

**Institute of System Science**

SA4108: SA50 - Mobile Application Development

Batch: SA50

Machine Learning Continuous Assessment

BY

Chua Khiong Kiat A0055647X

Rohan Ninad Mhatre A0214918B

Sheryl Teo Yuan Ting A0142771H

Shwe Yamone Oo A0214893X

Hlaing Thinn Phyu A0214902N

Li Xin Qi A0214921M

Arjun Chirumalle A0214879N

Zhong Yanbin A0214872A

Contents

[1. Linear Regression / Time Series Forecasting 5](#_Toc41145176)

[1.1. Introduction 5](#_Toc41145177)

[1.2. Data Dictionary 6](#_Toc41145178)

[1.3. The Aim of Analysis 6](#_Toc41145179)

[1.4. General Information of the Data 7](#_Toc41145180)

[1.5. Data Exploration 7](#_Toc41145181)

[1.6. Checking for null values 9](#_Toc41145182)

[1.7. Generating the correlation matrix 9](#_Toc41145183)

[1.8. Feature Engineering 10](#_Toc41145184)

[1.9. Linear Regression 10](#_Toc41145185)

[2. Classification Techniques 12](#_Toc41145186)

[2.1. Introduction 12](#_Toc41145187)

[2.2. The Aim of Analysis 13](#_Toc41145188)

[2.3. General Information of the Data 14](#_Toc41145189)

[2.4. Data Exploration 16](#_Toc41145190)

[2.5. Checking for Null Values 17](#_Toc41145191)

[2.6. Filling of the Row Data 17](#_Toc41145192)

[2.7. Getting information on the correlation list after filling null 17](#_Toc41145193)

[2.8. Logistic Regression 19](#_Toc41145194)

[2.8.1. Base Comparison 19](#_Toc41145195)

[2.8.2. Sensitivity Analysis 19](#_Toc41145196)

[2.8.3. Observation 20](#_Toc41145197)

[2.8.4. Discussion 20](#_Toc41145198)

[2.9. K-NN Classification 21](#_Toc41145199)

[2.9.1. Base Comparison 21](#_Toc41145200)

[2.9.2. Sensitivity Analysis 21](#_Toc41145201)

[2.9.3. Observation 22](#_Toc41145202)

[2.9.4. Discussion 23](#_Toc41145203)

[2.10. Decision Trees 24](#_Toc41145204)

[2.10.1. Base Comparison 24](#_Toc41145205)

[2.10.2. Sensitivity Analysis 24](#_Toc41145206)

[2.10.3. Observation 26](#_Toc41145207)

[2.10.4. Discussion 28](#_Toc41145208)

[2.10.5. Neural Network 30](#_Toc41145209)

[2.10.6. Base Comparison 30](#_Toc41145210)

[2.10.7. Sensitivity Analysis 30](#_Toc41145211)

[2.10.8. Observation 34](#_Toc41145212)

[2.10.9. Discussion 34](#_Toc41145213)

[2.11. KMeans 35](#_Toc41145214)

[2.11.1. Read the dataset 35](#_Toc41145215)

[2.11.2. Find the optimum number of clusters for k-means 35](#_Toc41145216)

[2.11.3. Create a k-means classifier with clusters = 3 and polt 36](#_Toc41145217)

[2.11.4. Plotting the centroids of the clusters 36](#_Toc41145218)

[2.11.5. View the relationship between quality and free sulfur dioxide/total sulfur dioxide(Actual Categorization) 37](#_Toc41145219)

[2.11.6. Relationship between quality range, free sulfur dioxide/total sulfur dioxide 38](#_Toc41145220)

[2.12. K-means 3D 39](#_Toc41145221)

[2.12.1. Read the dataset 39](#_Toc41145222)

[2.12.2. Find the optimum number of clusters for k-means 39](#_Toc41145223)

[2.12.3. create a k-means classifier with clusters = 3 and polt 39](#_Toc41145224)

[2.12.4. Plotting the centroids of the clusters 40](#_Toc41145225)

[2.12.5. View the relationship between quality and free sulfur dioxide/total sulfur dioxide/density (Actual Categorization) 40](#_Toc41145226)

[2.12.6. Divided quality into range 41](#_Toc41145227)

[2.12.7. View the relationship between quality in range and free sulfur dioxide/total sulfur dioxide(Actual Categorization) 42](#_Toc41145228)

[2.13. DBSCAN 43](#_Toc41145229)

[2.13.1. Read the dataset 43](#_Toc41145230)

[2.13.2. Set parameters and aggregate 43](#_Toc41145231)

[2.13.3. Plot 43](#_Toc41145232)

[2.14. DBSCAN -3D 44](#_Toc41145233)

[2.14.1. Read the dataset 44](#_Toc41145234)

[2.14.2. Set parameters and aggregate 44](#_Toc41145235)

[2.14.3. plot 44](#_Toc41145236)

[2.15. Comparison between Different Machine Learning (Classification) Model 46](#_Toc41145237)

[2.15.1. Observation 46](#_Toc41145238)

[2.15.2. Discussion / What we have learnt 49](#_Toc41145239)

[2.16. Comparison between Unsupervised Techniques 50](#_Toc41145240)

[2.16.1. Observation 50](#_Toc41145241)

[2.16.2. Discussion / What we have learnt 51](#_Toc41145242)

[2.17. Conclusion 52](#_Toc41145243)

[2.18. Appendix 53](#_Toc41145244)

[2.18.1. Decision Tree 53](#_Toc41145245)

[2.18.2. Feature Engineering 61](#_Toc41145246)

# Linear Regression / Time Series Forecasting

## Introduction

The dataset was downloaded from Kaggle.

The link from the file is as follows:

https://www.kaggle.com/swathiachath/kc-housesales-data

File Name: kc\_house\_data.csv

The dataset consisted of historic data of houses sold between May 2014 to May 2015 in King County, an area in Washington State, and Seattle, Washington. This data was published/released under CC0: Public Domain. The dataset consists of 21 variables and 21613 observations, and features data across May 2014 to May 2015.

The dataset is primarily viewed for regression analysis, where price is dependent on various features such as bedrooms, bathrooms, the total size of the property (in terms of living and lot) and floors. Other features related to geographical features, such as location and waterfront, also contribute to the price of the property.

It is to note that the dataset contains no missing values.

## Data Dictionary

|  |  |  |
| --- | --- | --- |
| S/N | Data Column Title | Description |
| 1 | Id | A notation for the house |
| 2 | Date | The date the house was sold |
| 3 | Price | The prediction target of the price of the house |
| 4 | Bedrooms | The number of bedrooms in the house sold |
| 5 | Bathrooms | The number of bathrooms in the house sold |
| 6 | Sqftliving | The square footage of the home |
| 7 | Sqftlot | The square footage of the lot |
| 8 | Floors | The total floors (levels) in the house |
| 9 | Waterfront | Whether the house has a view to a waterfront |
| 10 | View | Whether the house has been viewed before or not |
| 11 | Condition | How good the condition of the house is (overall) |
| 12 | Grade | The overall grade given to the housing unit (based on the King County grading system |
| 13 | Sqftabove | The square footage of the house apart from the basement |
| 14 | Sqftbasement | The basement square footage of the basement |
| 15 | Yrbuilt | The year the house was built |
| 16 | Yrrenovated | The year the house was renovated |
| 17 | Zipcode | The zipcode of the house |
| 18 | Lat | The latitude of the house |
| 19 | Long | The longitude of the house |
| 20 | Sqftliving15 | Living room area in 2015 |
| 21 | Sqftlot15 | The lot size area in 2015 |

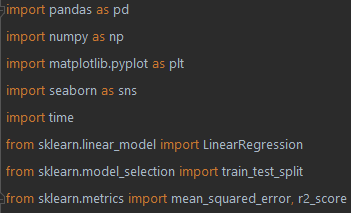
## The Aim of Analysis

The aim of the analysis is to predict the prices of house sold in King County. We start with running a simple linear regression analysis of prices on size of living space, and then running a multi-variate regression analysis of price on all other factors.

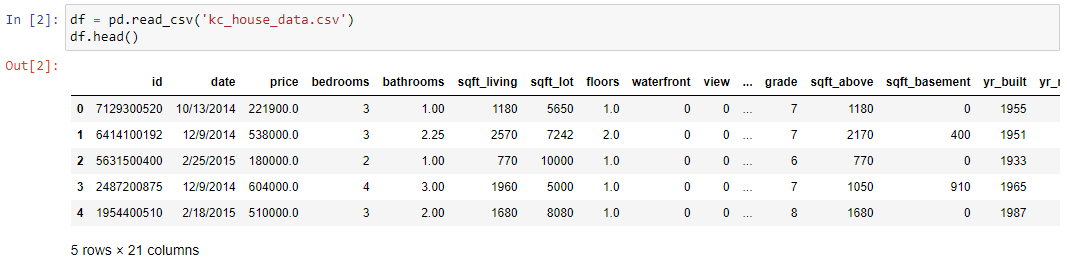
## General Information of the Data

## Data Exploration

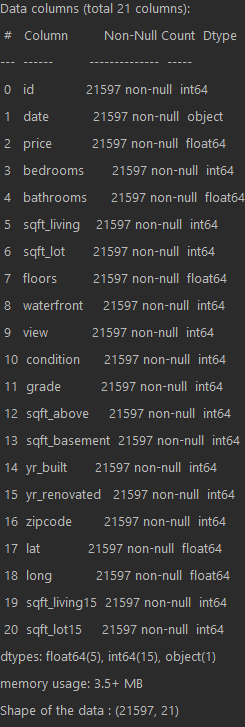
* Importing Packages



* Reading the csv file



* Getting information from the csv



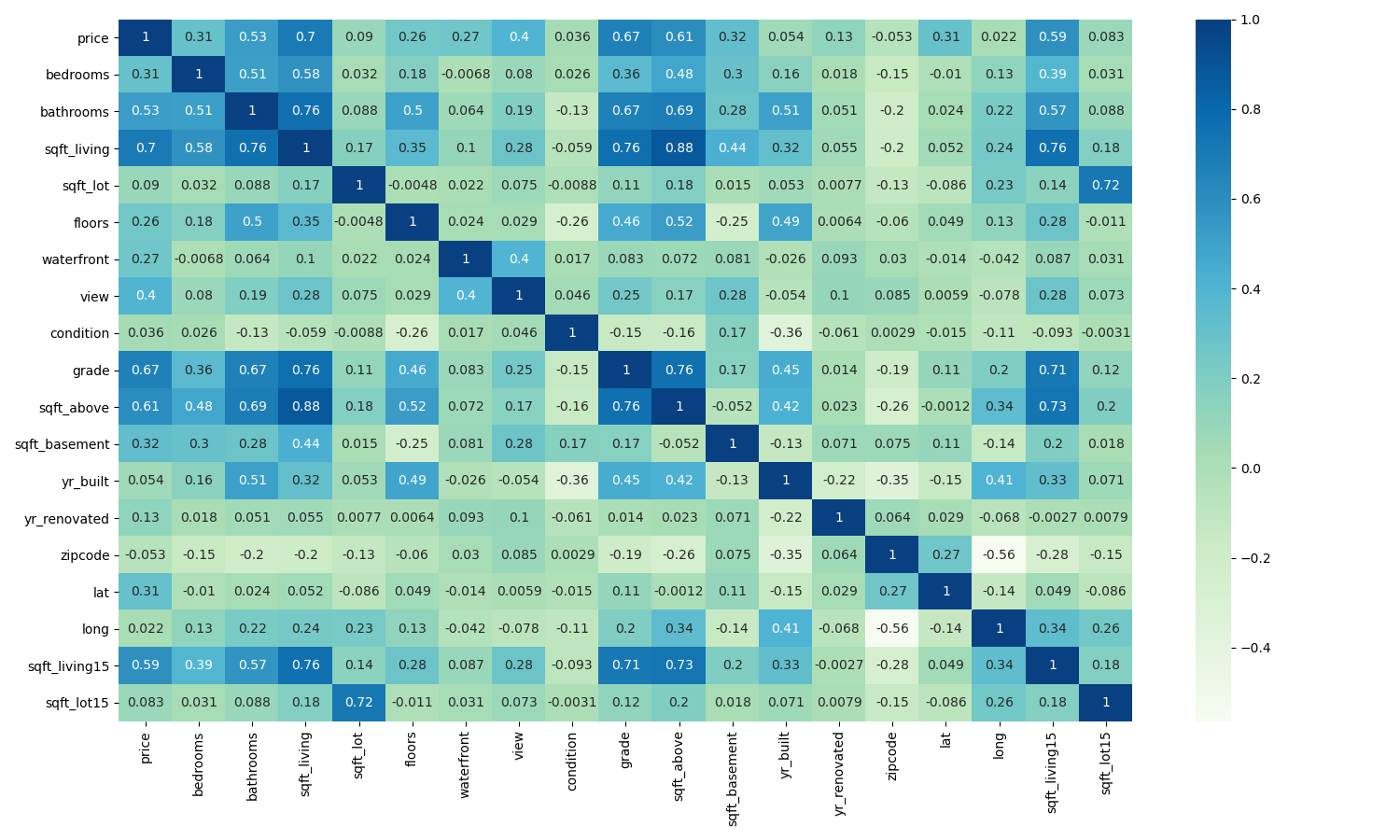
## Checking for null values



From here, we can see that the data has no missing values and hence, we are not required to perform cleaning of the database or data engineering for this dataset.

## Generating the correlation matrix

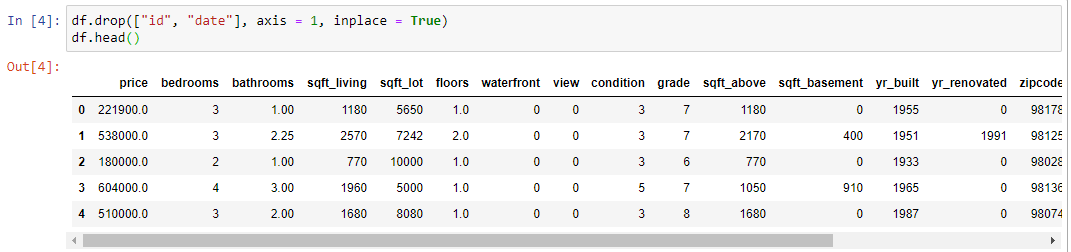




From the plot, we can see that there is a strong correlation between price and the size of the living space of a home (0.701917), followed by the grade of the housing unit (0.667951) and the size of the house apart from the basement (0.605368).

## Feature Engineering

We decided to drop variables that we are not using – id and date



## Linear Regression

Model 1: To Predict price of houses based on living area (sqft\_living)

Model 2: To Predict price of houses based on selected features (bedrooms, bathrooms, sqft\_living, sqft\_lot, floors, zipcode)

Model 3: To Predict price of houses based on all other features

Assumptions:

1. Duration of model is based on purely model fitting
2. Column expressed in array from 0 – 18  
   [0,1,2,3,4,5,6,7,8,9,10,11,18] is [“price”, ….. , “sqft\_lot15”] (after dropping unused rows)
3. All columns considered except for Price

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model 1 | Model 2 | Model 3 |
| Column Input | [3] | [1,2,3,4,5,14] | [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18] |
| Column Output | [0] | [0] | [0] |
| Duration (seconds) | 0.0857694149017334 | 0.06083083152770996 | 0.042882680892944336 |
| MSE (test) | 67361730627.54034 | 64099626980.09557 | 39350992633.3841 |
| R-squared (test) | 0.49820266085938314 | 0.5225030301497684 | 0.7068628847267018 |
| Train Accuracy | 0.490290636454328 | 0.5118544402491527 | 0.6970558577263126 |
| Test Accuracy | 0.49820266085938314 | 0.5225030301497684 | 0.7068628847267018 |

Observations/Analysis

1. Model 1 shows us that despite having a high correlation between price and living space, the model turned out a low r-squared value and very low accuracy in training and testing. This indicates to us that there are likely to be more features that affects the selling price of a house.
2. In Model 2, we see that by adding a few other variables, the results have only increased slightly. It is possible that the features used in the analysis for Model 2 did not greatly affect the prices of houses sold any more than living space did.
3. In Model 3, we note that after using all other features, our r-squared value has increased significantly (from 0.498 to 0.706). There is also a noted increase in training and testing accuracy.

It was initially suspected that with more features, we would see longer training time for our model. However, it is worth noting that the duration taken to train the model decreases with more features.

Yet, we note that the duration difference for the model is negligible.

Impact of data engineering and feature engineering

From our analysis of the model, we see that the mean squared error for the model is large as the value of prices in the model is large, hence we will use the r-squared value to evaluate our model. We have decided to present the last model (Model 3) as it showcases the highest r-squared value, hence showing that 70% of the variation in the various features of a house is able to explain the variation in house prices.

