Practical 6
2CS501
19BCE248

AIM: Implement decision tree with different variants such as ID3, C4.5 using sklearn

About Dataset:

The dataset was taken from an online source which was further categorized into two parts namely white and red wine. The dataset was predicting the quality of wine and whether it is safe to drink or not.

There were columns mentioned below:

```
['fixed acidity',
 'volatile acidity',
 'citric acid',
 'residual sugar',
 'chlorides',
 'free sulfur dioxide',
 'total sulfur dioxide',
 'density',
 'pH',
 'sulphates',
 'alcohol',
 'quality']
```

The Last column is the target variable.

Preprocessing:

- Firstly merging two dataset
- Formatting it to proper csv format
- Extracting X and Y from dataset
- Dividing it into training and testing.
- Now as the quality values were expected to be categorical whether good or bad so all outcome <=6 was considered as bad while others remained as good.

Applying Decision Tree with different variants:

```
acc=accuracy_score(y_test,y_pred)
acc
```

```
0.75
```

Maximum accuracy came out to be much lesser.

Increasing Accuracy:

1. By GridSearchCV

```
: acc=accuracy_score(y_test,y_pred)
  acc
```

```
: 0.7730769230769231
```

Still came out to be much less.

2. By CrossValidation:

```
acc=[]
for i,j in sf.split(X,y):
#     print(X.iloc[i,:])
    X_train,X_test,y_train,y_test=X.iloc[i,:],X.iloc[j,:],y.iloc[i,:],y.iloc[j,:]
#     print(len(X_train),len(X_test))
    y_pred=gsv.predict(X_test)
    acc1=accuracy_score(y_test,y_pred)
    acc.append(acc1)
acc
```

```
[0.9506001846722069, 0.9487534626038782, 0.9639722863741339]
```

Much better result as compared to above both.

Conclusion:

From the above practices we can infer that Decision tree can be used in circumstances like:

- when we want a simple model
- when we have limited computational power
- Dataset to be classified is in a continuous manner or categorical both accepted.
- Less data cleaning required