**U19EC046 | ML | LAB 5**

**AIM**

Write a program to implement the Decision Tree classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets. Also generate the random forest classifier.

**ALGORITHM**

1. Import necessary libraries
2. Read the dataset using pandas
3. Split the dataset into features and results
4. Split the features and results into training and testing dataset using scikit learn

test\_train\_split

1. Apply pre-processing if needed, use scikit-learn preprocessing class
2. Create an instance of Decision Tree classifier using gini or entropy
3. Train the model using fit method
4. Predict the result using predict method on model
5. Find the accuracy and confusion matrix using scikit-learn matrices

**CODE**

1. Decision Tree Classification

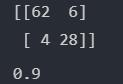
|  |
| --- |
| **import numpy as np**  **import matplotlib.pyplot as plt**  **import pandas as pd**  **dataset = pd.read\_csv('Social\_Network\_Ads.csv')**  **X = dataset.iloc[:, :-1].values**  **y = dataset.iloc[:, -1].values**  **from sklearn.model\_selection import train\_test\_split**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)**  **from sklearn.preprocessing import StandardScaler**  **sc = StandardScaler()**  **X\_train = sc.fit\_transform(X\_train)**  **X\_test = sc.transform(X\_test)**  **from sklearn.tree import DecisionTreeClassifier**  **classifier = DecisionTreeClassifier(criterion = 'gini', random\_state = 0)    *# or use criterion = 'gini'***  **classifier.fit(X\_train, y\_train)**  **y\_pred = classifier.predict(X\_test)**  **y\_pred = classifier.predict(X\_test)**  **print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))**  **from sklearn.metrics import confusion\_matrix, accuracy\_score**  **cm = confusion\_matrix(y\_test, y\_pred)**  **print(cm)**  **accuracy\_score(y\_test, y\_pred)** |

1. Random forest Classifier

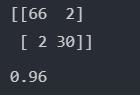
|  |
| --- |
| **import numpy as np**  **import matplotlib.pyplot as plt**  **import pandas as pd**  **dataset = pd.read\_csv('Social\_Network\_Ads.csv')**  **X = dataset.iloc[:, :-1].values**  **y = dataset.iloc[:, -1].values**  **from sklearn.model\_selection import train\_test\_split**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)**  **from sklearn.preprocessing import StandardScaler**  **sc = StandardScaler()**  **X\_train = sc.fit\_transform(X\_train)**  **X\_test = sc.transform(X\_test)**  **from sklearn.ensemble import RandomForestClassifier**  **classifier = RandomForestClassifier(n\_estimators = 10, criterion = 'entropy', random\_state = 2837)**  **classifier.fit(X\_train, y\_train)**  **y\_pred = classifier.predict(X\_test)**  **print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))**  **from sklearn.metrics import confusion\_matrix, accuracy\_score**  **cm = confusion\_matrix(y\_test, y\_pred)**  **print(cm)**  **accuracy\_score(y\_test, y\_pred)** |

**OUTPUT**

1. Decision tree



1. Random Forest Classifier



**CONCLUSION**

In this practical we have studied and implements decision tree classifier using scikit-learn library. We also learn about different attribute selection measures and usually gini index is used as it considers both positive and negative sample.