Other observations

While the change in training time for the model is negligible, we note that there is a decrease in training time for the model when we add in more features. This highlights to us that we should not think of omitting features from our model unless they are highly correlated with other independent features.

# Classification Techniques

## Introduction

The dataset was downloaded from the UCI Machine Learning Repository.

The link from the file is as follow:

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

File Name: winequalityN.csv

The dataset contains information related to red and white variants of the Portuguese "Vinho Verde" wine. The reference [Cortez et al., 2009]. 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.).

These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant.

The dataset contains null values which needs to do data cleaning to ensure a proper set of data can be used.

## The Aim of Analysis

There are a few aims to the current analysis:

* This study aims to search for the elements which effects WINE QUALITY/WINE TYPE by using multiclass decision classification methods.
* 3 outputs have been identified to be used as a machine learning target.
  + Type
  + Quality
  + Quality Group (Derived from Quality)
* Classification Methods to include:
  + Logistic Regression
  + K-NN Classification
  + Decision Trees
  + Neural Network
* Each Classification method will be accompanied with at least 1 Unsupervised Techniques
  + PCA
* A separate analysis with unsupervised learning using k-Means and Agglomerative Clustering will be used to find a good clustering pattern.

## General Information of the Data

**Type**: Two types of wines such as red wine and white wine.

**Fixed acidity**: Fixed acids include tartaric, malic, citric, and succinic acids which are found in grapes (except succinic)

Acids are one of the fundamental properties of wine and contribute greatly to the taste of the wine, Acidity in food and drink tastes tart and zesty. Tasting acidity is also sometimes confused with alcohol. Wines with higher acidity feel lighter-bodied because they come across as “spritzy”. Reducing acids significantly might lead to wines tasting flat. If you prefer a wine that is richer and rounder, you enjoy slightly less acidity.

**Volatile acidity**: These acids are to be distilled out from the wine before completing the production process. It is primarily constituted of acetic acid though other acids like lactic, formic and butyric acids might also be present. Excess of volatile acids are undesirable and lead to unpleasant flavour.

**Citric acid**: This is one of the fixed acids which gives a wine its freshness. Usually most of it is consumed during the fermentation process and sometimes it is added separately to give the wine more freshness.

**Residual sugar**: This typically refers to the natural sugar from grapes which remains after the fermentation process stops, or is stopped.

**Chlorides**: Chloride concentration in the wine is influenced by terroir and its highest levels are found in wines coming from countries where irrigation is carried out using salty water or in areas with brackish terrains.

**Free sulfur dioxide**: This is the part of the sulphur dioxide that when added to a wine is said to be free after the remaining part binds. Winemakers will always try to get the highest proportion of free sulphur to bind. They are also known as sulfites and too much of it is undesirable and gives a pungent odour.

**Total sulfur dioxide**: This is the sum total of the bound and the free sulfur dioxide. This is mainly added to kill harmful bacteria and preserve quality and freshness. There are usually legal limits for sulfur levels in wines and excess of it can even kill good yeast and give out undesirable odour.

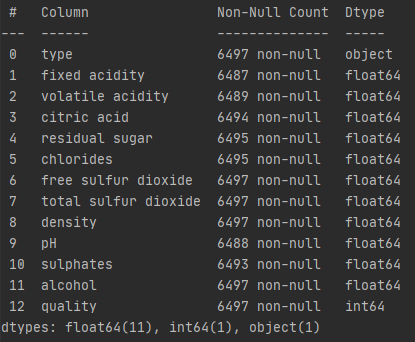
**Density**: This can be represented as a comparison of the weight of a specific volume of wine to an equivalent volume of water. It is generally used as a measure of the conversion of sugar to alcohol.

**pH**: Also known as the potential of hydrogen, this is a numeric scale to specify the acidity or basicity the wine. Fixed acidity contributes the most towards the pH of wines. You might know, solutions with a pH less than 7 are acidic, while solutions with a pH greater than 7 are basic. With a pH of 7, pure water is neutral. Most wines have a pH between 2.9 and 3.9 and are therefore acidic.

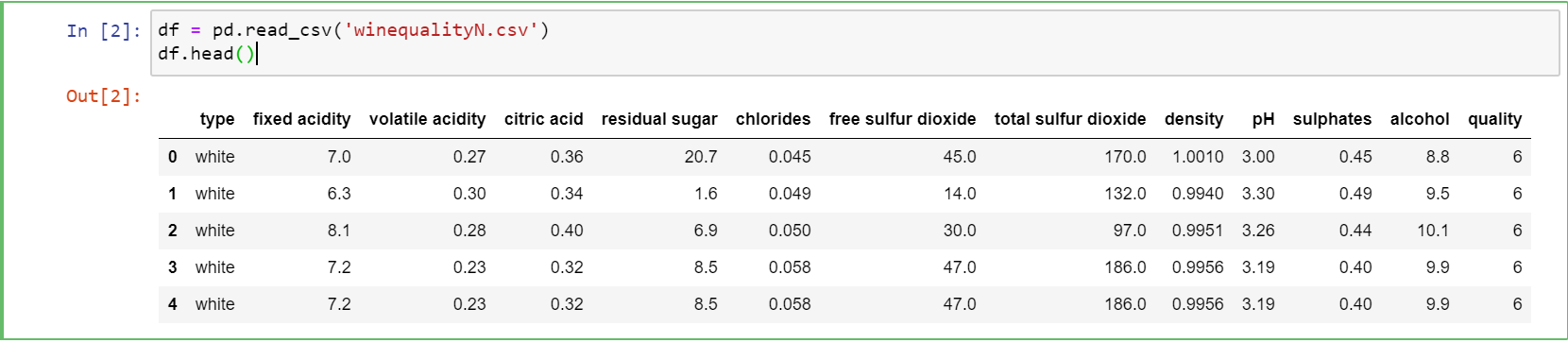
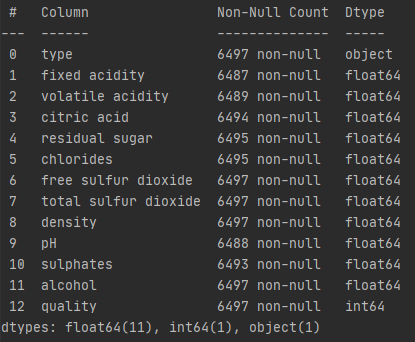
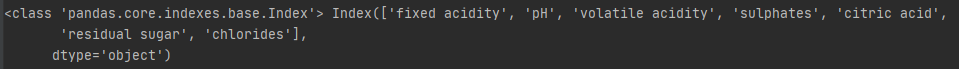
**Sulphates**: These are mineral salts containing sulfur. Sulphates are to wine as gluten is to food. They are a regular part of the winemaking around the world and are considered essential. They are connected to the fermentation process and affects the wine aroma and flavour.

**Alcohol**: It's usually measured in % vol or alcohol by volume (ABV).

**Quality**: Wine experts graded the wine quality between 0 (very bad) and 10 (very excellent). The eventual quality score is the median of at least three evaluations made by the same wine experts.



## Data Exploration

* Importing Packages  
  
* Reading the CSV file  
  
* Getting the information from the csv  
  
* Getting Column Head  
  

## Checking for Null Values

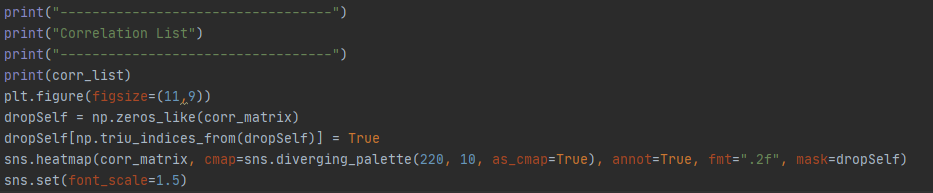
* Getting the number of columns that have null values and the percentage with respect to the total number rows  
  
* Checking NULL values in data.



## Filling of the Row Data



## Getting information on the correlation list after filling null





Wine quality has a higher correlation with alcohol i.e. 0.44. The correlation with density, volatile acidity, chlorides is -0.31, -0.27, -0.2 respectively. The quality label has no significant correlation with other features.

## Logistic Regression

### Base Comparison

Model 1: To Predict Wine Type

Model 2: To Predict Quality

Model 3: To Predict Quality Group

We have trained each model with all the input features in the dataset and measured the metrics for each model. The same is achieved by implementing the dimensionality reduction technique on the input features and feeding the newly generated principal components as input to each model. The findings and metrics are tabulated below.

Assumptions:

1. Duration of model is based on purely model fitting
2. Column expressed in array from 0 – 13  
   [0,1,2,3,4,5,6,7,8,9,10,11,12,13] is [“Wine Type”, ….., “Quality”, “Quality Group”]
3. PCA Refers to the no. of components for each model run.



### Sensitivity Analysis

#### Sensitivity Analysis 1

### Observation

#### Sensitivity Analysis 1

* It is noted that comparing models that utilize PCA has overall much lesser time required to train the model.
* Comparing Model 2 and 3, model 3 has an estimated 33% reduction in model fitting duration after binning the quality into 3 groups.
* It is noted that Model 2 which has 10 outputs (Quality Class) have a lower accuracy (estimated. 48% to 55%).
* The impact of PCA on the models in terms of accuracy is insignificant (this is true for all 3 models)

### Discussion

#### Sensitivity Analysis 1: Impact of Data Engineering and Feature Engineering

* It is noted that with feature engineering (e.g. PCA), PCA can reduce training time significantly as this is consistent as lesser data is being passed through to the model (11 columns of data compared to 3 columns).
* It is noted that PCA in general can increase accuracy (most of the time), this is especially true when the base training model has a high accuracy to start with (e.g. Model 1 and Model 3).
* It is evident that model training with PCA is beneficial when the base model inherently has a high accuracy value. Given the example in Model 1 and 3, the duration required to train the model differs as much as 500%. Given that we only use 6000 datapoints, during the event that 6 million datapoints is required. The training will be increased exponentially. Hence the conclusion that using feature engineering on Machine Learning Models is very important.

## K-NN Classification

### Base Comparison

Model 1: To Predict Wine Type

Model 2: To Predict Quality

Model 3: To Predict Quality Group

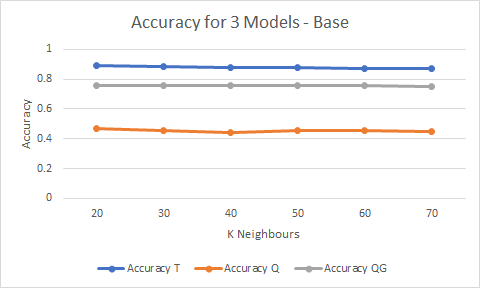
Assumptions:

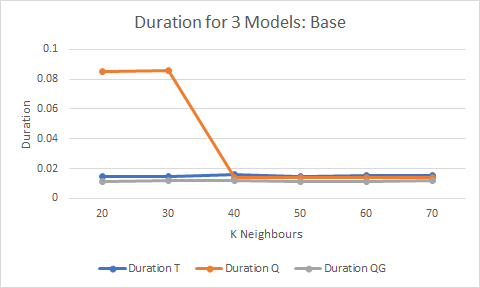
1. Duration of model is based on purely model fitting
2. Column expressed in array from 0 – 12  
   [0,1,2,3,4,5,6,7,8,9,10,11,12,13] is [“Wine Type”, ….., “Quality”,Quality\_Group]
3. All columns considered except for Wine Type, Quality & Quality Group
4. For Model 3, Quality Group is represented using data binning for quality scores [0(Low: 0 to 4),1 (Mid: 4 to 7) ,2 (High: 7 to 10)]

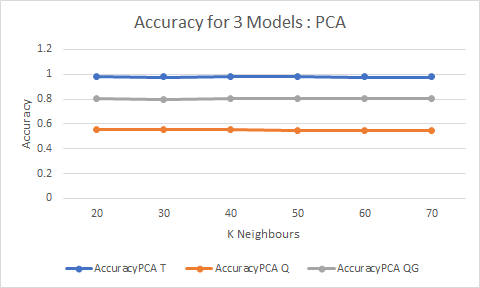
|  |  |  |  |
| --- | --- | --- | --- |
|  | Model 1 | Model 2 | Model 3 |
| Column Input | [1,2,3,4,5,6,7,8,9,10,11] | [1,2,3,4,5,6,7,8,9,10,11] | [1,2,3,4,5,6,7,8,9,10,11] |
| Column Output | [0] | [12] | [1] |
| K = number of neighbors | 50 | 50 | 50 |
| Duration | 0.014004945755004883 | 0.014005422592163086 | 0.008994102478027344 |
| Accuracy | 0.9317948717948717 | 0.4523076923076923 | 0.8046153846153846 |

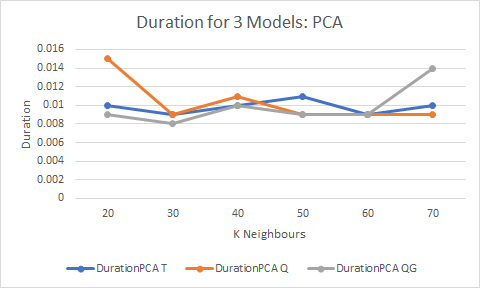
0.9897435897435898

### Sensitivity Analysis









### Observation

Our initial model only has an accuracy of 46% when predicting quality. After data engineering, accuracy of predicting quality is 75.4%.

Utilizing Principal Component Analysis for dimension reduction, we have improved accuracy of 78.8%

Explain limitations of KNN relative to results for Model 3, even with PCA.

Utilizing feature engineering, duration is reduced significantly, with 7 columns being considered instead of 11.

### Discussion

## Decision Trees

### Base Comparison

Model 1: To Predict Wine Type

Model 2: To Predict Quality

Model 3: To Predict Quality Group

Assumptions:

1. Duration of model is based on purely model fitting
2. Column expressed in array from 0 – 13  
   [0,1,2,3,4,5,6,7,8,9,10,11,12,13] is [“Wine Type”, ….., “Quality”, “Quality Group”]
3. All column inputs have been considered except wine type, quality, quality group.
4. Decision tree criterion is ‘Gini’.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model 1 | Model 2 | Model 3 |
| Column Input | [1,2,3,4,5,6,7,8,9,10,11] | [1,2,3,4,5,6,7,8,9,10,11] | [1,2,3,4,5,6,7,8,9,10,11] |
| Column Output | [0] | [12] | [13] |
| Max Depth | 3 | 3 | 3 |
| Duration | 0.0114 | 0.0089 | 0.01 |
| Accuracy | 94.9% | 53.64% | 93.33% |

### Sensitivity Analysis

### Observation

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy / Duration | | PCA (Components = 3) | Without PCA |
| Model 1 | Accuracy | 98.2% | 94.92% |
| Training Duration | 0.0198 | 0.0114 |
| Model 2 | Accuracy | 46.56% | 53.64% |
| Training Duration | 0.0019 | 0.0089 |
| Model 3 | Accuracy | 93.48% | 93.33% |
| Training Duration | 0.0009 | 0.0099 |

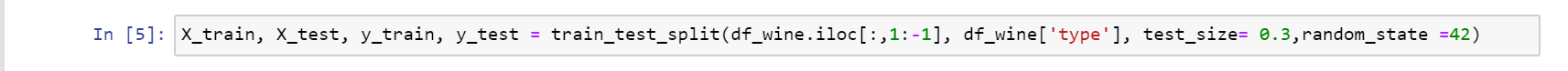
As mentioned in the above table,

* It is noted that we have better accuracy results in model 1 and model 3 by using PCA.
* In model 3 (Predicting the quality range), it gets 93.48% by using 3 components in PCA instead of 11 features in training dataset.
* Although, model 1 (predicting type) and model 3 (predicting quality range) can get higher accuracy results in comparing to without PCA, model 2 (predicting quality) have lower accuracy by using PCA.
* Using PCA reduce model training duration in model 2 and model 3.

#### Model 1 without PCA

Columns from dataset in array format [0,1,2,3,4,5,6,7,8,9,10,11,12,13

From this dataset columns, we train and test the data for model 1 decision tree classification as per below.



Then, by instantiating the decision tree model with the max\_depth = 3 , we got the accuracy score of 94.92%.

A screenshot of a social media post

Description automatically generatedA picture containing receipt

Description automatically generated

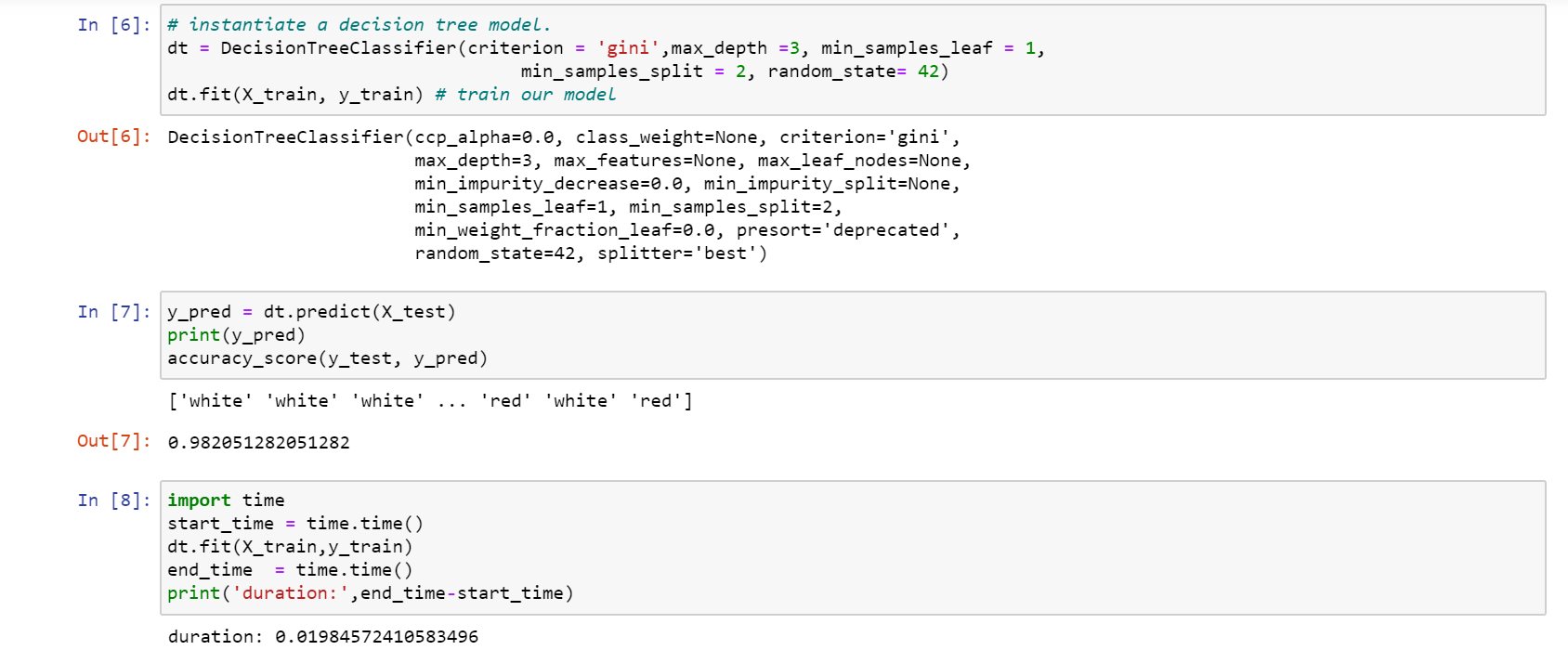
Duration of training the model



#### Model 1 with PCA

When analyzing our dataset for wine type, we found that for type to classify, many features are to be dependent variables. Thus, we tried with applying PCA(n-components = 3) to input variables before training our model. With the modified features by PCA, build a decision tree model and train the model. The accuracy score resulted is 0.982, which is better than training the model without PCA.

However, the duration of training this model with PCA(as per below) is not much different from without PCA.



When visualize the decision tree, depending on 3 features of X[0],X[1],X[2] which are extracted by PCA, it is classified which class of wine type.

A picture containing clock

Description automatically generated

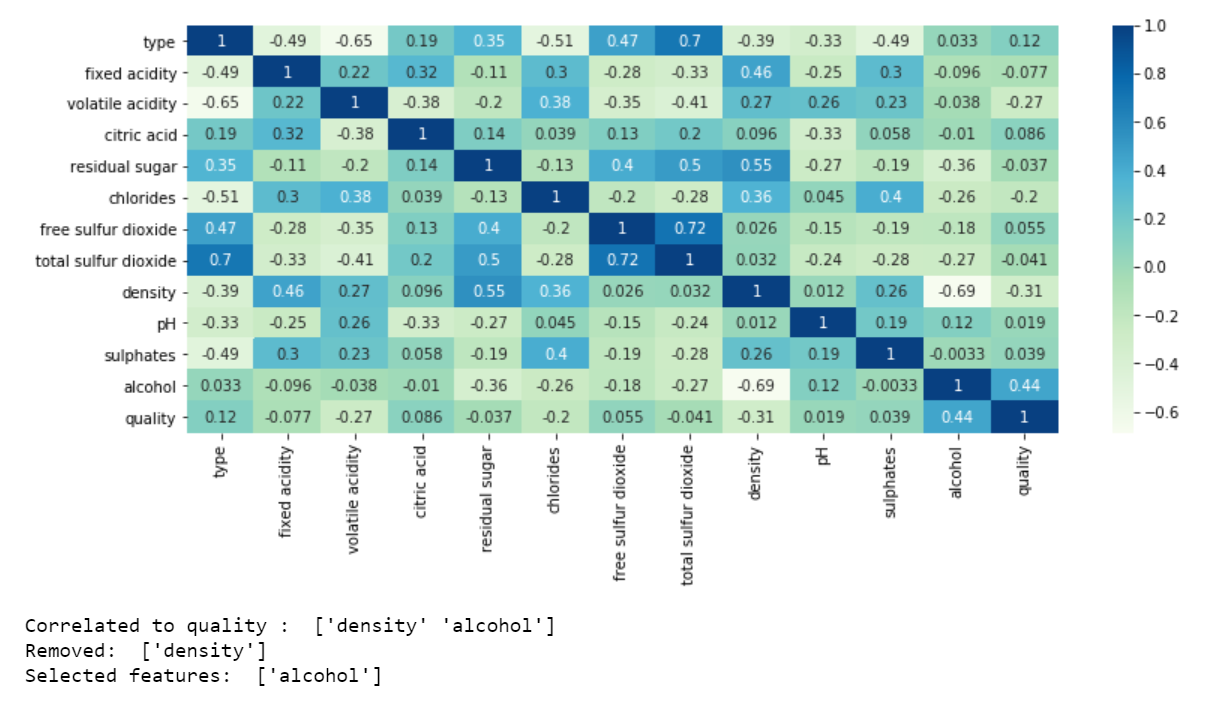
**\*\*\*Sharing the process of decision tree can be seen in appendix.**

### Discussion

#### Feature Selection by Pearson Correlation Matrix

By Using Pearson Correlation, we can know

* ‘quality’ is correlated with alcohol.
* ‘Free sulfur dioxide’ and ‘total sulfur dioxide’ is highly correlated (0.72 in our correlation matrix) and they can have almost the same effect on an independent variable.
* Therefore, when the two features are highly correlated, we can drop one of them in our feature set.
* Correlation of ‘total sulfur dioxide’ and ‘residual sugar’ is 0.5 and they have moderate correlation.
* Correlation of ‘density’ and ‘residual sugar’ is 0.55 and they also have moderate correlation.
* Therefore, our target variable(quality) is correlated with alcohol and low correlation with other features.
* Another target variable (type) is also have low correlation with citric acid and alcohol.



#### Impact of Data Engineering and Feature Engineering

In our dataset, we use two types of data engineering.

* The first one is categorical data engineering. By using categorial data engineering, we map ‘type’ category to a unique value such as white = 0 and red = 1. By using label encoding, we can easily visualize the relationship between the features and the target in our plots and matrix.
* The second one is data cleansing. Since our dataset contains missing data and NaN, we fill with a value of mean value on NaN column.

For the feature engineering, we use feature selection and feature extraction techniques.

* By using Pearson correlation matrix, we can know which features are correlated to each other.
* Since ‘free sulfur dioxide’ and ‘total sulfur dioxide’ are highly correlated with each other, removing one of these features doesn’t affect on our accuracy results.
* Moreover, dropping ‘residual sugar’ or ‘total sulfur dioxide’ will give better accuracy results since these are loosely correlated with our dependent variable ‘quality’.
* In classifying wine type, which is set as a target, dropping of highly correlated features in matrix , for example, free Sulphur dioxide and residual sugar or dropping the features which have low correlation with our target, citric acid and alcohol will give slightly better accuracy score of 98% rather than 95%.

## Neural Network

### Base Comparison

Model 1: To Predict Wine Type

Model 2: To Predict Quality

Model 3: To Predict Quality Group

Assumptions:

1. Duration of model is based on purely model fitting
2. Column expressed in array from 0 – 13  
   [0,1,2,3,4,5,6,7,8,9,10,11,12,13] is [“Wine Type”, ….., “Quality”, “Quality Group”]
3. All column inputs have been considered except wine type, quality, quality group.
4. Only 1 Hidden Layer of Activation Mode “Sigmoid” has been done for the base comparison for all data type.

Results:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model 1 | Model 2 | Model 3 |
| Column Input | [1,2,3,4,5,6,7,8,9,10,11] | [1,2,3,4,5,6,7,8,9,10,11] | [1,2,3,4,5,6,7,8,9,10,11] |
| Column Output | [0] | [12] | [13] |
| Dense Value | 500 | 500 | 500 |
| Duration | 12.161257028579712 | 12.11571193 | 11.330935001373291 |
| Accuracy | 0.9830769300460815 | 0.558461547 | 0.9215384721755981 |

### Sensitivity Analysis

#### Sensitivity Analysis 1

Graph below shows a summary of Models 1,2,3 are affected by the dense values.

Assumptions:

1. Column expressed in array from 0 – 13  
   [0,1,2,3,4,5,6,7,8,9,10,11,12,13] is [“Wine Type”, ….., “Quality”, “Quality Group”]
2. All column inputs have been considered except wine type [0], quality [12], quality group [13].

#### Sensitivity Analysis 2

Sensitivity Analysis 2 is tying to compare the same model but with implementation of PCA (e.g. PCA Component = 3 ) against the base model that didn’t apply PCA.

We have done a sensitivity analysis across all 3 models and each model with varying neurons.

### Observation

#### Sensitivity Analysis 1

* By binning the quality group, there is obvious improvement in terms of accuracy. Model 2 represents 10 classes of quality (0-10) while Model 3 represents 3 classes of quality as a group (0,1,2).
* Model 1 – No PCA shows that the no. of neurons doesn’t affect much of the accuracy.
* However, it is known that all Models will have increase duration when no. of neurons increase with no additional benefit.

#### Sensitivity Analysis 2

* For model 1 and Model 3, it is shown that PCA has higher accuracy values than the model that didn’t employ PCA.
* For model 2, the accuracy dropped.
* All models show more consistency in terms of accuracy across all no. of neurons.

### Discussion

#### Sensitivity Analysis 1: Impact of Data Engineering and Feature Engineering

* It is evident that by binning the quality group, it is found that combining certain classes (0-4,4-7,7-10) helps improve classification in terms of quality group. However, the Neural Network Model (NN) does not have enough data to learn whether it should be in a more specific group.
* The duration wasn’t affected much when increasing neurons.

#### Sensitivity Analysis 2: Impact of Data Engineering and Feature Engineering

* PCA should only be used when the base model have a relatively high accuracy (e.g. more than 0.7). When accuracy is lower as shown in Model 2 (est. 50%), the PCA does not help instead shows a lower accuracy.
* In overall, most models have slightly longer duration compared to their non PCA counterpart.
* PCA is able to show a more consistent accuracy when compared to models who does not use PCA.

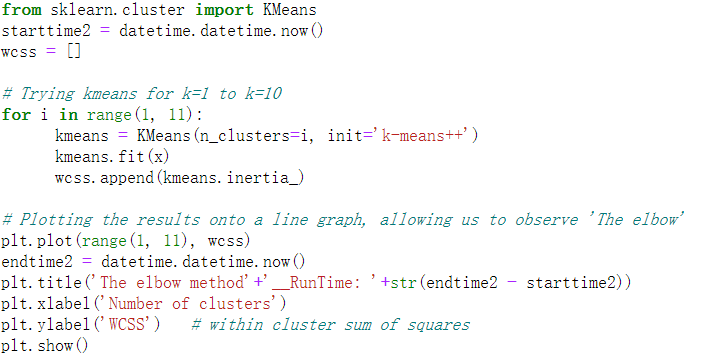
## KMeans

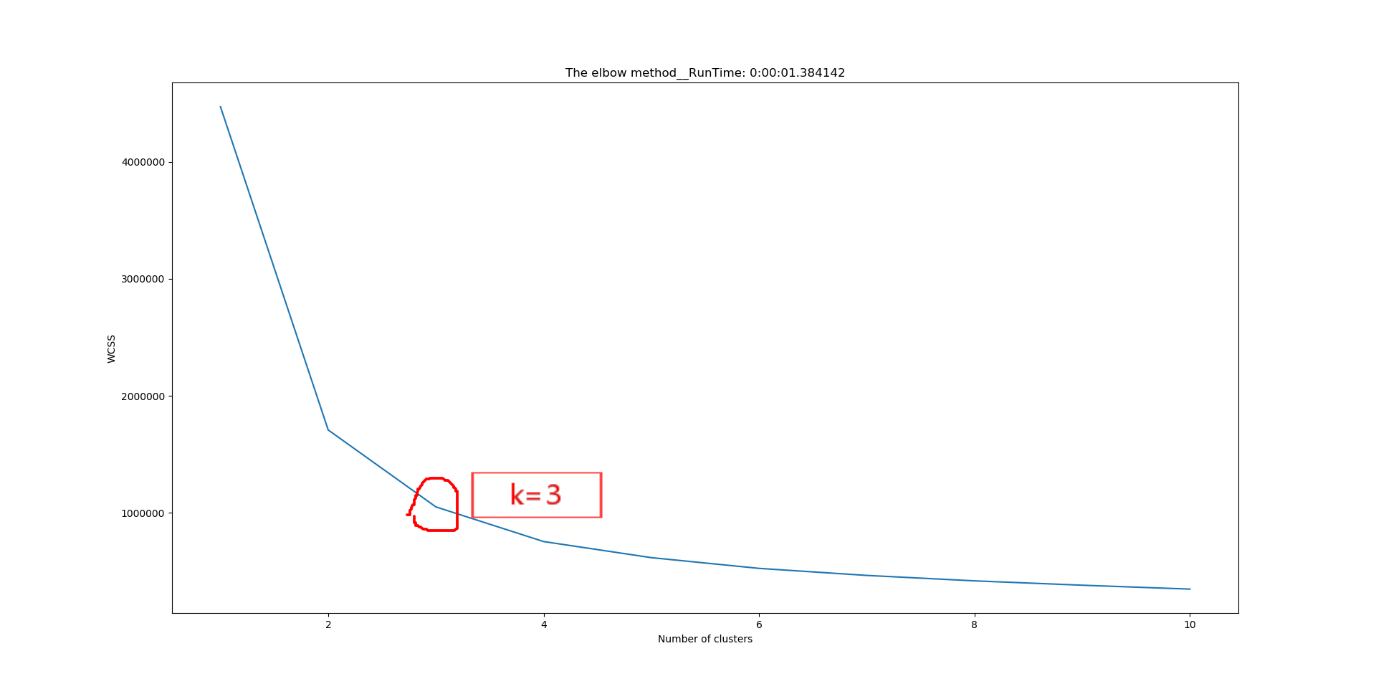
### Read the dataset



read the data in rows 0-2000, columns 6(free sulfur dioxide) and 7(total sulfur dioxide)

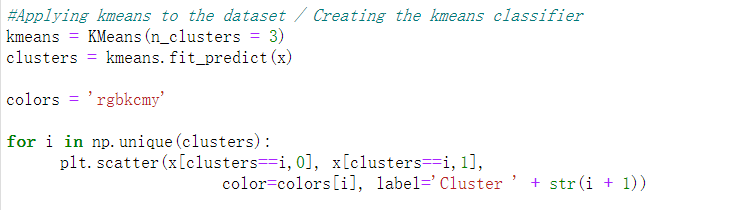
### Find the optimum number of clusters for k-means



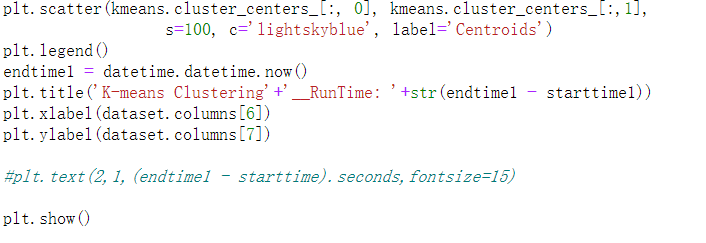


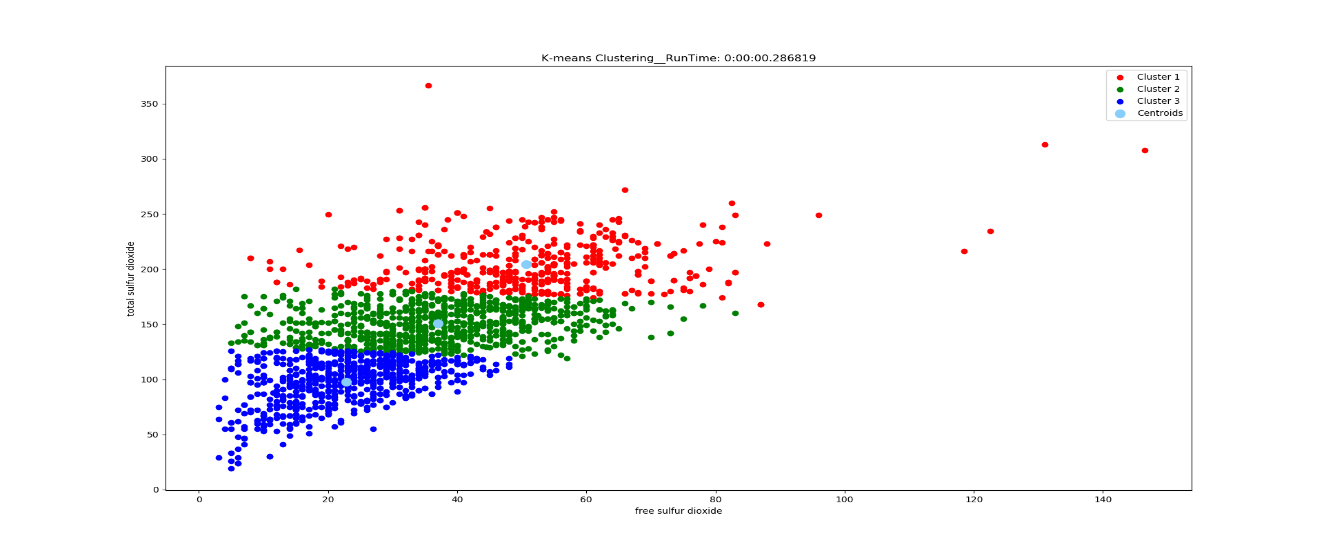
* It can be seen from the figure that the running time of the elbow graph is 1.384142 seconds.
* We can think that when K = 3, the elbow line starts to be gentle without rapid descending, so we would set the cluster value to 3.

