Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns


from warnings import filterwarnings
filterwarnings(action='ignore')
```

Loading Dataset

```
wine = pd.read_csv("winequality-red.csv")
wine.sample(25)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar
chlorides  \
591             6.6             0.390         0.49            1.70
0.070
1171            7.1             0.590         0.00            2.20
0.078
289            11.6             0.420         0.53            3.30
0.105
1424            8.3             0.260         0.37            1.40
0.076
948             8.9             0.120         0.45            1.80
0.075
623             7.9             0.510         0.25            2.90
0.077
818             7.1             0.715         0.00            2.35
0.071
558            10.9             0.530         0.49            4.60
0.118
351             9.1             0.795         0.00            2.60
0.096
683             8.1             0.780         0.23            2.60
0.059
1453            7.6             0.490         0.33            1.90
0.074
1377            5.2             0.490         0.26            2.30
0.090
627             8.8             0.600         0.29            2.20
0.098
1227            9.0             0.580         0.25            2.00
0.104
```

```
163             7.4             0.600         0.26            7.30
0.070
1535            7.0             0.550         0.13            2.20
0.075
0               7.4             0.700         0.00            1.90
0.076
1461            6.2             0.785         0.00            2.10
0.060
1083            8.7             0.420         0.45            2.40
0.072
162             7.8             0.530         0.04            1.70
0.076
1113            8.9             0.240         0.39            1.60
0.074
11              7.5             0.500         0.36            6.10
0.071
1156            8.5             0.180         0.51            1.75
0.071
504            10.5             0.240         0.42            1.80
0.077
568             9.8             0.500         0.49            2.60
0.250

      free sulfur dioxide  total sulfur dioxide  density    pH
sulphates  \
591                  23.0                 149.0  0.99220  3.12
0.50
1171                 26.0                  44.0  0.99522  3.42
0.68
289                  33.0                  98.0  1.00100  3.20
0.95
1424                  8.0                  23.0  0.99740  3.26
0.70
948                  10.0                  21.0  0.99552  3.41
0.76
623                  21.0                  45.0  0.99740  3.49
0.96
818                  21.0                  47.0  0.99632  3.29
0.45
558                  10.0                  17.0  1.00020  3.07
0.56
351                  11.0                  26.0  0.99940  3.35
0.83
683                   5.0                  15.0  0.99700  3.37
0.56
1453                 27.0                  85.0  0.99706  3.41
0.58
1377                 23.0                  74.0  0.99530  3.71
0.62
```

```
627                     5.0                   15.0  0.99880  3.36
0.49
1227                    8.0                   21.0  0.99769  3.27
0.72
163                    36.0                  121.0  0.99820  3.37
0.49
1535                   15.0                   35.0  0.99590  3.36
0.59
0                      11.0                   34.0  0.99780  3.51
0.56
1461                    6.0                   13.0  0.99664  3.59
0.61
1083                   32.0                   59.0  0.99617  3.33
0.77
162                    17.0                   31.0  0.99640  3.33
0.56
1113                    3.0                   10.0  0.99698  3.12
0.59
11                     17.0                  102.0  0.99780  3.35
0.80
1156                   45.0                   88.0  0.99524  3.33
0.76
504                     6.0                   22.0  0.99760  3.21
1.05
568                     5.0                   20.0  0.99900  3.31
0.79

      alcohol  quality
591      11.5        6
1171     10.8        6
289       9.2        5
1424      9.6        6
948      11.9        7
623      12.1        6
818       9.4        5
558      11.7        6
351       9.4        6
683      11.3        5
1453      9.0        5
1377     12.2        6
627       9.1        5
1227      9.6        5
163       9.4        5
1535      9.7        6
0         9.4        5
1461     10.0        4
1083     12.0        6
162      10.0        6
1113      9.5        6
```

```
11      10.5       5
1156    11.8       7
504     10.8       7
568     10.7       6
```

```
wine.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

Description

```
wine.describe()
```

```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide
density  \
count  1599.000000          1599.000000           1599.000000
1599.000000
mean      0.087467            15.874922             46.467792
0.996747
std       0.047065            10.460157             32.895324
```

```
0.001887
min        0.012000              1.000000                      6.000000
0.990070
25%        0.070000              7.000000                     22.000000
0.995600
50%        0.079000             14.000000                     38.000000
0.996750
75%        0.090000             21.000000                     62.000000
0.997835
max        0.611000             72.000000                    289.000000
1.003690

                pH    sulphates      alcohol      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
25%       3.210000     0.550000     9.500000     5.000000
50%       3.310000     0.620000    10.200000     6.000000
75%       3.400000     0.730000    11.100000     6.000000
max       4.010000     2.000000    14.900000     8.000000
```

