Nikit Gokhe

Class: TY Comp D1

Roll No. 324022

Gr No. 21810522

Assignment 3

Problem Statement:

Build a Data model in Python for the dataset chosen and apply Linear Regression/Logistic Regression. Infer the result using accuracy score.

Theory:

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things: (1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable? (2) Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable? These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula y = c + b*x, where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable

Objectives:

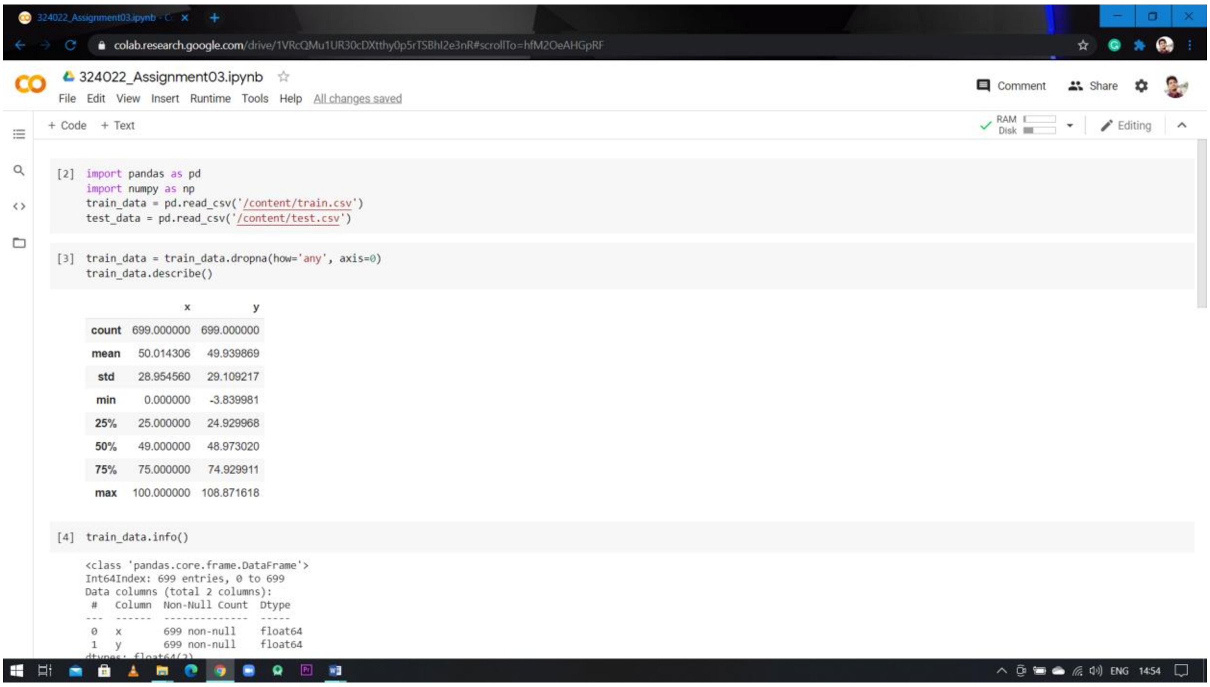
1. To apply linear regression
2. To understand significance of linear regression & Logistic Regression

Dataset:

Name: Train.csv

Test.csv

Expected Output/sample code:


324022_Assignment03.ipynb
colab.research.google.com/drive/1VRcQMu1UR30cDXtthy0p5rTSBhI2e3nR#scrollTo=hfM2OeAHGpRF
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
ENG 14:54

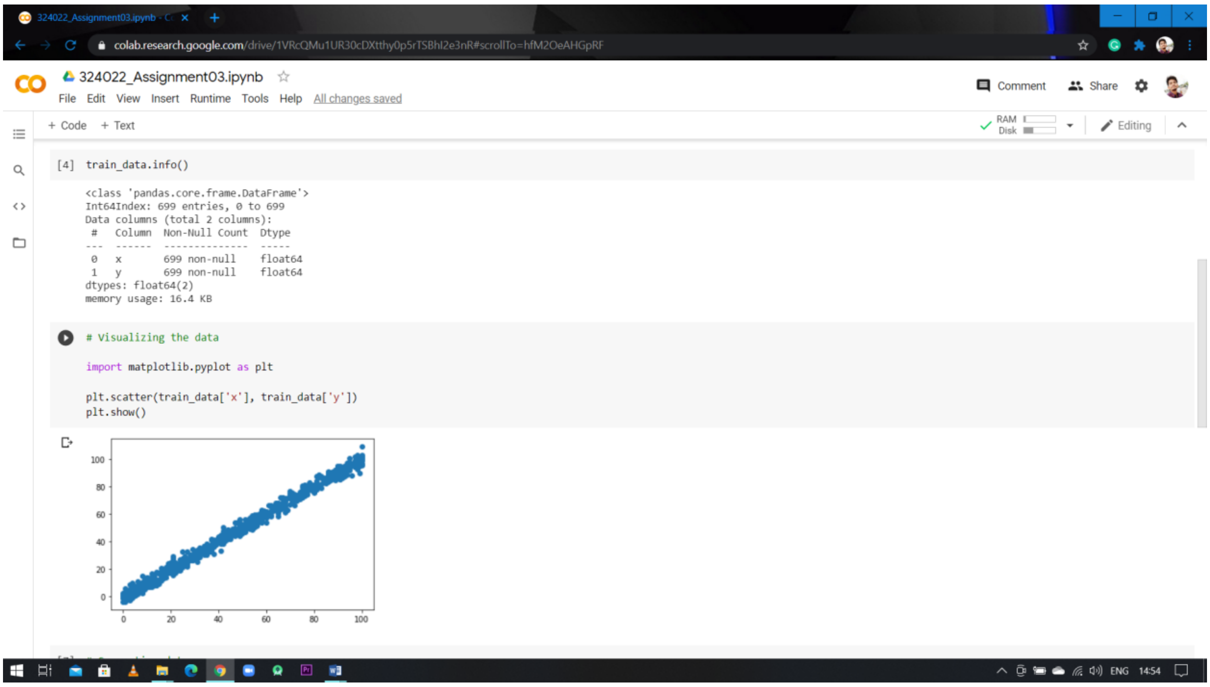
```
[2] import pandas as pd
    import numpy as np
    train_data = pd.read_csv('/content/train.csv')
    test_data = pd.read_csv('/content/test.csv')
```

```
[3] train_data = train_data.dropna(how='any', axis=0)
    train_data.describe()
```