### Create a k-means classifier with clusters = 3 and polt



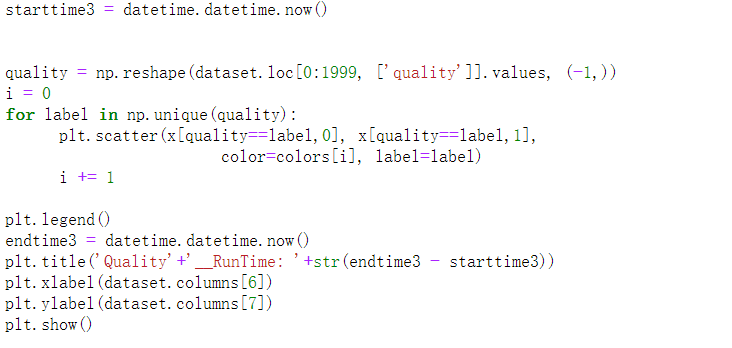
### Plotting the centroids of the clusters

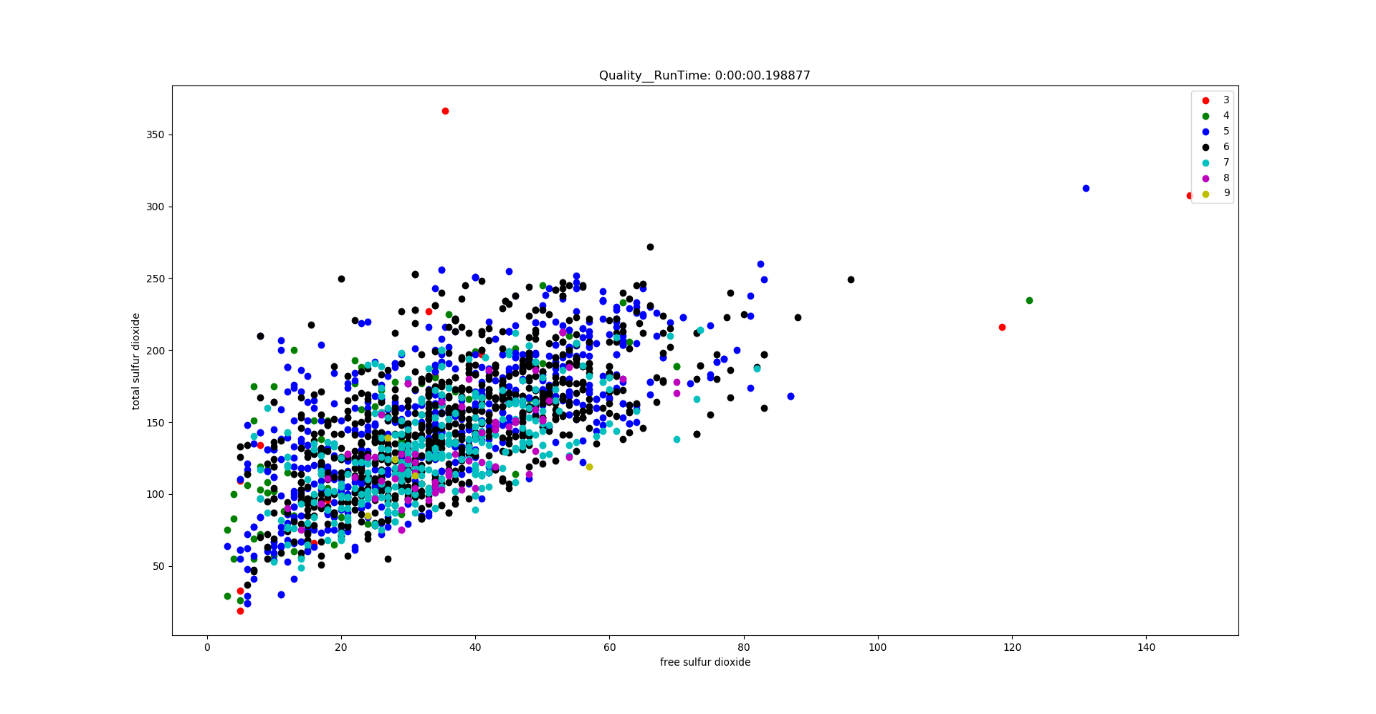




* The figure on title shows the running time is 0.286819 sec. And it contains the parameter relationship between the column 6 (free sulfur dioxide) and 7 (total sulfur dioxide).
* Calculated by the k-means algorithm, k-means cannot give us accurate conclusions because it shows no data plot group in the diagram, but through this pic we might can consider using linear regression to find the trend.

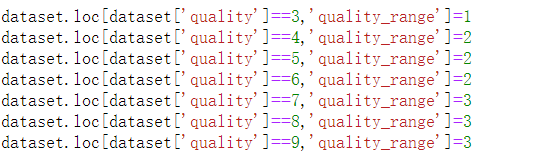
### View the relationship between quality and free sulfur dioxide/total sulfur dioxide(Actual Categorization)



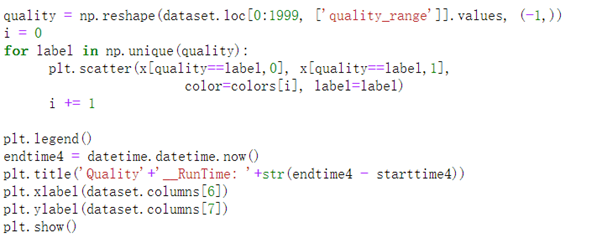


* The running time is 0.198877 sec.
* From the dataset we found that the quality of the wine ranges from 3 to 9. But it seems that there is no obvious relationship between quality and free sulfur dioxide/total sulfur dioxide.

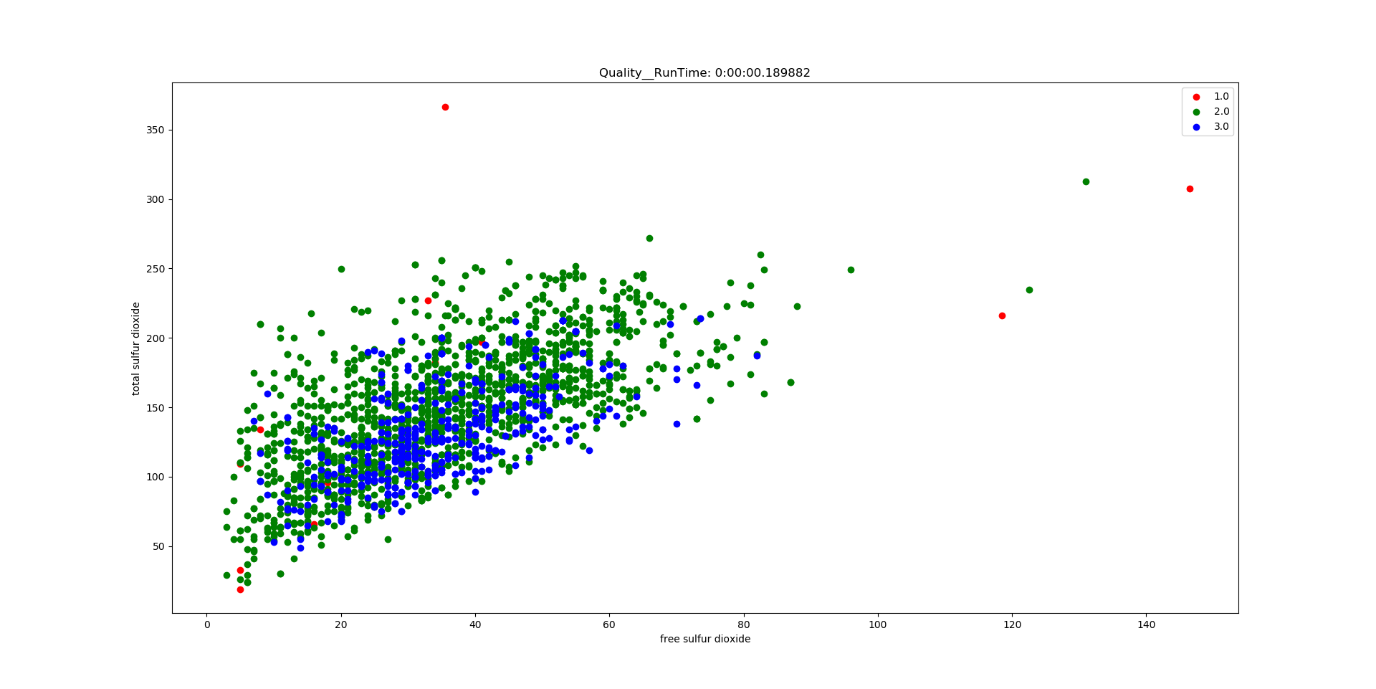
**6. Divided quality into range**



### Relationship between quality range, free sulfur dioxide/total sulfur dioxide



We try to find some extra info from using K-means algorithm. Now we are binning the quality from 3-9 into 3 ranges, quality [3-4) to range 1; quality [4-7) to range 2, quality [7-9) to range3.



When we use the ” quality ranges” to color the plot, the figure changes easier to be known, for example the plot of quality range 1 is discrete from the main group of data, and the other plots are concentrated in a group around 5-80 in X-label and 40-250 in Y-label.

But all the plot of quality range 2 and 3 are mixed together, we might need other way to find the regular.

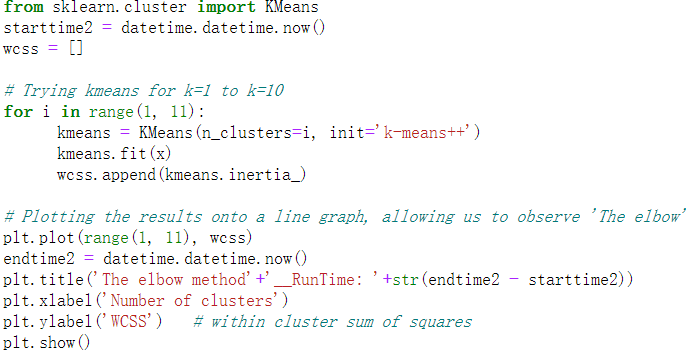
## K-means 3D

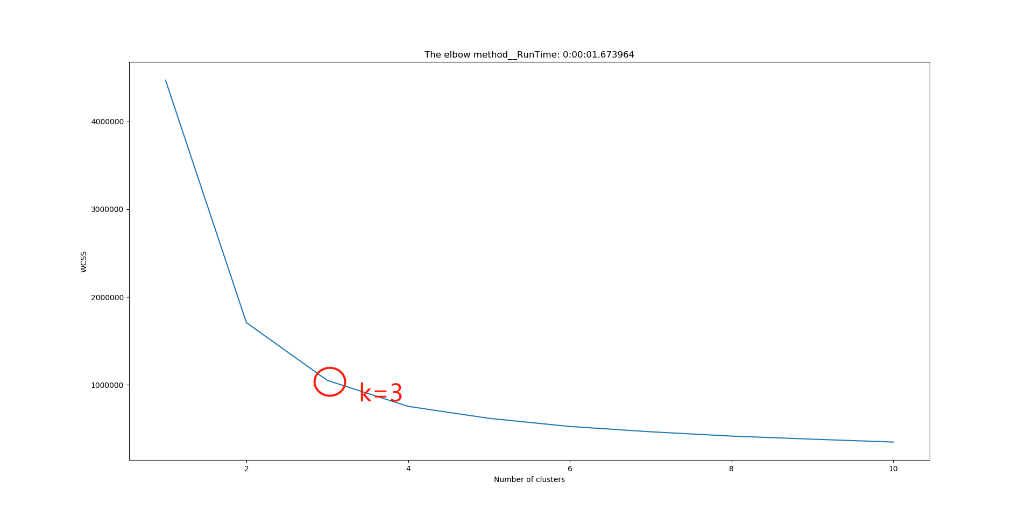
### Read the dataset



read the data in rows 0-2000, columns 6(free sulfur dioxide) and 7(total sulfur dioxide)and8(density)

### Find the optimum number of clusters for k-means

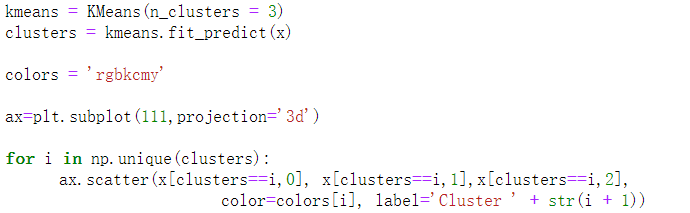




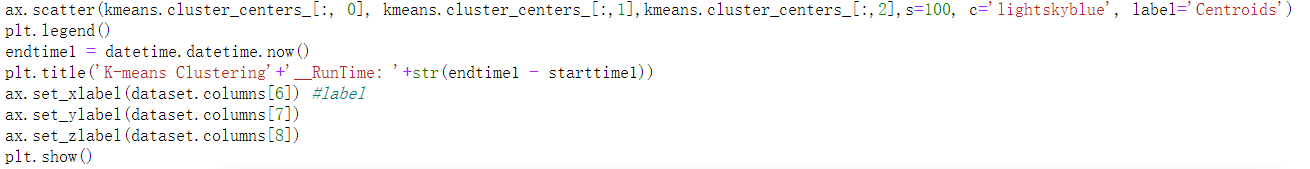
It can be seen from the figure that the running time of the elbow graph is 1.673964 seconds.

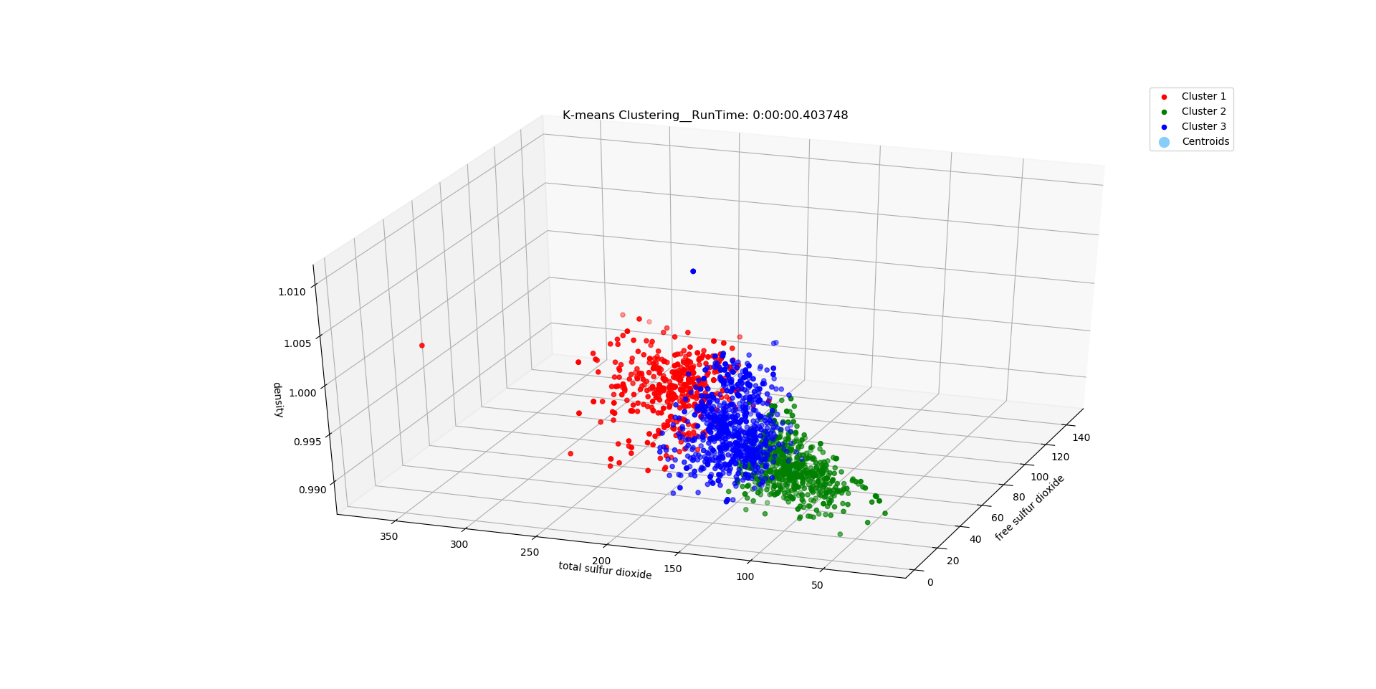
We can think that when K = 3, the elbow line starts to be gentle without rapid descending, so we would set the cluster value to 3.

### create a k-means classifier with clusters = 3 and polt



### Plotting the centroids of the clusters

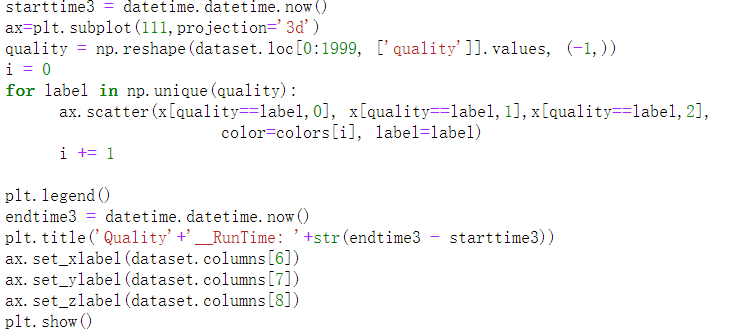


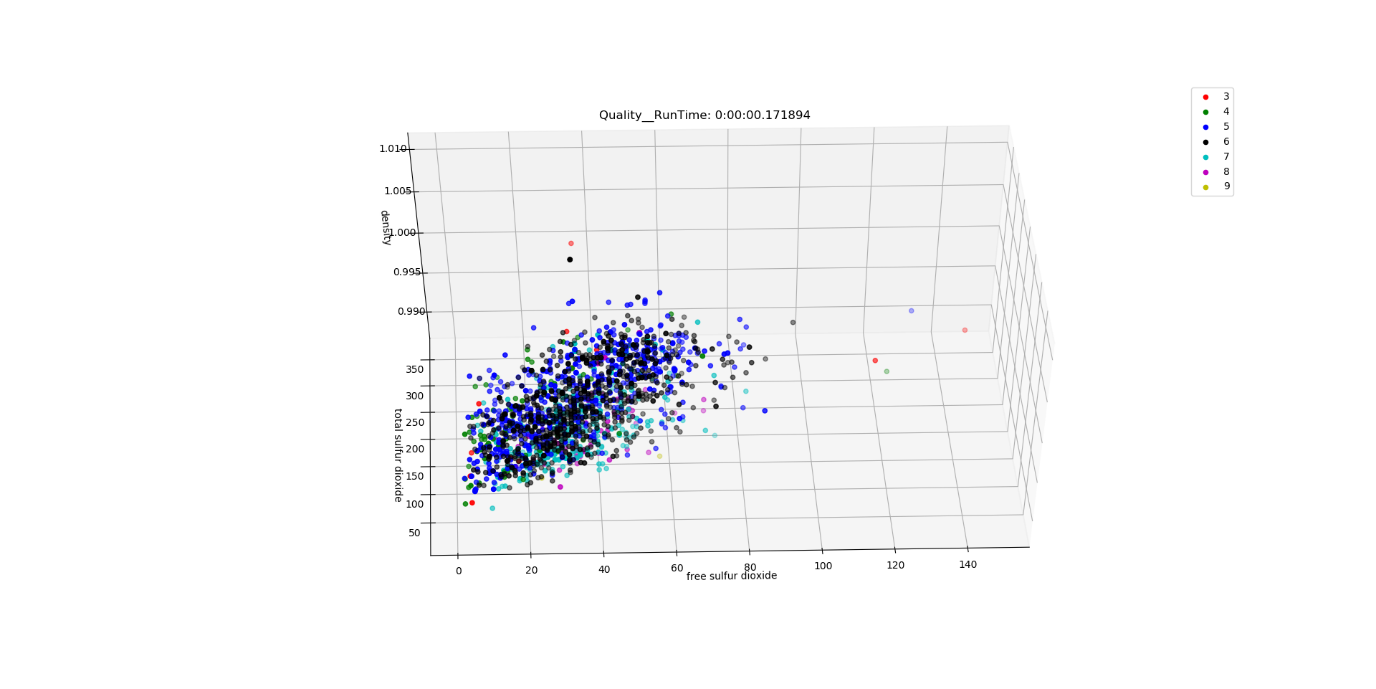


The figure on title shows the running time is 0. sec. And it contains the parameter relationship between the column 6 (free sulfur dioxide) and 7 (total sulfur dioxide)and 8(density)

Calculated by the k-means algorithm, We tried to add a variable, but still could not find data plot group in the diagram.

### View the relationship between quality and free sulfur dioxide/total sulfur dioxide/density (Actual Categorization)

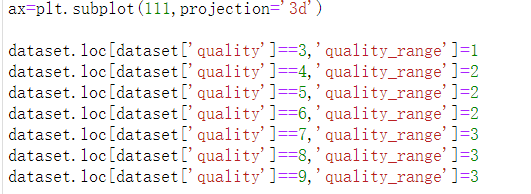




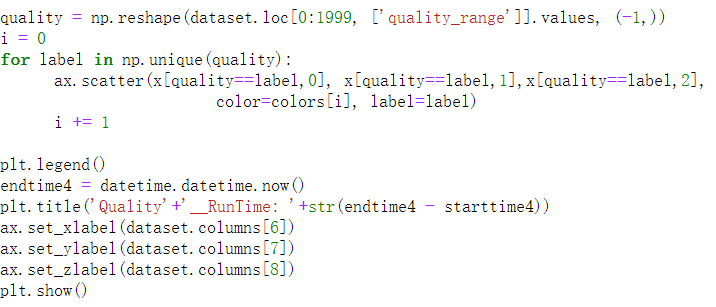
The running time is 0.171894 sec.

From the dataset we found that the quality of the wine ranges from 3 to 9. But it seems that there is no obvious relationship between quality and free sulfur dioxide/total sulfur dioxide/density.

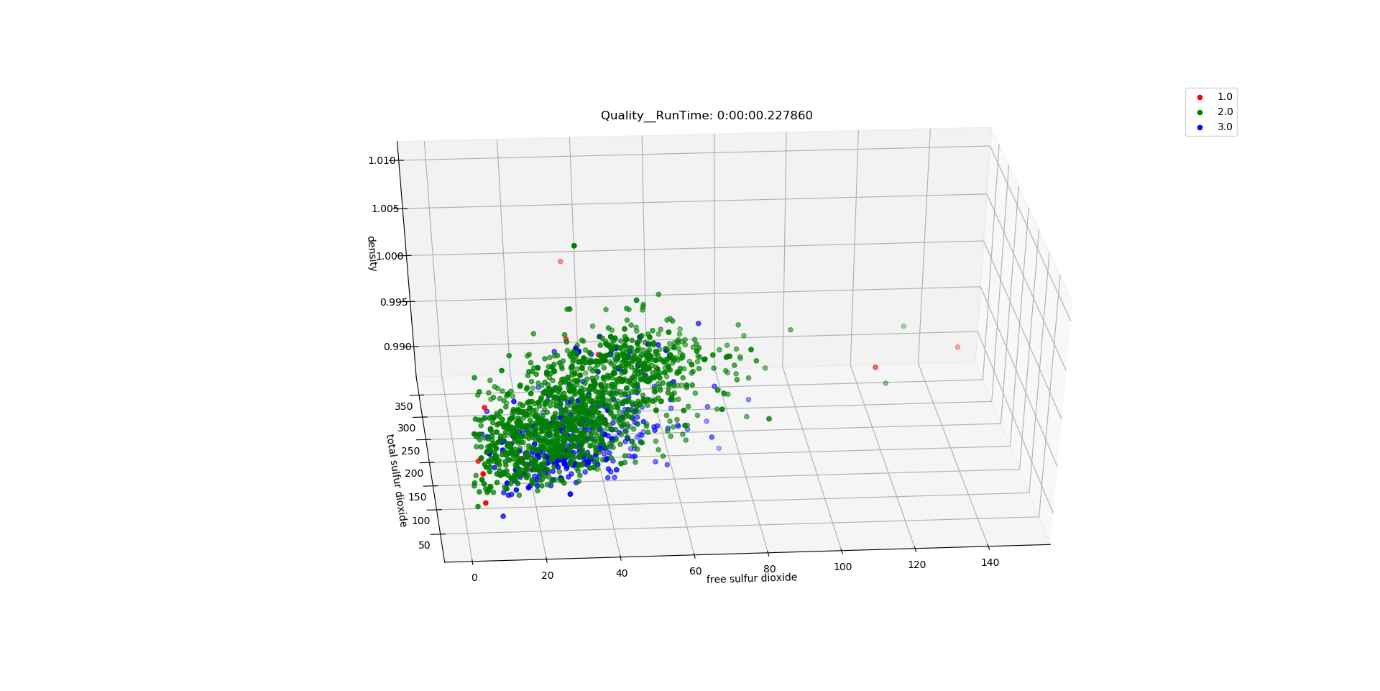
### Divided quality into range



### View the relationship between quality in range and free sulfur dioxide/total sulfur dioxide(Actual Categorization)



We try to find some extra info from using K-means algorithm. Now we are binning the quality from 3-9 into 3 ranges, quality [3-4) to range 1; quality [4-7) to range 2, quality [7-9) to range3.



When we use the ” quality ranges” to color the plot, the figure changes easier to be known, for example the plot of quality range 1 is discrete from the main group of data, and the other plots are concentrated in a group around 5-80 in X-label and 40-250 in Y-label and 0-1.1 in z-label.

But all the plot of quality range 2 and 3 are mixed together, we might need other way to find the regular.

## DBSCAN

### Read the dataset

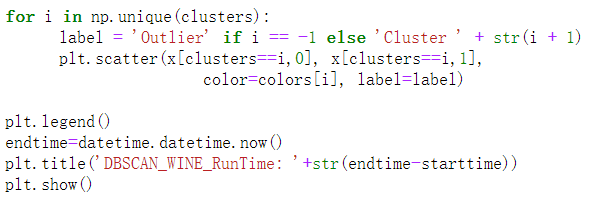


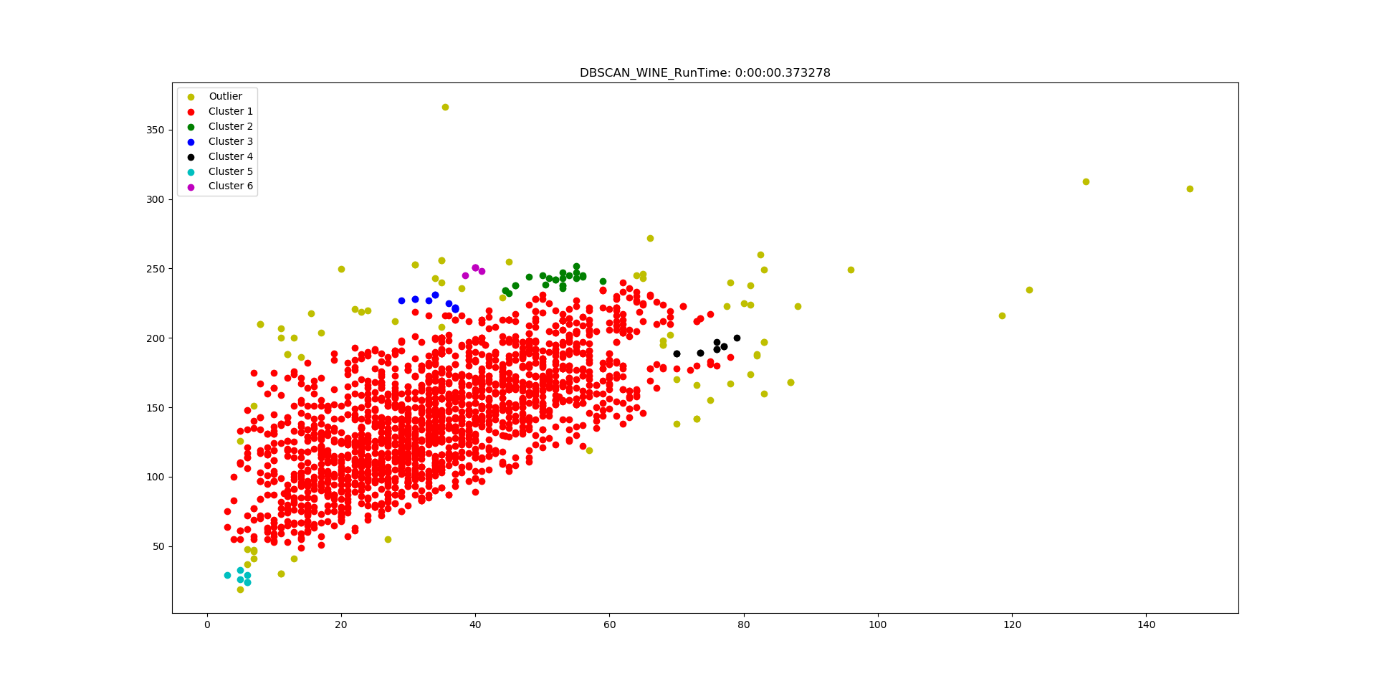
* Read the data in rows 0-2000, columns 6(free sulfur dioxide) and 7(total sulfur dioxide)，aim to compare with the result of the k- means and to find some similar points and different points

### Set parameters and aggregate



### Plot



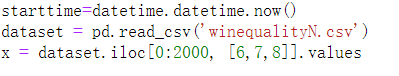


Running time is 0.373278 sec.

The picture shows that we aggregated 6 groups and found many scattered points at the same time. And there is an obvious large aggregation group, which proves that we have achieved basic aggregation.

