In [56]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```
wine_data=pd.read_csv('winequalityN.csv')
```

In [36]:

```
wine_data.shape
wine_data=wine_data.dropna()
wine_data['type']=wine_data['type'].replace({'white':0,'red':1})
```

In [50]:

```
wine_data
```

Out[50]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulph
0	0	7.0	0.270	0.36	20.7	0.045	45.0	170.0	1.00100	3.00	
1	0	6.3	0.300	0.34	1.6	0.049	14.0	132.0	0.99400	3.30	
2	0	8.1	0.280	0.40	6.9	0.050	30.0	97.0	0.99510	3.26	
3	0	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	
4	0	7.2	0.230	0.32	8.5	0.058	47.0	186.0	0.99560	3.19	
...	...	...	...	...	...	...	...	...	...	...	
6491	1	6.8	0.620	0.08	1.9	0.068	28.0	38.0	0.99651	3.42	
6492	1	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	
6494	1	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	
6495	1	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	
6496	1	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	

6463 rows × 13 columns

In [51]:

```
x=wine_data.drop('type',axis=1)
y=wine_data['type']
```

In [54]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20)
```

In [55]:

```
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier()
classifier.fit(x_train,y_train)
```

Out[55]:

```
DecisionTreeClassifier()
```

In [58]:

```
y_pred=classifier.predict(x_test)
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[964  10]
 [  7 312]]
```

In [59]:

```
wine_data=pd.DataFrame({'Actual':y_test,'Predict':y_pred})
wine_data
```

Out[59]:

	Actual	Predict
3875	0	0
5676	1	1
6136	1	1
1952	0	0
3807	0	0
...	...	...
1537	0	0
4777	0	0
442	0	0
3699	0	0
3936	0	0

1293 rows × 2 columns

In [66]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       974
           1       0.97      0.98      0.97       319

    accuracy                           0.99      1293
   macro avg       0.98      0.98      0.98      1293
weighted avg       0.99      0.99      0.99      1293
```

In [68]:

```
from sklearn.metrics import accuracy_score
```

In [69]:

```
accuracy=accuracy_score(y_test,y_pred)
print('Accuracy:',accuracy)
```

```
Accuracy: 0.9868522815158546
```

In [ ]: