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.

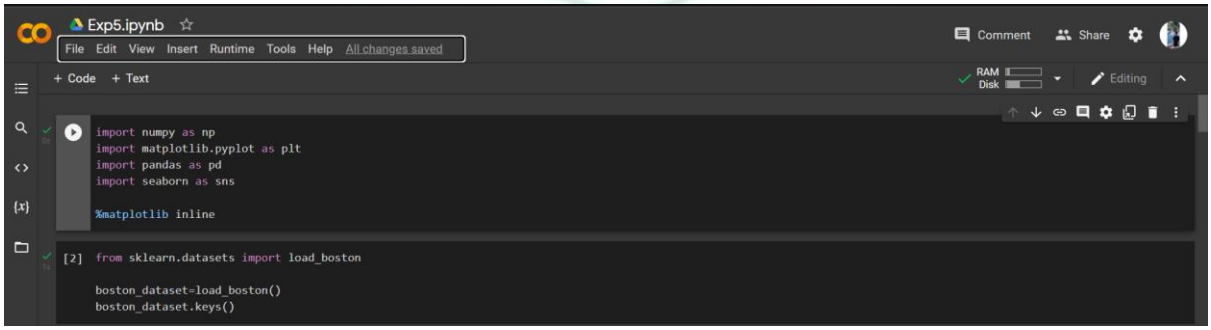
1. **Aim:** To implement Linear and Logistic Regression to find out relation between variables.
2. **Objectives:** After study of this experiment, the student will be able to
   - Understand linear regression
   - Understand logistic regression
3. **Outcomes:** After study of this experiment, the student 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.



Exp5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[3] boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
    boston.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

```
[4] boston['MEDV'] = boston_dataset.target
```

```
[5] # DATA PREPROCESSING
    #Check for missing values
    boston.isnull().sum()
```

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
```

Exp5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

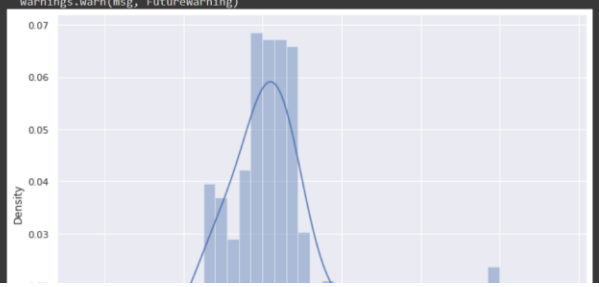
+ Code + Text

RAM Disk Editing

```
[6] #DATA VISUALIZATION
    #set size of figure
    sns.set(rc={'figure.figsize':(11,8)})

    # plot histogram
    sns.distplot(boston['MEDV'],bins=30)
    plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Plea
  warnings.warn(msg, FutureWarning)
```



Exp5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[7] # compute the pair wise correlation for all columns
    correlation_matrix=boston.corr().round(2)
```

```
[8] # using heatmap to display matrix
    # annot = True to print the values inside the square
    sns.heatmap(data=correlation_matrix, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f07d1581910>
```

Exp5.ipynb

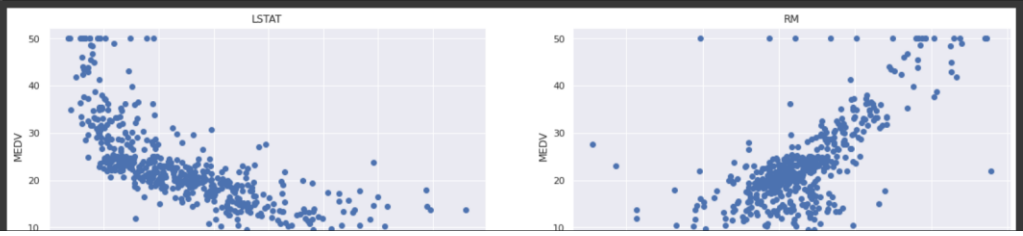
File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
plt.figure(figsize=(20,5))
features=['LSTAT','RM']
target=boston['MEDV']
for i, col in enumerate(features):
    plt.subplot(1, len(features) , i+1)
    x = boston[col]
    y = target
    plt.scatter(x, y, marker='o')
    plt.title(col)
    plt.xlabel(col)
    plt.ylabel('MEDV')
```



Exp5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[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

Comment Share

+ Code + Text

RAM Disk Editing

