

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
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 = pd.read_csv('Wine.csv')
wine.head()
```

	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	5.0
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5.0
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5.0
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6.0
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5.0

```
In [3]: wine.shape

Out[3]: (1599, 12)
```

```
In [5]: wine.isna().sum()

Out[5]: fixed acidity      0
volatile acidity      0
citric acid           0
residual sugar        0
chlorides             0
free sulfur dioxide    0
total sulfur dioxide   1
density              0
pH                   1
sulphates             0
alcohol              0
quality              1
dtype: int64
```

```
In [7]: wine.dropna(subset=['total sulfur dioxide', 'pH', 'quality'], inplace=True)
wine.isna().sum()

Out[7]: 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 [8]: wine.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1596 entries, 0 to 1598
Data columns (total 12 columns):
#   Column              Non-Null Count  Dtype
---  --
0   fixed acidity        1596 non-null   float64
1   volatile acidity     1596 non-null   float64
2   citric acid          1596 non-null   float64
3   residual sugar       1596 non-null   float64
4   chlorides            1596 non-null   float64
5   free sulfur dioxide  1596 non-null   float64
6   total sulfur dioxide 1596 non-null   float64
7   density              1596 non-null   float64
8   pH                  1596 non-null   float64
9   sulphates            1596 non-null   float64
10  alcohol              1596 non-null   float64
11  quality              1596 non-null   float64
dtypes: float64(12)
memory usage: 162.1 KB
```

```
In [9]: wine.describe()

Out[9]:
```

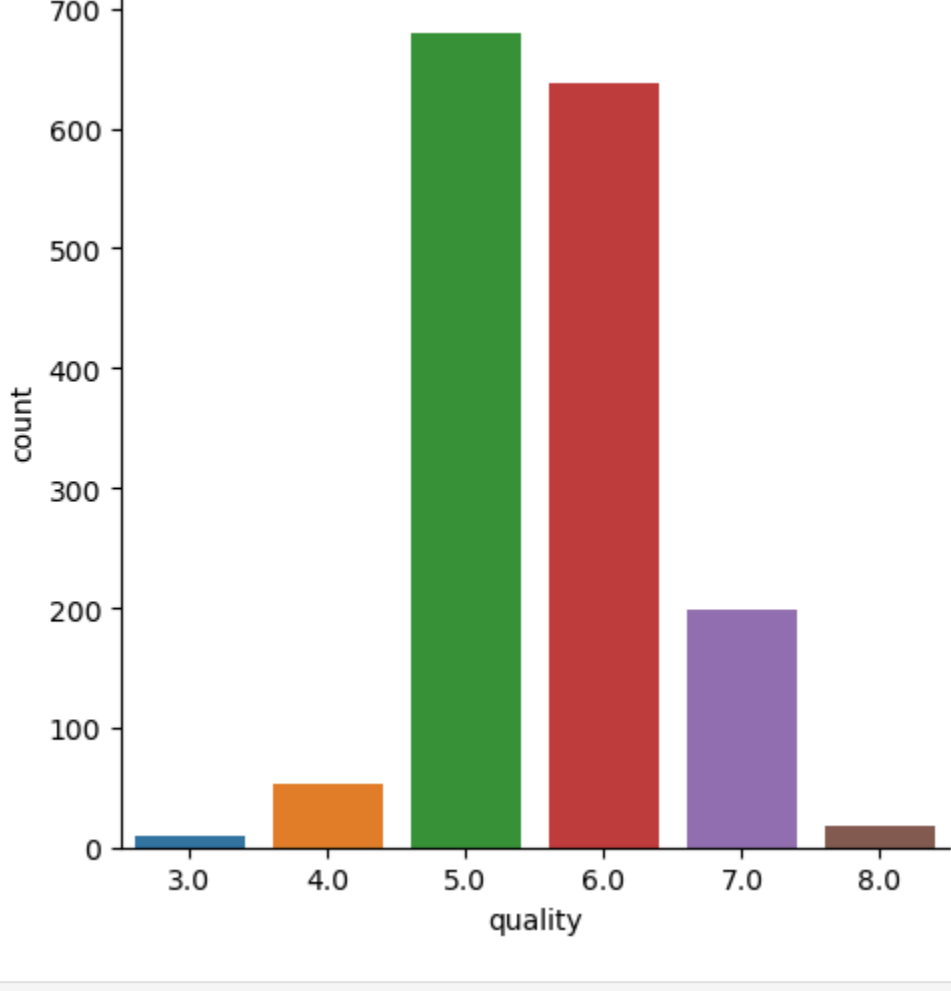
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000	1596.000000
mean	8.321366	0.527666	0.271128	2.536936	0.087487	15.882206	46.431078	0.996745	3.498716	0.658189	10.424217	5.636591
std	1.742121	0.179154	0.194847	1.408341	0.047107	10.467380	32.893072	0.001889	0.080297	0.169587	1.066046	0.807963
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.520000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996745	3.520000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997842	3.520000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	3.900000	2.000000	14.900000	8.000000