Finding Null Values

```
wine.isnull().sum()
```

```
fixed acidity           0
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
```

```
wine.groupby('quality').mean()
```

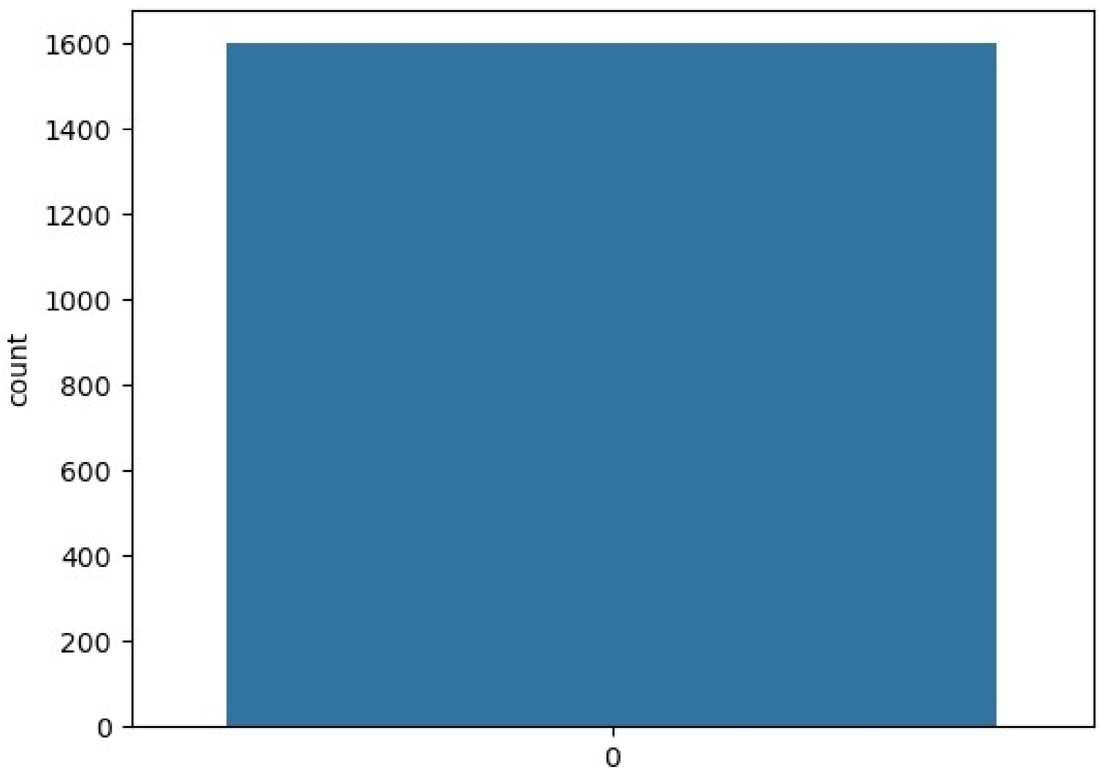
```
         fixed acidity  volatile acidity  citric acid  residual sugar
\
quality

3             8.360000          0.884500     0.171000        2.635000
```

```
4               7.779245            0.693962     0.174151        2.694340

5               8.167254            0.577041     0.243686        2.528855

6               8.347179            0.497484     0.273824        2.477194

7               8.872362            0.403920     0.375176        2.720603

8               8.566667            0.423333     0.391111        2.577778


         chlorides  free sulfur dioxide  total sulfur dioxide
density  \
quality

3         0.122500            11.000000             24.900000
0.997464
4         0.090679            12.264151             36.245283
0.996542
5         0.092736            16.983847             56.513950
0.997104
6         0.084956            15.711599             40.869906
0.996615
7         0.076588            14.045226             35.020101
0.996104
8         0.068444            13.277778             33.444444
0.995212

               pH  sulphates    alcohol
quality
3        3.398000   0.570000   9.955000
4        3.381509   0.596415  10.265094
5        3.304949   0.620969   9.899706
6        3.318072   0.675329  10.629519
7        3.290754   0.741256  11.465913
8        3.267222   0.767778  12.094444
```