```
[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()
```

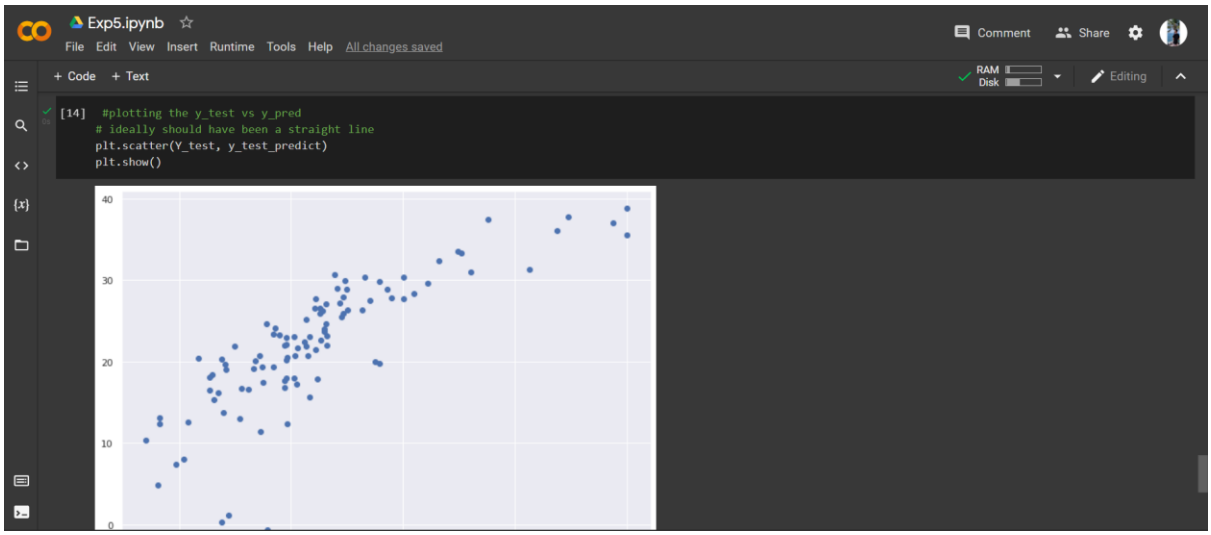
```
[13] # model evaluation for training set

