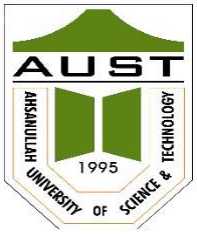
AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY DHAKA-1208, BANGLADESH.



Department of Computer Science and Engineering

Spring 2019

Program: Bachelor of Science in Computer Science and Engineering

Course No: CSE 4108

Course Title: Artificial Intelligence Lab

Term Project No: 03

Topic: Linear Regression & Random Forest Regression

Date of Submission: 15.10.2019

Submitted to

Dr. S.M. Abdullah Al-Mamun
Professor, Department of CSE, AUST.

Md. Siam Ansary
Adjunct Faculty, Department of CSE, AUST.

Submitted By:

Name : Raihan Tanvir

Student ID: 16.01.04.140

Lab Group: C-2

Linear Regression

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

Random Forest Regression

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.[1][2] Random decision forests correct for decision trees' habit of overfitting to their training set.

Implementation of Linear Regression & Random Forest Regression in Python:

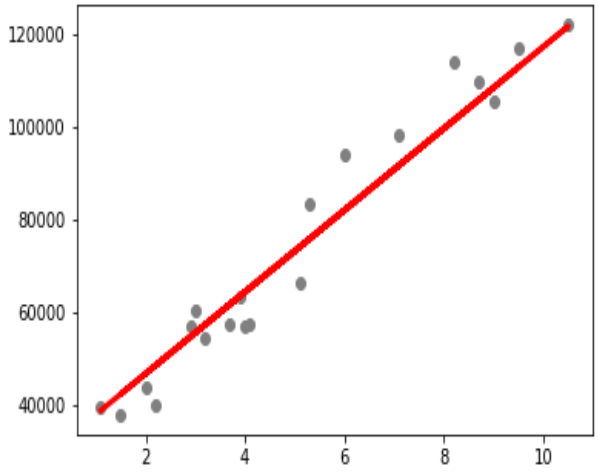
```
import numpy
import matplotlib.pyplot as plt
import pandas
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
import warnings
warnings.simplefilter("ignore")

dataset = pandas.read_csv('salaryData.csv')

# Differentiate attribute and target columns
x = dataset['YearsExperience'].values
y = dataset['Salary'].values

X = x.reshape(len(x),1)
Y = y.reshape(len(y),1)

xTrain, xTest, yTrain, yTest = train_test_split(X, Y, test_size = 2/3)

linearRegressor = LinearRegression()
linearRegressor.fit(xTrain, yTrain)
lrPredict = linearRegressor.predict(xTest)
plt.scatter(xTest, yTest,  color='gray')
```

```
plt.plot(xTest, lrPredict, color='red', linewidth=3)
plt.show()

accuracy = linearRegressor.score(xTest, yTest)
print("Accuracy: {}%".format(int(round(accuracy * 100))))
lrAcc=int(round(accuracy * 100))

kf = KFold(n_splits=5)

print("\nLinear Regression:\n")
for train_index, test_index in kf.split(x):
   x_train, x_test = X[train_index], X[test_index]
   y_train, y_test = Y[train_index], Y[test_index]
   linearRegressor.fit(x_train, y_train)
   prediction = linearRegressor.predict(x_test)
   print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, prediction))
   print('Mean Squared Error:', metrics.mean_squared_error(y_test, prediction))
   print('Root Mean Squared Error:', numpy.sqrt(metrics.mean_squared_error(y_test,
prediction)))
   print('\n')

randForest = RandomForestRegressor(n_estimators=10,random_state=0)
randForest.fit(xTrain, yTrain)
rfPredict = randForest.predict(xTest)

plt.scatter(xTest, yTest,  color='gray')
plt.plot(xTest, rfPredict, color='blue', linewidth=2)
plt.show()

accuracy = randForest.score(xTest, yTest)
print("Accuracy: {}%".format(int(round(accuracy * 100))))
rfAcc=int(round(accuracy * 100))

print("Random Forest Regression:\n")
for train_index, test_index in kf.split(x):
   x_train, x_test = X[train_index], X[test_index]
   y_train, y_test = Y[train_index], Y[test_index]
   randForest.fit(x_train, y_train)
   prediction = randForest.predict(x_test)
   print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, prediction))
   print('Mean Squared Error:', metrics.mean_squared_error(y_test, prediction))
   print('Root Mean Squared Error:', numpy.sqrt(metrics.mean_squared_error(y_test,
prediction)))
   print('\n')
```

```
left = [1, 2]
height = [lrAcc, rfAcc]
tick_label = ['Linear', 'Random-Forest']
plt.bar(left, height, tick_label = tick_label, width = 0.5,color = ['blue', 'red'])
plt.ylabel('Accuracy')
plt.title('Linear vs Random-Forest')
plt.show()
```

Sample Input and Output:

Linear Regression



Random Forest Regression