```
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 [2]:  wine_dataset=pd.read_csv('winequality-red.csv')
```

```
In [3]:  wine_dataset.shape
```

Out[3]: (1599, 12)

```
In [4]:  wine_dataset.head()
```

Out[4]:

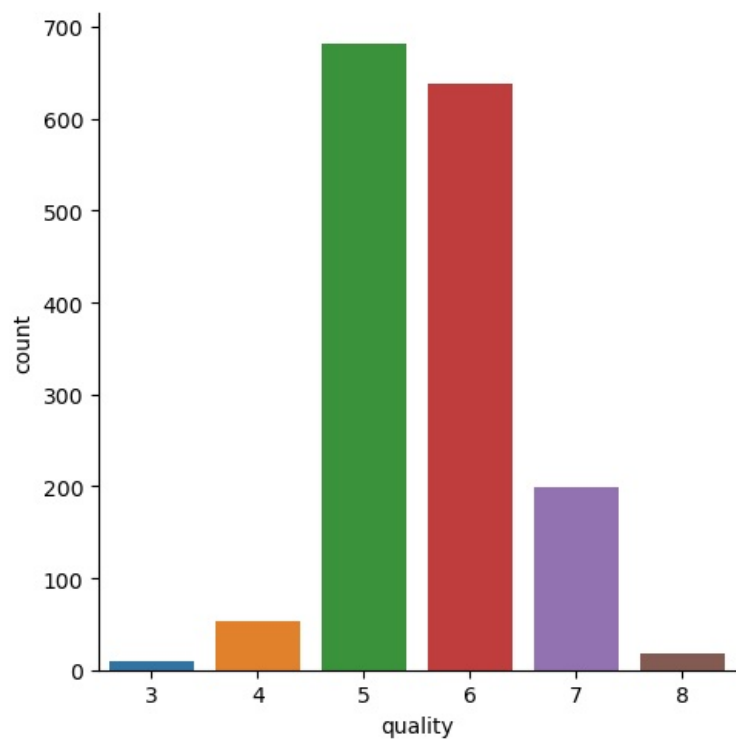|	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

```
In [5]:  wine_dataset.describe()
```

Out[5]:

|	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	

```
In [6]:  sns.catplot(x='quality',data=wine_dataset,kind='count')
```

Out[6]: <seaborn.axisgrid.FacetGrid at 0x21a129ffa00>


```
In [7]:  plot=plt.figure(figsize=(5,5))
         sns.barplot(x='quality',y='volatile acidity',data=wine_dataset)
```