Data Analysis

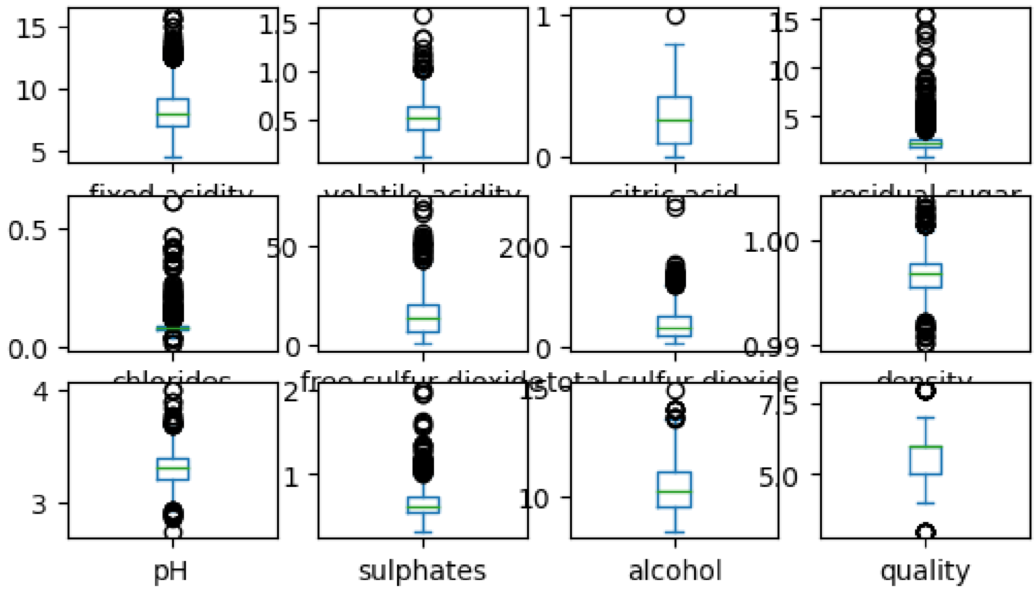
Countplot:

```
sns.countplot(wine['quality'])
plt.show()
```



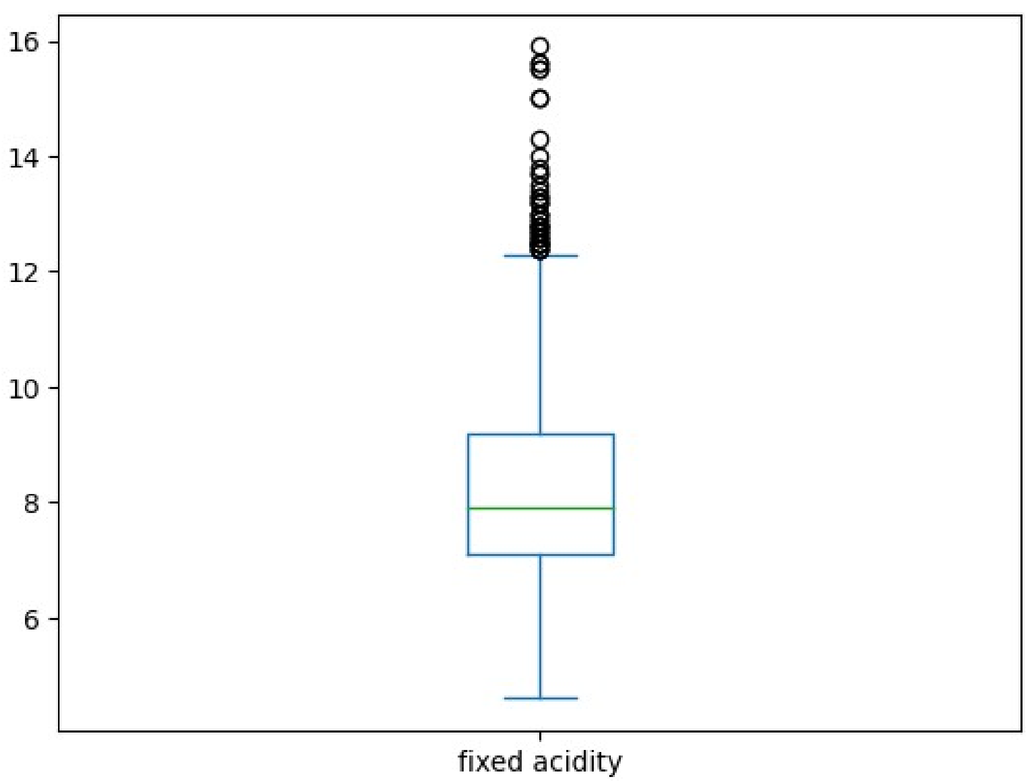

```
wine.plot(kind ='box',subplots = True, layout =(4,4),sharex = False)
```

```
fixed acidity              Axes(0.125,0.712609;0.168478x0.167391)
volatile acidity        Axes(0.327174,0.712609;0.168478x0.167391)
citric acid             Axes(0.529348,0.712609;0.168478x0.167391)
residual sugar          Axes(0.731522,0.712609;0.168478x0.167391)
chlorides                  Axes(0.125,0.511739;0.168478x0.167391)
free sulfur dioxide     Axes(0.327174,0.511739;0.168478x0.167391)
total sulfur dioxide    Axes(0.529348,0.511739;0.168478x0.167391)
density                 Axes(0.731522,0.511739;0.168478x0.167391)
pH                          Axes(0.125,0.31087;0.168478x0.167391)
sulphates                Axes(0.327174,0.31087;0.168478x0.167391)
alcohol                  Axes(0.529348,0.31087;0.168478x0.167391)
quality                  Axes(0.731522,0.31087;0.168478x0.167391)
dtype: object
```



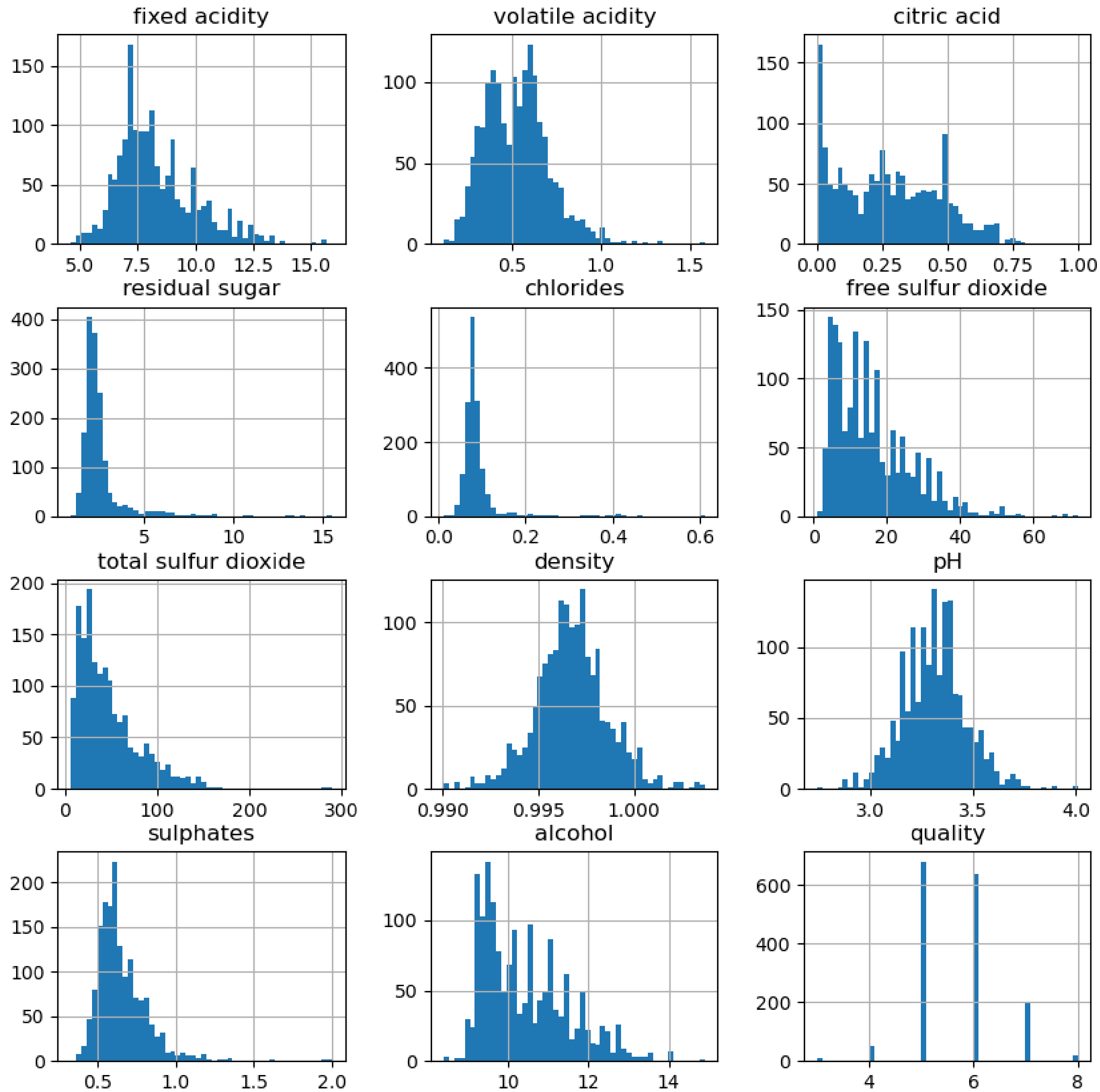

```
wine['fixed acidity'].plot(kind ='box')
```

```
<Axes: >
```



Histogram

```
wine.hist(figsize=(10,10),bins=50)
plt.show()
```



Feature Selection

```
wine.sample(5)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar  
chlorides  \
```

```
1081            7.9              0.30         0.68             8.3
0.050
471             9.6              0.54         0.42             2.4
0.081
908             7.4              0.52         0.13             2.4
0.078
1008            8.9              0.35         0.40             3.6
0.110
580            12.3              0.50         0.49             2.2
0.089

      free sulfur dioxide  total sulfur dioxide  density    pH
sulphates  \
1081                 37.5                 289.0  0.99316  3.01
0.51
471                  25.0                  52.0  0.99700  3.20
0.71
908                  34.0                  61.0  0.99528  3.43
0.59
1008                 12.0                  24.0  0.99549  3.23
0.70
580                   5.0                  14.0  1.00020  3.19
0.44

      alcohol  quality
1081     12.3        7
471      11.4        6
908      10.8        6
1008     12.0        7
580       9.6        5
```

```python
wine['quality'].unique()
```

```
array([5, 6, 7, 4, 8, 3], dtype=int64)
```

```python
# If wine quality is 7 or above then will consider as good quality
wine
wine['goodquality'] = [1 if x >= 7 else 0 for x in wine['quality']]
wine.sample(5)
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar
chlorides  \
451             8.4              0.37         0.53             1.8
0.413
235             7.2              0.63         0.00             1.9
0.097
1128           10.0              0.43         0.33             2.7
0.095
304             8.4              0.65         0.60             2.1
0.112
```

```
1276            8.5              0.40        0.40              6.3
0.050

      free sulfur dioxide  total sulfur dioxide  density    pH
sulphates  \
451                   9.0                  26.0  0.99790  3.06
1.06
235                  14.0                  38.0  0.99675  3.37
0.58
1128                 28.0                  89.0  0.99840  3.22
0.68
304                  12.0                  90.0  0.99730  3.20
0.52
1276                  3.0                  10.0  0.99566  3.28
0.56

      alcohol  quality  goodquality
451       9.1        6            0
235       9.0        6            0
1128     10.0        5            0
304       9.2        5            0
1276     12.0        4            0
```

```python
# See total number of good vs bad wines samples
wine['goodquality'].value_counts()
```

```
goodquality
0    1382
1     217
Name: count, dtype: int64
```

```python
# Separate depedent and indepedent variables
X = wine.drop(['quality','goodquality'], axis = 1)
Y = wine['goodquality']

X
```

```
      fixed acidity  volatile acidity  citric acid  residual sugar
chlorides  \
0               7.4             0.700         0.00             1.9
0.076
1               7.8             0.880         0.00             2.6
0.098
2               7.8             0.760         0.04             2.3
0.092
3              11.2             0.280         0.56             1.9
0.075
4               7.4             0.700         0.00             1.9
0.076
...             ...               ...          ...             ...
...
```

```
1594            6.2             0.600        0.08             2.0
0.090
1595            5.9             0.550        0.10             2.2
0.062
1596            6.3             0.510        0.13             2.3
0.076
1597            5.9             0.645        0.12             2.0
0.075
1598            6.0             0.310        0.47             3.6
0.067

      free sulfur dioxide  total sulfur dioxide  density    pH
sulphates  \
0                    11.0                  34.0  0.99780  3.51
0.56
1                    25.0                  67.0  0.99680  3.20
0.68
2                    15.0                  54.0  0.99700  3.26
0.65
3                    17.0                  60.0  0.99800  3.16
0.58
4                    11.0                  34.0  0.99780  3.51
0.56
...                   ...                   ...      ...   ...
...
1594                 32.0                  44.0  0.99490  3.45
0.58
1595                 39.0                  51.0  0.99512  3.52
0.76
1596                 29.0                  40.0  0.99574  3.42
0.75
1597                 32.0                  44.0  0.99547  3.57
0.71
1598                 18.0                  42.0  0.99549  3.39
0.66

      alcohol
0         9.4
1         9.8
2         9.8
3         9.8
4         9.4
...       ...
1594     10.5
1595     11.2
1596     11.0
1597     10.2
1598     11.0

[1599 rows x 11 columns]
```

```
print(Y)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1594    0
1595    0
1596    0
1597    0
1598    0
Name: goodquality, Length: 1599, dtype: int64
```

