Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

Data Set

```
wine_set=pd.read_csv('/content/WineQT.csv')

wine_set.shape

(1143, 12)

wine_set.head()

   fixed acidity  volatile acidity  citric acid  residual sugar
chlorides  \
0            7.4              0.70         0.00             1.9
0.076
1            7.8              0.88         0.00             2.6
0.098
2            7.8              0.76         0.04             2.3
0.092
3           11.2              0.28         0.56             1.9
0.075
4            7.4              0.70         0.00             1.9
0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates
\
0                 11.0                  34.0   0.9978  3.51       0.56

1                 25.0                  67.0   0.9968  3.20       0.68

2                 15.0                  54.0   0.9970  3.26       0.65

3                 17.0                  60.0   0.9980  3.16       0.58

4                 11.0                  34.0   0.9978  3.51       0.56


   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
```

```
3      9.8        6
4      9.4        5
```

```
wine_set.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
```

Data Analysis and Visualization

```
wine_set.describe()
```

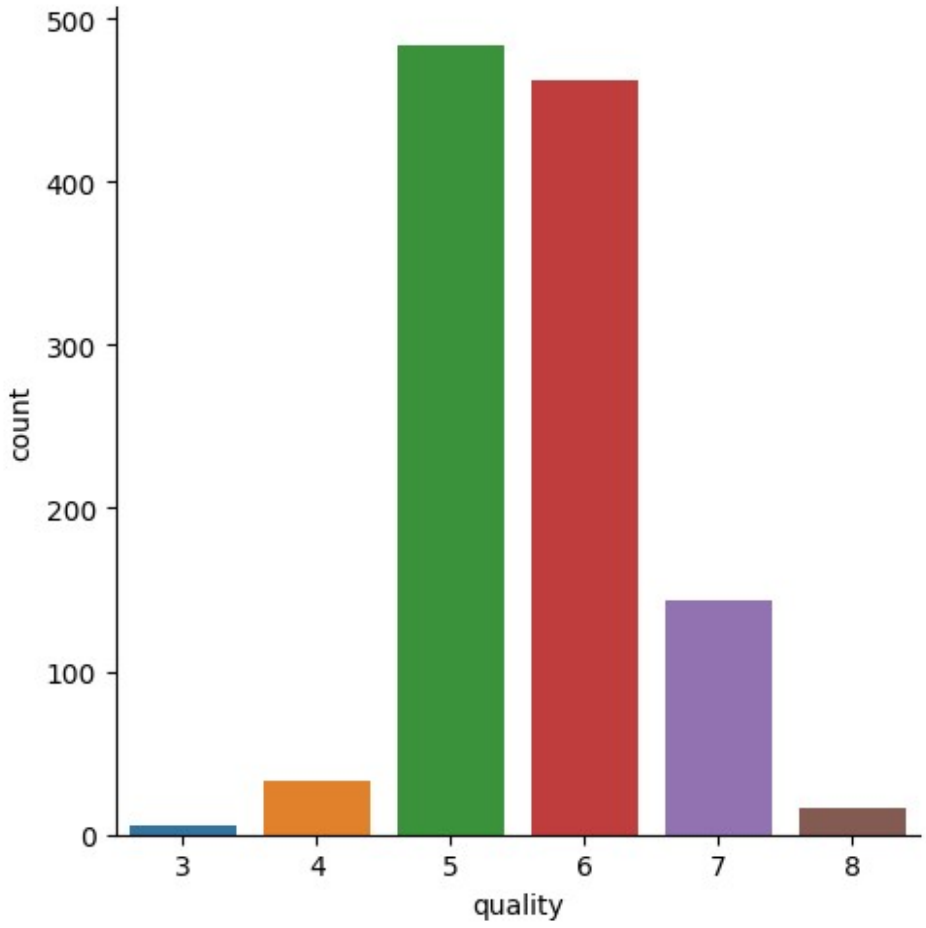
```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1143.000000       1143.000000  1143.000000     1143.000000
mean        8.311111          0.531339     0.268364        2.532152
std         1.747595          0.179633     0.196686        1.355917
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.392500     0.090000        1.900000
50%         7.900000          0.520000     0.250000        2.200000
75%         9.100000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide
density  \
count  1143.000000          1143.000000           1143.000000
1143.000000
mean      0.086933            15.615486             45.914698
0.996730
std       0.047267            10.250486             32.782130
0.001925
min       0.012000             1.000000              6.000000
0.990070
25%       0.070000             7.000000             21.000000
0.995570
50%       0.079000            13.000000             37.000000
0.996680
75%       0.090000            21.000000             61.000000
0.997845
max       0.611000            68.000000            289.000000
```

```
1.003690
```

