

NAME: VANDHITHA R S

EMAIL: vandithachinnu77777@gmail.com
(<mailto:vandithachinnu77777@gmail.com>)

In [1]:

```
pip install sklearn
```

Requirement already satisfied: sklearn in c:\programdata\anaconda3\lib\site-packages (0.0)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\lib\site-packages (from sklearn) (0.24.1)

Requirement already satisfied: numpy>=1.13.3 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.20.1)

Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.0.1)

Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sklearn) (2.1.0)

Requirement already satisfied: scipy>=0.19.1 in c:\programdata\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.6.2)

In [1]:

```
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
```

In [4]:

```
# Loading the dataset to a Pandas DataFrame
wine_dataset = pd.read_csv('winequality.csv')
```

In [5]:

```
# number of rows & columns in the dataset
wine_dataset.shape
```

Out[5]:

```
(1599, 12)
```

In [6]:

```
#first 5 rows of the dataset
wine_dataset.head()
```

Out[6]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9

In [8]:

```
# checking for missing values
wine_dataset.isnull().sum()
```

Out[8]:

```
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
```

In [9]:

```
# statistical measures of the dataset
wine_dataset.describe()
```

Out[9]:

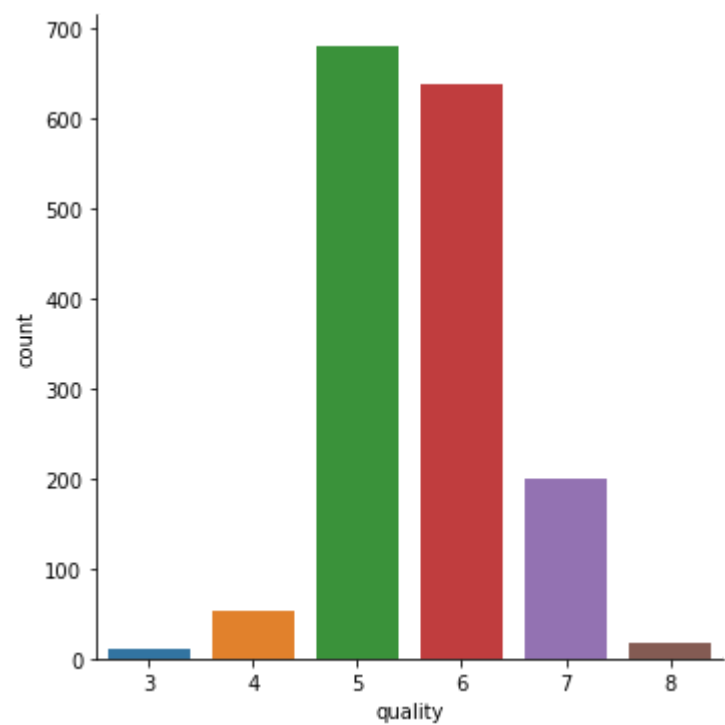
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467091
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895794
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000

In [10]:

```
# number of values for each quality
sns.catplot(x='quality', data = wine_dataset, kind = 'count')
```

Out[10]:

<seaborn.axisgrid.FacetGrid at 0x2267012d4c0>

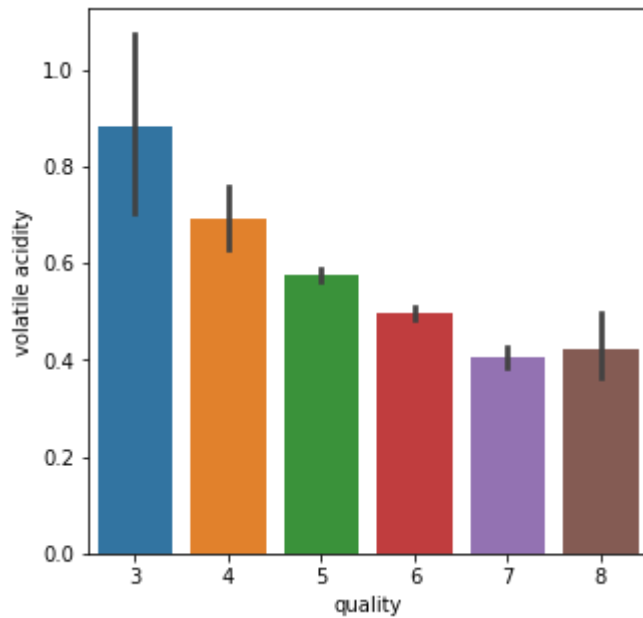


In [11]:

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

Out[11]:

<AxesSubplot:xlabel='quality', ylabel='volatile acidity'>

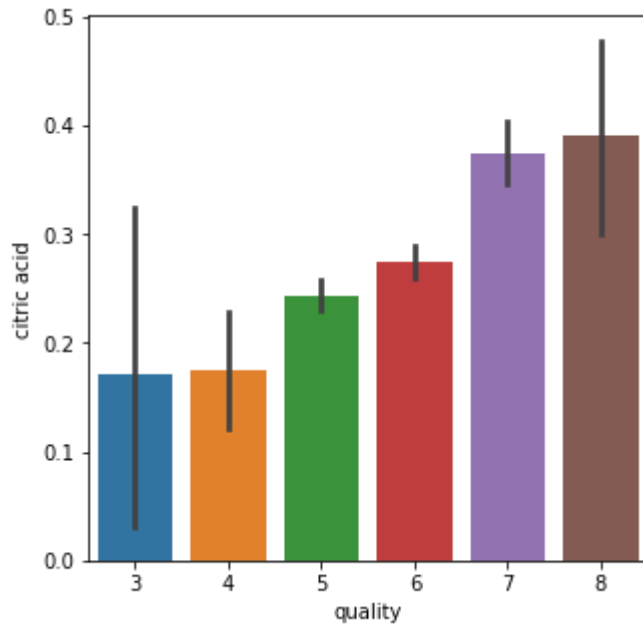


In [12]:

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

Out[12]:

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



In [14]:

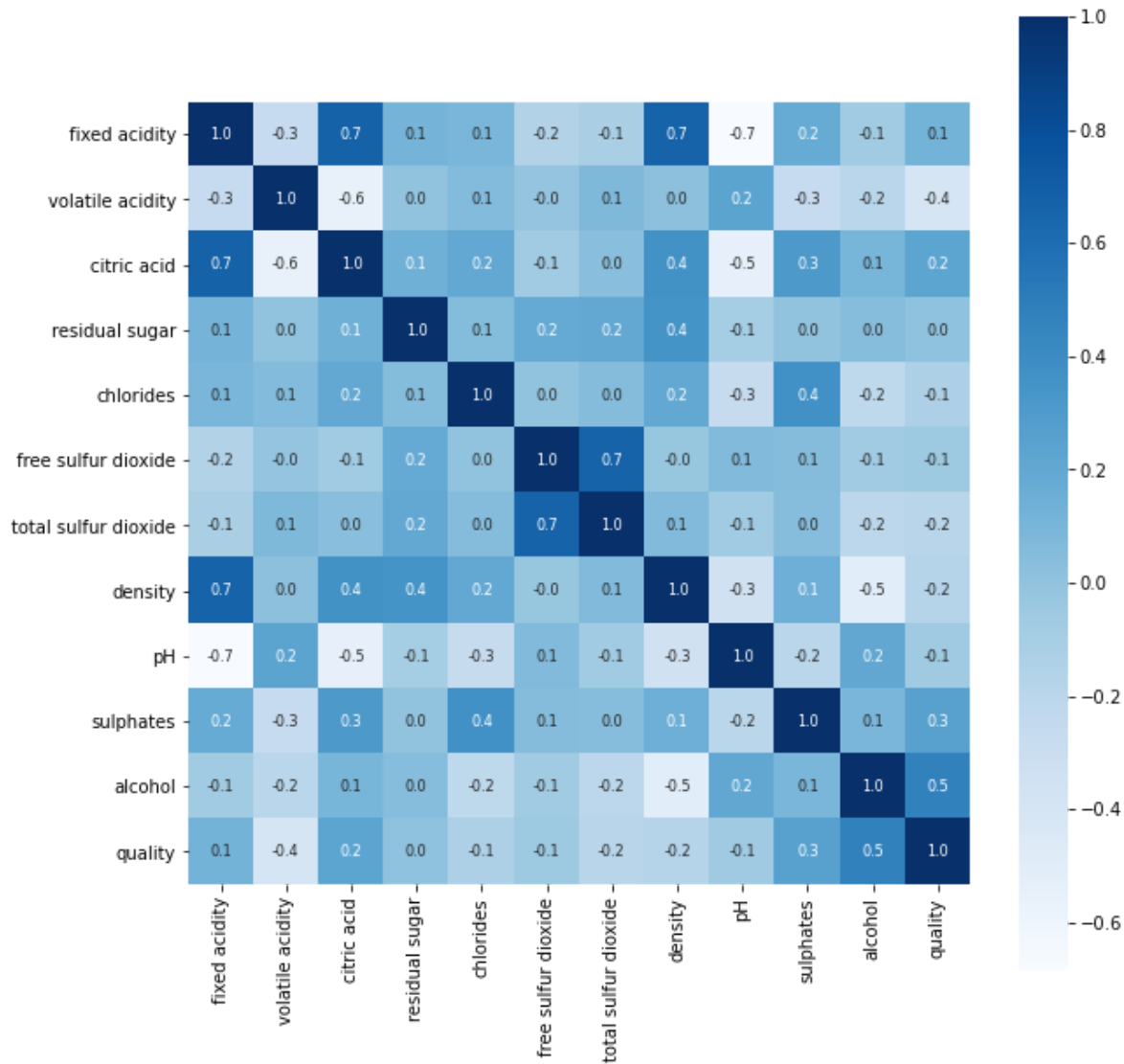
```
correlation = wine_dataset.corr()
```

In [15]:

```
# constructing a heatmap to understand the correlation between the columns
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':
```

Out[15]:

<AxesSubplot:>



In [17]:

```
# separate the data and Label  
X= wine_dataset.drop('quality',axis=1)
```

In [18]:

```
print(X)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chloride
0	7.4	0.700	0.00	1.9	0.07
1	7.8	0.880	0.00	2.6	0.09
2	7.8	0.760	0.04	2.3	0.09
3	11.2	0.280	0.56	1.9	0.07
4	7.4	0.700	0.00	1.9	0.07
...
1594	6.2	0.600	0.08	2.0	0.09
1595	5.9	0.550	0.10	2.2	0.06
1596	6.3	0.510	0.13	2.3	0.07
1597	5.9	0.645	0.12	2.0	0.07
1598	6.0	0.310	0.47	3.6	0.06

	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]

In [21]:

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


In [22]:

```
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
```

In [27]:

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

In [28]:

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

```
(1599,) (1279,) (320,)
```

In [29]:

```
model = RandomForestClassifier()
```

In [30]:

```
model.fit(X_train, Y_train)
```

Out[30]:

```
RandomForestClassifier()
```

In [33]:

```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

In [34]:

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

```
Accuracy :  0.925
```

In [37]:

```
input_data = (7.5,0.5,0.36,6.1,0.071,17.0,102.0,0.9978,3.35,0.8,10.5)

# changing the input data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the data as we are predicting the Label for only one instance
input_data_resaped = input_data_as_numpy_array.reshape(1,-1)

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

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

```
[0]
Bad Quality Wine
```