In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
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	sulpha
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000

In [6]:

```
wine_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

In [7]:

```
wine_dataset.isnull().sum()
# check missing values
```

Out[7]:

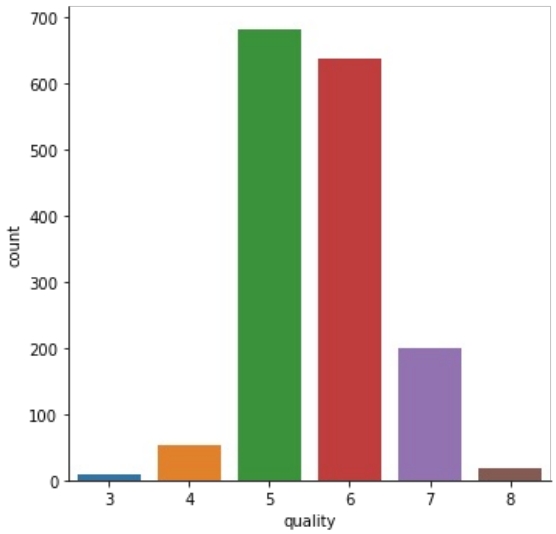
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality   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

In [8]:

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

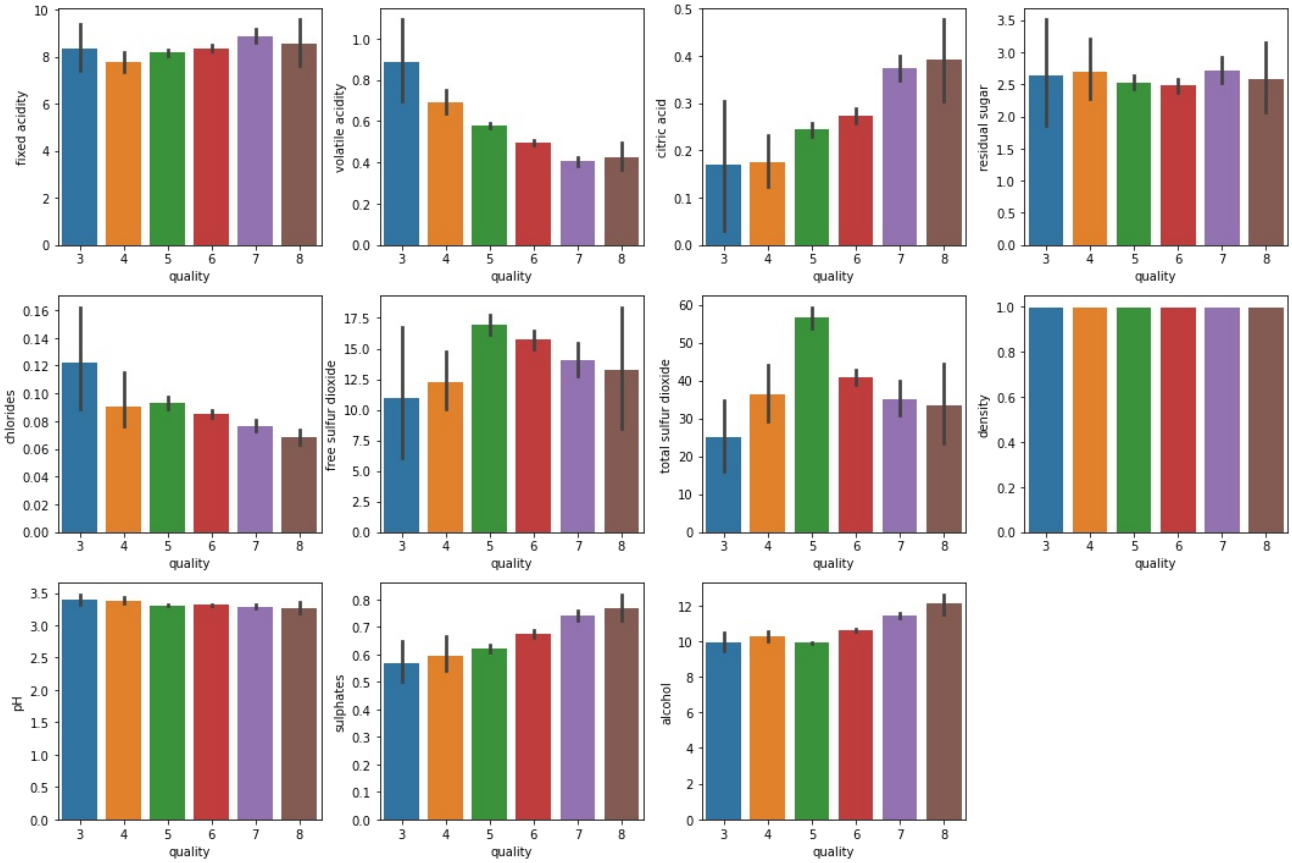
Out[8]:

```
<seaborn.axisgrid.FacetGrid at 0x1645d5ac8b0>
```



In [9]:

```
fig = plt.figure(figsize=(15,10))

plt.subplot(3,4,1)
sns.barplot(x='quality',y='fixed acidity',data=wine_dataset)

plt.subplot(3,4,2)
sns.barplot(x='quality',y='volatile acidity',data=wine_dataset)

plt.subplot(3,4,3)
sns.barplot(x='quality',y='citric acid',data=wine_dataset)

plt.subplot(3,4,4)
sns.barplot(x='quality',y='residual sugar',data=wine_dataset)

plt.subplot(3,4,5)
sns.barplot(x='quality',y='chlorides',data=wine_dataset)

plt.subplot(3,4,6)
sns.barplot(x='quality',y='free sulfur dioxide',data=wine_dataset)

plt.subplot(3,4,7)
sns.barplot(x='quality',y='total sulfur dioxide',data=wine_dataset)

plt.subplot(3,4,8)
sns.barplot(x='quality',y='density',data=wine_dataset)

plt.subplot(3,4,9)
sns.barplot(x='quality',y='pH',data=wine_dataset)

plt.subplot(3,4,10)
sns.barplot(x='quality',y='sulphates',data=wine_dataset)

plt.subplot(3,4,11)
sns.barplot(x='quality',y='alcohol',data=wine_dataset)

plt.tight_layout()
```



In [10]:

```
#correlation
```

In [11]:

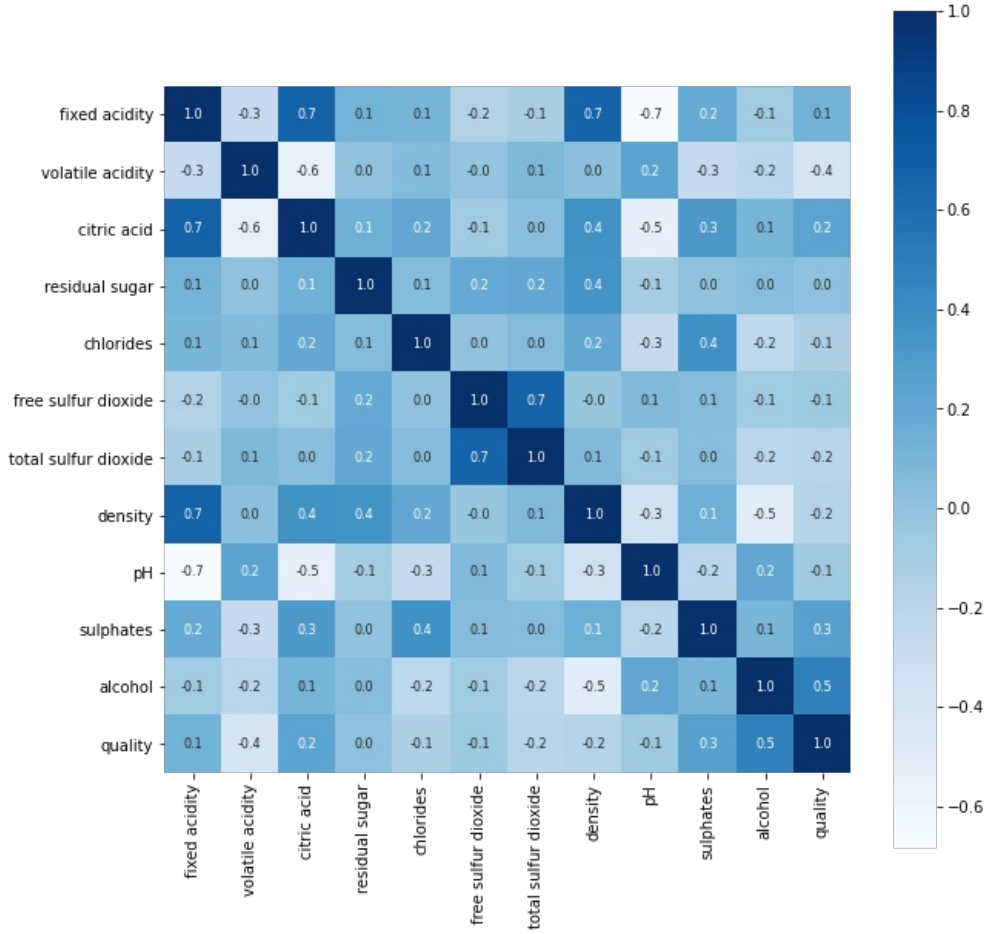
```
correlation = wine_dataset.corr()
```

In [12]:

```
# 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':8}, cmap = 'Blues')
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1645e4830d0>
```



In [14]:

```
# Data Pre processing
```

In [15]:

```
# seperate the data and label
```

In [16]:

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

In [17]:

```
X
```

Out[17]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0

1599 rows × 11 columns

In [18]:

```
# label Binarization
```

In [19]:

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

In [20]:

```
Y
```

Out[20]:

```
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 [21]:

```
# spiliting into train and test data
```

In [22]:

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

In [23]:

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

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

In [24]:

```
# model training using random forest classifier
```

In [25]:

```
model = RandomForestClassifier()
```

In [26]:

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

Out[26]:

```
RandomForestClassifier()
```

In [27]:

```
# training the model using fit function
```

In [28]:

```
# model Evaluation
```

In [29]:

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

In [30]:

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

```
Accuracy :  0.928125
```

In [31]:

```
# for prediction our model can predict 93 correct values out of 100
```

In [32]:

```
# building a prediction system
```

In [33]:

```
input_data = (7.3,0.65,0.0,1.2,0.065,15.0,21.0,0.9946,3.39,0.47,10.0)
```

In [34]:

```
# changing the input data to a numpy array }
```

In [35]:

```
input_data_as_numpy_array = np.asarray(input_data)

# reshape the data as we are predicting the label for only one instance
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
```