**Random Forest Algorithm**

The following are the basic steps involved in performing the random forest algorithm:

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

import pandas as pd

import numpy as np

dataset = pd.read\_csv(‘C:/Users/Admin/Desktop/petrol\_consumption.csv')

dataset.head()

X = dataset.iloc[:, 0:4].values

y = dataset.iloc[:, 4].values

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, 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 RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators=20, random\_state=0) regressor.fit(X\_train, y\_train)

y\_pred = regressor.predict(X\_test)

from sklearn import metrics

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

### Random Forest for Classification

import pandas as pd

dataset = pd.read\_csv(‘C:/Users/Admin/Desktop/bill\_authentication.csv’)

dataset.head()

X = dataset.iloc[:, 0:4].values

y = dataset.iloc[:, 4].values

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

from sklearn.ensemble import RandomForestClassifier

classifier= RandomForestClassifier(n\_estimators=20, random\_state=0)   
classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

print(confusion\_matrix(y\_test,y\_pred))

print(classification\_report(y\_test,y\_pred))

print(accuracy\_score(y\_test, y\_pred))