`<Axes: xlabel='quality', ylabel='volatile acidity'>`
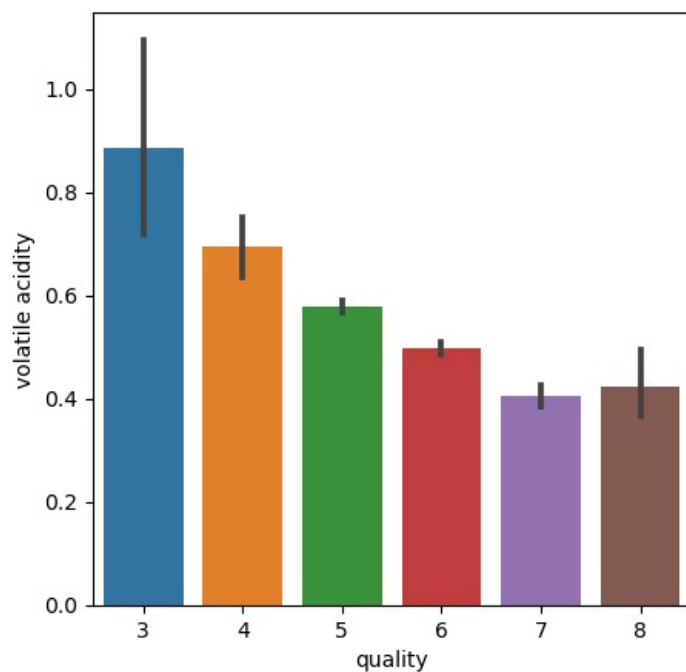
```python
plot=plt.figure(figsize=(5,5))
sns.barplot(x='quality',y='citric acid',data=wine_dataset)
```
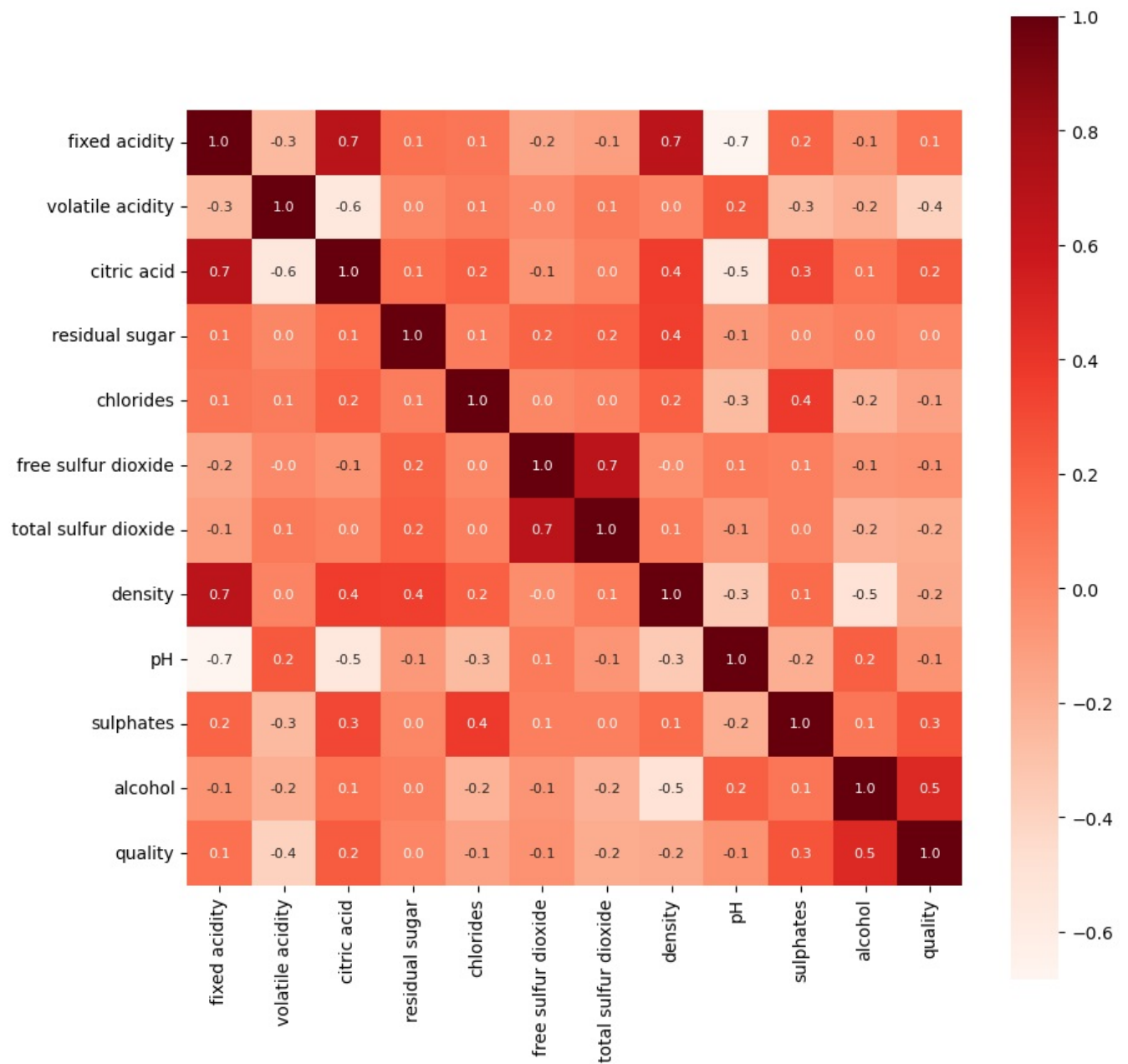
`<Axes: xlabel='quality', ylabel='citric acid'>`
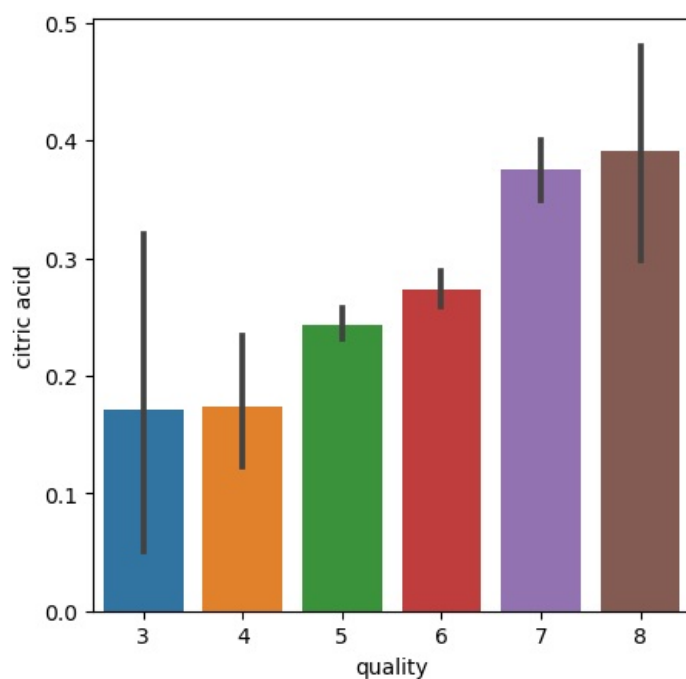
```python
correlation=wine_dataset.corr()
```

