



A PROJECT REPORT ON “WINE QUALITY”

SANJANA M
TLS21A481

INDEX

Topic	Page no:
Abstract	3
Introduction	4
Discussion on Tasks	5-10
Python Code	11-19
Conclusion	20

ABSTRACT

Wine classification is a difficult task since taste is the least understood of the human senses. A good wine quality prediction can be very useful in the certification phase, since currently the sensory analysis is performed by human tasters, being clearly a subjective approach. An automatic predictive system can be integrated into a decision support system, helping the speed and quality of the performance. Furthermore, a feature selection process can help to analyze the impact of the analytical tests. If it is concluded that several input variables are highly relevant to predict the wine quality, since in the production process some variables can be controlled, this information can be used to improve the wine quality.

INTRODUCTION

The aim of this project is to predict the quality of wine on a scale of 0–10 given a set of features as inputs. Input variables are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulphur dioxide, total sulphur dioxide, density, pH, sulphates, alcohol. And the output variable is quality (score between 0 and 10). We are dealing only with white wine. We have quality being one of these values: [5, 6, 7, 8]. The higher the value the better the quality. In this project we will treat each class of the wine separately and their aim is to be able and find decision boundaries that work well for new unseen data. These are the classifiers.

In this project we are explaining the steps we followed to build our models for predicting the quality of white wine in a simple non-technical way. We would follow similar process for red wine or we could even mix them together and include a binary attribute red/white, but our domain knowledge about wines suggests that we shouldn't. Classification is used to classify the wine as good or bad. Before examining the data, it is often referred to as supervised learning because the classes are determined. For better understanding, 30 records of the wine quality dataset are used in the following code.

TASKS

- DATA ACQUISITION AND CLEANING
- DATA VISUALIZATION
- DATA MODELLING
- TESTING
- COMPARISON AND MEASUREMENT

GROUP MEMBERS:

- Anusha B P(TLS21A008)
- Sanjana M(TLS21A481)
- Sannidhi N Hegde(TLS21A009)

Task Performed by Me:

- DATA MODELLING

DATA ACQUISITION AND CLEANING

Two datasets were created, using white wine samples.

The inputs are physicochemical information, such as PH values, and the output is based on sensory data which is the median of at least 3 evaluations made by wine experts. Each expert graded the wine quality between 0 (very bad) and 10 (excellent).

The dataset providers alleges that due to privacy and logistic issues, only physicochemical and sensory variables are available.

There's no information about how the dataset was created, such as the distribution of grape types, wine brands and whether the same experts graded all wines. This information is important to identify possible bias that may distort the analysis results. For example, the physicochemical properties may vary among different wines of the same type, affecting the distribution in the dataset.

The classes are ordered and not balanced. There are much more normal wines than excellent or poor ones.

Input variables based on physicochemical tests:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulphur dioxide
- total sulphur dioxide
- density
- pH
- sulphates
- alcohol

Output variable based on sensory data:

- quality (score between 0 and 10)

DATA VISUALIZATION

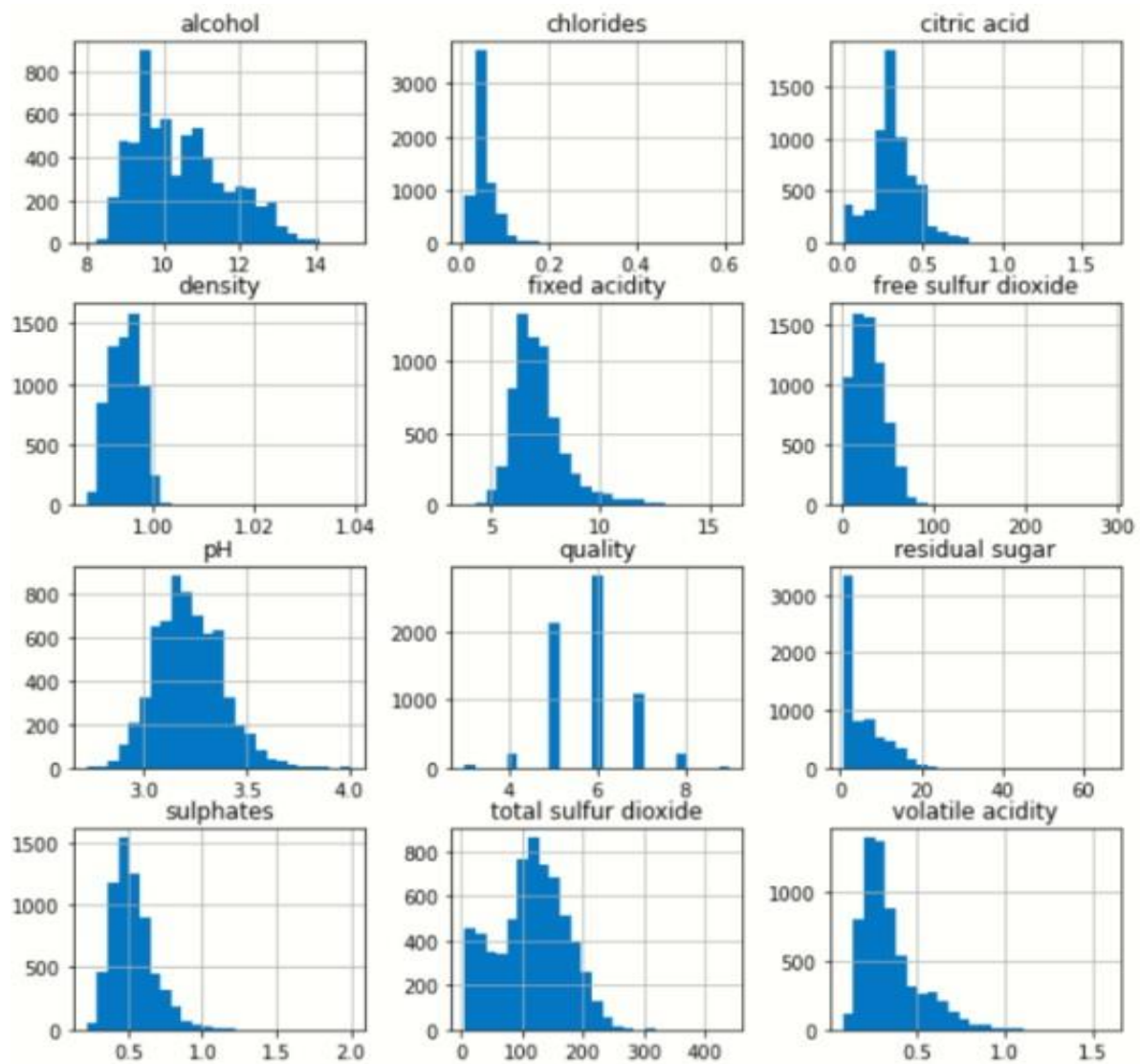
In this checks the dataset structure, look for possible problems in the data and provides a clear understanding of the data. The information acquired in this section may be used during the modelling phase.

