

```
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
```

```
wine=pd.read_csv("/content/winequality-red.csv")
wine.shape
```

```
(1599, 12)
```

```
wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulph
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	

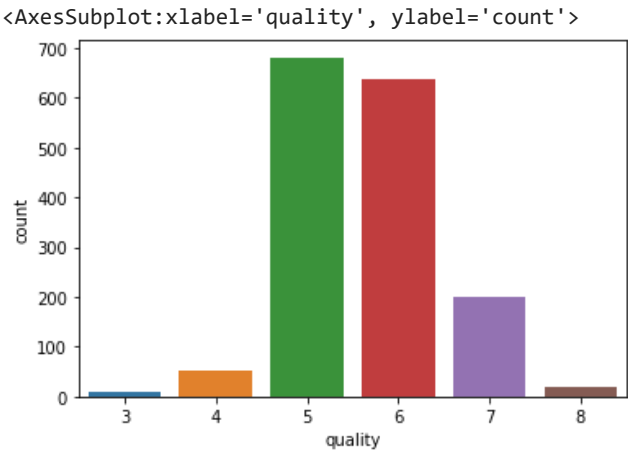
```
wine.isnull().sum()
```

```
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
```

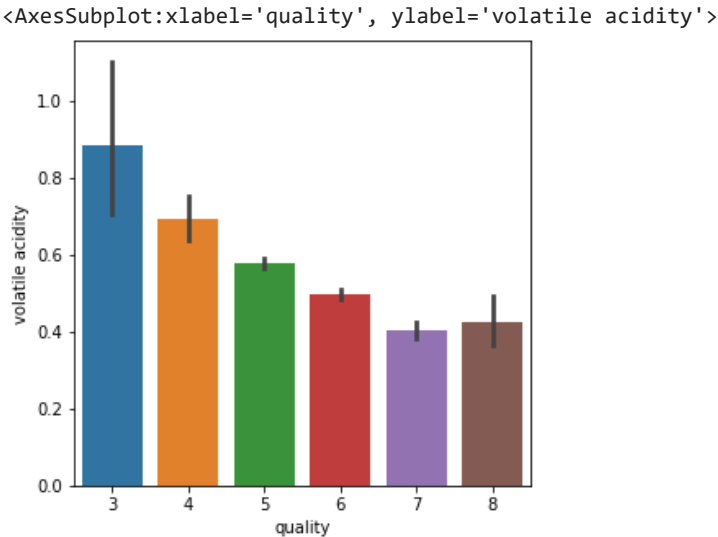
```
wine.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	15
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	

```
sns.countplot(x='quality',data=wine)
```



```
plot=plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='volatile acidity',data=wine)
```