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

`<Axes: >`

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.0	-0.3	0.7	0.1	0.1	-0.2	-0.1	0.7	-0.7	0.2	-0.1	0.1
volatile acidity	-0.3	1.0	-0.6	0.0	0.1	-0.0	0.1	0.0	0.2	-0.3	-0.2	-0.4
citric acid	0.7	-0.6	1.0	0.1	0.2	-0.1	0.0	0.4	-0.5	0.3	0.1	0.2
residual sugar	0.1	0.0	0.1	1.0	0.1	0.2	0.2	0.4	-0.1	0.0	0.0	0.0
chlorides	0.1	0.1	0.2	0.1	1.0	0.0	0.0	0.2	-0.3	0.4	-0.2	-0.1
free sulfur dioxide	-0.2	-0.0	-0.1	0.2	0.0	1.0	0.7	-0.0	0.1	0.1	-0.1	-0.1
total sulfur dioxide	-0.1	0.1	0.0	0.2	0.0	0.7	1.0	0.1	-0.1	0.0	-0.2	-0.2
density	0.7	0.0	0.4	0.4	0.2	-0.0	0.1	1.0	-0.3	0.1	-0.5	-0.2
pH	-0.7	0.2	-0.5	-0.1	-0.3	0.1	-0.1	-0.3	1.0	-0.2	0.2	-0.1
sulphates	0.2	-0.3	0.3	0.0	0.4	0.1	0.0	0.1	-0.2	1.0	0.1	0.3
alcohol	-0.1	-0.2	0.1	0.0	-0.2	-0.1	-0.2	-0.5	0.2	0.1	1.0	0.5
quality	0.1	-0.4	0.2	0.0	-0.1	-0.1	-0.2	-0.2	-0.1	0.3	0.5	1.0

In [14]:
```python
x=wine_dataset.drop('quality',axis=1)
```

In [15]:
```python
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]
```

In [17]:
```python
y=wine_dataset['quality'].apply(lambda y_value: 1 if y_value>=7 else 0)
```

In [18]:
```python
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 [20]:
```python
x_train, x_test,y_train, y_test=train_test_split(x,y,test_size=0.2,random_state=3)
```

In [22]:
```python
print(y.shape,y_train.shape,y_test.shape)
```
```
(1599,) (1279,) (320,)
```

In [23]:
```python
model=RandomForestClassifier()
```

In [24]:
```python
model.fit(x_train,y_train)
```

Out[24]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [25]:
```python
x_test_prediction=model.predict(x_test)
test_data_accuracy=accuracy_score(x_test_prediction,y_test)
```

In [26]:
```python
print('Accuracy:',test_data_accuracy)
```
```
Accuracy: 0.921875
```

In [29]:
```python
input_data= (7.3,0.65,0.0,1.2,0.065,15.0,21.0,0.9946,3.39,0.47,10.0)

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):
```

```python
    print("Good Quality Wine")
else:
    print("Bad Quality Wine")
```

[1]
Good Quality Wine

In [ ]:

In [ ]: