

In []:

```

1  '''
2
3  Decision Tree is a Supervised learning technique that can be used for both classification and regression problems,
4  but mostly it is preferred for solving Classification problems. It is a tree-structured model where internal nodes
5  represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
6
7
8  Attribute Selection Measures
9  While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for
10 internal nodes. So, to solve such problems there is a technique which is called as Attribute selection measures.
11 By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular
12 techniques for ASM, which are:
13
14
15 1.Information Gain
16 2.Gini Index
17 '''

```

In [23]:

```
1 from sklearn.datasets import load_iris
```

In [24]:

```
1 X,y = load_iris(return_X_y=True)
```

In [25]:

```
1 X # input features
2 y # target output
```

Out[25]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [26]:

```
1 from sklearn.tree import DecisionTreeClassifier
```

In [30]:

```
1 # clf = DecisionTreeClassifier(criterion='gini',max_depth=5)
```

In [82]:

```
1 clf = DecisionTreeClassifier(criterion='entropy',max_depth=5)
```

In [94]:

```
1 from sklearn.model_selection import train_test_split
2 # for splitting training data and testing data from dataset
```

In [95]:

```
1 x_train,x_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=0)
```

In [96]:

```
1 clf.fit(x_train,y_train) # model training
```

Out[96]:

DecisionTreeClassifier(criterion='entropy', max_depth=5)

In [110]:

```
1 y_test # acutual testing output
```

Out[110]:

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 1, 1, 1, 2, 0, 2, 0,
       0])
```

In [114]:

```
1 y_pred = clf.predict(x_test)
2 y_pred # our model predicted output
```

Out[114]:

```
array([2, 1, 0, 2, 0, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 0, 0, 2, 1,
       0, 0, 2, 0, 0, 1, 1, 0, 2, 1, 0, 2, 2, 1, 0, 2, 1, 1, 2, 0, 2, 0,
       0])
```

In [99]:

```
1 # Check model accuracy
2
3 acc = clf.score(x_test,y_test)
4
5 acc = acc*100
6 acc = "{:.2f}".format(acc)
7
8 print('Accuracy : ',acc,'%')
```

Accuracy : 97.78 %

In [100]:

```
1 # confusuion matrix
2
3 from sklearn.metrics import confusion_matrix
4
5 cm = confusion_matrix(y_test,y_pred)
6 print('Confusion matrix \n',cm)
```

Confusion matrix

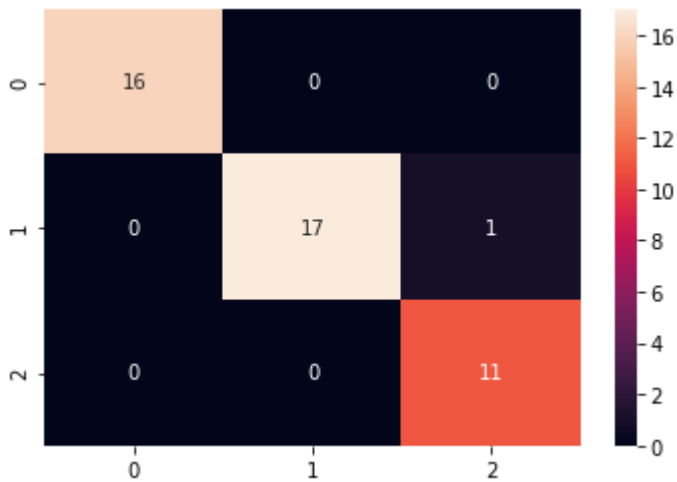
```
[[16  0  0]
 [ 0 17  1]
 [ 0  0 11]]
```

In [101]:

```
1 # seaborn heatmap
2
3 import seaborn as sns
4
5 sns.heatmap(cm,annot=True)
6 sns
```

Out[101]:

<module 'seaborn' from 'C:\\Users\\UmarKhan pathan\\anaconda3\\lib\\site-packages\\seaborn__init__.py'>



In [102]:

```
1 # accuracy score
2
3 from sklearn.metrics import accuracy_score
4
5 score = accuracy_score(y_test,y_pred)
6 score = score*100
7
8 score = "{:.2f}".format(score)
9
10 print('Accuracy score : ',score,'%')
```

Accuracy score : 97.78 %

In [103]:

```

1 '''
2
3 Classification report
4 to display the model precision , recall , f1-score and support
5
6 '''

```

Out[103]:

```

'\n\nClassification report \n\n
to display the model precision , recall , f1-score and support\n\n'

```

In [104]:

```

1 # importing Classification report from sklearn.metrics
2
3 from sklearn.metrics import classification_report
4
5 cr = classification_report(y_test,y_pred)
6 print('Classification report: \n',cr)

```

Classification report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 1.00 | 1.00 | 1.00 | 16 |
| 1 | 1.00 | 0.94 | 0.97 | 18 |
| 2 | 0.92 | 1.00 | 0.96 | 11 |
| accuracy | | | 0.98 | 45 |
| macro avg | 0.97 | 0.98 | 0.98 | 45 |
| weighted avg | 0.98 | 0.98 | 0.98 | 45 |

In []:

1

In []:

1