Although UCI provides the dataset ready for analysis, it's good practice to check it for possible issues we may encounter in the future.

We start by counting the number of empty values, NAs, in the entire dataset, i.e., in both the training and testing sets. Empty values may cause calculation errors that can be avoided if we identify them in advance. Next, we check the predictors that have very few unique values relative to the number of samples or a large difference in the frequency of the most common and the second most common values. Predictors with these characteristics provide little value in the analysis and may be discarded. No variable has near zero variance. The dataset structure confirms the information provided by UCI and provides additional information. All variables are numeric except type which we added as a factor with two levels. There are 5847 observations.

The statistics summary provides a clue of each variable distribution. Median close to the mean may be an indication of normal distribution. The variables `free_sulfur_dioxide`, `total_sulfur_dioxide`, `density`, `pH`, `sulphates` and `alcohol` seem to follow this rule that we will confirm later

Wine quality



DATA MODELLING

During data exploration we learned that total sulphur dioxide, chlorides and volatile acidity are good candidates for wine type prediction. The AUC is very high, the distributions have low overlapping areas and the correlations are low.

On the other hand, the quality prediction of red wines may be challenging. Although some features with high AUC have low correlations, all features present large distribution overlapping areas. Besides this, the prevalence of wines with qualities 3, 4 and 8 is very low.

In this section we discuss several modelling approaches to predict red and white wines and wine quality.

Simple Model:

The simplest model is just assuming that all wines are red or white. Since there's more white than red wines, the model predicts all wines are white. The accuracy at 75.4 is higher than 50%, the specificity is 1 and sensitivity 0. The model is good at predicting white wines, but it isn't very useful at predicting red wines. A better model should have higher sensitivity, although it may lower specificity and accuracy.

Random Model:

Another model is to use the sample probabilities of each wine type as a predictor. We know that the prevalence of red and white wines is 24.6% and 75.4% respectively, so we just randomly assign red or white using these proportions. The problem with this approach is that we don't know the actual distribution in the population, and the actual distribution may fluctuate over time. Any machine learning model should do better than both models.

Random Forest:

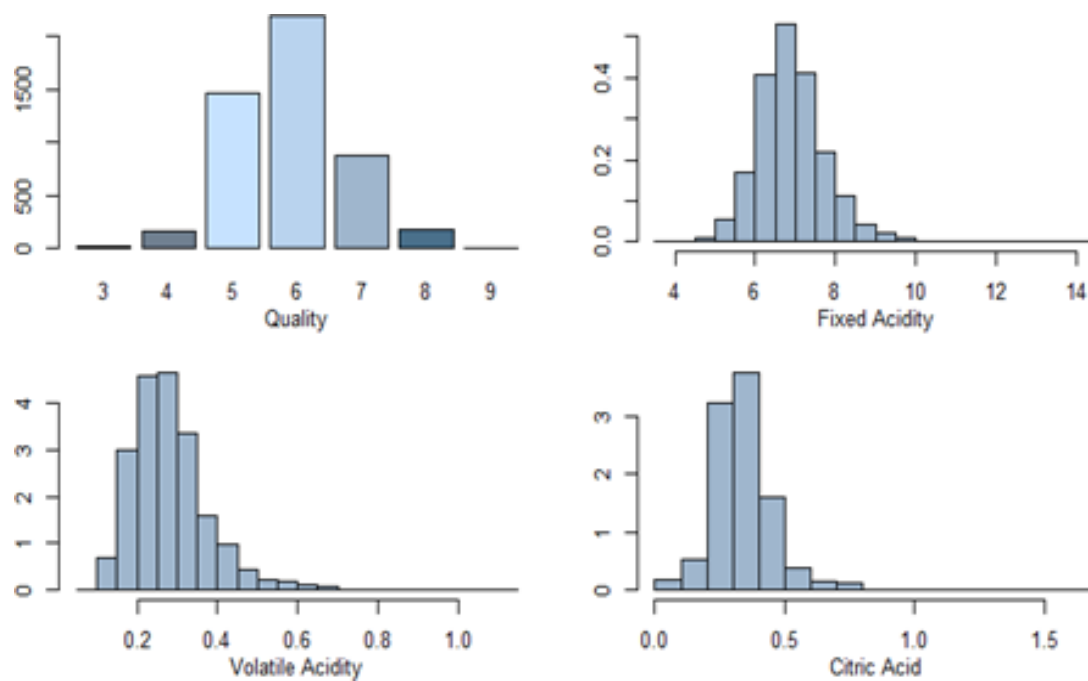
Random forests improve prediction performance over classification trees by averaging multiple decision trees. The algorithm creates several random subsets of the original data, in this case the training set, and calculates the classification trees, then the final result is the average of all trees.

The name random forest derives from the random process of splitting the data and creating many trees, or a forest.

TESTING

We are keeping 20% of our dataset to treat it as unseen data and be able to test the performance of our models. We are splitting our dataset in a way such that all of the wine qualities are represented proportionally equally in both training and testing dataset.

Other than that the selection is being done randomly with uniform distribution. Various classification and regression algorithms are used to fit the model.



PYTHON CODE

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('winequali.csv')

print(data.head())
```

Output :

```
Run: WineQualityCheck X
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
   type  fixed acidity  volatile acidity  ...  sulphates  alcohol  quality
0  white           7.0             0.27  ...     0.45     8.8         6
1  white           6.3             0.30  ...     0.49     9.5         6
2  white           8.1             0.28  ...     0.44    10.1         6
3  white           7.2             0.23  ...     0.40     9.9         6
4  white           7.2             0.23  ...     0.40     9.9         6

[5 rows x 13 columns]

Process finished with exit code 0
```

```
print(data.describe())
```

Output:

```
Run: WineQualityCheck X
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
   fixed acidity  volatile acidity  ...  alcohol  quality
count    29.000000      30.000000  ...  30.000000  30.000000
mean       7.134483       0.315000  ...  10.493333   6.133333
std       0.708759       0.134440  ...   1.138642   0.860366
min       6.200000       0.160000  ...   8.800000   5.000000
25%       6.600000       0.242500  ...   9.625000   6.000000
50%       7.000000       0.275000  ...  10.100000   6.000000
75%       7.600000       0.317500  ...  11.225000   6.000000
max       8.600000       0.670000  ...  12.800000   8.000000

[8 rows x 12 columns]

Process finished with exit code 0
```

Wine quality

```
print(data.dtypes)
```

Output:

```
Run: WineQualityCheck X
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
type          object
fixed acidity  float64
volatile acidity float64
citric acid    float64
residual sugar float64
chlorides      float64
free sulfur dioxide int64
total sulfur dioxide int64
density        float64
pH             float64
sulphates      float64
alcohol        float64
quality        int64
dtype: object

Process finished with exit code 0
```

```
print(data.isnull().sum())
```

Output:

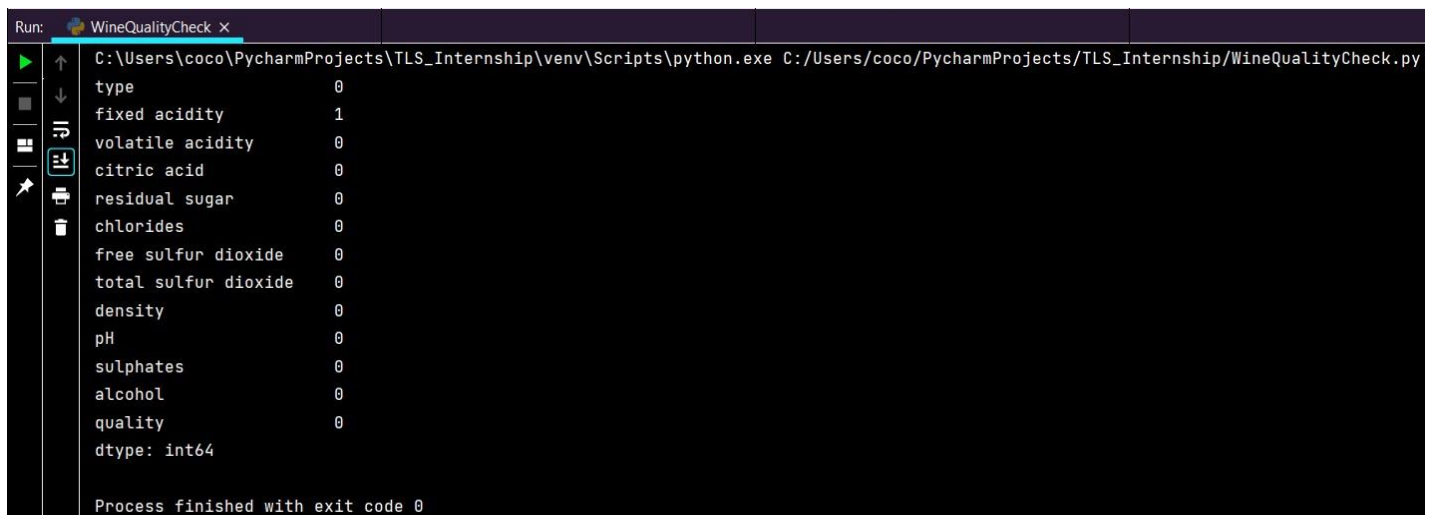
```
Run: WineQualityCheck X
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
type          0
fixed acidity  1
volatile acidity 0
citric acid    0
residual sugar 0
chlorides      0
free sulfur dioxide 0
total sulfur dioxide 0
density        0
pH             0
sulphates      0
alcohol        0
quality        0
dtype: int64

