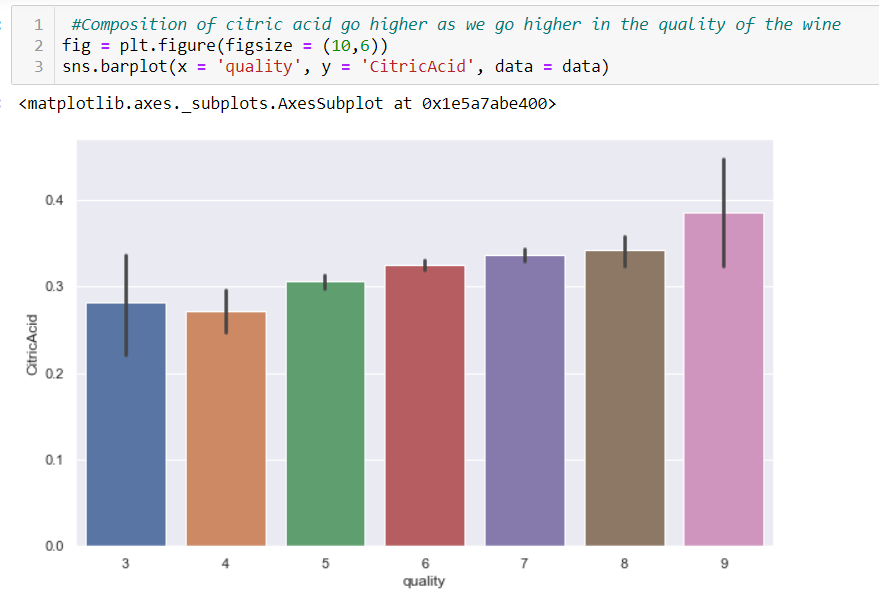


Fig-3.5 Citric Acid vs. Quality Bar plot

**Chlorides vs Quality** : Composition of chlorides go down as we go higher with the quality.

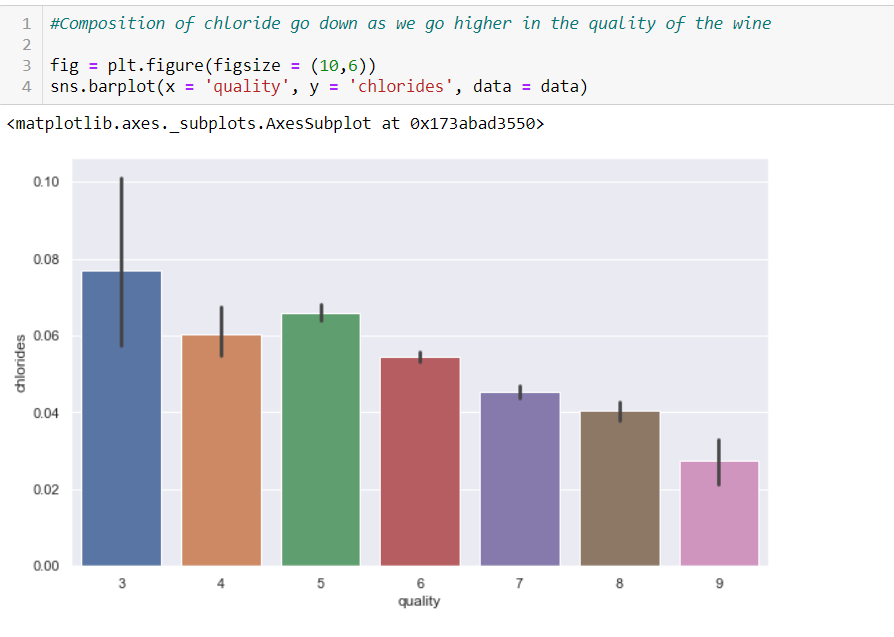


Fig-3.6 Chlorides vs. Quality Bar Plot

**Sulphates Vs Quality** : Composition of sulphates first increase with increase in quality then decrease after 7

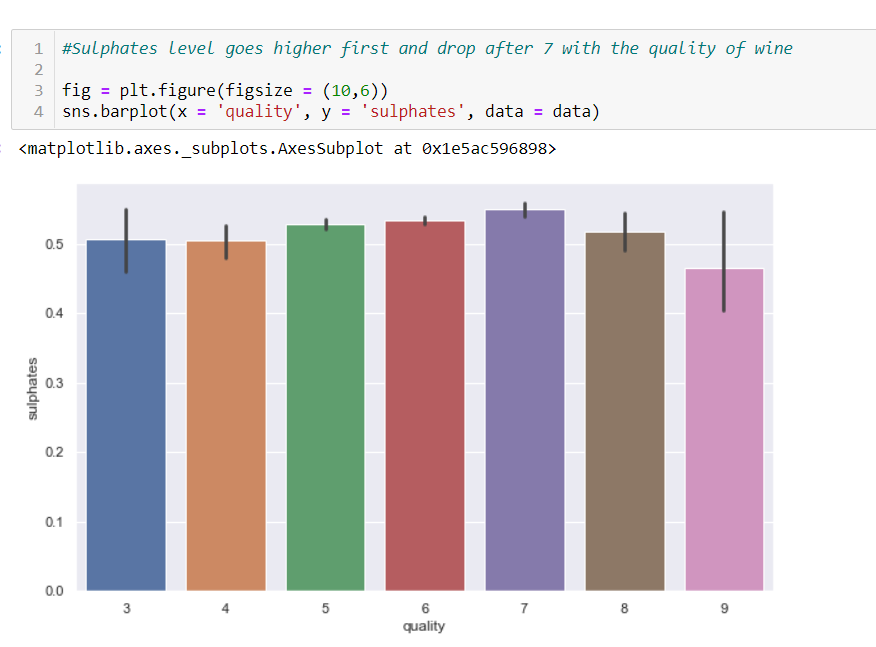


