EAST WEST UNIVERSITY

Lab Report-08

Course Title: Artificial Intelligence

Course Code: CSE 366

Semester: Fall 2021

Section No: 01

Submitted By

Name: Syeda Tamanna Sheme

ID:2018-2-60-010

Submitted To

Md Al-Imran

Lecturer

Department of Computer Science & Engineering

Date of Submission: 31 December, 2021

Lab-08: Implementation of Linear Regression and Logistic Regression

Theory

Linear Regression

Linear regression is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous or real or numeric variables such as **sales, salary, age, product price,** etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

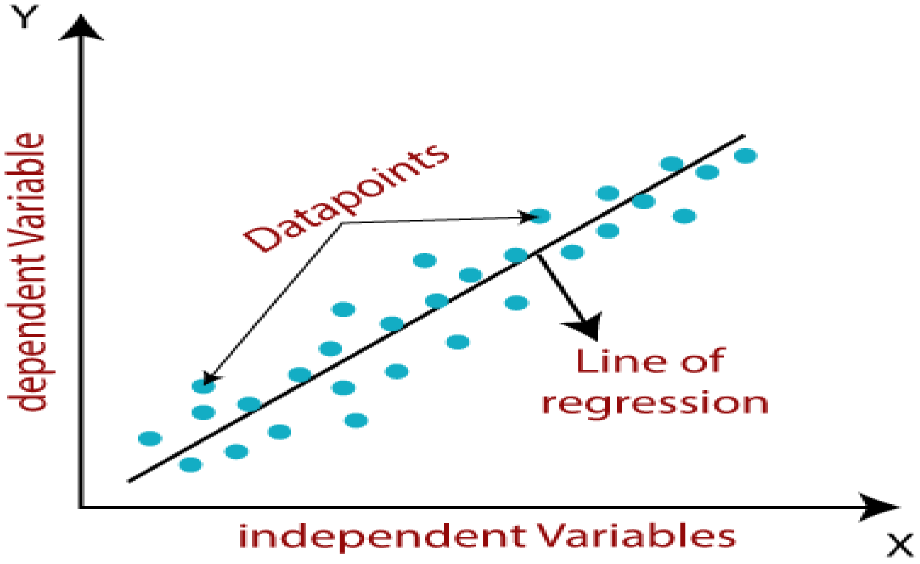


Figure: Linear Regression

Mathematically, we can represent a linear regression as:

$$\mathbf{y= a_0+a_1x+ \varepsilon}$$

Here,

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

Advantages And Disadvantages

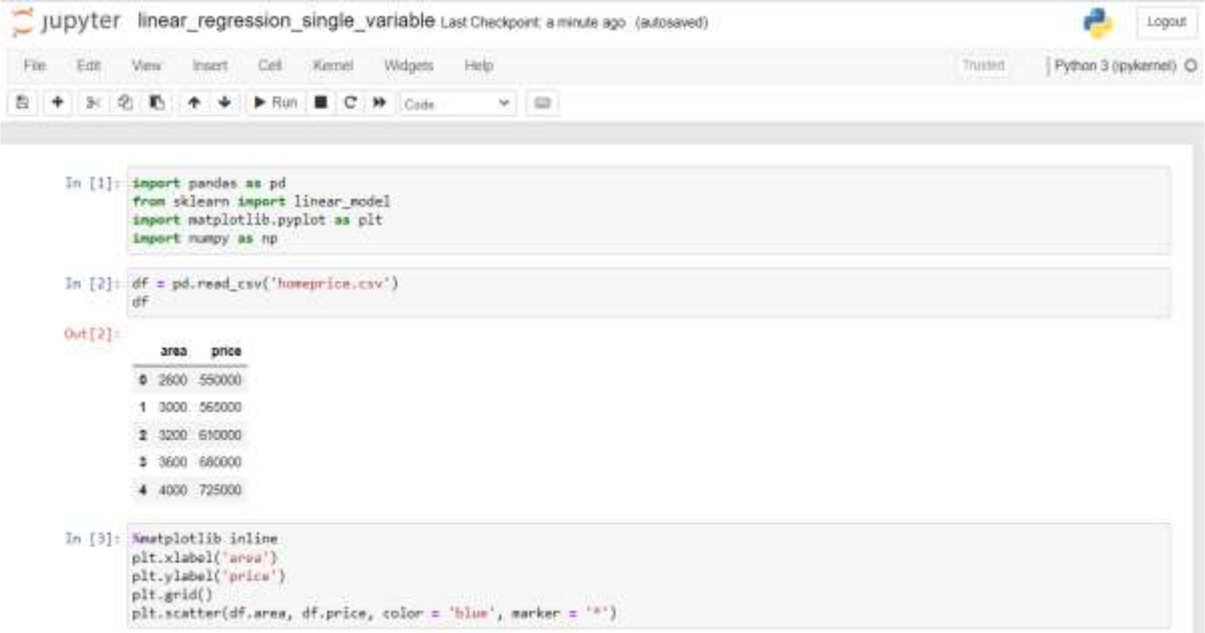
Advantages	Disadvantages
Linear regression performs exceptionally well for linearly separable data	The assumption of linearity between dependent and independent variables
Easier to implement, interpret and efficient to train	It is often quite prone to noise and overfitting
It handles overfitting pretty well using dimensionally reduction techniques, regularization, and cross-validation	Linear regression is quite sensitive to outliers
One more advantage is the extrapolation beyond a specific data set	It is prone to multicollinearity