Process finished with exit code 0
```

Data Cleaning:

```
data.dropna(inplace=True)
typ=pd.get_dummies(data['type'],drop_first=True)
data.drop(['type'],axis=1,inplace=True)
data=pd.concat([data,typ],axis=1)
print(data.isnull().sum())
```

Output:



```
Run: WineQualityCheck X
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
type      0
fixed acidity  1
volatile acidity  0
citric acid  0
residual sugar  0
chlorides  0
free sulfur dioxide  0
total sulfur dioxide  0
density  0
pH  0
sulphates  0
alcohol  0
quality  0
dtype: int64
Process finished with exit code 0
```

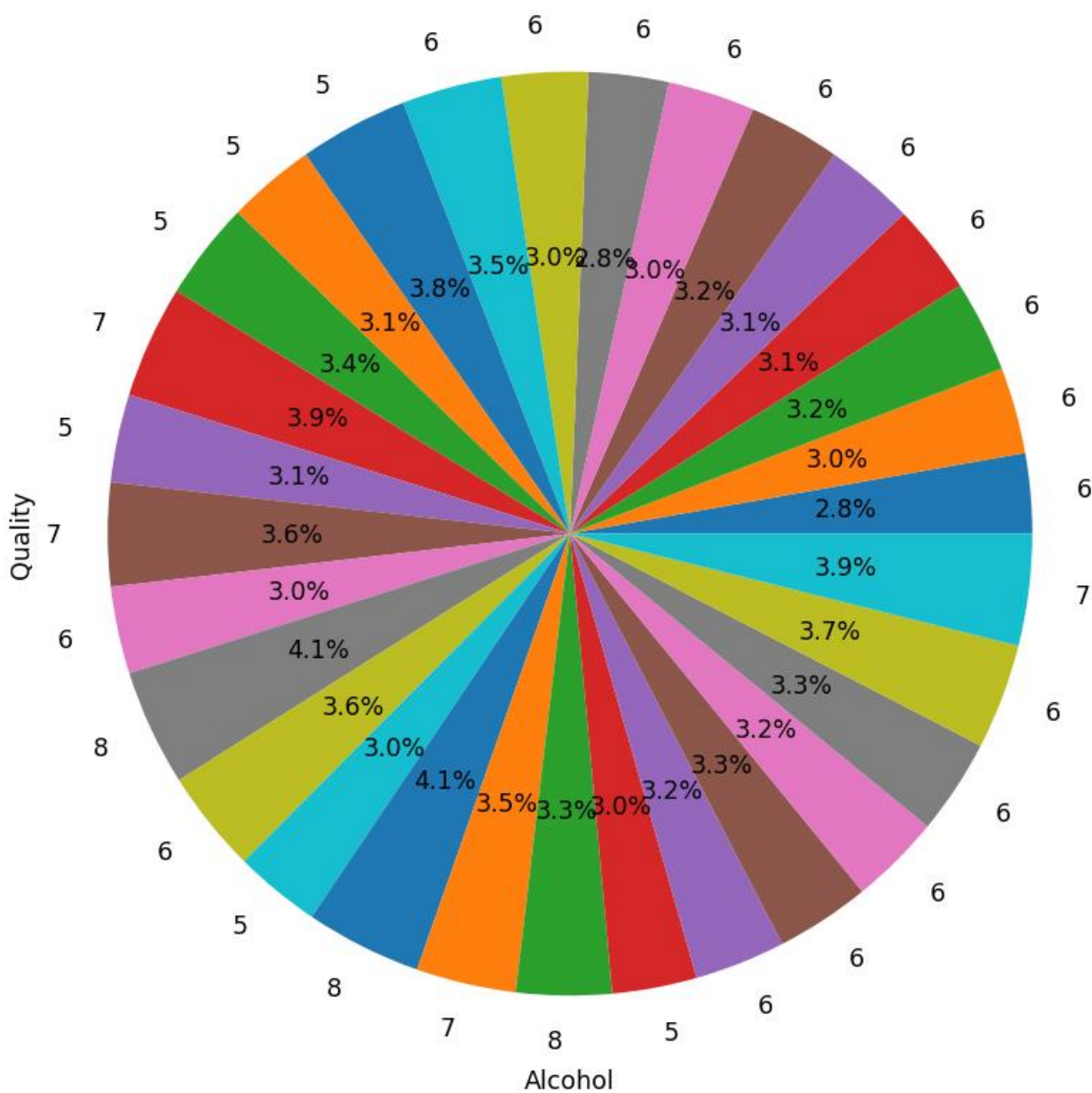
Data Visualization : Graphs plotted - Quality against alcohol content

