CSE 4020 - MACHINE LEARNING

Lab 29+30

Random Forest

Submitted by: Alokam Nikhitha(19BCE2555)

Question:

Use random forest regression as part of ensemble learning to predict the amount of petrol consumption by studying different traits of a particular place.

Dataset Used:

petrol_consumption.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

Code Snippets and Explanation:

```
In [1]: #Importing Libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

Here we are importing the required Libraries

In [2]:
```python
#Importing the Dataset
dataset = pd.read_csv("petrol_consumption.csv")
```

Using Pandas we are importing the data

In [3]:
```python
#First few rows of our dataset
dataset.head(10)
```

Out[3]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
0	9.0	3571	1976	0.525	541
1	9.0	4092	1250	0.572	524
2	9.0	3865	1586	0.580	561
3	7.5	4870	2351	0.529	414
4	8.0	4399	431	0.544	410
5	10.0	5342	1333	0.571	457
6	8.0	5319	11868	0.451	344
7	8.0	5126	2138	0.553	467
8	8.0	4447	8577	0.529	464
9	7.0	4512	8507	0.552	498

Printing the first few rows.

In [4]:
```python
#Checcking for null values
print(dataset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 5 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Petrol_tax                    48 non-null     float64
 1   Average_income                48 non-null     int64
 2   Paved_Highways                48 non-null     int64
 3   Population_Driver_licence(%)  48 non-null     float64
 4   Petrol_Consumption            48 non-null     int64
dtypes: float64(2), int64(3)
memory usage: 2.0 KB
None
```

In [5]:
```
#Set of independent and dependent attributes
X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, -1].values
```

In [6]:
```
#Training our Random Forest Regression Model
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators=200, random_state=0)
regressor.fit(X, y)
```

Out[6]:
```
RandomForestRegressor(n_estimators=200, random_state=0)
```

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.

In [7]:
```
#Predictions by Regressor
y_pred = regressor.predict(X)
```

In [8]:
```
#Printing Mean Absolute Error
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y, y_pred)
```

Out[8]:
```
16.542083333333327
```

Printing the Mean Absolute Error

In [9]:
```
#Printing Mean Absolute Error
from sklearn.metrics import mean_squared_error
mean_squared_error(y, y_pred)
```

Out[9]:
```
676.4954427083334
```

Printing the Mean Squared Error

In [10]:
```
#Printing Root Mean Squared Error
np.sqrt(mean_squared_error(y, y_pred))
```

Out[10]:
```
26.00952599930136
```

Printing the Root Mean Squared Error

```
In [11]: #Printing Root Mean Sqaured Log Error
         np.log(np.sqrt(mean_squared_error(y, y_pred)))

Out[11]: 3.258462855507552
```

Printing the Root Mean Sqaured Log Error

```
In [12]: #Printing R-square value
         from sklearn.metrics import r2_score
         r2_score(y, y_pred)

Out[12]: 0.9448102799874128
```

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 3.25846285550 7552
R-square value from cell12 is 0.9448102799874128