

Name: Allan Rodrigues
Class: TE IT A
Roll no: 59
Pid:191104

St. Francis Institute of Technology, Mumbai-400 103
Department of Information Technology

A.Y. 2021-2022
Class: TE-ITA/B, Semester: VI

Subject: **Data Science Lab**

Experiment – 5: To implement Regression.

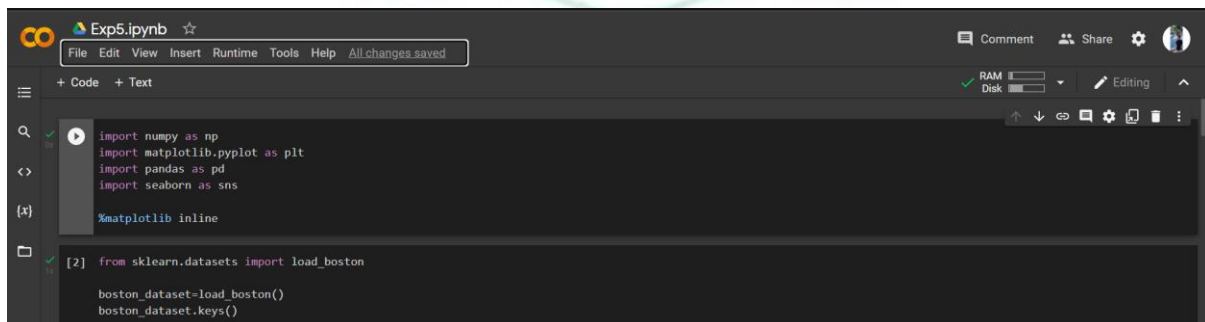
1. **Aim:** To implement Linear and Logistic Regression to find out relation between variables.
2. **Objectives:** After study of this experiment, the student will be able to
 - Understand linear regression
 - Understand logistic regression
3. **Outcomes:** After study of this experiment, the student will be able to
 - Understand concepts of regression
4. **Prerequisite:** Fundamentals of Python Programming and Database Management System.
5. **Requirements:** Python Installation, Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

Brief Theory:

- Concept of regression in machine learning.

Laboratory Exercise

- A. **Procedure:** (the sheet for commands in attached with the file)
- B. Paste Screenshots of above commands.



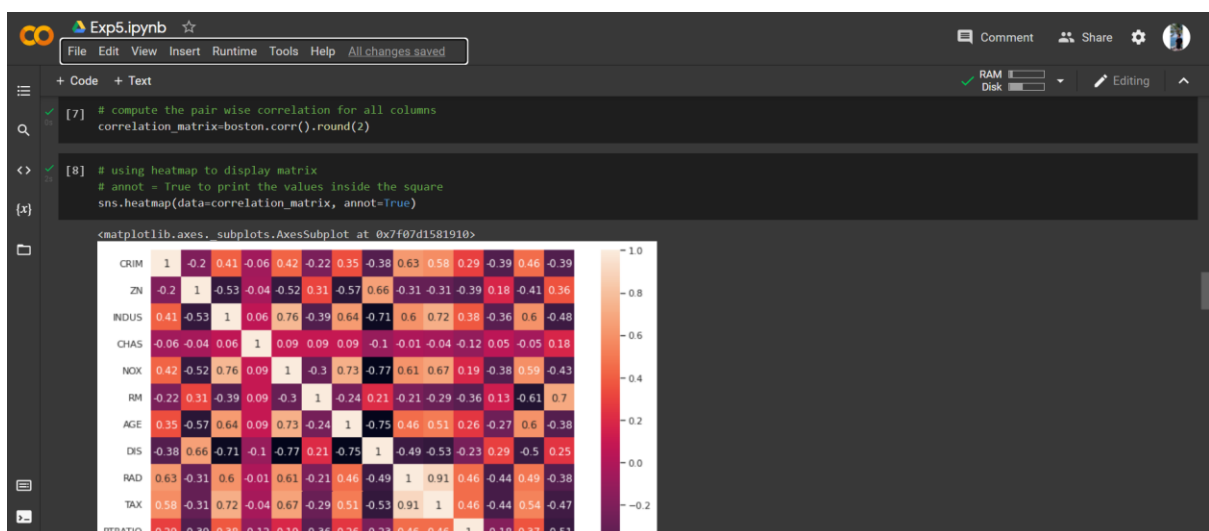
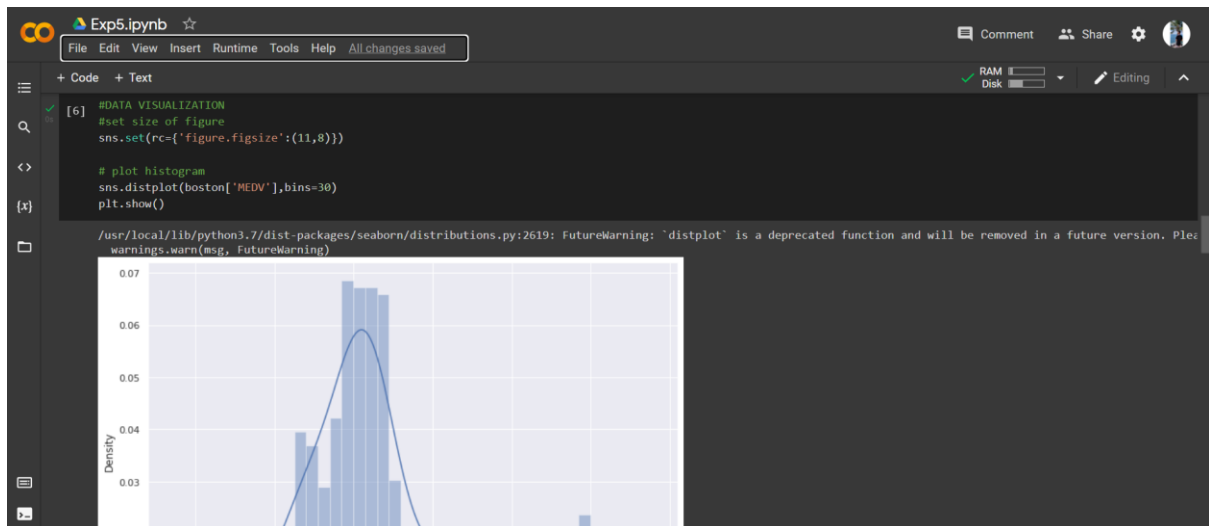
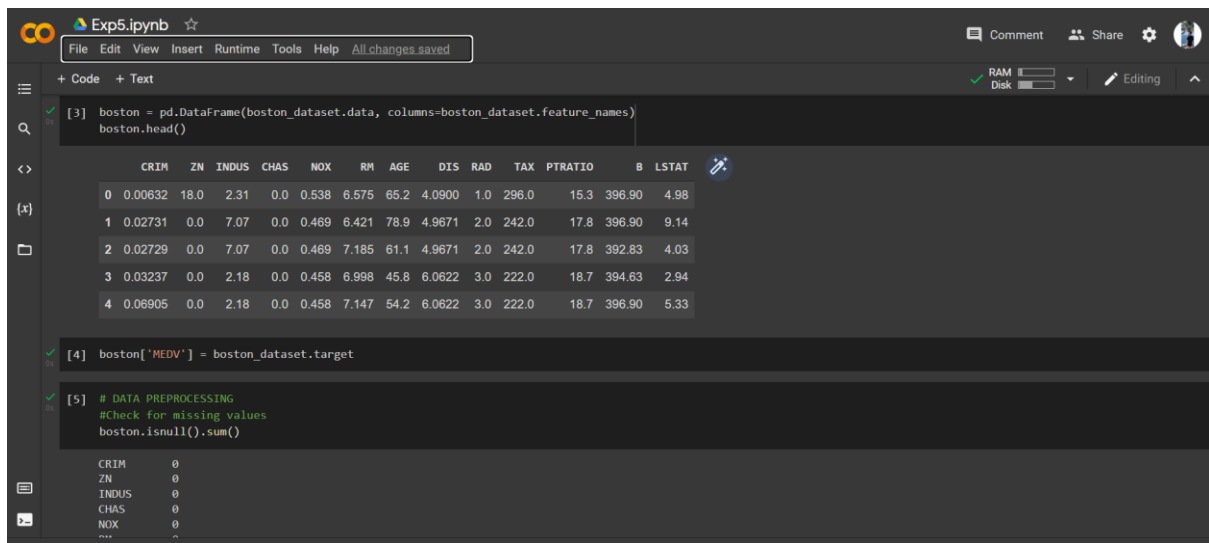
The screenshot shows a Jupyter Notebook titled 'Exp5.ipynb'. The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with icons for running code, saving, and other functions. The code is written in a dark-themed editor. The first cell contains imports for numpy, matplotlib, pandas, and seaborn. The second cell contains code to load the Boston dataset from sklearn.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