```
x=data['alcohol']
y=data['quality']

plt.xlabel('Alcohol')
plt.ylabel('Quality')

plt.pie(x, labels=y, radius=1.1, autopct='%0.01f%%')
```

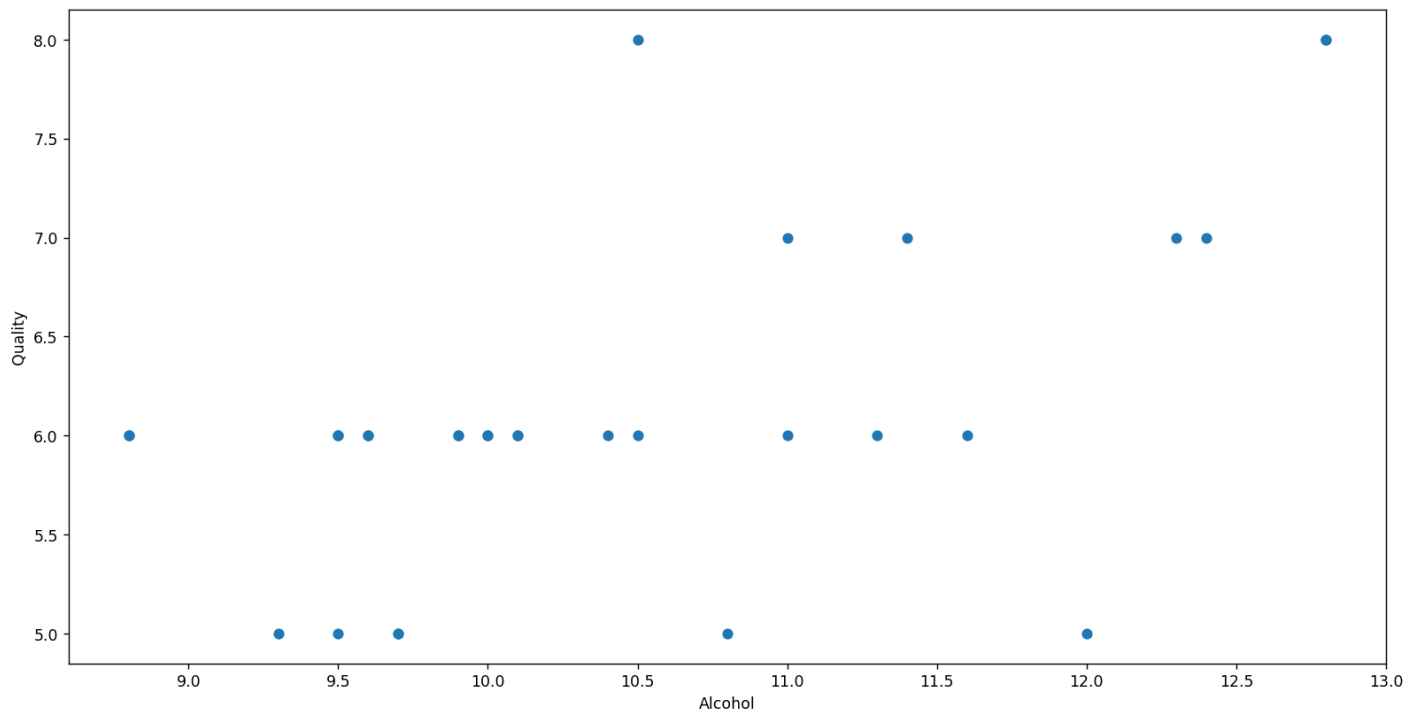
Output:



Wine quality

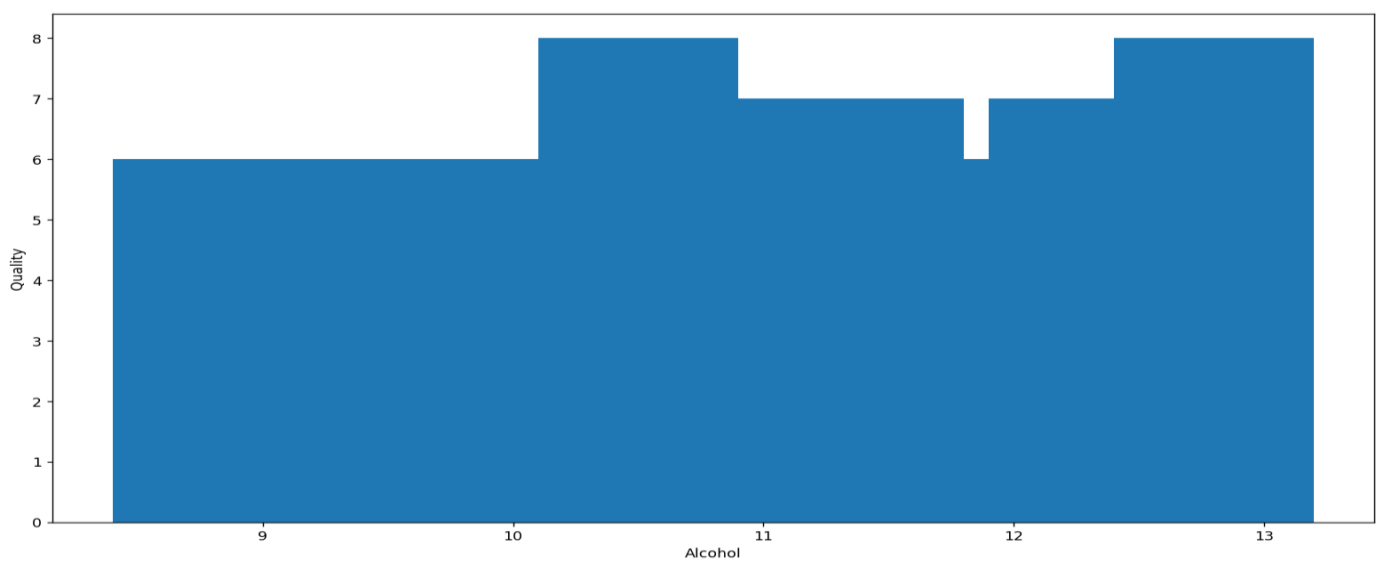
```
plt.scatter(x, y)
```

Output:



