**Practical – 4: Application of Decision Tree, SVM and Random Forest**

**3) Random Forest**

**Description:** The problem here is to predict the gas consumption (in millions of gallons) in 48 of the US states based on petrol tax (in cents), per capita income (dollars), paved highways (in miles) and the proportion of population with the driving license. To solve this regression problem we will use the random forest algorithm via the Scikit-Learn Python library.

**Structure of Dataset:**



**Code:**

**# import libraries and dataset**

import pandas as pd

import numpy as np

dataset = pd.read\_csv('petrol\_consumption.csv')

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

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

**# split data**

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)

# it looks like the BareNuc column includes some values that are not numerical.

# We can drop those rows:

cell\_df = cell\_df[pd.to\_numeric(cell\_df['BareNuc'], errors='coerce').notnull()]

cell\_df['BareNuc'] = cell\_df['BareNuc'].astype('int')

cell\_df.dtypes

feature\_df = cell\_df[['Clump', 'UnifSize', 'UnifShape', 'MargAdh', 'SingEpiSize', 'BareNuc', 'BlandChrom', 'NormNucl', 'Mit']]

X = np.asarray(feature\_df)

X[0:5]

**# Feature Scaling**

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

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

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

**# Training the Algorithm**

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)

**# Evaluating the Algorithm**

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)))

**Output:**

Mean Absolute Error: 51.76500000000001

Mean Squared Error: 4216.166749999999

Root Mean Squared Error: 64.93201637097064