Linear Regression Use Cases

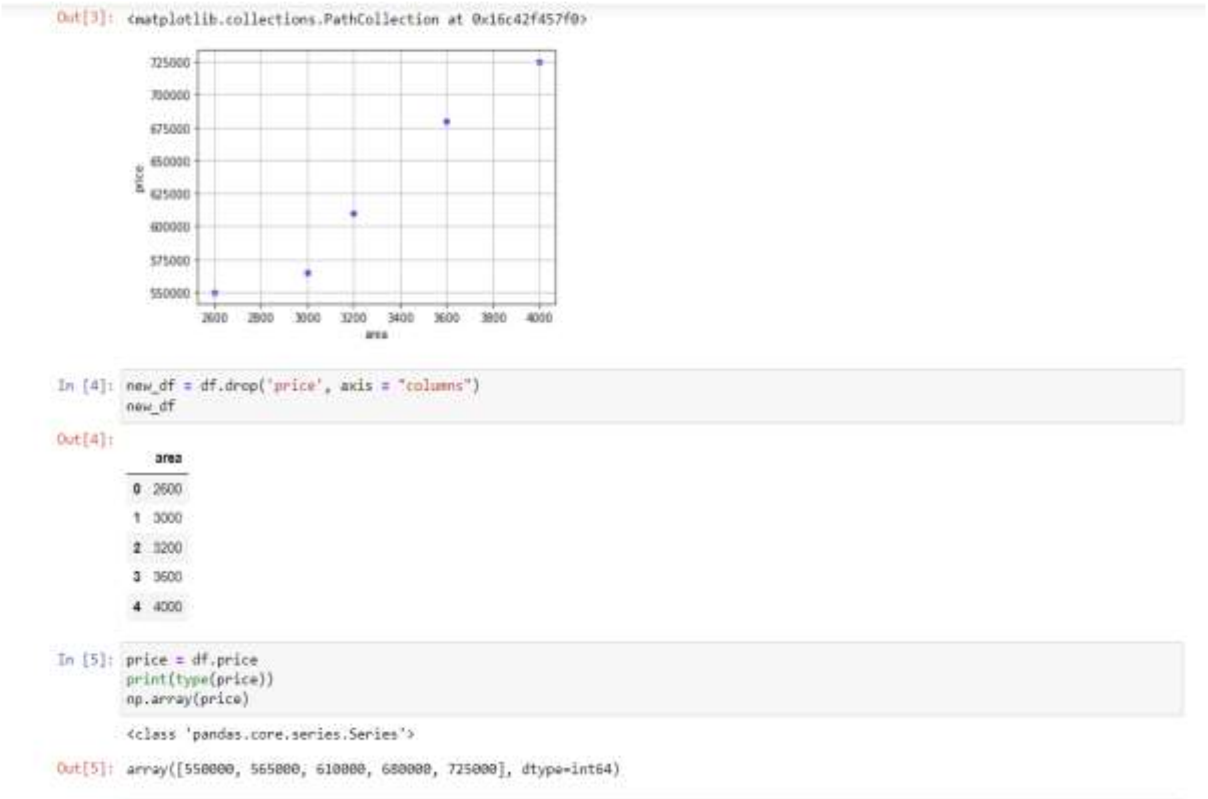
- Sales Forecasting
- Risk Analysis
- Housing Applications to Predict the prices and other factors
- Finance Applications to Predict Stock prices, investment evaluation, etc.

The basic idea behind linear regression is to find the relationship between the dependent and independent variables. It is used to get the best fitting line that would predict the outcome with the least error. We can use linear regression in simple real-life situations, like predicting the SAT scores with regard to the number of hours of study and other decisive factors.