```
plt.bar(x, y)
```

Output:



```
plt.show()
```

Train Test Algorithm:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```

Linear Regression:

```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)
y_pred=regressor.predict(x_test)
print(y_pred)
```

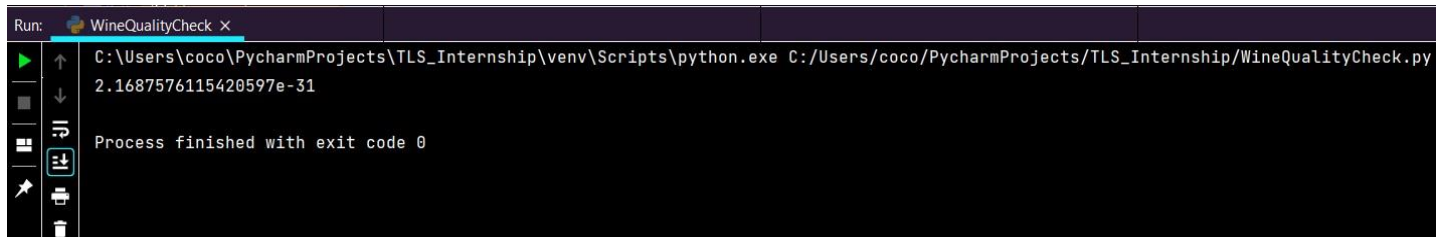
```
Run: WineQualityCheck x
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
[0.05 0.038 0.052 0.044 0.035 0.052]
Process finished with exit code 0
```

