**CSE-3024 Web Mining**

**Random Forest**

**Alokam Nikhitha**

**19BCE2555**

**Question:**

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.

**Dataset Used:**

petrol\_consumption.csv, bill\_authentication.csv.

**Procedure:**

-Using pandas, we first import the dataset into our workspace.

-Next we define the set of dependent and independent attributes.

- We then import the random forest regressor from sklean rn.ensemble and train our model using the independent and dependent attributes.

- Next, we have printed the results of independent set as predicted by our regressor.

- Lastly, To check for the performance of our dataset, we have printed all the evaluation metrics

Since it has less Number of Rows we haven’t split the dataset

**Petrol\_consumption dataset**

**Code**

#Importing Libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

#Importing the Dataset

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

#First few rows of our dataset

dataset.head(10)

#Checcking for null values

print(dataset.info())

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

y = dataset.iloc[:, -1].values

#Training our Random Forest Regression Model

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators=200, random\_state=0)

regressor.fit(X, y)

#Predictions by Regressor

y\_pred = regressor.predict(X)

#Printing Mean Absolute Error

from sklearn.metrics import mean\_absolute\_error

mean\_absolute\_error(y, y\_pred)

#Printing Mean Absolute Error

from sklearn.metrics import mean\_squared\_error

mean\_squared\_error(y, y\_pred)

#Printing Root Mean Squared Error

np.sqrt(mean\_squared\_error(y, y\_pred))

#Printing Root Mean Sqaured Log Error

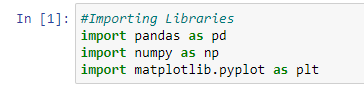
np.log(np.sqrt(mean\_squared\_error(y, y\_pred)))

#Printing R-square value

from sklearn.metrics import r2\_score

r2\_score(y, y\_pred)

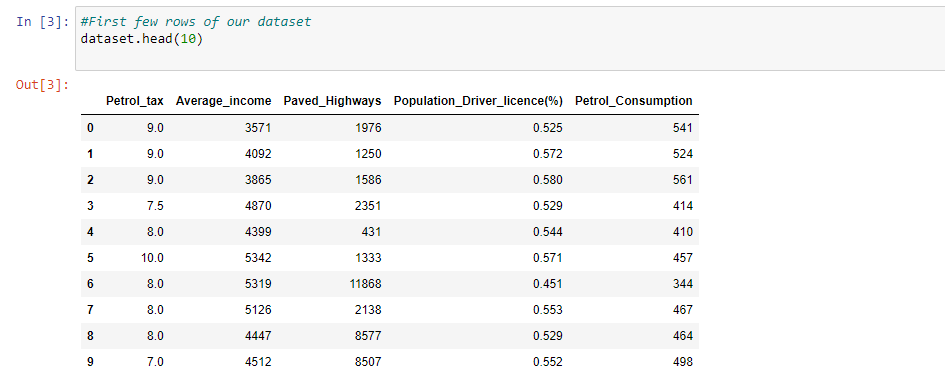
**Code Snippets and Explanation:**

****

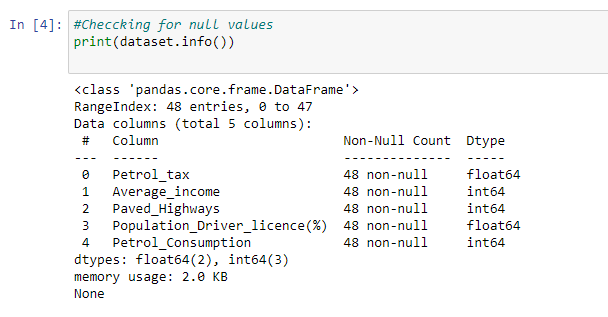
**Here we are importing the required Libraries**

****

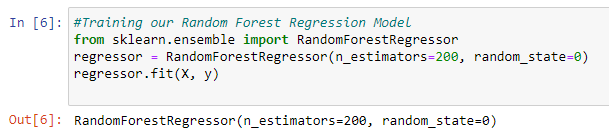
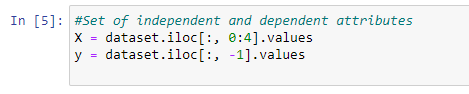
**Using Pandas we are importing the data**

****

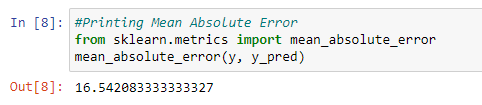
**Printing the first few rows.**

****

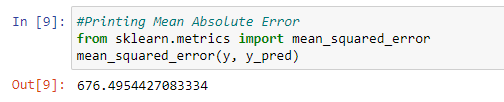
**Here we are checking for the null values.**

****

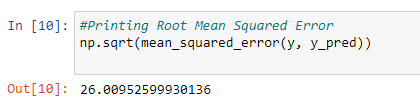
We have Defined set of Dependent and Independent attributes.The n\_estimators here indicate the number of decision trees that we are using to train our random forest regressor. Hence we are using 200 decision trees for prediction. For final value we have used the average value of each decision tree to find the final consumption of petrol of a particular region.