Code Screenshot:



```
In [1]: import pandas as pd
        from sklearn import linear_model
        import matplotlib.pyplot as plt
        import numpy as np

In [2]: df = pd.read_csv('homeprice.csv')
        df

Out[2]:
           area   price
        0  2600  550000
        1  3000  565000
        2  3200  610000
        3  3600  680000
        4  4000  725000

In [3]: %matplotlib inline
        plt.xlabel('area')
        plt.ylabel('price')
        plt.grid()
        plt.scatter(df.area, df.price, color = 'blue', marker = '*')
```



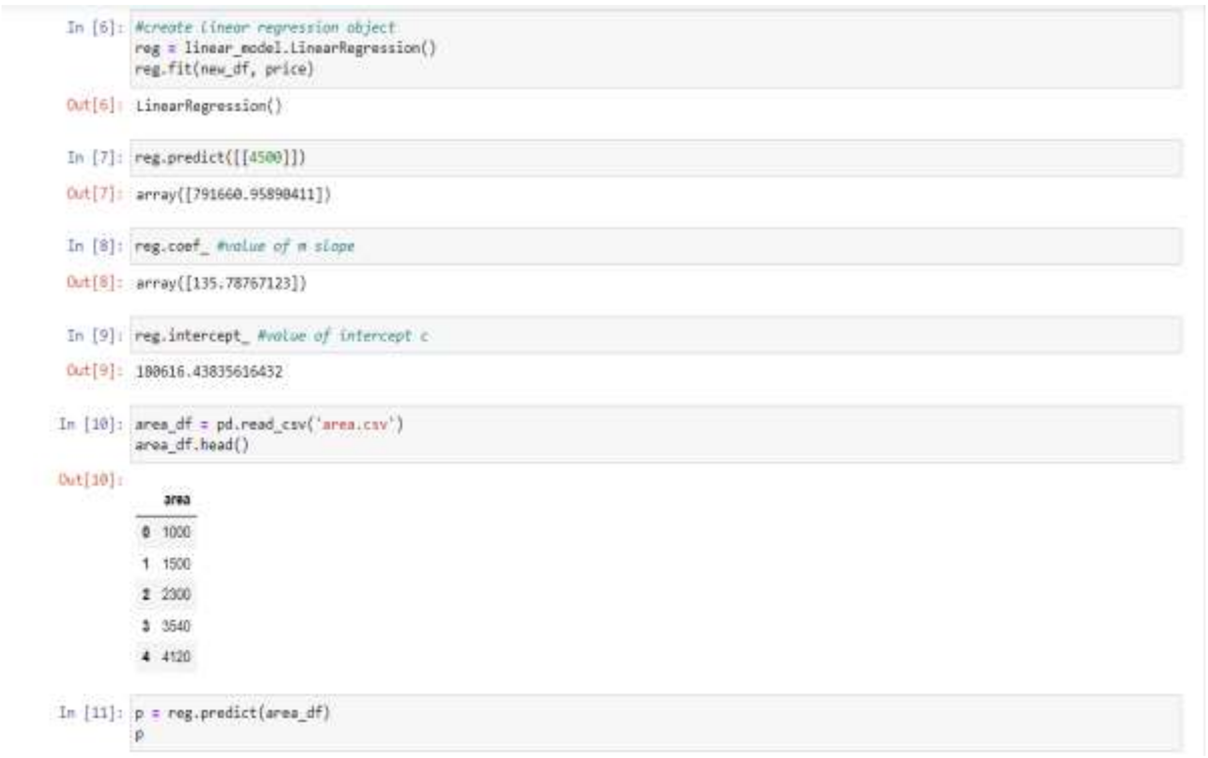
```
Out[3]: <matplotlib.collections.PathCollection at 0x16c42f457f0>
```

```
In [4]: new_df = df.drop('price', axis = "columns")
        new_df
```

Out[4]:

	area
0	2600
1	3000
2	3200
3	3600
4	4000

```
In [5]: price = df.price
        print(type(price))
        np.array(price)

        <class 'pandas.core.series.Series'>

Out[5]: array([550000, 565000, 610000, 680000, 725000], dtype=int64)
```