```
original=pd.DataFrame(x_test,y_test)
print(original)
```

```
Run: WineQualityCheck x
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
      0    1    2    3    4    5    6    7    8    9   10
0.050  8.1  0.28  0.4  6.9  0.05  30  97  0.9951  3.26  0.44  10.1
0.038  7.4  0.27  0.48  1.1  0.047  17  132  0.9914  3.19  0.49  11.6
0.052  6.6  0.16  0.4  1.5  0.044  48  143  0.9912  3.54  0.52  12.4
0.044  8.1  0.27  0.41  1.45  0.033  11  63  0.9908  2.99  0.56  12.0
0.032  6.9  0.24  0.35  1.0  0.052  35  146  0.993  3.45  0.44  10.0
0.052  6.6  0.27  0.41  1.3  0.052  16  142  0.9951  3.42  0.47  10.0
Process finished with exit code 0
```


Mean Squared root:

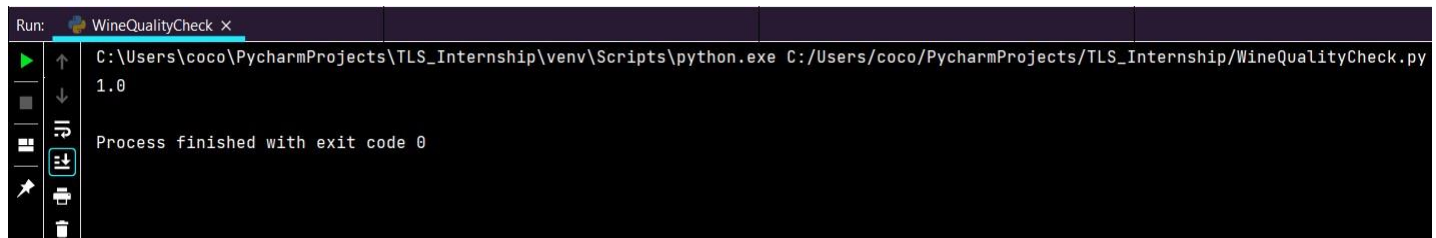
```
from sklearn.metrics import mean_squared_error  
print(mean_squared_error(y_test,y_pred))
```



The screenshot shows a PyCharm Run console window titled 'WineQualityCheck'. The command prompt shows the execution of a Python script at the path 'C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py'. The output is '2.1687576115420597e-31'. Below the output, it says 'Process finished with exit code 0'.

Regression Score :

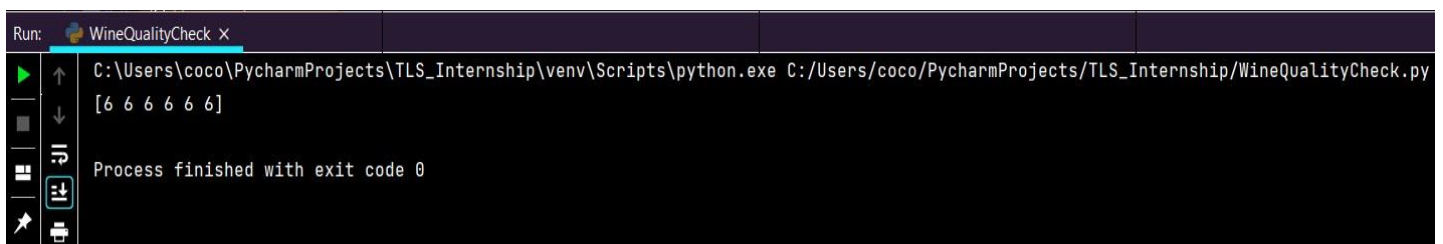
```
print(regressor.score(x_test,y_test))
```



The screenshot shows a PyCharm Run console window titled 'WineQualityCheck'. The command prompt shows the execution of a Python script at the path 'C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py'. The output is '1.0'. Below the output, it says 'Process finished with exit code 0'.

Logistical Regression :

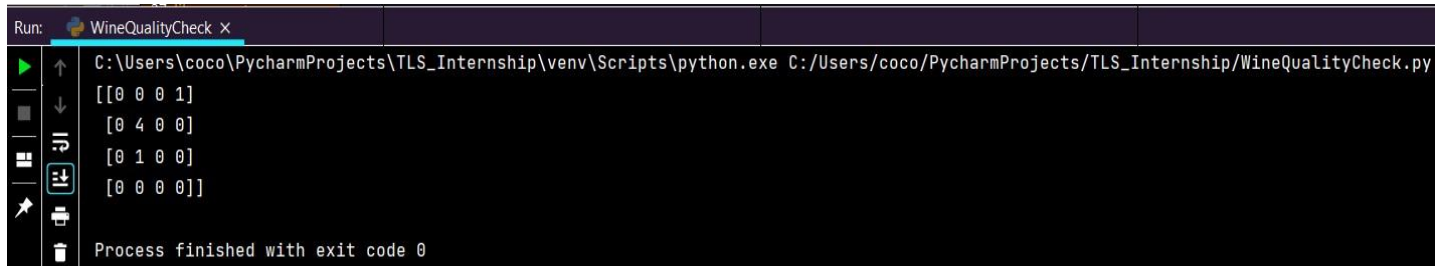
```
x=data[['chlorides', 'total sulfur dioxide','density','pH','sulphates','alcohol']]  
y=data.quality  
  
from sklearn.linear_model import LogisticRegression  
logreg=LogisticRegression()  
logreg.fit(x_train,y_train)  
y_pred=logreg.predict(x_test)  
print(y_pred)
```



The screenshot shows a PyCharm Run console window titled 'WineQualityCheck'. The command prompt shows the execution of a Python script at the path 'C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py'. The output is '[6 6 6 6 6 6]'. Below the output, it says 'Process finished with exit code 0'.

Printing Confusion matrix:

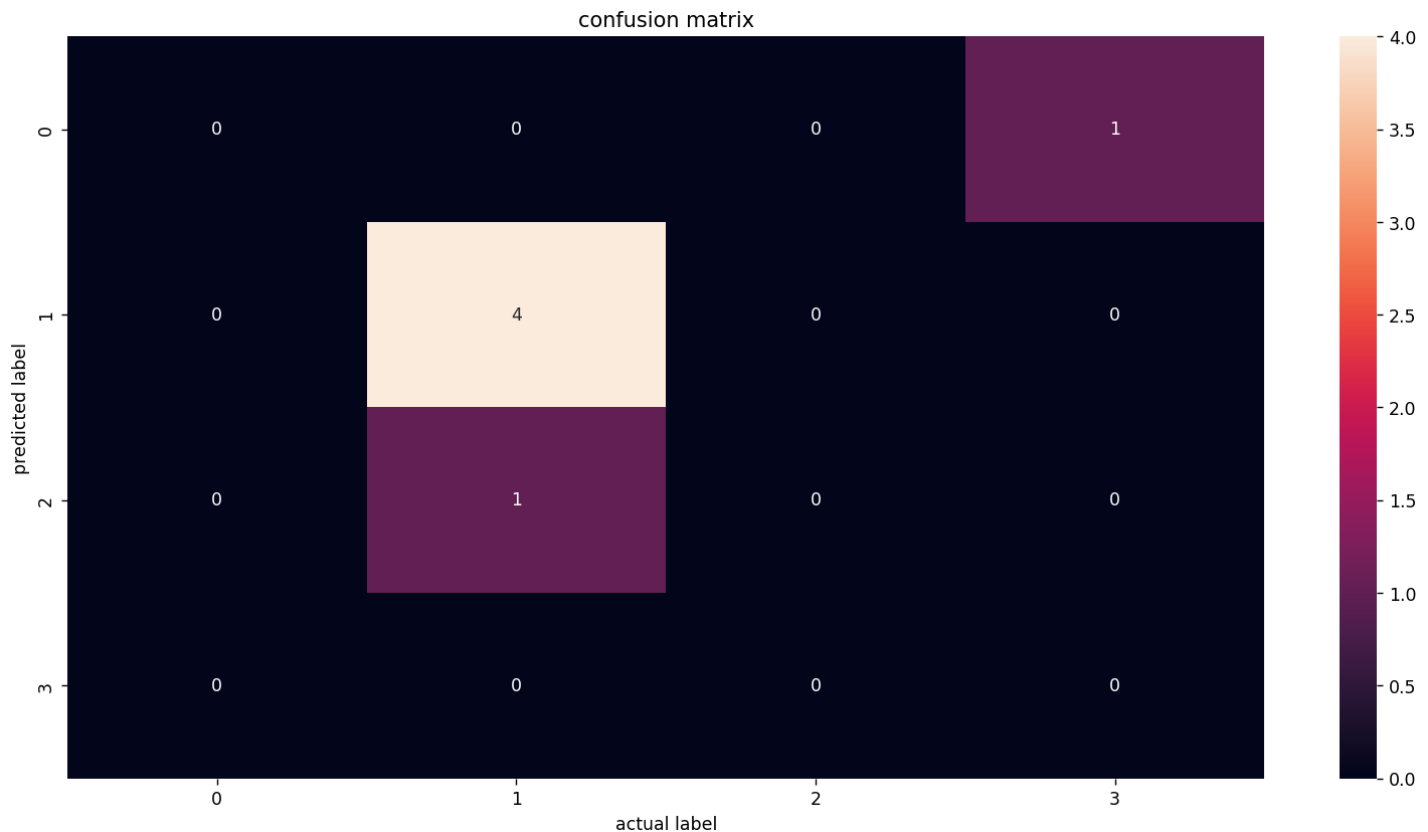
```
from sklearn import metrics
cnf_matrix=metrics.confusion_matrix(y_test,y_pred)
print(cnf_matrix)
```



```
Run: WineQualityCheck X
C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py
[[0 0 0 1]
 [0 4 0 0]
 [0 1 0 0]
 [0 0 0 0]]
Process finished with exit code 0
```

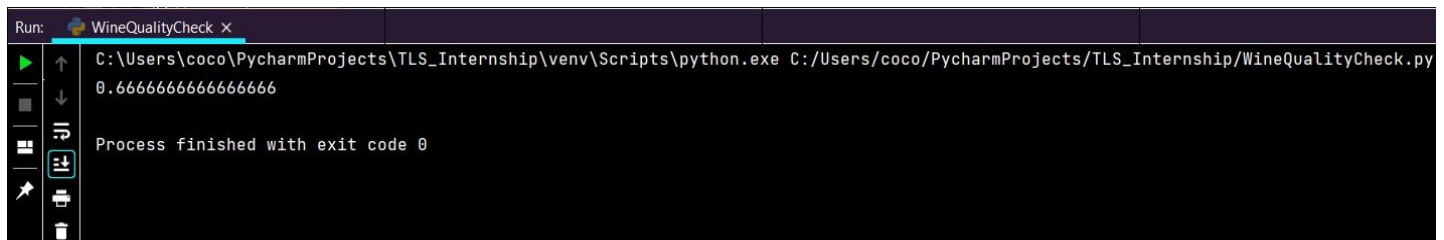
Heat Map:

```
sns.heatmap(pd.DataFrame(cnf_matrix),annot=True,fmt='g')
plt.title("confusion matrix")
plt.xlabel("actual label")
plt.ylabel("predicted label")
plt.show()
```



Accuracy :

```
print(metrics.accuracy_score(y_test,y_pred))
```



The screenshot shows a PyCharm Run console window titled "Run: WineQualityCheck x". The console output displays the command being executed: "C:\Users\coco\PycharmProjects\TLS_Internship\venv\Scripts\python.exe C:/Users/coco/PycharmProjects/TLS_Internship/WineQualityCheck.py". The output of the script is "0.6666666666666666". Below the command, it states "Process finished with exit code 0". The left sidebar of the console window contains standard icons for running, debugging, and other development actions.

Conclusion

This report uses two datasets of wine to predict the wine type, red or white, and the quality based on physicochemical properties. Quality is a subjective measure, given by the average grade of three experts. Before starting the predictions, the report makes a brief summary of model evaluation, explaining the most common metrics used in categorical problems in machine learning.

In data preparation, the training and testing sets are created and they will be used during the model building.

In data exploration and visualization, we look for features that may provide good prediction results. The best predictors have low distribution overlapping area and low correlation among them.

Modelling starts explaining very simple models and gradually moves to more complex ones. There's a brief explanation on some of the models used in this report.

The results section presents the modelling results and discusses the model performance. The algorithms are used to predict wine type; there's a section for predicting quality and the last part demonstrates clustering.

The physicochemical properties of wine differ between red and white wines, but the difference is not so evident when evaluating red wine quality. Maybe there are other properties not considered in this dataset that are better indicators for quality.