****

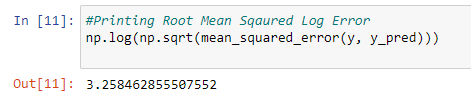
**Printing the Mean Absolute Error**

****

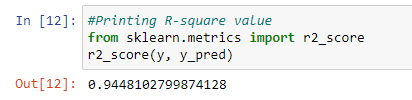
**Printing the Mean Squared Error**

****

**Printing the Root Mean Squared Error**

****

**Printing the Root Mean Sqaured Log Error**

****

**Printing the R-square value**

**Results and Conclusions:**

**Mean Absolute Error from cell8 is** 16.542083333333327

**Mean absolute error from cell 9 is** 676.4954427083334

**Root Mean Squared Error from cell10 is** 26.00952599930136

**Root Mean Squared Log Error from cell11 is** 3.258462855507552

**R-square value from cell12 is** 0.9448102799874128

**Bill\_authentication dataset**

**Code**

#Importing Libraries

import pandas as pd

#importing the bill\_authentication dataset

dataset = pd.read\_csv('bill\_authentication.csv')

#Displaying the first few rows of the dataset

dataset.head()

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

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

#Training our Random Forest Regression Model

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

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))

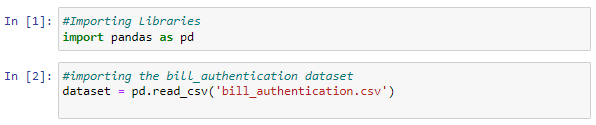
#printing classification\_report

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

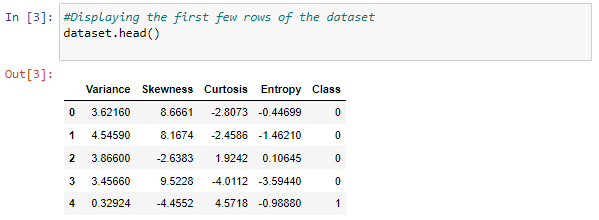
#printing Accuracy

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

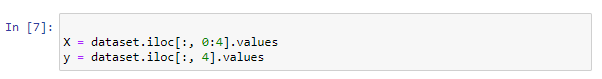
**Code Snippets and Explaination**

****

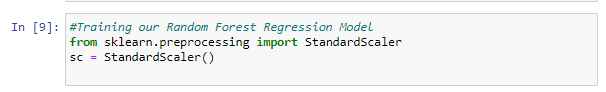
**Here we are importing the required Libraries. Using Pandas we are importing the data**

****

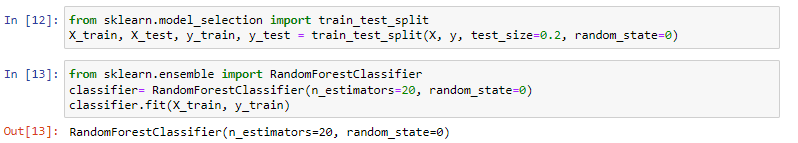
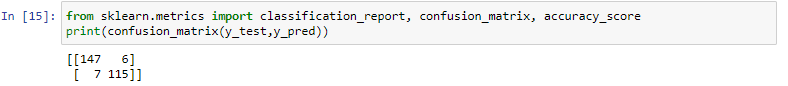
**Printing the first few rows.**

****

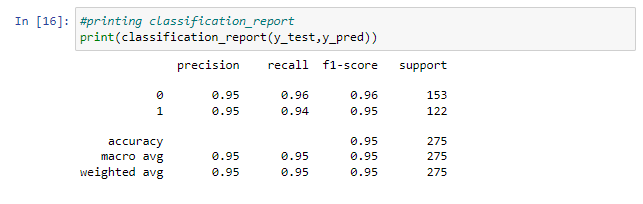
**Defining the Dependent and Independent variables**

****

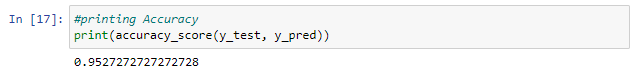
Here we are training our Random forest Regression model

****  

Here we are printing the Confusion Matrix



Here we are printing the Classification Report

****

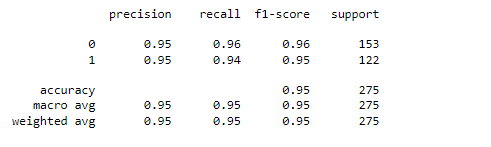
**The Accuracy of the model is** 0.9527272727272728

**Results and Conclusion**

**Confusion Matrix**

****

**Classification Report**

****

**Accuracy of the dataset is:** 0.9527272727272728