Feature Importance

```
from sklearn.ensemble import ExtraTreesClassifier
classifiern = ExtraTreesClassifier()
classifiern.fit(X,Y)
score = classifiern.feature_importances_
print(score)
```

```
[0.07484542 0.10041076 0.098194   0.07704065 0.06799978 0.06684995
 0.08334131 0.08620213 0.06586187 0.10751625 0.17173788]
```

Splitting Dataset

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test =
train_test_split(X,Y,test_size=0.3,random_state=7)
```

Result

```
model_res=pd.DataFrame(columns=['Model', 'Score'])
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train,Y_train)
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score,confusion_matrix
# accuracy_score(Y_test,Y_pred)
model_res.loc[len(model_res)] = ['LogisticRegression',
accuracy_score(Y_test,y_pred)]
model_res
```

```
                Model     Score
0  LogisticRegression  0.872917
```

SVC

```
from sklearn.svm import SVC
model = SVC()
model.fit(X_train,Y_train)
y_pred = model.predict(X_test)

from sklearn.metrics import accuracy_score
print("Accuracy Score:",accuracy_score(Y_test,y_pred))
model_res.loc[len(model_res)] = ['SVC', accuracy_score(Y_test,y_pred)]
model_res
```

```
Accuracy Score: 0.86875
```

```
                Model     Score
0  LogisticRegression  0.872917
1                 SVC  0.868750
```

Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion='entropy',random_state=7)
model.fit(X_train,Y_train)
y_pred = model.predict(X_test)

from sklearn.metrics import accuracy_score
print("Accuracy Score:",accuracy_score(Y_test,y_pred))
model_res.loc[len(model_res)] = ['DecisionTreeClassifier',
accuracy_score(Y_test,y_pred)]
model_res
```

```
Accuracy Score: 0.8645833333333334
```

```
                    Model     Score
0      LogisticRegression  0.872917
1                     SVC  0.868750
2  DecisionTreeClassifier  0.864583
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
model2 = RandomForestClassifier(random_state=1)
model2.fit(X_train, Y_train)
y_pred = model2.predict(X_test)

from sklearn.metrics import accuracy_score
print("Accuracy Score:",accuracy_score(Y_test,y_pred))
model_res.loc[len(model_res)] = ['RandomForestClassifier',
accuracy_score(Y_test,y_pred)]
model_res
```

```
Accuracy Score: 0.89375
```

	Model	Score
0	LogisticRegression	0.872917
1	SVC	0.868750
2	DecisionTreeClassifier	0.864583
3	RandomForestClassifier	0.893750

```
model_res = model_res.sort_values(by='Score', ascending=False)
model_res
```

	Model	Score
3	RandomForestClassifier	0.893750
0	LogisticRegression	0.872917
1	SVC	0.868750
2	DecisionTreeClassifier	0.864583