```
import pandas as pd
df=pd.read_csv("/content/winequality-red.csv")
df
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.
...	...	...	...	...	...	...	...	...	
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.

```
import warnings
warnings.filterwarnings("ignore")
```

```
df.dtypes
```

```
    fixed acidity           float64
    volatile acidity        float64
    citric acid             float64
    residual sugar          float64
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
    density                 float64
    pH                      float64
    sulphates               float64
    alcohol                 float64
    quality                   int64
    dtype: object
```

```
df.isna().sum()
```

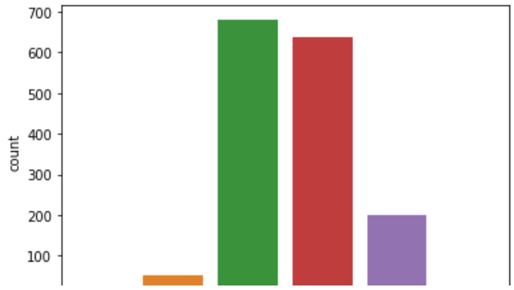
```
    fixed acidity           0
    volatile acidity        0
    citric acid             0
    residual sugar          0
    chlorides               0
    free sulfur dioxide     0
    total sulfur dioxide    0
    density                 0
    pH                      0
    sulphates             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

```
df["quality"].value_counts()
```

```
    5    681
    6    638
    7    199
    4     53
    8     18
    3     10
    Name: quality, dtype: int64
```

```
import seaborn as sns
sns.countplot(df["quality"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f84b8eeb430>
```



```
X=df.iloc[:,:-1].values
Y=df.iloc[:,-1].values
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_new=sc.fit_transform(X)
X_new
```

```
array([[-0.52835961,  0.96187667, -1.39147228, ...,  1.28864292,
        -0.57920652, -0.96024611],
       [-0.29854743,  1.96744245, -1.39147228, ..., -0.7199333 ,
         0.1289504 , -0.58477711],
       [-0.29854743,  1.29706527, -1.18607043, ..., -0.33117661,
        -0.04808883, -0.58477711],
       ...,
       [-1.1603431 , -0.09955388, -0.72391627, ...,  0.70550789,
         0.54204194,  0.54162988],
       [-1.39015528,  0.65462046, -0.77526673, ...,  1.6773996 ,
         0.30598963, -0.20930812],
       [-1.33270223, -1.21684919,  1.02199944, ...,  0.51112954,
         0.01092425,  0.54162988]])
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=1)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=100)
rf.fit(X_train,Y_train)
Y_pred=rf.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print("Accuracy Score",accuracy_score(Y_test,Y_pred))
```

```
Accuracy Score 0.6729166666666667
```

```
from sklearn.metrics import classification_report
print("Classification Report",classification_report(Y_test,Y_pred))
```

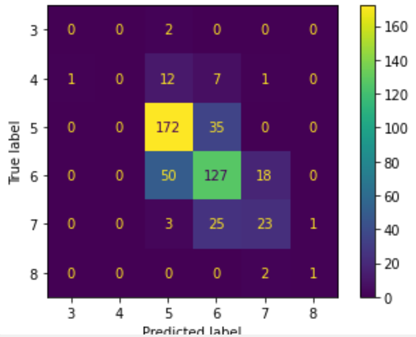
```
Classification Report               precision    recall  f1-score   support

           3       0.00      0.00      0.00         2
           4       0.00      0.00      0.00        21
           5       0.72      0.83      0.77       207
           6       0.65      0.65      0.65       195
           7       0.52      0.44      0.48        52
           8       0.50      0.33      0.40         3

    accuracy                           0.67       480
   macro avg       0.40      0.38      0.38       480
weighted avg       0.64      0.67      0.65       480
```

```
from sklearn.metrics import ConfusionMatrixDisplay
print("Confusion Matrix",ConfusionMatrixDisplay.from_predictions(Y_test,Y_pred))
```

```
Confusion Matrix <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay obje
```



Colab paid products - Cancel contracts here

✓ 0s completed at 10:55 AM

Automatic saving failed. This file was updated remotely or in another tab. Show diff