	pH	sulphates	alcohol	quality
count	1143.000000	1143.000000	1143.000000	1143.000000
mean	3.311015	0.657708	10.442111	5.657043
std	0.156664	0.170399	1.082196	0.805824
min	2.740000	0.330000	8.400000	3.000000
25%	3.205000	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

```
sns.catplot(x='quality',data=wine_set,kind='count')
```

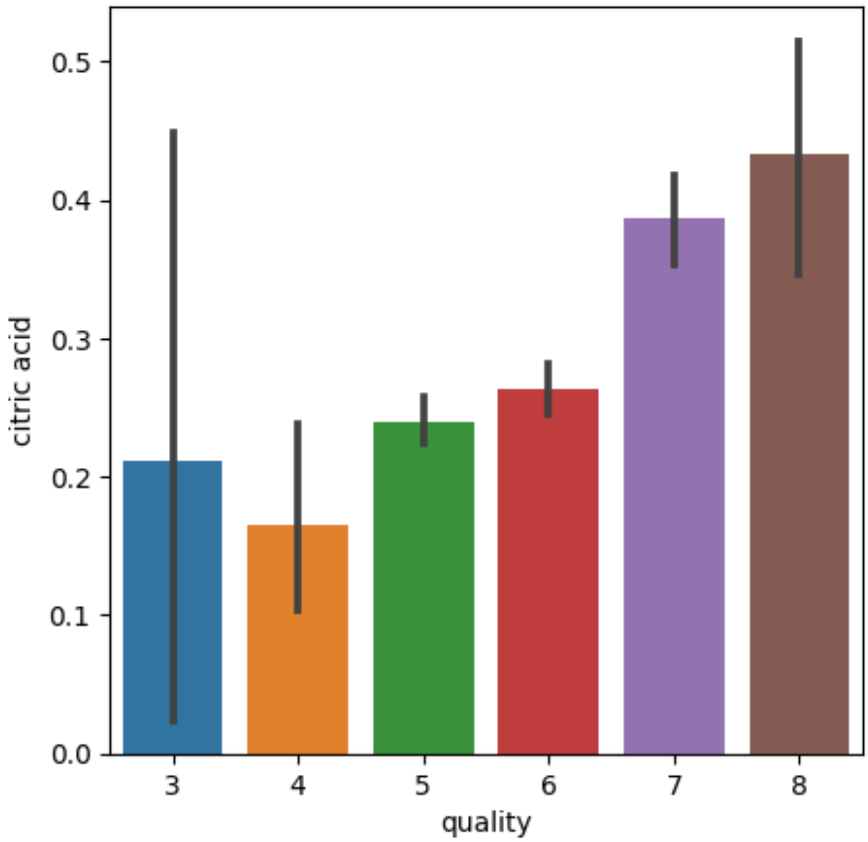
```
<seaborn.axisgrid.FacetGrid at 0x7a5543dfed10>
```