## DBSCAN -3D

### Read the dataset

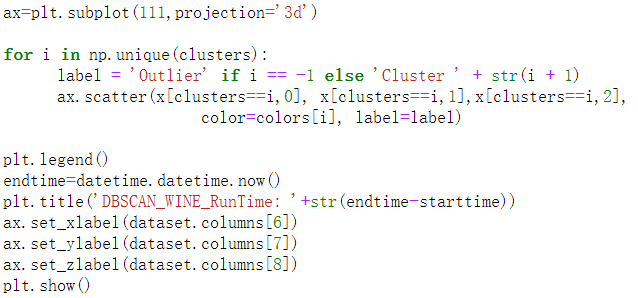


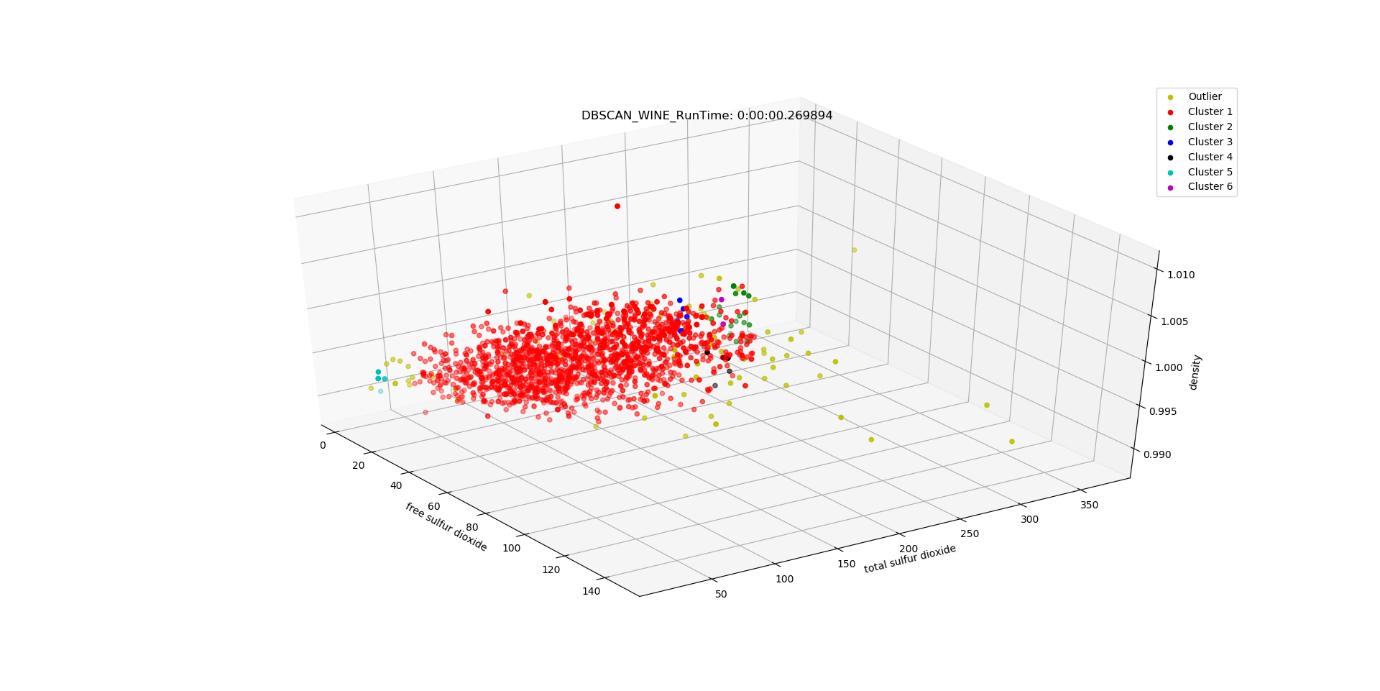
read the data in rows 0-2000, columns 6(free sulfur dioxide) ，7(total sulfur dioxide) and 8(density)，we want to add one columns to show the results if the result will be better or worse.

### Set parameters and aggregate



### plot





Running time is 0.269894 sec.

We aggregated the same number of clusters. Compared with 2D, this effect can not clearly prove the relationship. This shows that our newly added column destroys the original state, indicating that it is not suitable for increasing this relationship.

If we want to further find this relationship, we can continue to use the data of other columns, such as 1, 2, 3 instead of 8 columns.

But what's surprising is that the running time in 3D is shorter than that in 2D. We haven't found a reason yet, but we will continue to learn.

## Comparison between Different Machine Learning (Classification) Model

### Observation

A tabulation has been created across different machine learning models.

Legend:

* DT = Decision Tree
* KNN = K-NN Classification Model
* NN = Neural Network
* LR = Logistic Regression model
* PCA = No. of PCA Components used for Comparison
  + NA = No PCA (unsupervised learning) was used
  + PCA = 3 (3 PCA Components were used as comparison)
* **Value = Worst Expected Result**
* **Value = 2nd Best/Optimal Result**
* **Value = Best Result**

|  |  |  |
| --- | --- | --- |
| Model | Accuracy (Max = 1.0) | Duration (seconds) |
| Model 1 - PCA = 3 - DT | 0.820 | 0.0200 |
| Model 1 - PCA = 3 - KNN | **0.986** | **0.0090** |
| Model 1 - PCA = 3 - LR | **0.986** | 0.0150 |
| Model 1 - PCA = 3 - NN | **0.987** | **12.3662** |
| Model 1 - PCA = NA - DT | **0.490** | **0.0114** |
| Model 1 - PCA = NA - KNN | 0.932 | 0.0140 |
| Model 1 - PCA = NA - LR | 0.981 | 0.0858 |
| Model 1 - PCA = NA - NN | 0.983 | 12.1613 |
| Model 2 - PCA = 3 - DT | 0.466 | **0.0019** |
| Model 2 - PCA = 3 - KNN | 0.497 | **0.0050** |
| Model 2 - PCA = 3 - LR | 0.483 | 0.2064 |
| Model 2 - PCA = 3 - NN | 0.483 | **12.5280** |
| Model 2 - PCA = NA - DT | **0.536** | 0.0089 |
| Model 2 - PCA = NA - KNN | **0.452** | 0.0140 |
| Model 2 - PCA = NA - LR | 0.470 | 0.2264 |
| Model 2 - PCA = NA - NN | **0.558** | 12.1157 |
| Model 3 - PCA = 3 - DT | **0.993** | **0.0009** |
| Model 3 - PCA = 3 - KNN | **0.780** | **0.0060** |
| Model 3 - PCA = 3 - LR | **0.931** | 0.0289 |
| Model 3 - PCA = 3 - NN | **0.932** | **13.2756** |
| Model 3 - PCA = NA - DT | **0.933** | 0.0100 |
| Model 3 - PCA = NA - KNN | 0.805 | **0.0090** |
| Model 3 - PCA = NA - LR | **0.931** | 0.1406 |
| Model 3 - PCA = NA - NN | 0.922 | 11.3309 |

* As shown above, this is a graphical representation of how accurate each ML Model is compared to those of the same Base Model
* As shown above, this is the graphical plot based on different ML Models.
* Neural Network models was taken out separately as majority of the neural network models are within the 10-16 seconds range.
* Hence the neural network models will be treated as a separate discussion. The tabulation of each Neural network models can be shown in the Table above.

### Discussion / What we have learnt

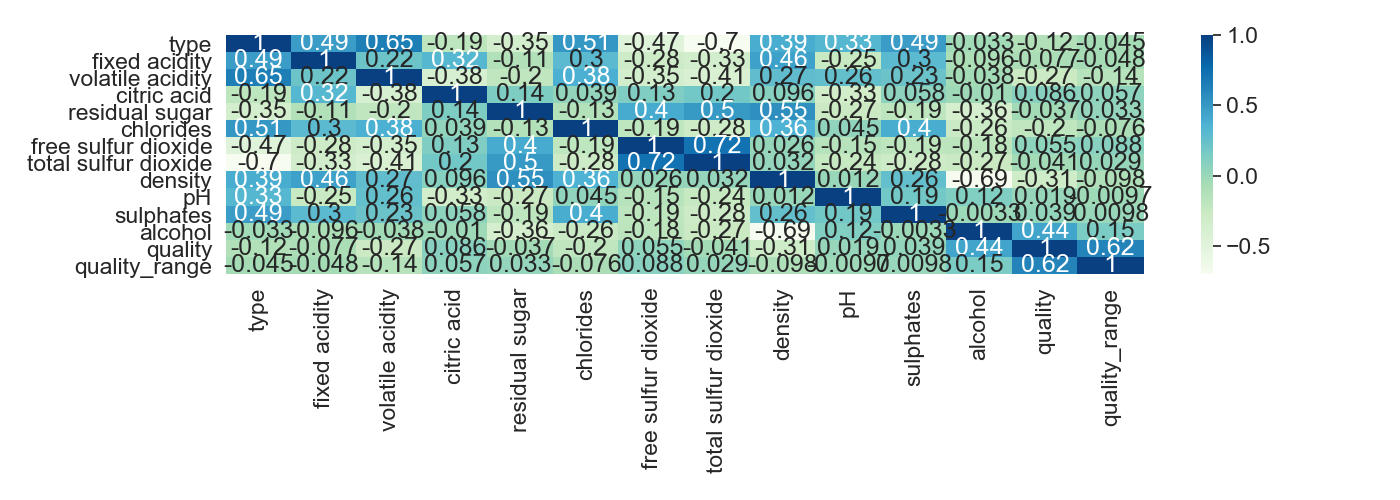
* In overall, it is shown that binning certain classes within the dependent variable will improve ML Models accuracy.
* Based on the wine group data that were used to identify wine type. It is found out that there is no direct correlation between any 1 column to the “wine type”, instead it is dependent on all factors other than quality and quality group that provides the ML model to identify which kind of wine type it is.
* Based on the current data, KNN throughout all 3 Cases is not a good model to do machine learning for this dataset. The reason is due to the fact that there is a large sample set (estimated 6500 data points). In addition to that, all the datapoints are close to each other, hence determining the correct sample set k is not easy.
* On overall, neural networks shows the longest duration to do machine learning, it does not improve when using PCA on top of neural networks. Neural networks’ duration is highly dependent on the number of neurons and the no. of hidden layers used and the no. of epochs used.
* Instead increasing the no. of layers might not be a good suggestion and is very dependent on the dataset.
* Decision tree is a reasonable machine learning model to use across all 3 cases of determining “Wine type”, “quality group” and “quality”.
* Due to the fact that quality has low accuracy score might be due to the lack of datapoints at the two ends of the quality category (0-4 and 7 to 10) as majority of the data is concentrated at 6. Hence the models are accurate enough to predict if the wine has a quality of value of 6 but not good enough to predict other datasets.
* In using PCA, it is known that PCA can definitely reduce the duration of model fitting drastically. This is shown when applied to Case 1 and Case 3(Wine type and Quality Group), however it is not efficient to use it for models that have inherently low accuracy scores (such as Model 2 – Wine Quality)

## Comparison between Unsupervised Techniques

### Observation

### Discussion / What we have learnt

### Conclusion

* There is a high correlation between free Sulphur dioxide and total Sulphur dioxide, this is evident when the unsupervised technique (KMeans) were used to identify clusters.
* It is possible to determine a wine’s type other than its colour, these factors include fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free Sulphur dioxide, total Sulphur dioxide, density, pH, sulphates, alcohol.
* However, DBSCAN and KMEANS is ineffective as there are not 1 governing factor that is used to determine the quality/winetype/quality group of the wine. Most of the clusters found were mixed with each other and is hard to interpret.
* Based on the correlation matrix shown, not 1 single factor has a high correlation rate to wine type/quality/quality group as shown below.  
  

## Appendix

### Decision Tree

Sharing the process

#### Model 1(without PCA)

Import the required libraries and read the file. In our dataset, there are some null values and fill with mean values. Then, train the dataset with 70%,30% ratio.

A screenshot of a cell phone

Description automatically generated

Instantiate the decision tree model and after it, the accuracy score and duration are calculated.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Visualize the decision tree

A picture containing receipt

Description automatically generated

Predict the data

A screenshot of a social media post

Description automatically generated

Confusion Matrix

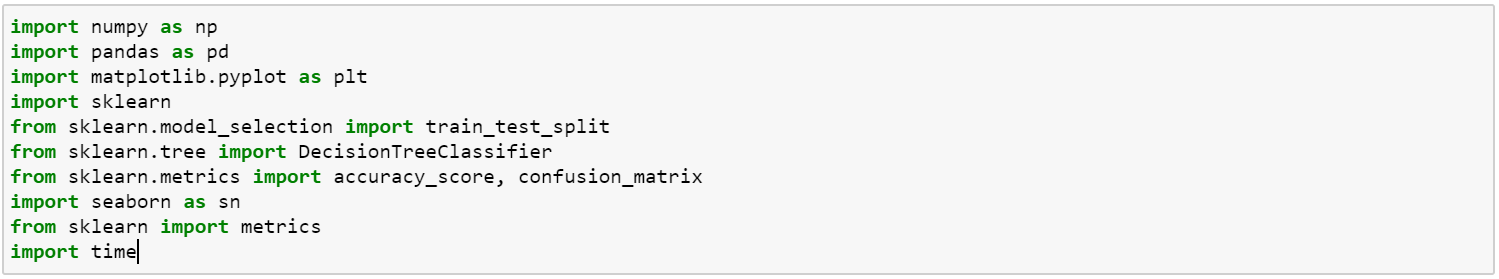
A screenshot of a cell phone

Description automatically generated

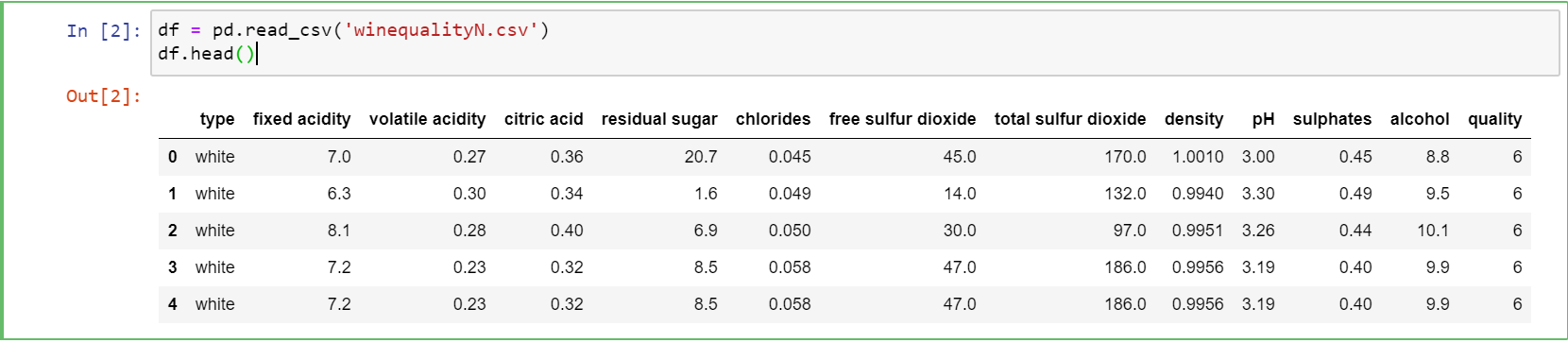
#### Model 2 (Without PCA)

**Data before Data Engineering**

Import the necessary libraries such as numpy, pandas.



Use read\_csv() function to read our dataset.



**Data after Data Engineering**

Checking NULL values in data.



Copy the original dataframe to a new dataframe to replace null values with the mean of each null values columns.



The first column ‘Type’ is categorial values called white, red.

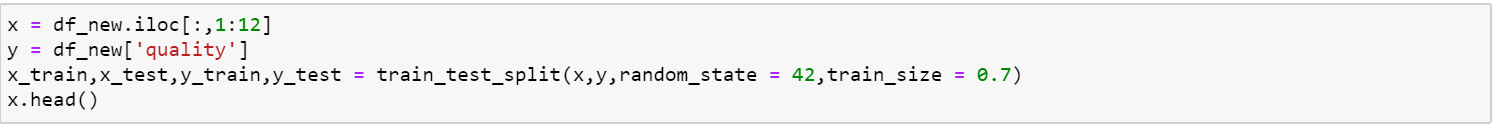
Therefore, we map each type with class 0 and 1.



After that we split training and testing data by using train\_test\_split.

We use all of the columns except quality,type as our training data.

In this case, we take training size 70% of data and 30% is for testing.



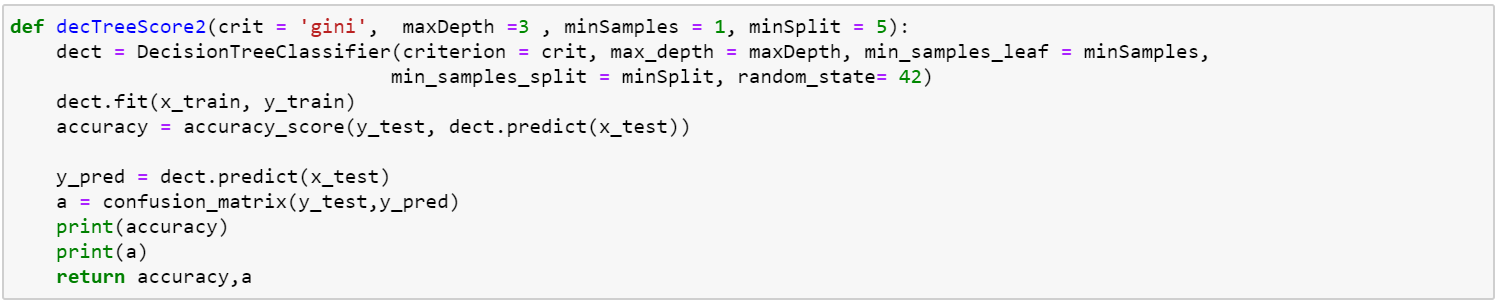
The below function is to feed our training data to the model and get the accuracy results.

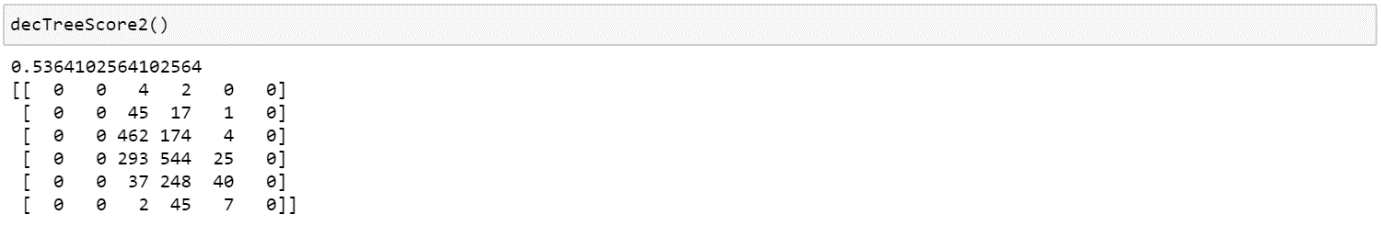
In this decision tree classifier, we use maximun depth = 3 that gives accuracy 0.5364.