     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)))
```



8. Post-Experiments Exercise

A. Extended Theory: (Soft Copy)

- Logistic regression

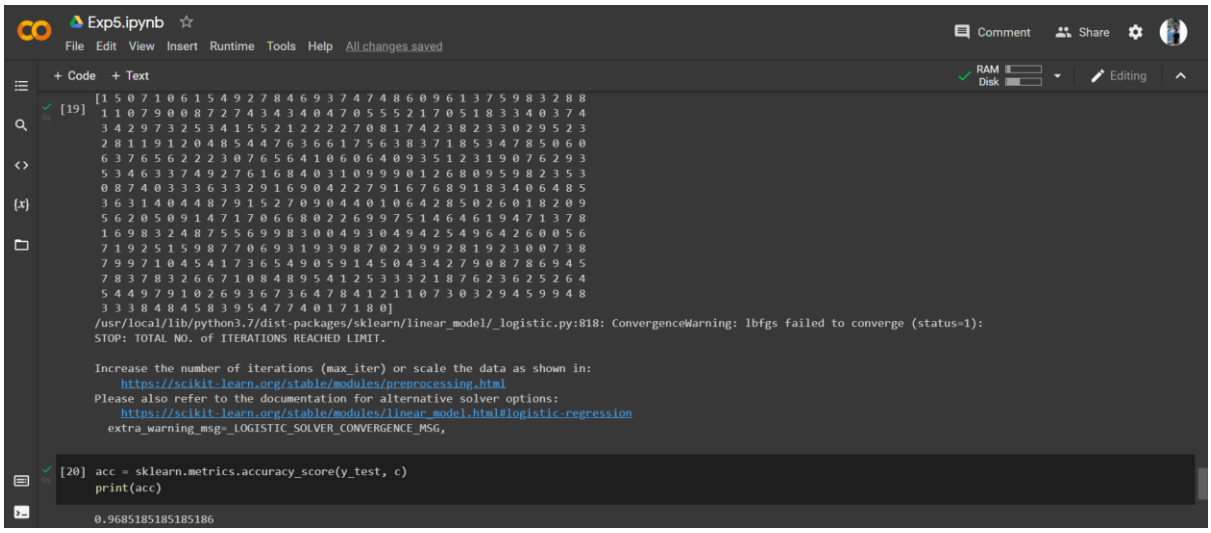
Logistic Regression is a kind of parametric classification model, despite having the word ‘regression’ in its name.

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.

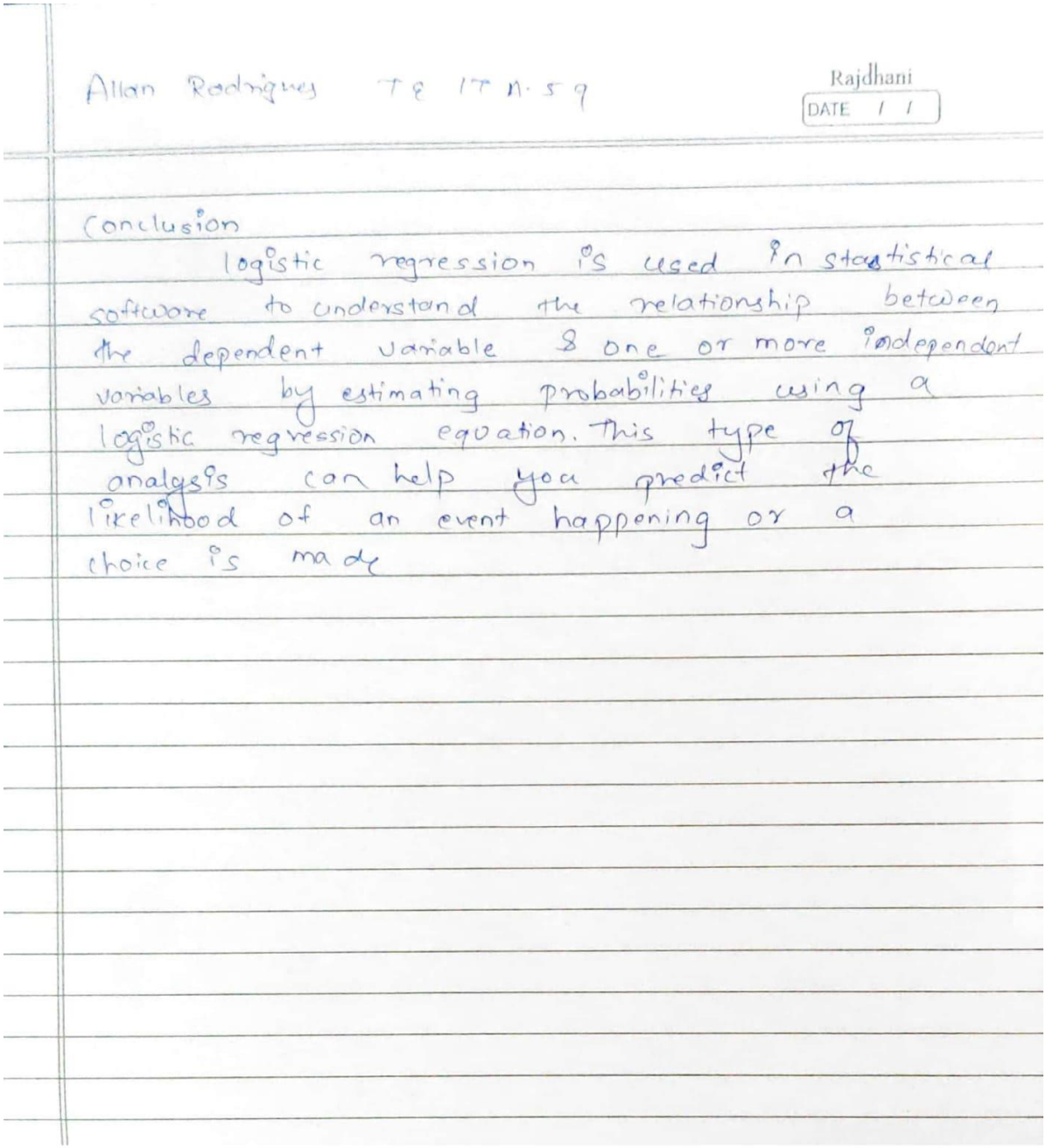
B. Questions:

- Use MNIST Dataset and apply logistic regression.



C. Conclusion:

Write the significance of the topic studied in the experiment.



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)