Fig 3.7 Sulphates vs. Quality Bar Plot

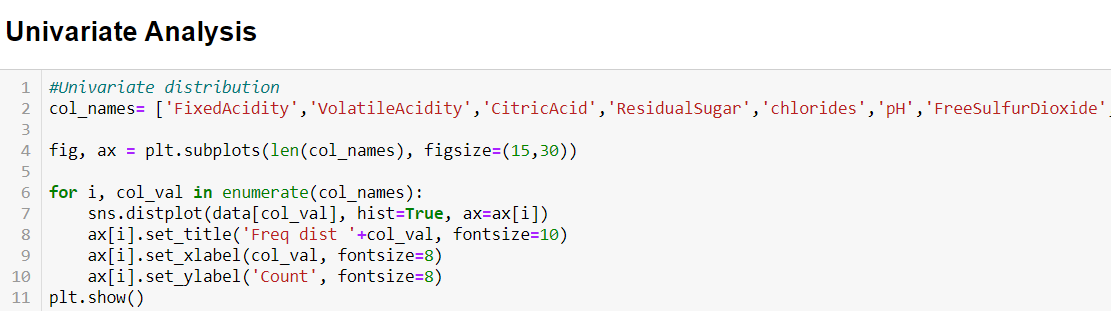
**Univariate and Bivariate Analysisà**

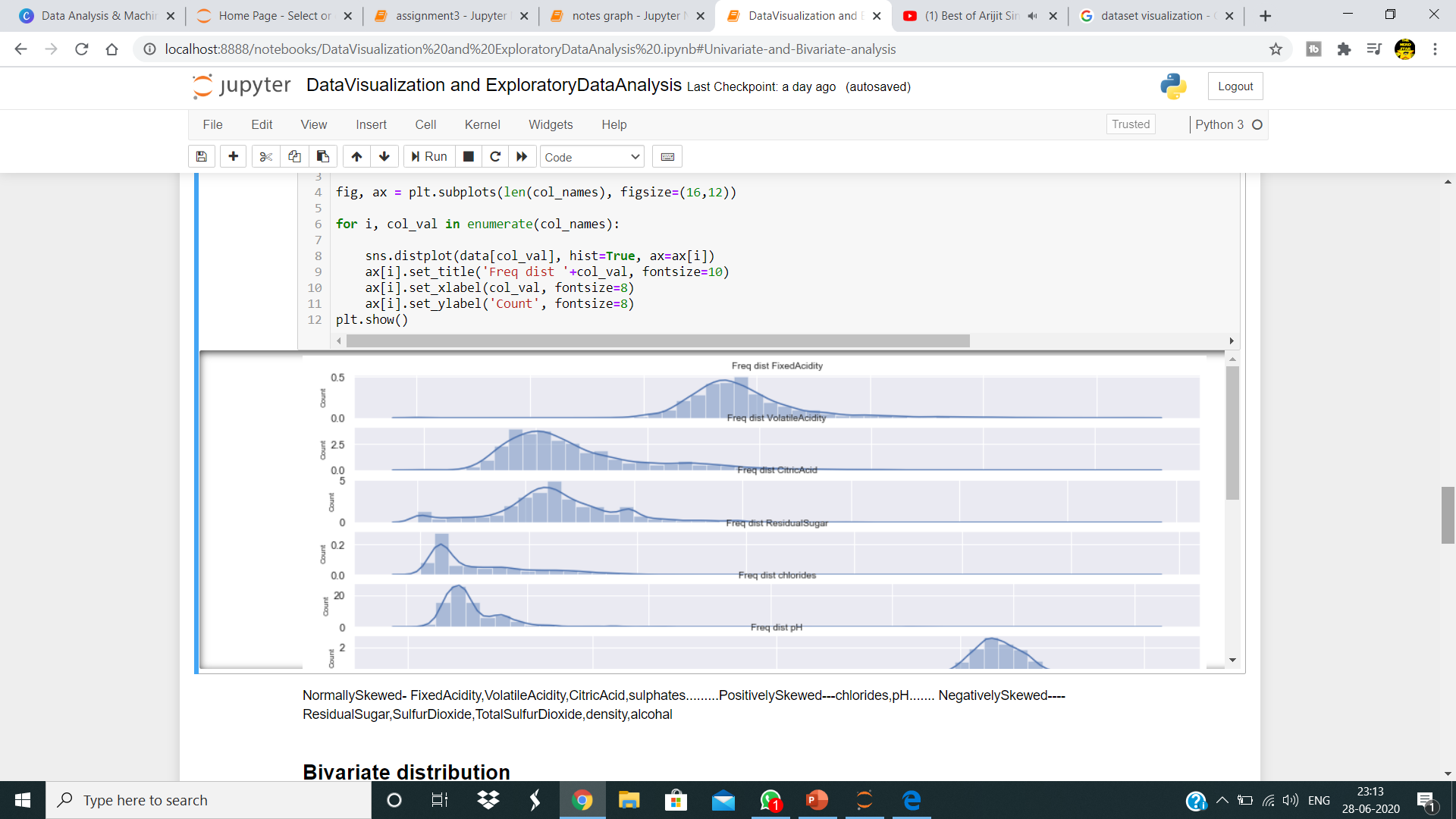
**Univariate Analysisà** It is the simplest form of statistical analysis.The key fact is that only one variable is involved. With the help of the univariate analysis we could also meaure skewness  by looking at graph whether it is positive negative or normally skewed.You could see here that