Increasing maximun depth will give **better** accuracy results.

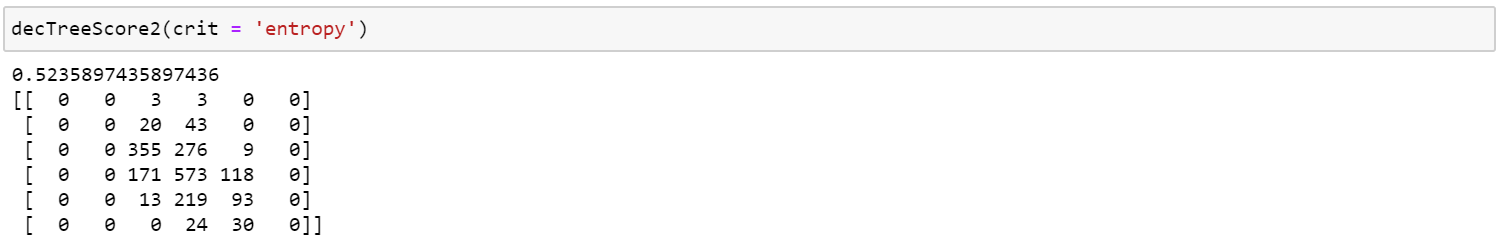
depth = 3 , accuracy = 0.5364

depth = 5, accuracy = 0.5605



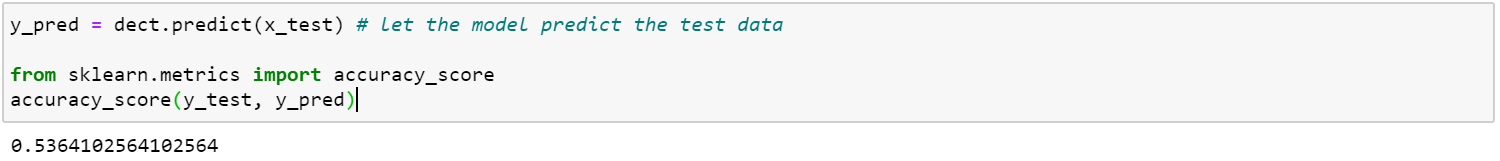


Although criterion ‘entropy’ and ‘gini’ have different calculations, we have very near result accuracy.



This is our prediction for testing data.

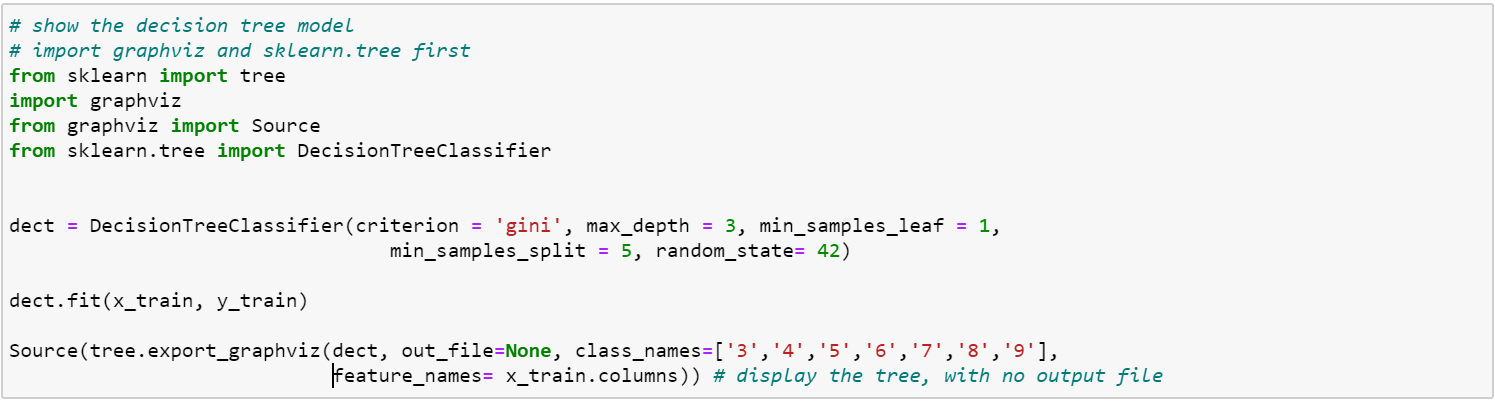
Accuracy is 0.53.

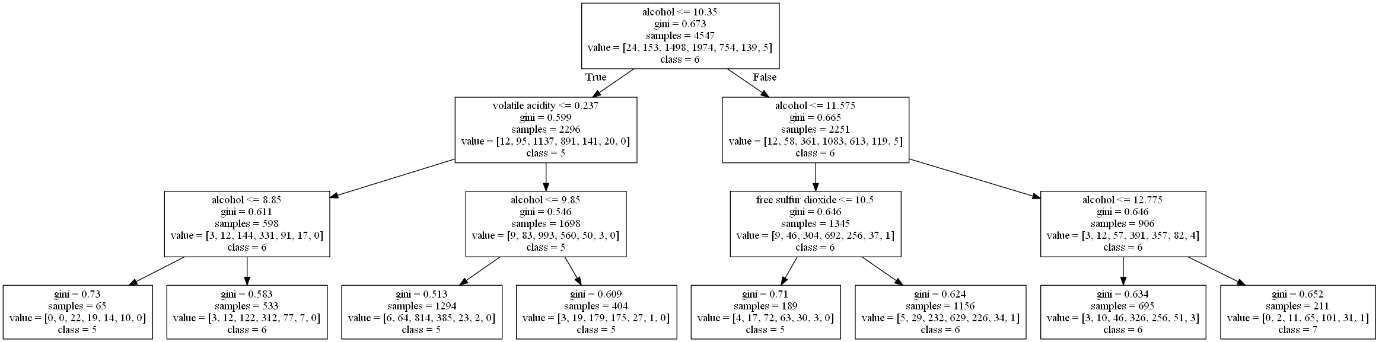


Compare the result of actual values and prediction values.



Generating decision tree result with export\_graphviz() function.

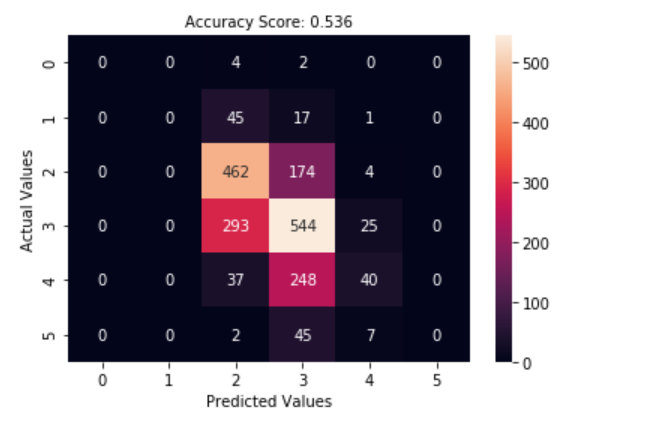




Training Duration



A confusion matrix that **describe the performance of a classification model** on a set of test data for which the true values are known.



**In conclusion, the classification accuracy for wine quality is nearly 60%.**

**To get better accuracy results, we will implement new model by using binning method.**

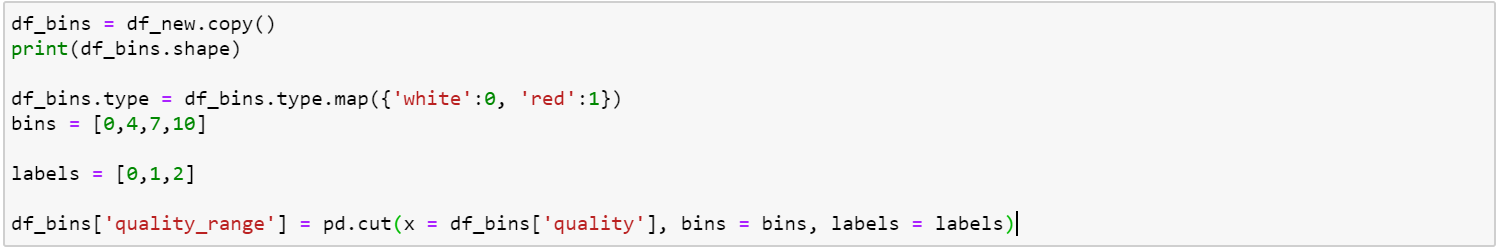
#### Model 3 (Without PCA)

Since the accuracy score is not good in model 2, we use **binning method** to get the higher accuracy. We divide the quality column into quality group (0,1,2) and the quality range is

0 – 4 is quality group 1 (Low)

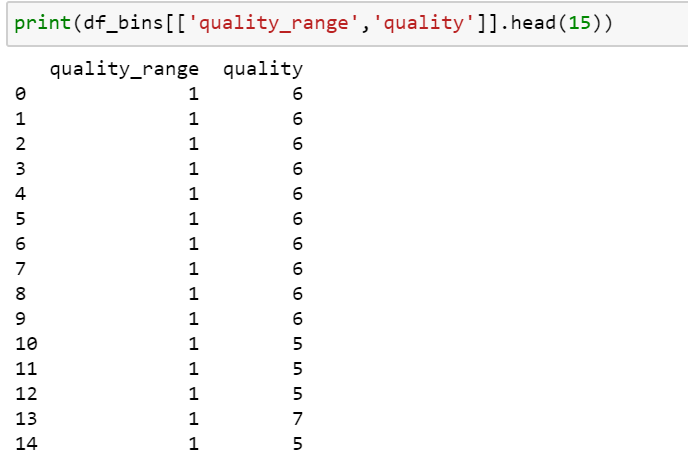
4 -7 is quality group 2 (Median)

7- 10 is quality group 3 (High)

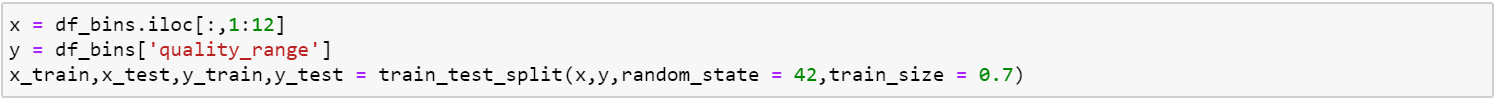


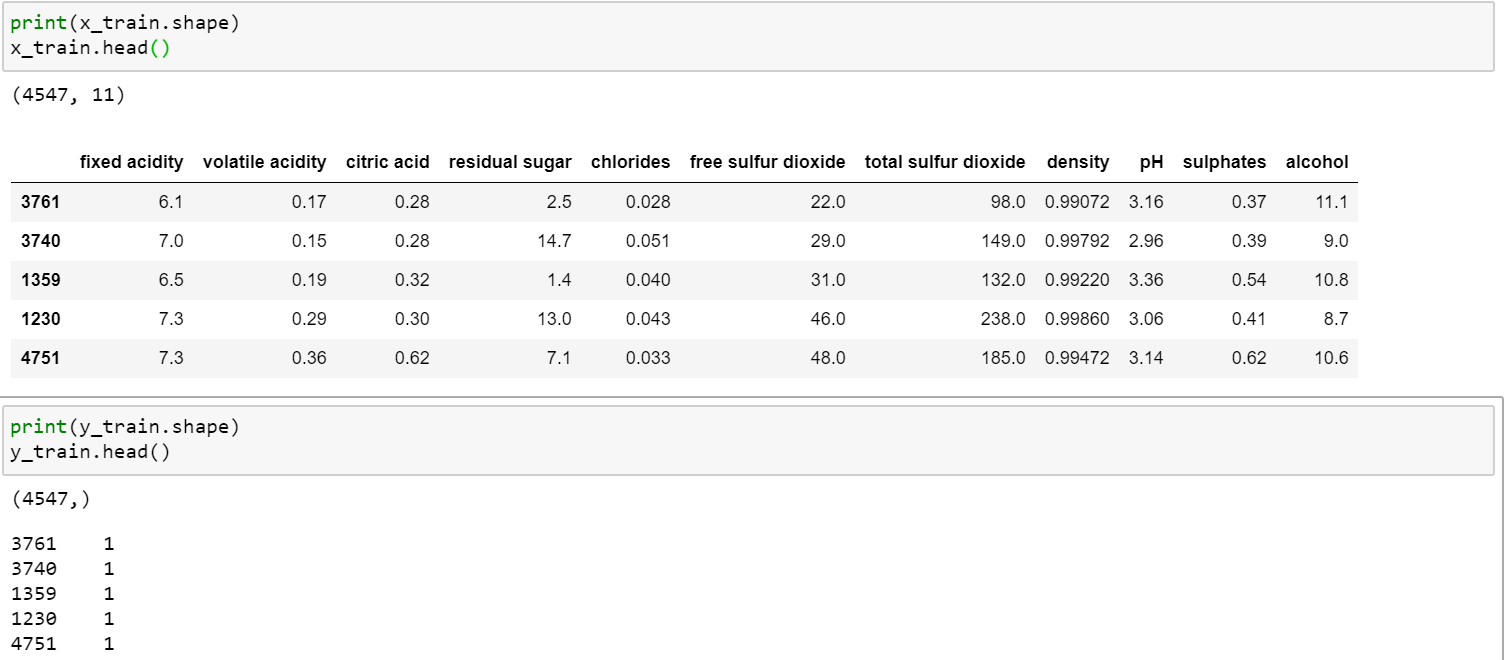
The result of using binning looks like this.

Quality 6 involves in quality range 1.



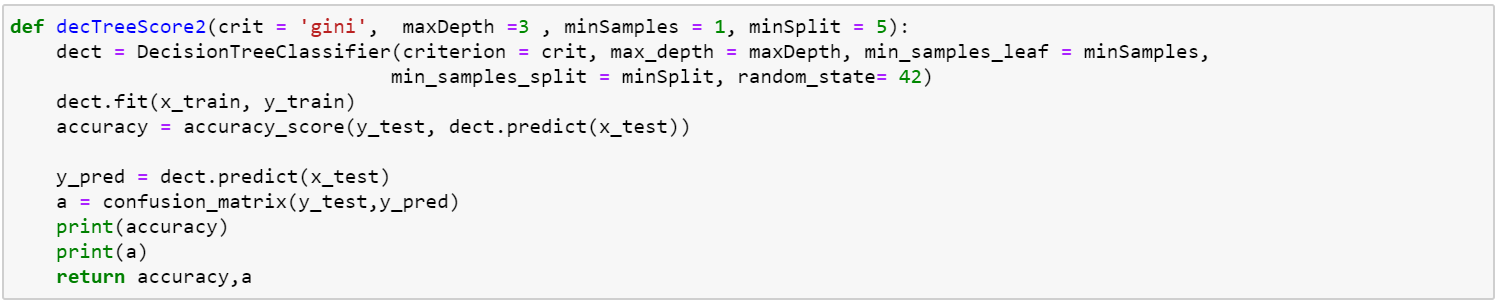
After that we split training and testing data by using train\_test\_split. In this case, we take training size 70% of data and 30% is for testing.





This decTreeScore2 function use maxDepth 3, minSamples = 1 and minSplit = 5 gives the accuracy of 0.93.

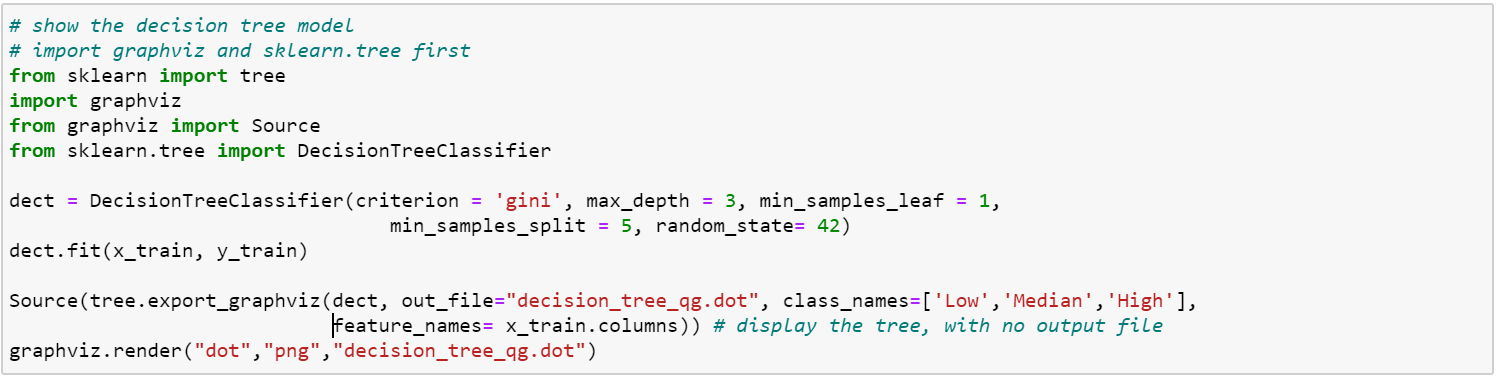
We got a better result by comparing the model 2 accuracy results.