```
In [10]: wine.dtypes

Out[10]: fixed acidity      float64
volatile acidity      float64
citric acid           float64
residual sugar        float64
chlorides             float64
free sulfur dioxide    float64
total sulfur dioxide   float64
density              float64
pH                   float64
sulphates            float64
alcohol              float64
quality              float64
dtype: object
```

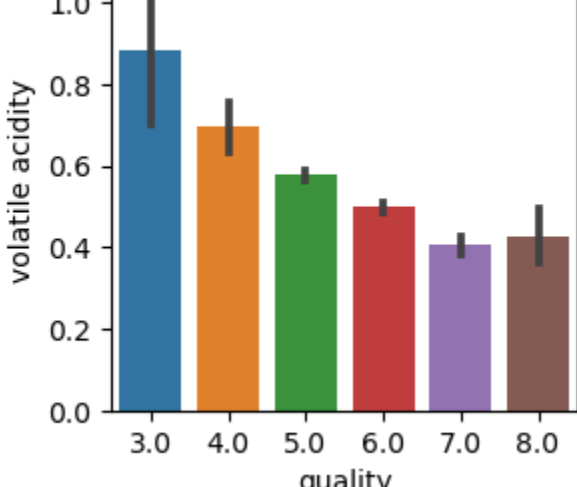
```
In [11]: sns.catplot(x='quality', data=wine, kind='count')

C:\Users\rames_ykh6ekx\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)
<seaborn.axisgrid.FacetGrid at 0x16872e5c610>
```



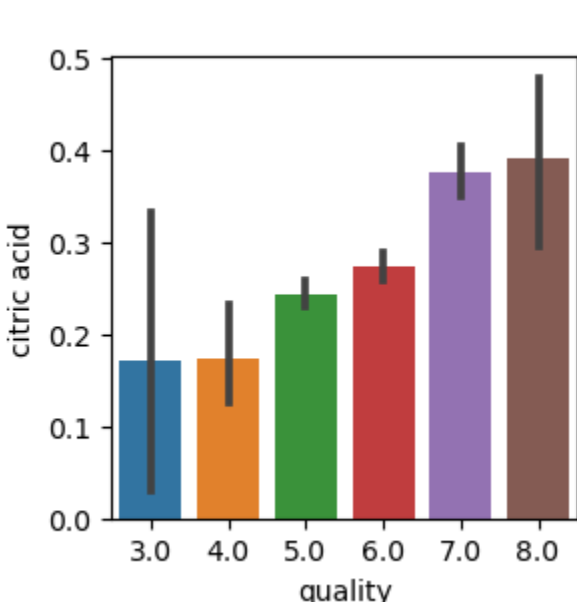
```
In [13]: plot=plt.figure(figsize=(3,3))
sns.barplot(x='quality', y='volatile acidity', data=wine)