Normally skewed-Graph of Fixed Acidity

Positively skewed-Graph of pH

Negatively skewed – rest all graphs





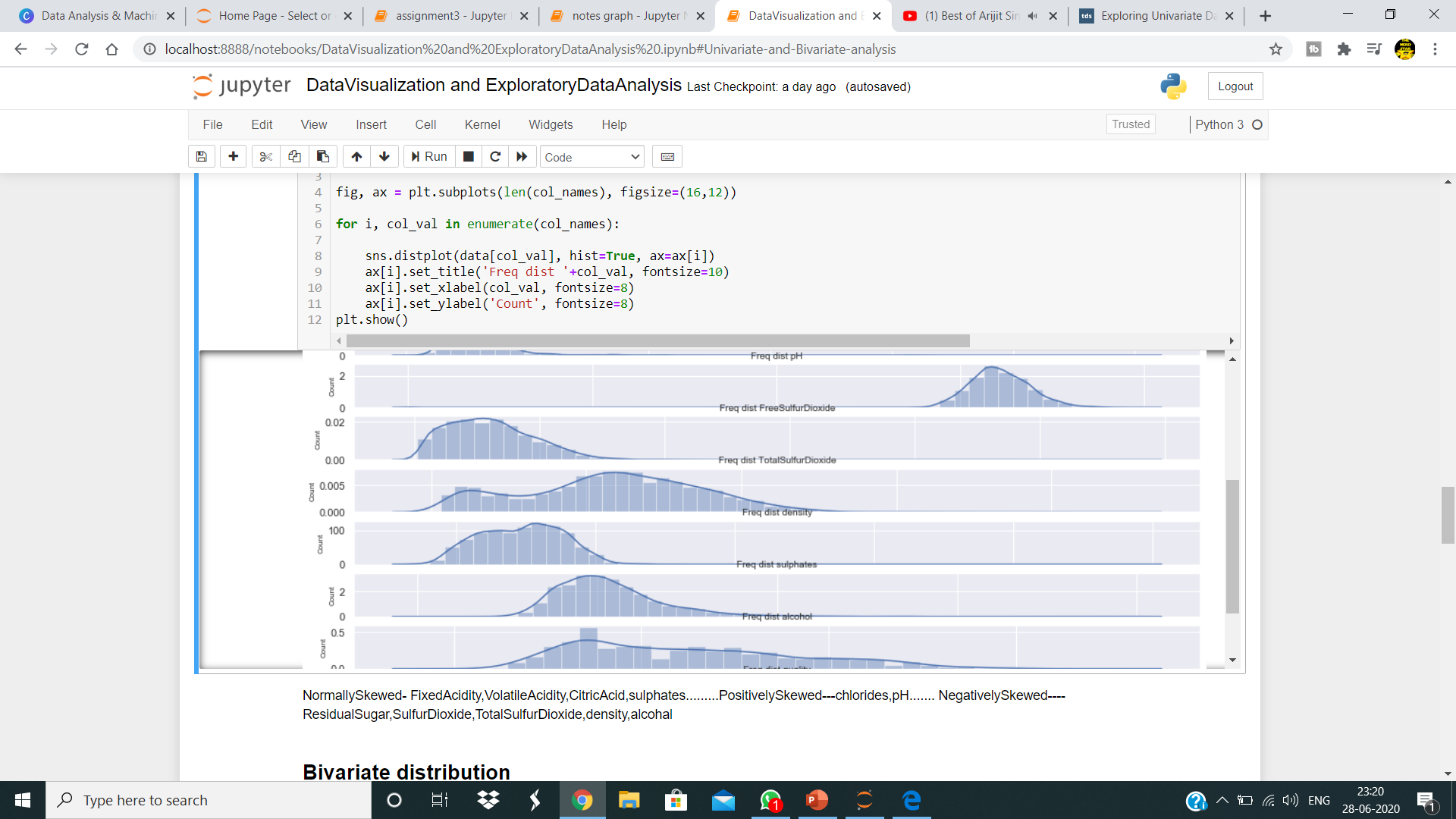


Fig-3.8 Distplot Histogram for Univariate Analysis

**Bivariate Analysisà** Bivariate analysis is the simultaneous analysis of two variables (attributes). It explores the concept of relationship between two variables. We have plot a pair plot to understand the concept of relationship between two variables for each attributes.



Fig-3.9 Pair Plot for Bivariate Analysis

**Outliersà** Outliers in a graph are those points that don’t lie in same place with other points in the graph due to which we get to see great variations in our dataset.To get more accurate data we need to remove these outliers from our data.

**Checking for outliersà**We plotted a box plot for each attribute to identify the outliers in our dataset

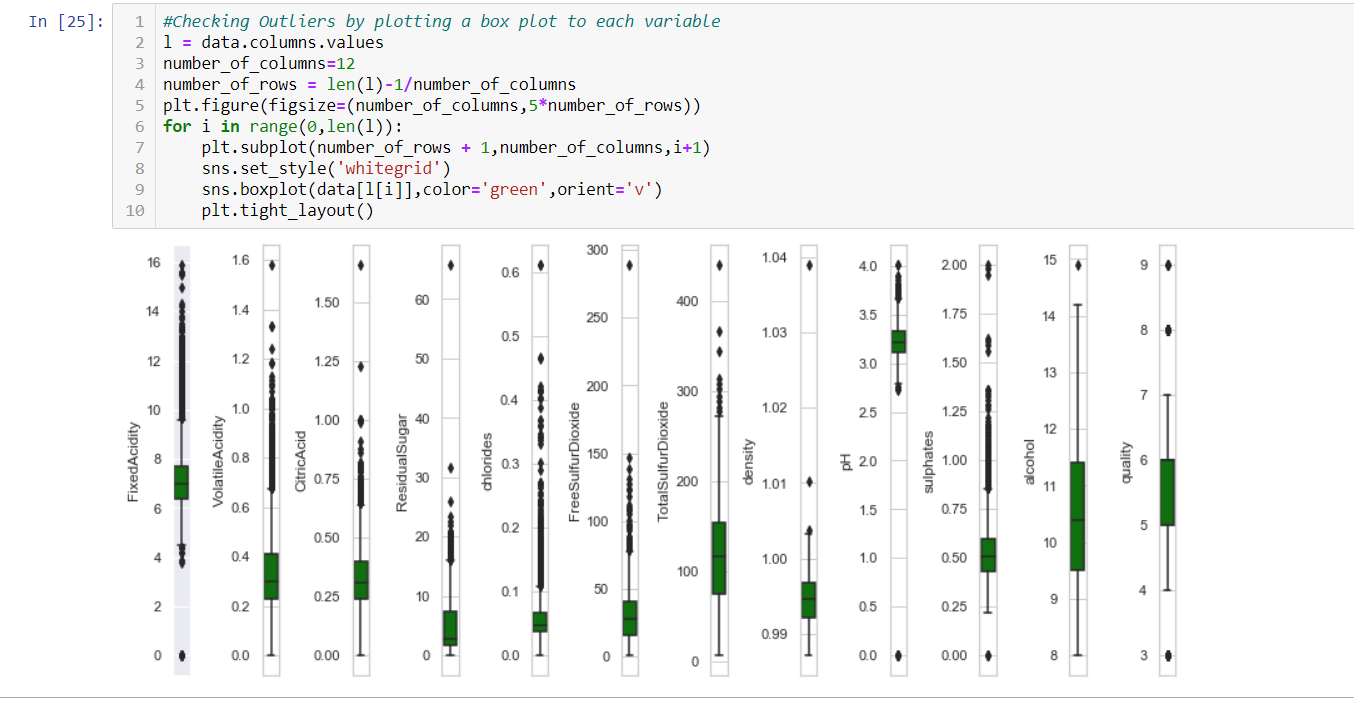


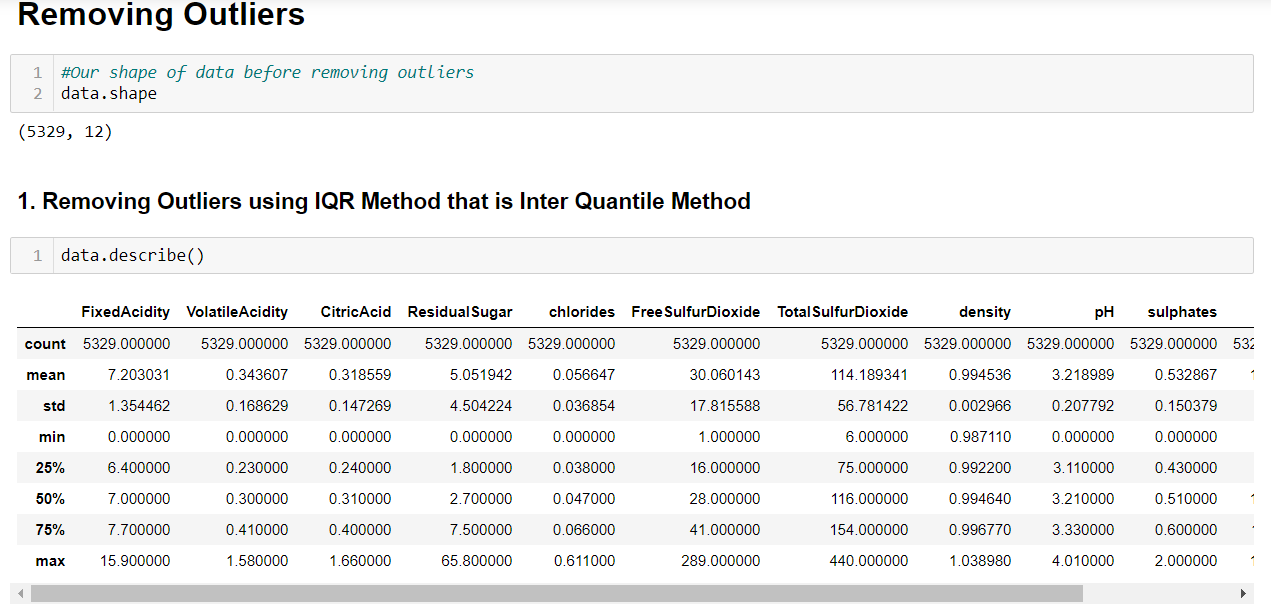
Fig-3.10 Box Plot for Outlier Detection

You can see here that in each box plot there are some points or valueswhich are outliers .So we need to remove tthem.

**Removing Outliers**à We used 2 methods to remove our outliers

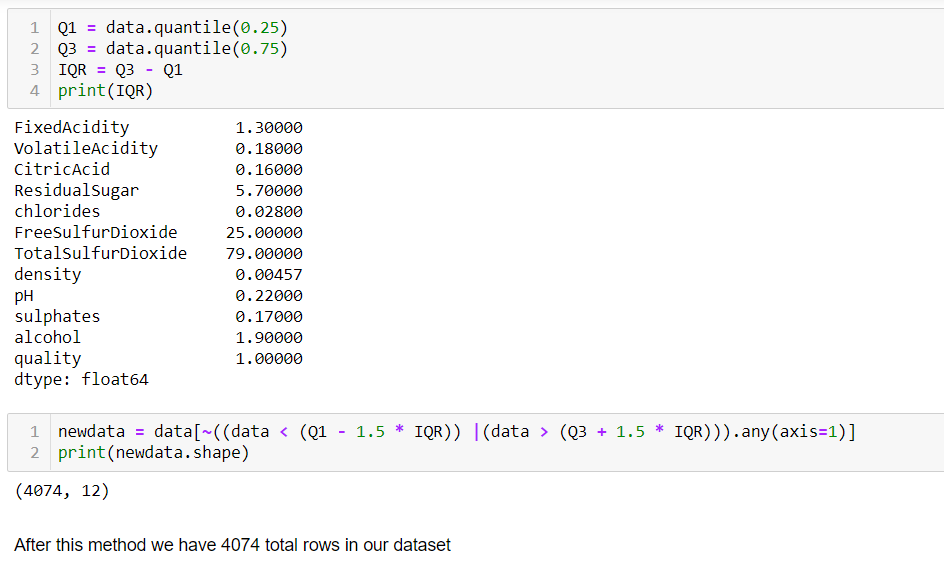
Inter Quantile Method(IQR) Z Score Method