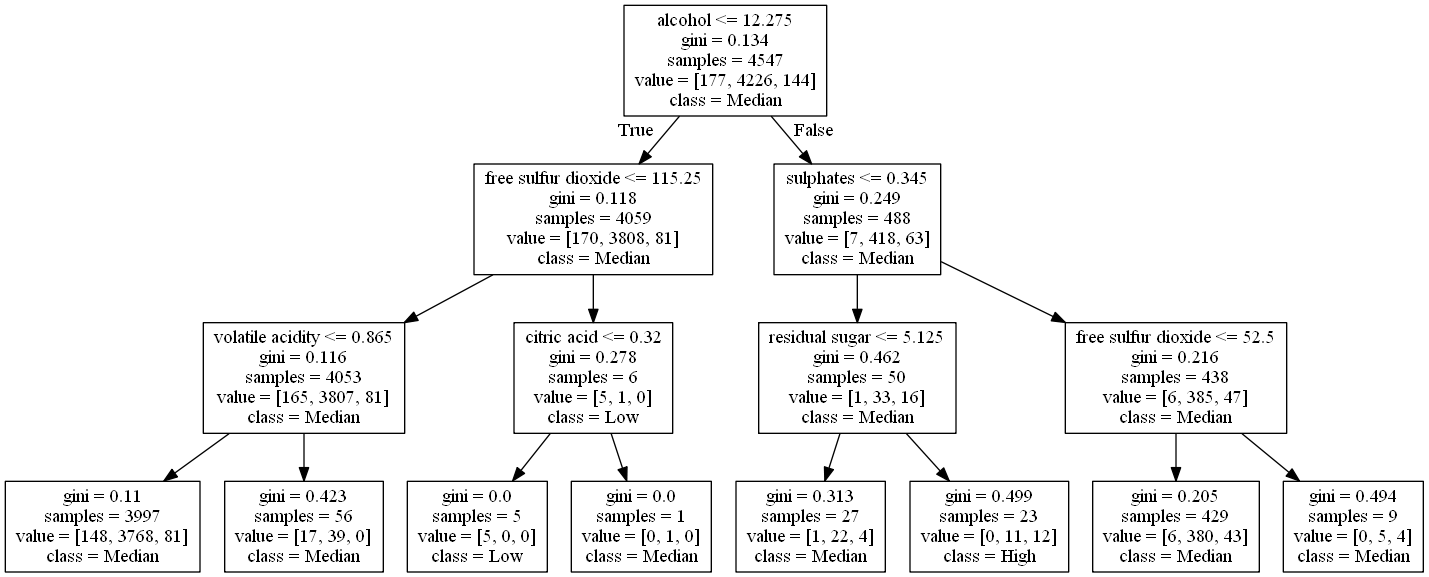


Decision Tree with criterion ‘Entropy’ will give similar accuracy with ‘Gini’





By using export\_graphviz, our decision tree with depth = 3 gives the class results.



Training Duration



Accuracy is 0.93 for testing data.

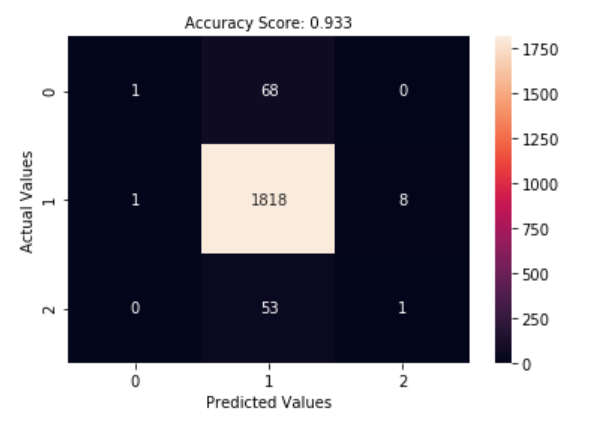


We can also see the actual testing data and prediction results.



This is our confusion metrix that displays the accuracy score of 0.93.





### Feature Engineering

**Feature Extraction**

#### Model 1(with PCA)

This model analyses the dataset with PCA to reduce features on our features.

After PCA(n\_components =3) ,dataset with 11 features changed to 3 features. After that, we trained the dataset with 3 features.

A screenshot of a cell phone

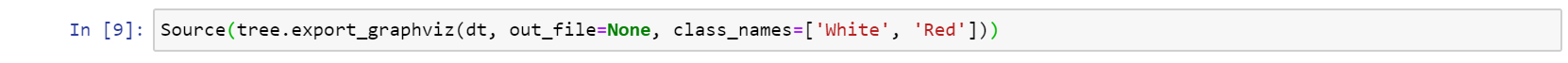
Description automatically generated

Instantiate the decision tree classifier after getting the train data.

Its accuracy score of applying PCA is better than without applying it. And also the duration of time to train has become slightly shorter than before.

A screenshot of a social media post

Description automatically generated



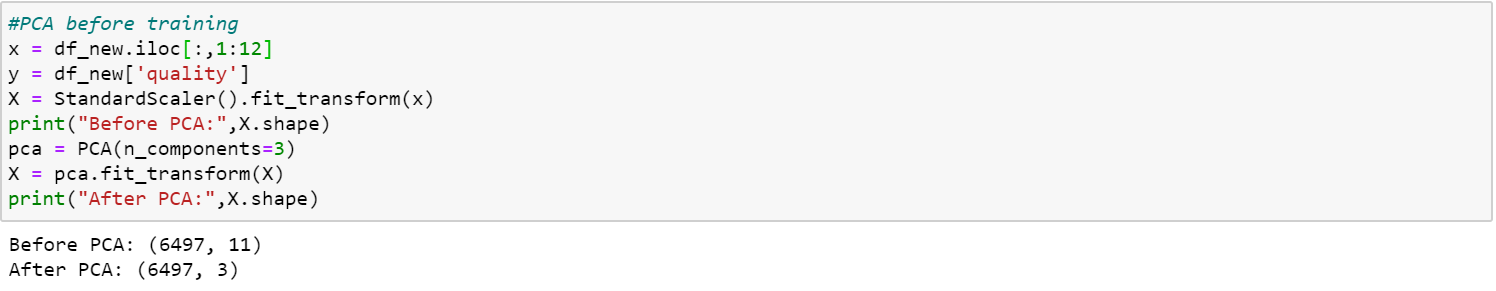
A picture containing clock

Description automatically generated

#### Model 2 (With PCA)

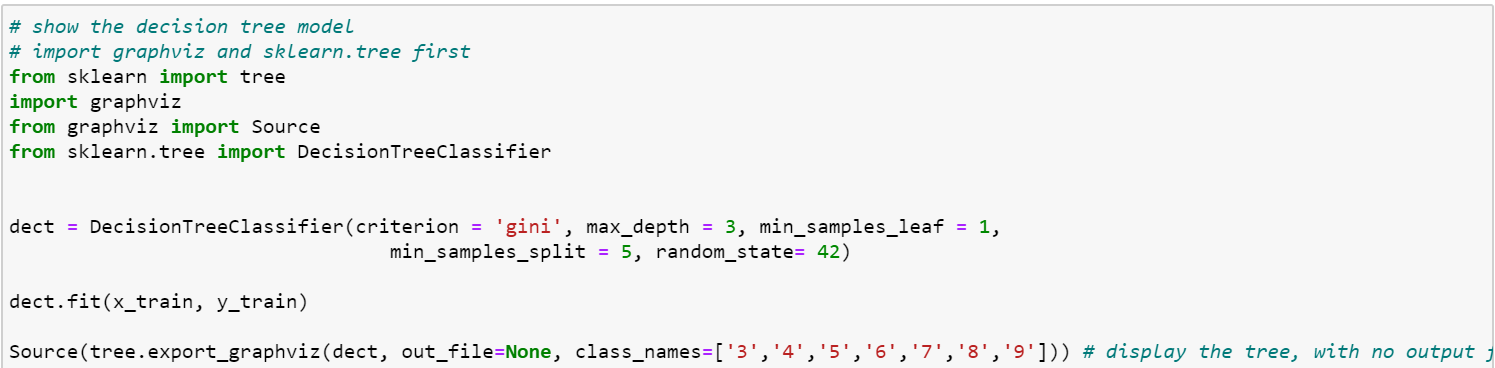
We use PCA on model-1 to reduce features on our data.

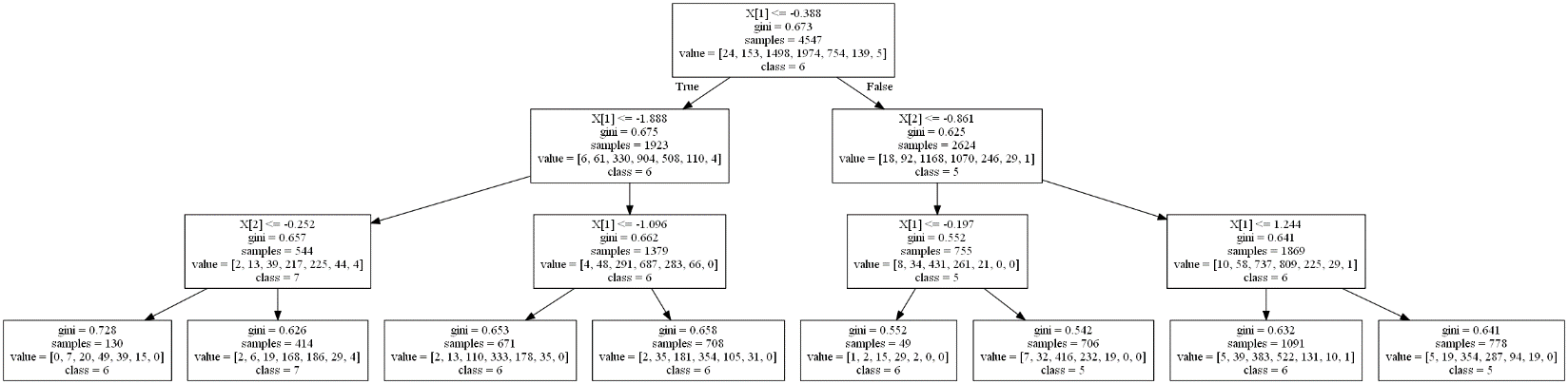
Before PCA, dataset has 12 features on training and 3 features after PCA.



Now we use new X after PCA on train\_test\_split to get training and testing data based on this PCA features.



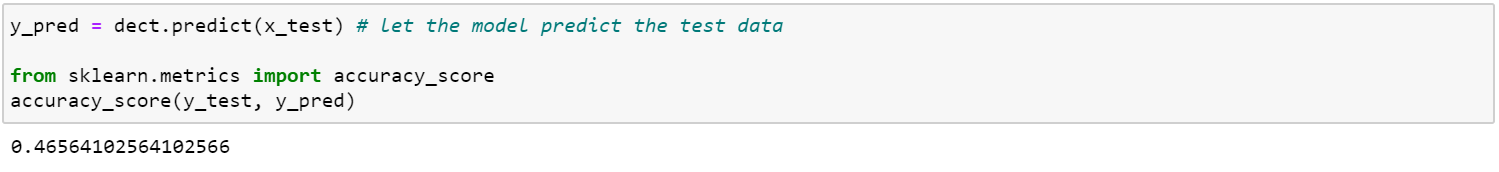




Training Duration



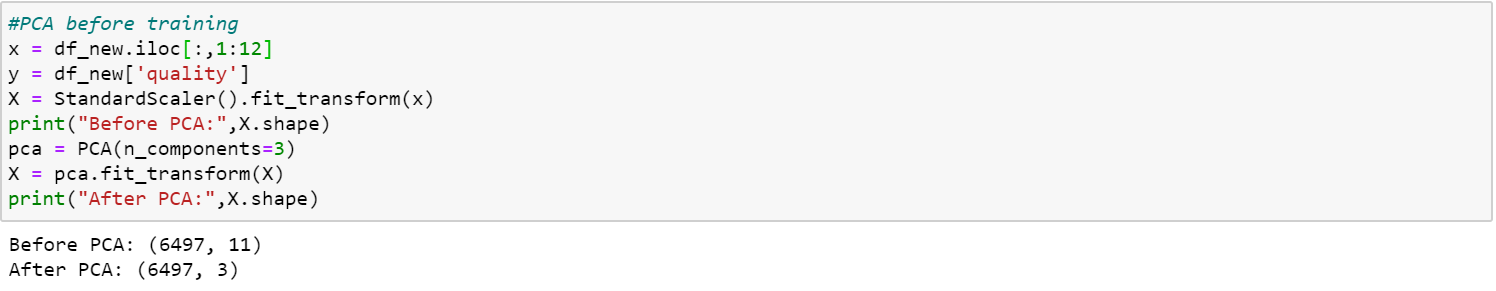
Although PCA is applied to model, we don’t have better accuracy result.



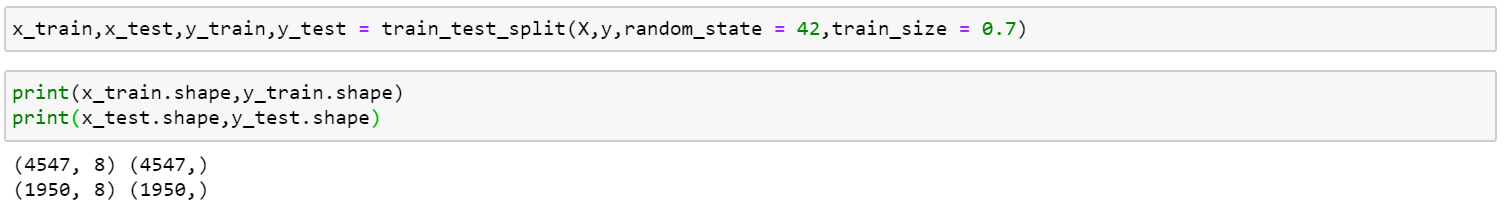
#### Model 3 (With PCA)

We also use PCA on model 3 that use binning method in our model to reduce features.

Before PCA, 12 features on training dataset and get 5 features after PCA.



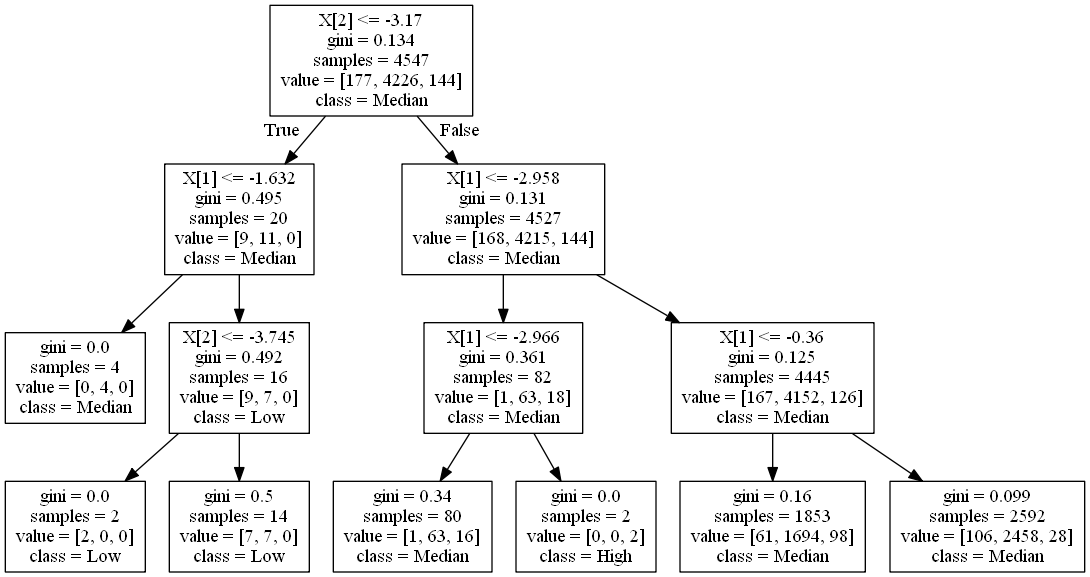
Now we use new X after PCA on train\_test\_split to get training and testing data based on this PCA features.



Training Duration







Although we train our model with 3 features on PCA we can get better accuracy result.