%matplotlib inline

[2] from sklearn.datasets import load_boston

boston_dataset=load_boston()
boston_dataset.keys()
```





Exp5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[10] #Prepare data for training
X = pd.DataFrame(np.c_[boston['LSTAT'],boston['RM']],columns=['LSTAT','RM'])
Y= boston['MEDV']

[11] #Split data into training and testing
from sklearn.model_selection import train_test_split

# splits the training and test data set in 80% : 20%
# assign random_state to any value.This ensures consistency.
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=5)
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

(404, 2)
(102, 2)
(404,)
(102,)

Exp5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[12] #Train thye model using sklearn LinearRegression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score

lin_model= LinearRegression()
lin_model.fit(X_train,Y_train)

LinearRegression()

[13] # model evaluation for training set

y_train_predict = lin_model.predict(X_train)
rmse = (np.sqrt(mean_squared_error(Y_train, y_train_predict)))
r2 = r2_score(Y_train, y_train_predict)

print("The model performance for training set")
print("-----")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
print("\n")

# model evaluation for testing set

y_test_predict = lin_model.predict(X_test)
# root mean square error of the model
rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))
```

```
Exp5.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
[13] print("\n")

# model evaluation for testing set

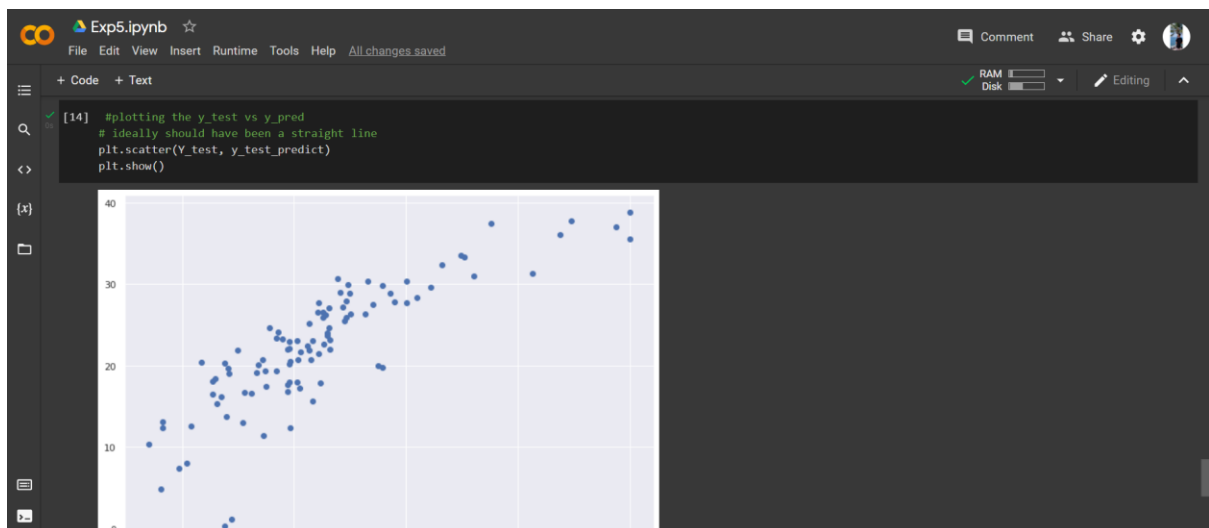
y_test_predict = lin_model.predict(X_test)
# root mean square error of the model
rmse = (np.sqrt(mean_squared_error(Y_test, y_test_predict)))

# r-squared score of the model
r2 = r2_score(Y_test, y_test_predict)

print("The model performance for testing set")
print("-----")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))

The model performance for training set
-----
RMSE is 5.6371293350711955
R2 score is 0.6300745149331701

The model performance for testing set
-----
RMSE is 5.137400784702911
R2 score is 0.6628996975186952
```



8. Post-Experiments Exercise

A. Extended Theory: (Soft Copy)

- Logistic regression


Logistic Regression is a kind of parametric classification model, despite having the word ‘regression’ in its name.

This means that logistic regression models are models that have a certain fixed number of parameters that depend on the number of input features, and they output categorical prediction, like for example if a plant belongs to a certain species or not.

In Logistic Regression, we don’t directly fit a straight line to our data like in linear regression. Instead, we fit a S shaped curve, called Sigmoid, to our observations.

B. Questions:

- Use MNIST Dataset and apply logistic regression.



```
Exp5.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
RAM Disk
Editing

[15] import sklearn
      from sklearn.datasets import load_digits
      from sklearn.model_selection import train_test_split
      from sklearn.linear_model import LogisticRegression

[16] digits = load_digits()

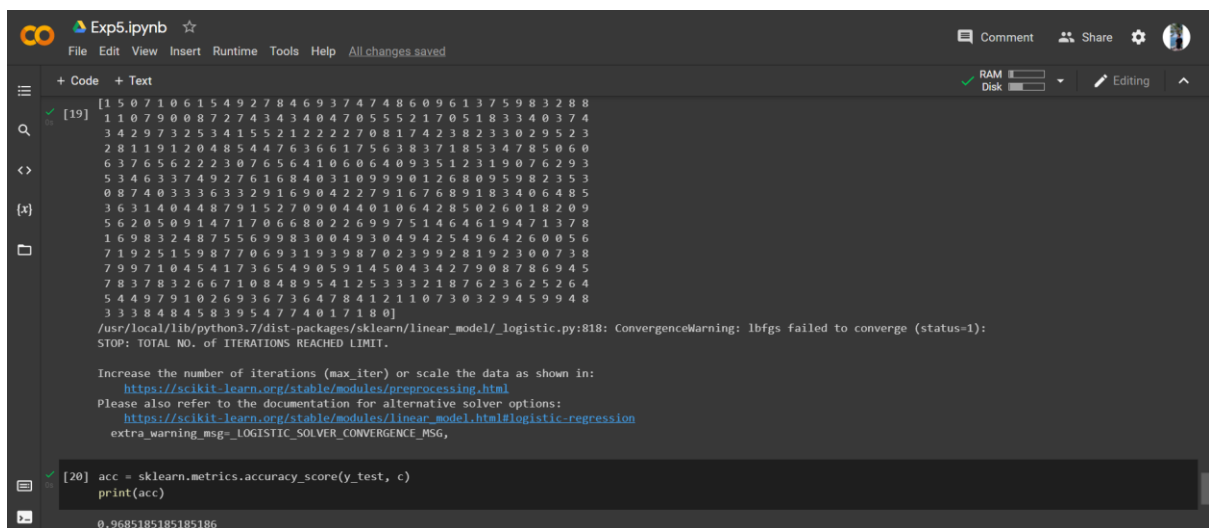
      x = digits.data
      y = digits.target

[17] X_train, X_test, y_train, y_test = train_test_split(x,y,test_size=0.3, random_state=1)

[18] log_reg = LogisticRegression()

[19] b = log_reg.fit(X_train,y_train)
      c = log_reg.predict(X_test )

      print(c)
```



```
Exp5.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
RAM Disk
Editing

[19] [1 5 0 7 1 0 6 1 5 4 9 2 7 8 4 6 9 3 7 4 7 4 8 6 0 9 6 1 3 7 5 9 8 3 2 8 8
      1 1 0 7 9 0 0 8 7 2 7 4 3 4 3 4 0 4 7 0 5 5 2 1 7 0 5 1 8 3 3 4 0 3 7 4
      3 4 2 9 7 3 2 5 3 4 1 5 5 2 1 2 2 2 7 0 8 1 7 4 2 3 8 2 3 3 0 2 9 5 2 3
      2 8 1 1 9 1 2 0 4 8 5 4 4 7 6 3 6 6 1 7 5 6 3 8 3 7 1 8 5 3 4 7 8 5 0 6 0
      6 3 7 6 5 6 2 2 2 3 0 7 6 5 6 4 1 0 6 0 6 4 0 9 3 5 1 2 3 1 9 0 7 6 2 9 3
      5 3 4 6 3 7 4 9 2 7 6 1 6 8 4 0 3 1 0 9 9 9 0 1 2 6 8 0 9 5 9 8 2 3 5 3