. Our data set shape currently is (5329,12)

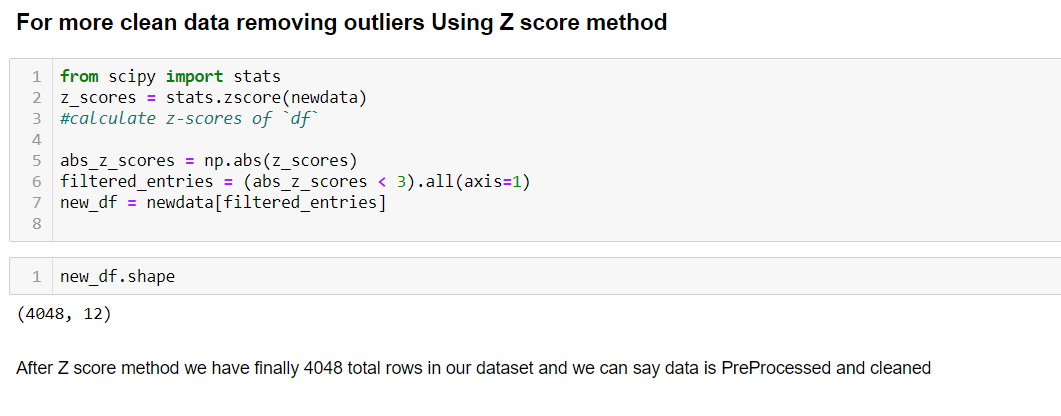


**Inter Quantile Method (IQR)-**

After applying this method you can see that our new data has only 4074 values left.



We cleaned our data more by applying Z-Score outlier removal method



Now finally our data is cleaned and have 4048 total rows in our dataset.

**Correlation Matrix** : Correlation Matrix-A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

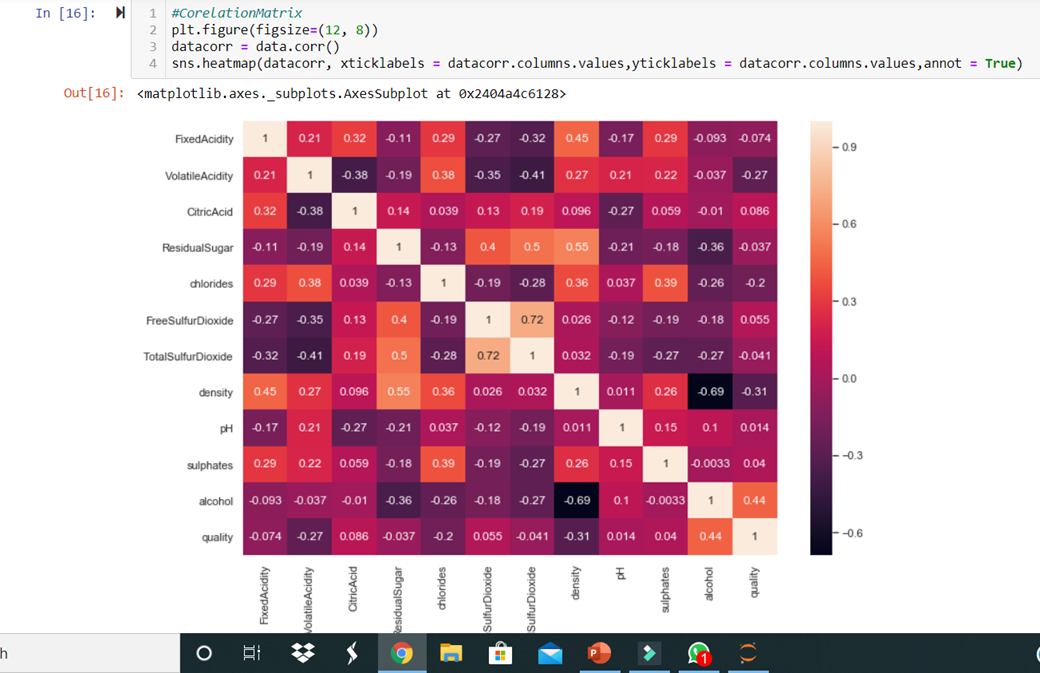


Fig-3.11 Correlation Matrix

**Outcome :**

The Lighter color shows that the value is more closer to 1 that means it is more corelated and darker value tells it is less corelated we have more darker values in our matrix which show less correlation.