	x	y
count	699.000000	699.000000
mean	50.014306	49.939869
std	28.954560	29.109217
min	0.000000	-3.839981
25%	25.000000	24.929968
50%	49.000000	48.973020
75%	75.000000	74.929911
max	100.000000	108.871618

```
[4] train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 699 entries, 0 to 699
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x       699 non-null    float64
 1   y       699 non-null    float64
dtypes: float64(2)
```


324022_Assignment03.ipynb
colab.research.google.com/drive/1VRcQMu1UR30cDXtthy0p5rTSBhI2e3nR#scrollTo=hfM2OeAHGpRF
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
100
80
60
40
20
0
0
20
40
60
80
100
ENG 14:54

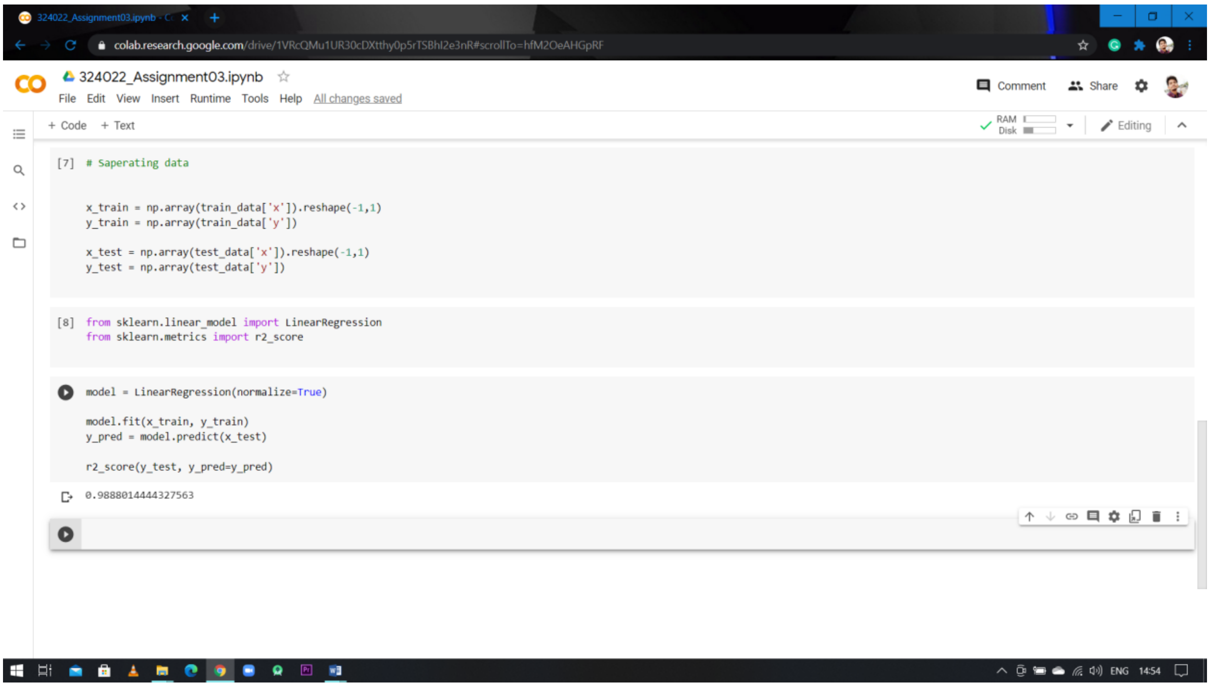
```
[4] train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 699 entries, 0 to 699
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   x       699 non-null    float64
 1   y       699 non-null    float64
dtypes: float64(2)
memory usage: 16.4 KB
```

```
# Visualizing the data

import matplotlib.pyplot as plt

plt.scatter(train_data['x'], train_data['y'])
plt.show()
```



324022_Assignment03.ipynb - Co
colab.research.google.com/drive/1VRcQMu1UR30cDXtthy0p5rTSBhI2e3nR#scrollTo=hfM2OeAHGpRF
324022_Assignment03.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
ENG
14:54

```
[7] # Saperating data


    x_train = np.array(train_data['x']).reshape(-1,1)
    y_train = np.array(train_data['y'])

    x_test = np.array(test_data['x']).reshape(-1,1)
    y_test = np.array(test_data['y'])
```

```
[8] from sklearn.linear_model import LinearRegression
    from sklearn.metrics import r2_score
```

```
model = LinearRegression(normalize=True)

model.fit(x_train, y_train)
y_pred = model.predict(x_test)

r2_score(y_test, y_pred=y_pred)
```

```
0.9888014444327563
```

Conclusion:

Linear regression attempts to model the relationship between two variables by fitting a **linear** equation to observed data.
One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable.
Understood and Applied linear regression.