Citric Acid Vs Quality

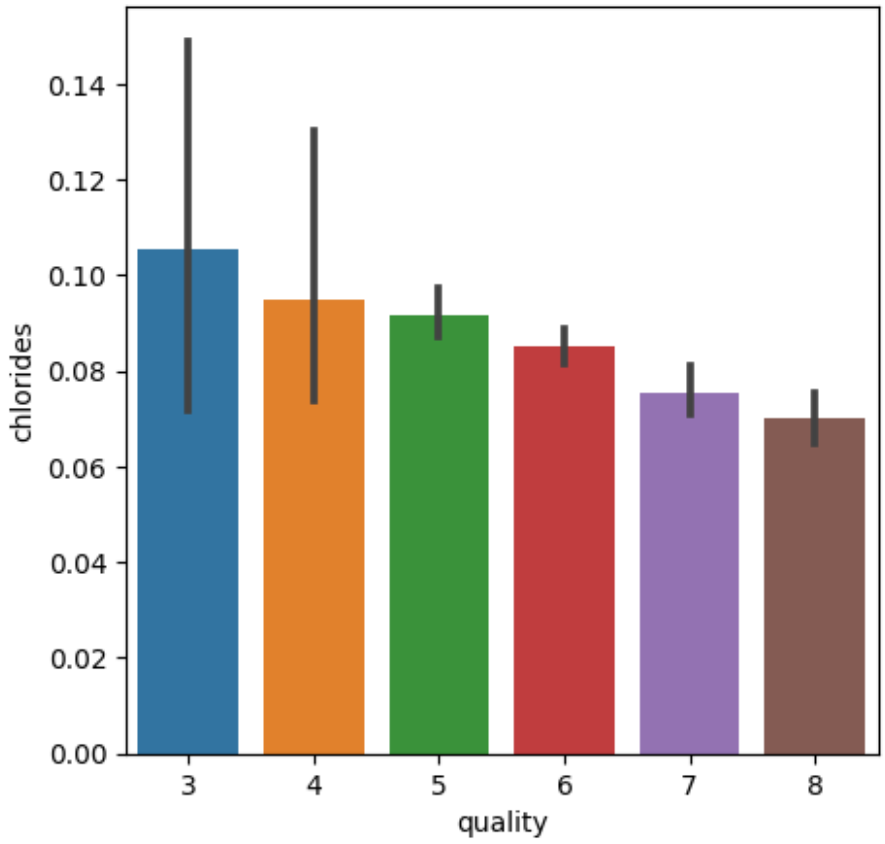
```
plot=plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='citric acid',data=wine_set)
```

```
<Axes: xlabel='quality', ylabel='citric acid'>
```



Chlorides vs Quality

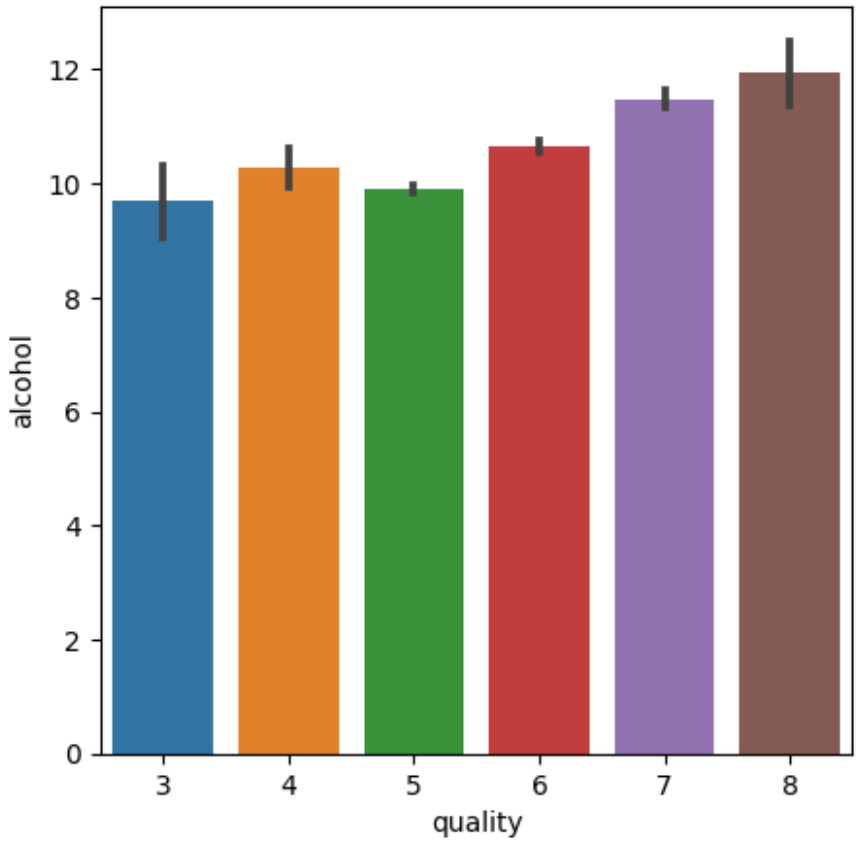
```
plot=plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='chlorides',data=wine_set)

<Axes: xlabel='quality', ylabel='chlorides'>
```



