**Decision tree**

Theoretical concept:

1. Is a supervised learning algorithm.

2. It models probable outcomes, cost of resources, utilities, and possible consequences.

3. For decision tree python uses sklearn package to use some of its model like train\_test\_split, DecisionTreeClasifier and accuracy\_score.

4.from sklearn.tree import DecisionTreeClasifier

Types of decision tree:

There are mainly two kind of decision tree and those are-

#### 1. Categorical variable decision tree:- A categorical variable decision tree target a variable that are divided into two categories and those are Yes or No.

#### 2. Continuous variable decision tree:- It is depends on more than two parameters. Example: As like weather forecasting.

**Working:**

1. Consider the whole training set as the root.

2. Then split the training set into subset.(**Note: it subsets should have the same value for an attribute**)

**Process:**

1. Building face:

* Process the datasets
* Split the dataset from the train and test using python sklearn packege.
* Train the classifier.

2. Operational face:

* Make prediction
* Calculate the accuracy
* Before training the model we have to split the data by using(from sklearn.cross\_validation import train\_test\_split)
* Separate the target variable

x\_train, x\_test, y\_train, y\_test = train\_test\_split (x,y,test\_size = 0.3, random state = 75)

test\_size = percentage of train and test data.

Random\_state = use for random sampling

We can find,

1. Gini index:

* is a metric
* we prefer lower Gini index
* how often it measure how often randomly chosen element would be identified incorrectly.
* **Def train\_using\_gini(x\_train, x\_test,y\_train):**

**s = DecisionTreeClassifier(criterion = ‘gini’,random\_state = 75, max\_date = 3,min\_samples\_leaf = 2)**

**s.fit(x\_train, y\_train)**

**return s**

2. Entropy:

* Measures the uncertainty of our random variable.
* Categories the impurity of the data
* High entropy-more information

Def train\_using\_entropy(x\_train, x\_test,y\_train):

s = DecisionTreeClassifier(criterion= ‘entropy’,random\_state = 75, max\_date = 3, min\_samples\_leaf = 2)

s.fit(x\_train, y\_train)

return s

3. Information gain:

* It measures the changes in entropy

4. Accuracy score:

* Use to calculate the accuracy of the classifier
* from sklearn.metric import accuracy\_score

5. Confusion matrix:

* used to understand the trained classifier behavior over the test data sets.
* from sklearn.metrics import confusion\_matrix