      0 8 7 4 0 3 3 3 6 3 3 2 9 1 6 9 0 4 2 2 7 9 1 6 7 6 8 9 1 8 3 4 0 6 4 8 5
      3 6 3 1 4 0 4 4 8 7 9 1 5 2 7 0 9 0 4 4 0 1 0 6 4 2 8 5 0 2 6 0 1 8 2 0 9
      5 6 2 0 5 0 9 1 4 7 1 7 0 6 6 8 0 2 2 6 9 9 7 5 1 4 6 4 6 1 9 4 7 1 3 7 8
      1 6 9 8 3 2 4 8 7 5 5 6 9 9 8 3 0 0 4 9 3 0 4 9 4 2 5 4 9 6 4 2 6 0 0 5 6
      7 1 9 2 5 1 5 9 8 7 7 0 6 9 3 1 9 3 9 8 7 0 2 3 9 9 2 8 1 9 2 3 0 0 7 3 8
      7 9 9 7 1 0 4 5 4 1 7 3 6 5 4 9 0 5 9 1 4 5 0 4 3 4 2 7 9 0 8 7 8 6 9 4 5
      7 8 3 7 8 3 2 6 6 7 1 0 8 4 8 9 5 4 1 2 5 3 3 3 2 1 8 7 6 2 3 6 2 5 2 6 4
      5 4 4 9 7 9 1 0 2 6 9 3 6 7 3 6 4 7 8 4 1 2 1 1 0 7 3 0 3 2 9 4 5 9 9 4 8
      3 3 8 4 8 4 5 8 3 9 5 4 7 7 4 0 1 7 1 8 0]

      /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):
      STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

      Increase the number of iterations (max_iter) or scale the data as shown in:
      https://scikit-learn.org/stable/modules/preprocessing.html
      Please also refer to the documentation for alternative solver options:
      https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
      extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

[20] acc = sklearn.metrics.accuracy_score(y_test, c)
      print(acc)

0.9685185185185186
```

C. Conclusion:

Write the significance of the topic studied in the experiment.

Conclusion

logistic regression is used in statistical software to understand the relationship between the dependent variable & one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help you predict the likelihood of an event happening or a choice is made

D. References:

1. [Logistic Regression using Python \(scikit-learn\) | by Michael Galarnyk | Towards Data Science](#)
2. [Logistic Regression in Python | Techniques for Logistic Regression \(educba.com\)](#)