Alcohol vs Quality

```
plot=plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='alcohol',data=wine_set)
```

```
<Axes: xlabel='quality', ylabel='alcohol'>
```



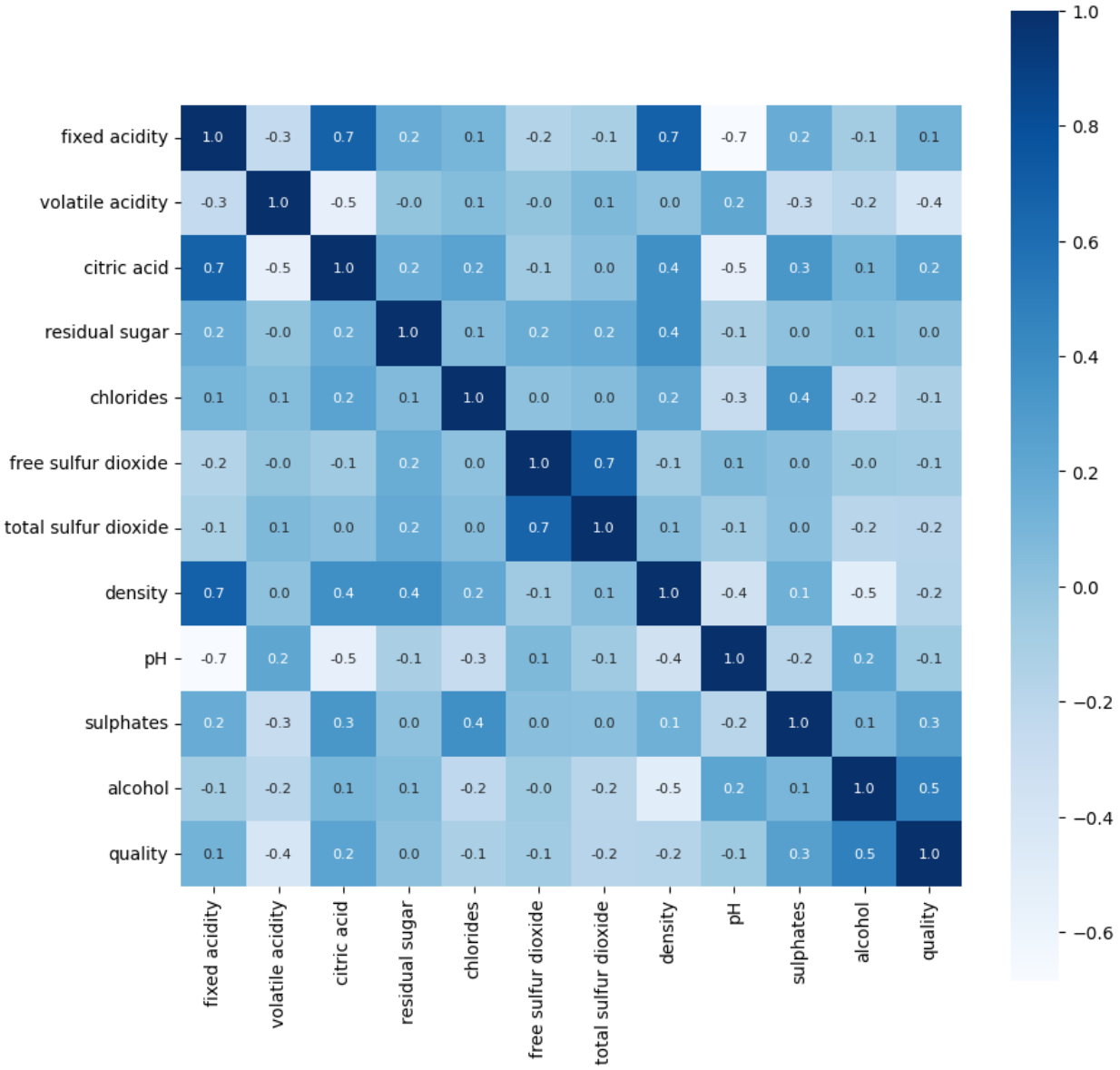
Correlation

Positive Correlation

Negative Correlation

```
correlation=wine_set.corr()

plt.figure(figsize=(10,10))
sns.heatmap(correlation,cbar=True,square=True,fmt='.1f',annot=True,ann
ot_kws={'size':8},cmap='Blues')
```

```
<Axes: >
```



Data Preprocessing

```
X = wine_set.drop('quality',axis=1)

print(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
...              ...               ...          ...               ...
...
1138             6.3             0.510         0.13              2.3
0.076
1139             6.8             0.620         0.08              1.9
0.068
1140             6.2             0.600         0.08              2.0
0.090
1141             5.9             0.550         0.10              2.2
0.062
1142             5.9             0.645         0.12              2.0
0.075

      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
...                   ...                   ...      ...   ...
...
1138                 29.0                  40.0  0.99574  3.42
0.75
1139                 28.0                  38.0  0.99651  3.42
0.82
1140                 32.0                  44.0  0.99490  3.45
0.58
1141                 39.0                  51.0  0.99512  3.52
0.76
1142                 32.0                  44.0  0.99547  3.57
0.71

      alcohol
0         9.4
1         9.8
2         9.8
3         9.8
4         9.4
...       ...
```

```
1138     11.0
1139      9.5
1140     10.5
1141     11.2
1142     10.2

[1143 rows x 11 columns]
```

Label Binarization

```
Y = wine_set['quality'].apply(lambda y_value: 1 if y_value>=7 else 0)

print(Y)
```

```
0       0
1       0
2       0
3       0
4       0
       ..
1138    0
1139    0
1140    0
1141    0
1142    0
Name: quality, Length: 1143, dtype: int64
```

Train and Test Split

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,rando
m_state=3)

print(Y.shape,Y_train.shape,Y_test.shape)
```

```
(1143,) (914,) (229,)
```

Model Training: Random Forest Classifier

```
model = RandomForestClassifier()

model.fit(X_train,Y_train)
```

```
RandomForestClassifier()
```

Model Evaluation

Accuracy Score

```
X_test_prediction=model.predict(X_test)
test_data_accuracy=accuracy_score(X_test_prediction,Y_test)
```

```
print('Accuracy: ',test_data_accuracy)
```

```
Accuracy:  0.8951965065502183
```

Predictive System

```
input_data=(8.5,0.28,0.56,1.8,0.092,35,103,0.9969,3.3,0.75,10.5)

input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if(prediction[0]==1):
  print('Good Quality Wine')
else:
  print('Bad Quality Wine')
```

```
[1]
Good Quality Wine

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but
RandomForestClassifier was fitted with feature names
  warnings.warn(
```

```
input_data=(7.9,0.32,0.51,1.8,0.341,17,56,0.9969,3.04,1.08,9.2)

input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if(prediction[0]==1):
  print('Good Quality Wine')
else:
  print('Bad Quality Wine')
```

```
[0]
Bad Quality Wine

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but
RandomForestClassifier was fitted with feature names
  warnings.warn(
```