**CHAPTER-4**

**EXPERIMENT AND RESULT ANALYSIS**

In our data set we tried different ML Models for predictions like

1) Linear Regression

2) Multi Linear Regression

3) Logistic Regression

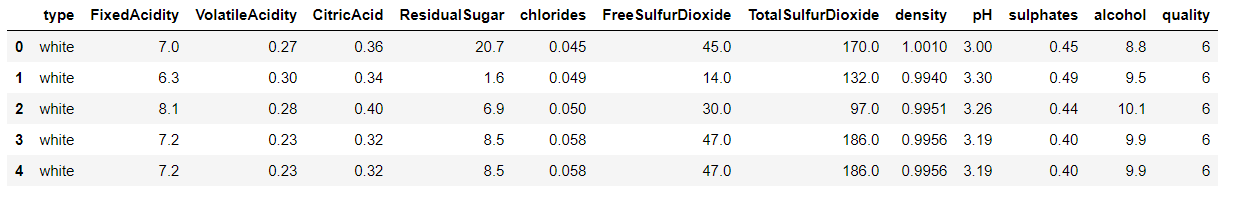
4) Decision Tree

5)Random Forest Classifier

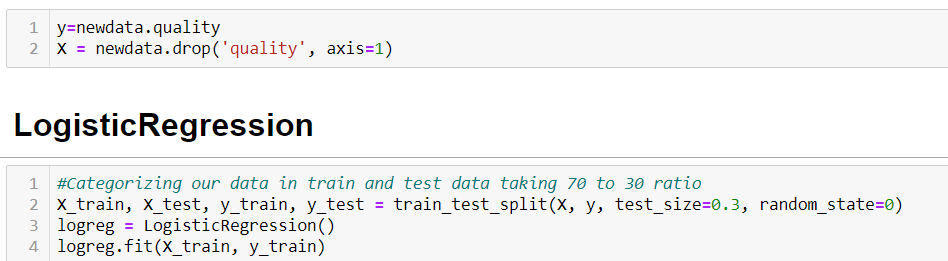
Linear Regression and Multi Linear Regression models were not that effective as they give a prediction score of nearly 30 % in our dataset so we didn’t add them.

We used Logistic Regression, Decision tree &Random Forest Models who also don’t give much accuracy score but descent amt of 57, 55 & 57%. And at some values it is around 60-65% also. It is been seen accuracy above 50% is considered fine

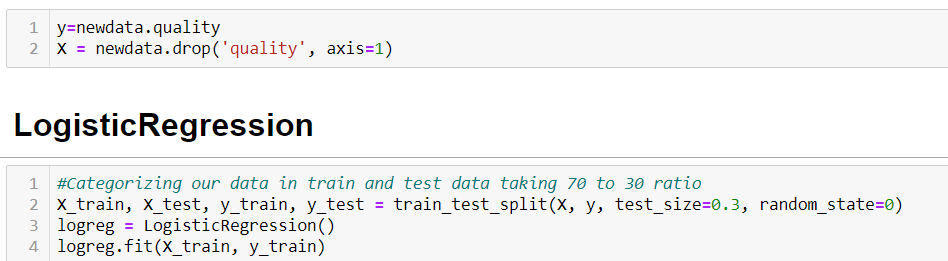
**Train Test dataset :**



In our dataset we have taken dependent variable y as quality and independent variable x as rest of the attributes of data dropping quality



We have taken test train ratio of 30 / 70



**Logistic Regression :**

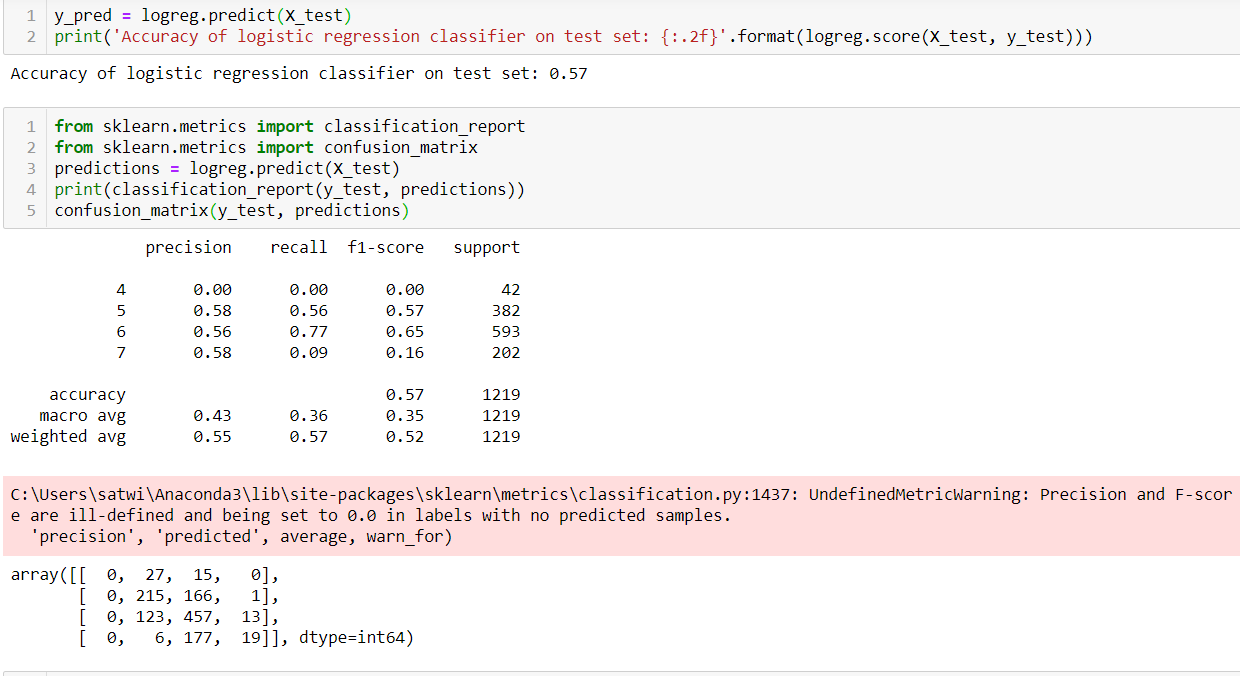
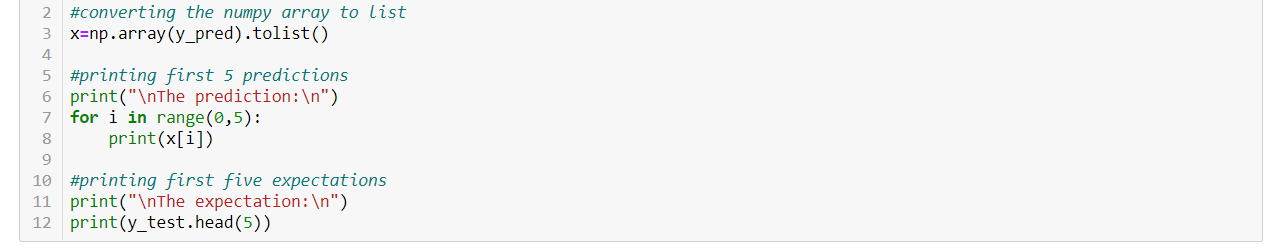
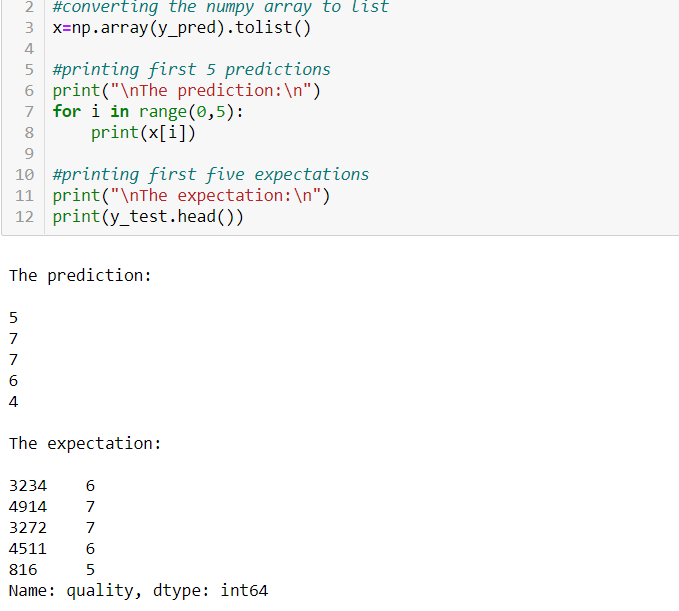


Table-1.3 Logistic Regression Result

For analyzing our prediction more easily we converted numpy array to list and print out first 5 values and compared to expected values.



To check we took out first 5 expected datavalues and matched with predicted values of first 5 we get 3 datavalues that are predicted right and rest values have slight difference which makes it upto 60% prediction of our dataset. At quality of 6 its accuracy score is 65%



**DecisionTree:**

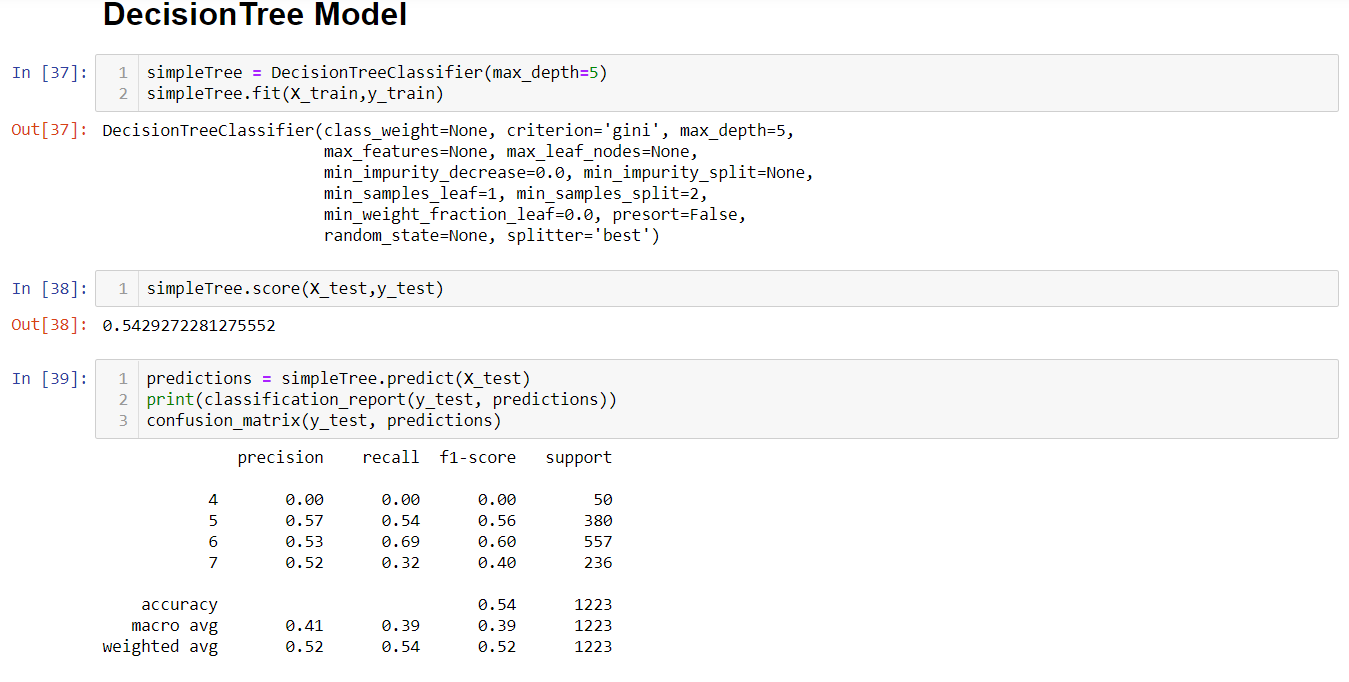


Table-1.4 Decision Tree Result

We get an accuracy of 54% with the help of Decision Tree model. At somevalues of quality it is even 60 % accuracy

**Random Forest Model :**

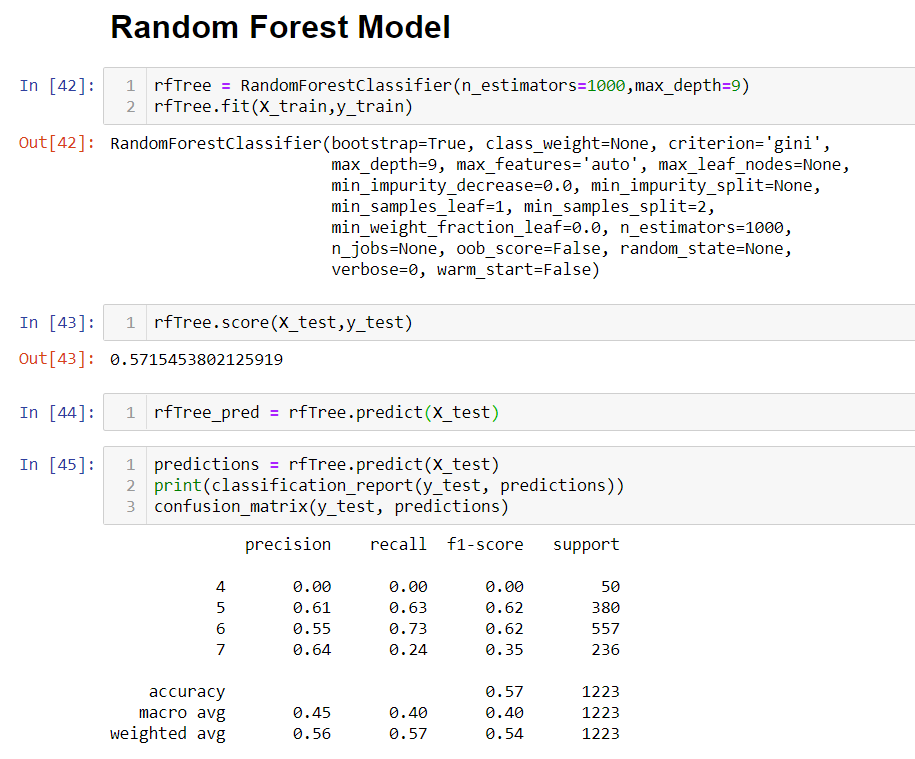


Table-1.5 Random Forest Result

We get an accuracy of 57% with the help of Random Forest Classifier model. At some values of quality it is even 60 -62% accuracy.

**CHAPTER 5**

**CONCLUSION**

**DISCUSSION AND FUTURE WORK :**

The specific objective of this study is to analyze how physicochemical properties as percentage of alcohol, The content of chlorides, sulfates, etc., varies the quality of the wine. This study analyzes the types and quality of wine with the Various physicochemical variables. Two data sets were created, using samples of red and white wine. Of thirteen attributes, the statistically significant attribute that influences.The quality of the wine is an essential finding. The model that highlights the significant attribute in both sets. This result useful in production and quality prediction when studying those attributes Analyze the type of wine using logistics regression and quality by three machine learning algorithms like decision tree, random forest and extreme gradient Boost The results obtained are more accurate than Previous techniques.

The interest has been increased in wine industry in recent years which demands growth in this industry. Therefore, companies are investing in new technologies to improve wine production and selling. In this direction,wine quality certification plays a very important role for both processes and it requires wine testing by human experts.

In end we get an accuracy of 57% with the help of Random Forest Classifier model. At some values of quality it is even 60 -62% accuracy.

In future, large dataset can be taken for experiments and other machine learning techniques may be explored for wine quality prediction. Also we can take dataset of different countries to predict the accuracy of this model.

**REFERENCES**

1- Ough, C. S., and V. L. Singleton. "Wine Quality Prediction from Juice Brix/Acid Ratio and Associated Compositional Changes forWhite Riesling'andCabernet Sauvignon'." *American Journal of Enology and Viticulture* 19, no. 3 (1968): 129-138.

2- Frank, I. E., and Bruce R. Kowalski. "Prediction of wine quality and geographic origin from chemical measurements by parital least-squares regression modeling." *Analytica Chimica Acta* 162 (1984): 241-251.

3- Marais, J., F. Calitz, and P. D. Haasbroek. "Relationship between microclimatic data, aroma component concentrations and wine quality parameters in the prediction of Sauvignon blanc wine quality." *South African Journal of Enology and Viticulture* 22, no. 1 (2001): 22-26.

4- Morakul, Sumallika, Jean-Roch Mouret, Pamela Nicolle, Ioan Cristian Trelea, Jean-Marie Sablayrolles, and Violaine Athes. "Modelling of the gas–liquid partitioning of aroma compounds during wine alcoholic fermentation and prediction of aroma losses." *Process Biochemistry* 46, no. 5 (2011): 1125-1131.

5- Gasso, Gilles. "Logistic regression." (2019).

6- Wright, Raymond E. "Logistic regression." (1995).

7- Utgoff, Paul E., Neil C. Berkman, and Jeffery A. Clouse. "Decision tree induction based on efficient tree restructuring." *Machine Learning* 29, no. 1 (1997): 5-44.

8- Magerman, David M. "Statistical decision-tree models for parsing." *arXiv preprint cmp-lg/9504030* (1995).

9- Pal, Mahesh. "Random forest classifier for remote sensing classification." *International journal of remote sensing* 26, no. 1 (2005): 217-222.

10-Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." *R news* 2, no. 3 (2002): 18-22.

40