```
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
```

```
#loading dataframe to a pandas data frame
wine_dataset = pd.read_csv("./winequality-red.csv")
```

```
wine_dataset.shape
```

```
(1599, 12)
```

```
wine_dataset.head()
```

	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

```
wine_dataset.isnull().sum()
```

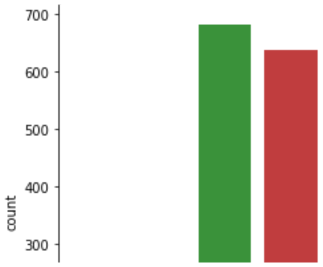
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality   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

```
wine_dataset.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	dens
count	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	0.996
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003

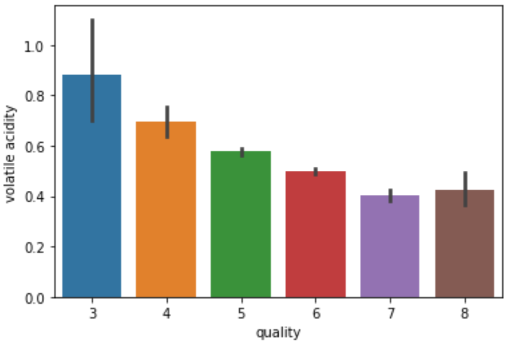
```
#plotting quality and count of each quality
sns.catplot(x="quality",data=wine_dataset,kind="count")
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd2cf35c340>
```



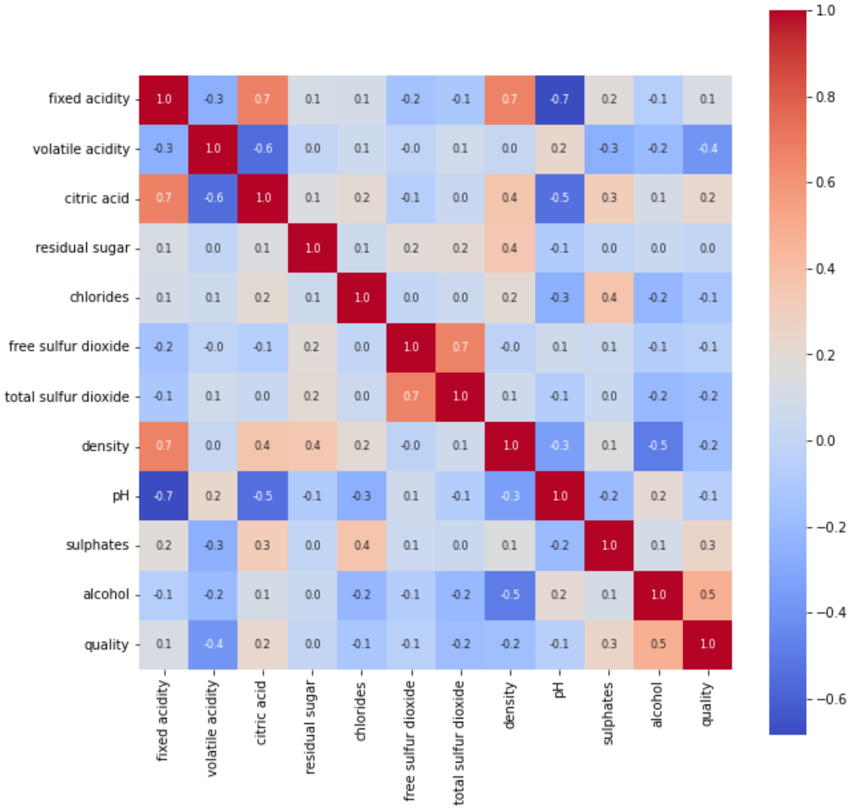
```
#finding which column has co-relation with the quality column
sns.barplot(x="quality",y="volatile acidity",data=wine_dataset)
```

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



```
corelation = wine_dataset.corr()
plot = plt.figure(figsize=(10,10))
sns.heatmap(corelation,cbar=True,square=True,cmap='coolwarm',annot=True,annot_kws={'size':8},fmt='.1f')
```

```
<AxesSubplot:>
```



	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

```
#Data Preprocessing
#seperating the quality and other columns
X = wine_dataset.drop('quality',axis=1)
```

```
X.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	s
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	

```
#binaryizing the values of quality data set
Y = wine_dataset['quality'].apply(lambda value:1 if value>=7 else 0)
Y.head()
```

```
0    0
1    0
2    0
3    0
4    0
Name: quality, dtype: int64
```

```
#Training and Testing data
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=3)


#Model training
#RandomForestClassifierModel
model = RandomForestClassifier()


#training the model
model.fit(X_train,Y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
#model evaluation
X_test_prediction = model.predict(X_test)
test_data_score = accuracy_score(X_test_prediction,Y_test)


print("Test Data Accuracy ",test_data_score)
```

```
Test Data Accuracy  0.9375
```

```
#predicting system
input_data = (7.3,0.65,0.0,1.2,0.065,15.0,21.0,0.9946,3.39,0.47,10.0)
#changing into numpy array
input_data_numpy_array = np.asarray(input_data)
#input reshape
reshaped_data = input_data_numpy_array.reshape(1,-1)

prediction = model.predict(reshaped_data)
if(prediction[0]==1):
  print("Good Quality Wine")
else:
  print("Bad Quality")
```

```
Good Quality Wine
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:420: UserWarning: X does not have valid feature names, but RandomForestClass
  warnings.warn(
```

✓ 0s completed at 21:26 ● ×