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

<AxesSubplot:xlabel='quality', ylabel='citric acid'>



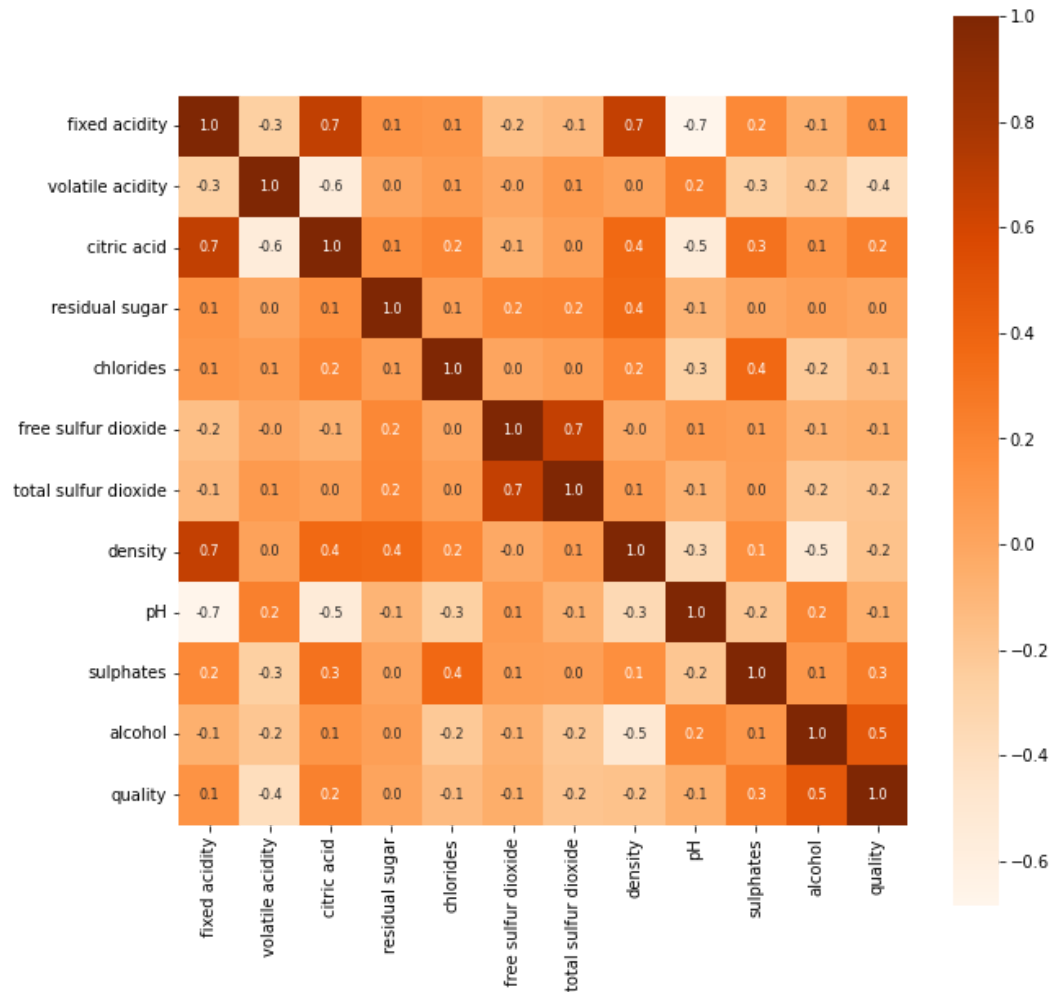
```
correlation=wine.corr()
```



```
plt.figure(figsize=(10,10))
```

```
sns.heatmap(correlation,cbar=True,square=True,fmt='.1f',annot=True,annot_kws={'size':8},cmap='Oranges')
```

<AxesSubplot:>



```
X=wine.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	
...	
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]

```
Y=wine['quality'].apply(lambda yval: 1 if yval>=7 else 0)
print(Y)
```

0	0
1	0
2	0
3	0
4	0
...	..
1594	0
1595	0
1596	0
1597	0
1598	0

Name: quality, Length: 1599, dtype: int64

```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=3)
```

```
print(x_train.shape,x_test.shape,X.shape)
```

(1279, 11) (320, 11) (1599, 11)

```
model=RandomForestClassifier()
```

```
model.fit(x_train,y_train)
```

```
▼ RandomForestClassifier
RandomForestClassifier()
```

```
predict=model.predict(x_test)
accuracy=accuracy_score(predict,y_test)
```

```
print("The Accuracy of the model:",accuracy)
```

```
The Accuracy of the model: 0.925
```

```
data=[7.5,2.5,3.09,3.81,6.2,8.3,0.9978,0.2,2.7,10.5,2.5]
arr=np.asarray(data)
data_arr=arr.reshape(1,-1)
prediction=model.predict(data_arr)
print(prediction)
if(prediction[0]==1):
    print("Quality of Wine is Good")
else:
    print("Quality of Wine is Bad")

[0]
Quality of Wine is Bad
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:420: UserWarning: X does not have valid feature names,
warnings.warn(
```

```
import pickle
```

```
filename='trained_model.sav'
```

```
pickle.dump(model,open(filename, 'wb'))
```

```
loaded_model=pickle.load(open('trained_model.sav','rb'))
```

```
data=[7.5,2.5,3.09,3.81,6.2,8.3,0.9978,0.2,2.7,10.5,2.5]
arr=np.asarray(data)
data_arr=arr.reshape(1,-1)
prediction=loaded_model.predict(data_arr)
print(prediction)
if(prediction[0]==1):
    print("Quality of Wine is Good")
else:
    print("Quality of Wine is Bad")

[0]
Quality of Wine is Bad
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:420: UserWarning: X does not have valid feature names,
warnings.warn(
```