Out[13]: <Axes: xlabel='quality', ylabel='volatile acidity'>
```



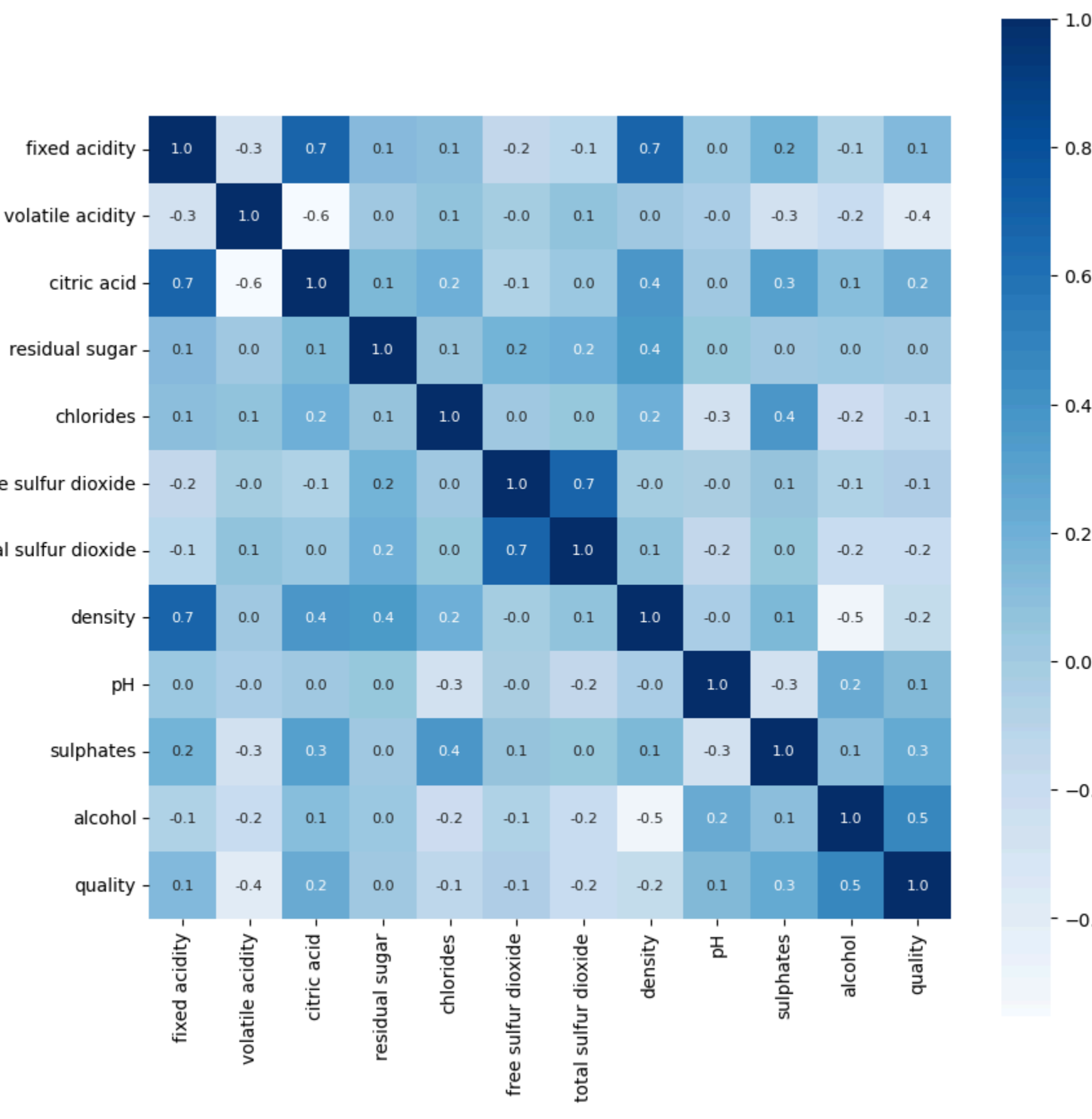
```
In [14]: plot=plt.figure(figsize=(3,3))
sns.barplot(x='quality', y='citric acid', data=wine)

Out[14]: <Axes: xlabel='quality', ylabel='citric acid'>
```



```
In [15]: correlation=wine.corr()
plt.figure(figsize=(10,10))
sns.heatmap(correlation, square=True, fmt='.1f', annot=True, annot_kws={'size':8}, cmap='Blues')

Out[15]: <Axes: >
```



```
In [16]: X=wine.drop('quality', axis=1)
X.head()

Out[16]:
```

	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	9.4
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4

```
In [17]: Y=wine['quality'].apply(lambda y_values: 1 if y_values>=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: 1596, dtype: int64
```

```
In [21]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.14, random_state=2)
print(Y.shape, Y_train.shape, Y_test.shape)

(1596,) (1372,) (224,)
```

```
In [22]: model=RandomForestClassifier()
model.fit(X_train, Y_train)
RandomForestClassifier()
X_test_prediction = model.predict(X_test)
test_data_accuracy=accuracy_score(X_test_prediction, Y_test)
print('Accuracy:', test_data_accuracy)
Accuracy: 0.8794642857142857
input=(11.2,0.28,0.56,1.9,0.075,17.0,60.0,0.9980,3.16,0.58,9.8)
input_data=np.asarray(input)
reshaped=input_data.reshape(1,-1)
prediction=model.predict(reshaped)
print(prediction)
if prediction==1:
    print('Good')
else:
    print('Bad')

Accuracy: 0.8883928571428571
[0]
Bad

C:\Users\rames_ykh6ekx\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
warnings.warn(
```

```
In [ ]:
```