**Decision Tree using python programming .**

**import matplotlib.pyplot as plt**

**#from sklearn.datasets import load\_iris**

**from sklearn.datasets import load\_breast\_cancer**

**#from sklearn.datasets import fetch\_california\_housing**

**from sklearn.tree import DecisionTreeClassifier**

**from sklearn import metrics**

**#Import scikit.learn metrics module for accuracy calculation**

**#from sklearn.ensemble import RandomForestClassifier**

**from sklearn.model\_selection import train\_test\_split**

**import pandas as pd**

**import numpy as np**

**from sklearn import tree**

**data\_b = load\_breast\_cancer()**

**print(data\_iris)**

**#housing = fetch\_california\_housing()**

**#housing**

**#boston=load\_boston()**

**df=pd.DataFrame(data\_b.data,columns=data\_b.feature\_names)**

**df['target'] = data\_b.target**

**#df['target']**

**print(df)**

**X\_train, X\_test, Y\_train, y\_test = train\_test\_split(df[data\_b.feature\_names], df['target'], random\_state=1)**

**print(X\_train)**

**print(X\_test)**

**# Step 1: Import the model you want to use**

**# This was already imported earlier in the notebook so commenting out**

**#from sklearn.tree import DecisionTreeClassifier**

**# Step 2: Make an instance of the Model**

**#max\_depth = 4**

**clf = DecisionTreeClassifier(max\_depth = 3,random\_state = 0, criterion='gini') #'gini'**

**# Step 3: Train the model on the data**

**clf.fit(X\_train, Y\_train)**

**# Step 4: Predict labels of unseen (test) data**

**# Not doing this step in the tutorial**

**#clf.predict(X\_test)**

**tree.plot\_tree(clf);**

**y\_pred = clf.predict(X\_test)**

**print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))**