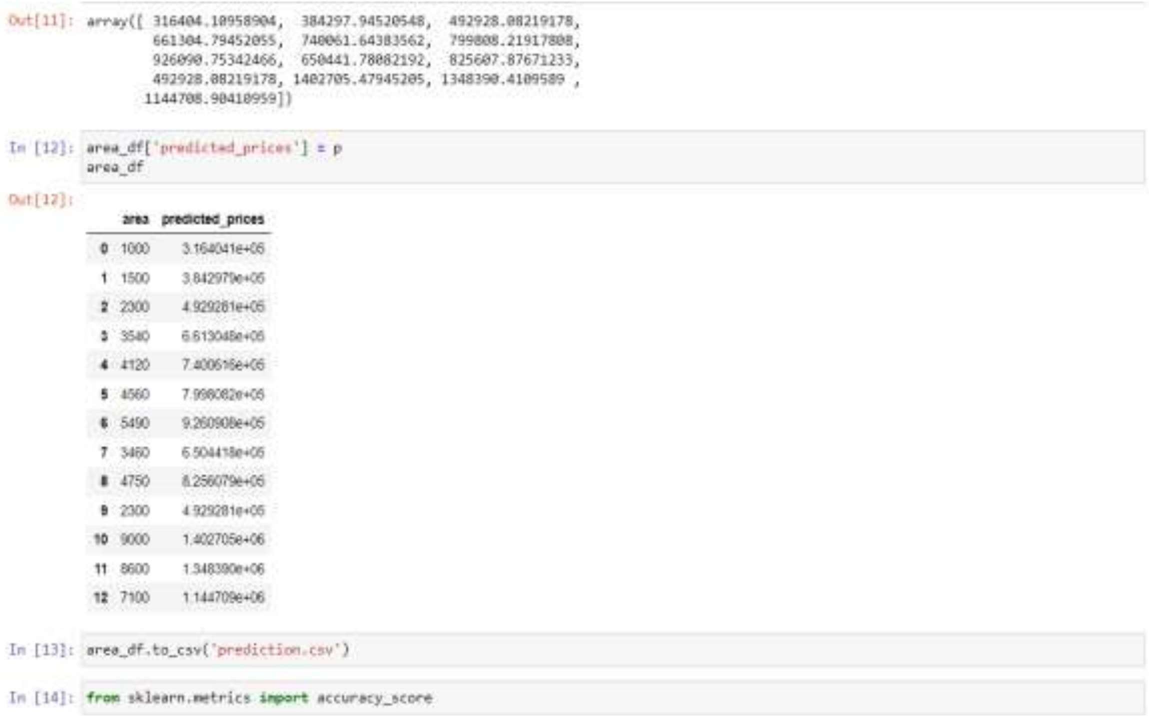


```
In [6]: #create linear regression object
        reg = linear_model.LinearRegression()
        reg.fit(new_df, price)

Out[6]: LinearRegression()

In [7]: reg.predict([[4500]])

Out[7]: array([791660.95890411])

In [8]: reg.coef_ #value of m slope

Out[8]: array([135.78767123])

In [9]: reg.intercept_ #value of intercept c

Out[9]: 180616.43835616432

In [10]: area_df = pd.read_csv('area.csv')
         area_df.head()
```

Out[10]:

	area
0	1000
1	1500
2	2300
3	3540
4	4120

```
In [11]: p = reg.predict(area_df)
         p
```



```
Out[11]: array([ 316404.10958904,  384297.94520548,  492928.08219178,
                 661304.79452055,  740061.64383562,  799808.21917808,
                 926090.75342466,  650441.78082192,  825607.87671233,
                 492928.08219178, 1402705.47945205, 1348390.4109589 ,
                1144708.90410959])

In [12]: area_df['predicted_prices'] = p
         area_df

Out[12]:
```

	area	predicted_prices
0	1000	3.164041e+05
1	1500	3.842979e+05
2	2300	4.929281e+05
3	3540	6.613048e+05
4	4120	7.400616e+05
5	4560	7.998082e+05
6	5490	9.260908e+05
7	3460	6.504418e+05
8	4750	8.256079e+05
9	2300	4.929281e+05
10	9000	1.402705e+06
11	8600	1.348390e+06
12	7100	1.144709e+06

```
In [13]: area_df.to_csv('prediction.csv')

In [14]: from sklearn.metrics import accuracy_score
```

Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

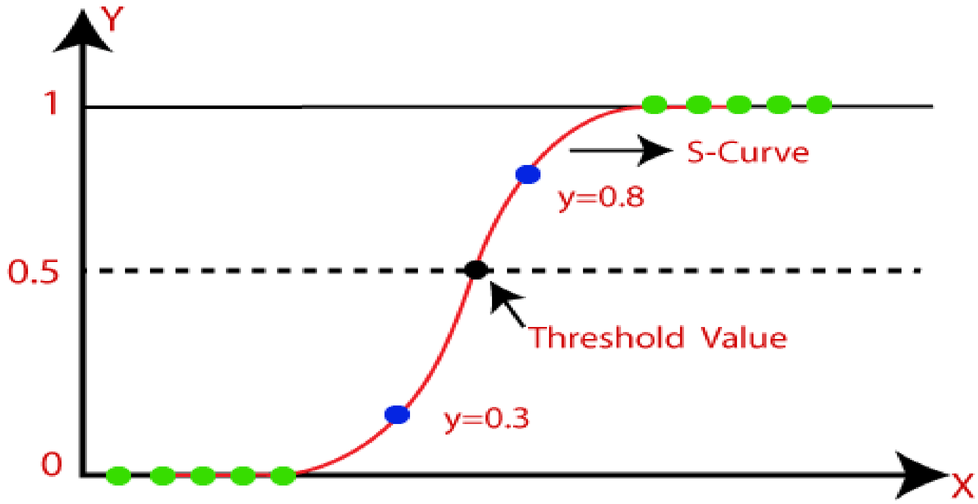


Figure: Logistic Regression

Applications of Logistic Regression

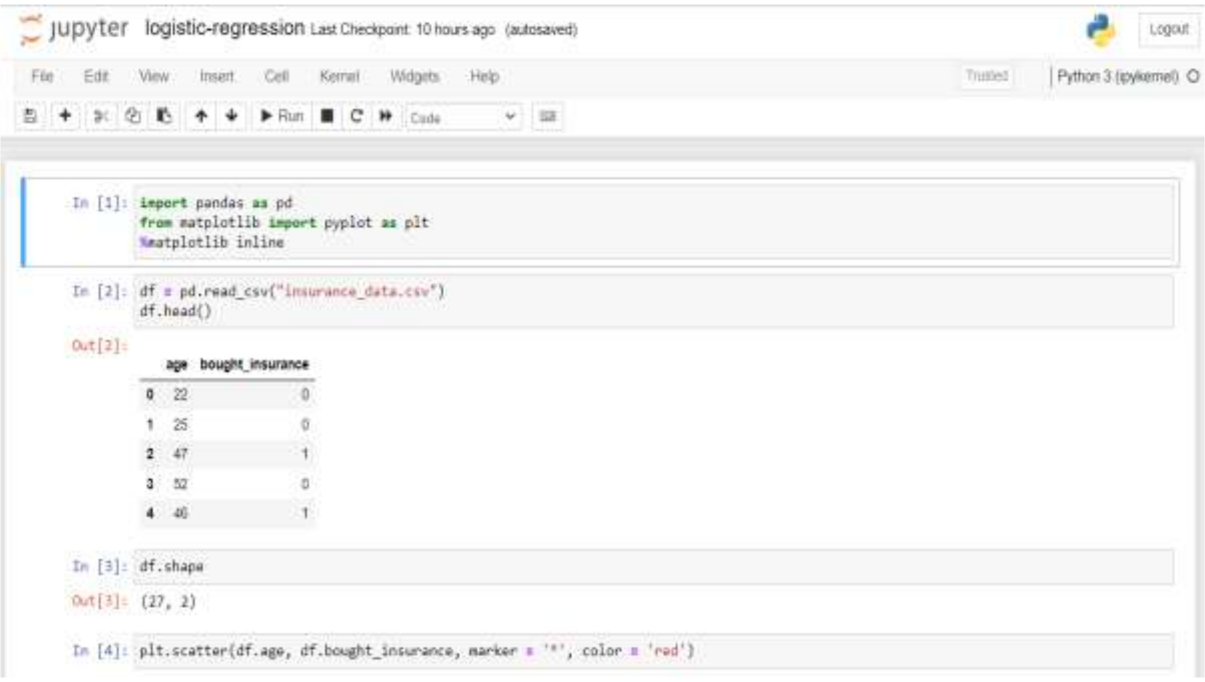
Some of the applications of logistic regression. Predicting a probability of a person having a heart attack Predicting a customer’s propensity to purchase a product or halt a subscription Predicting the probability of failure of a given process or product.

Uses of Logistic Regression:

- When your data is binary: 0/1, True/False, Yes/No
- When you need probabilistic results
- When your data is linearly separable
- When you need to understand the impact of the feature.

Many other classification algorithms are widely used other than logistic regression like kNN, decision trees, random forest, and clustering algorithms like k-means clustering. But logistic regression is a widely used algorithm and also easy to implement.

Code Screenshot:


jupyter
logistic-regression Last Checkpoint: 10 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        %matplotlib inline
```

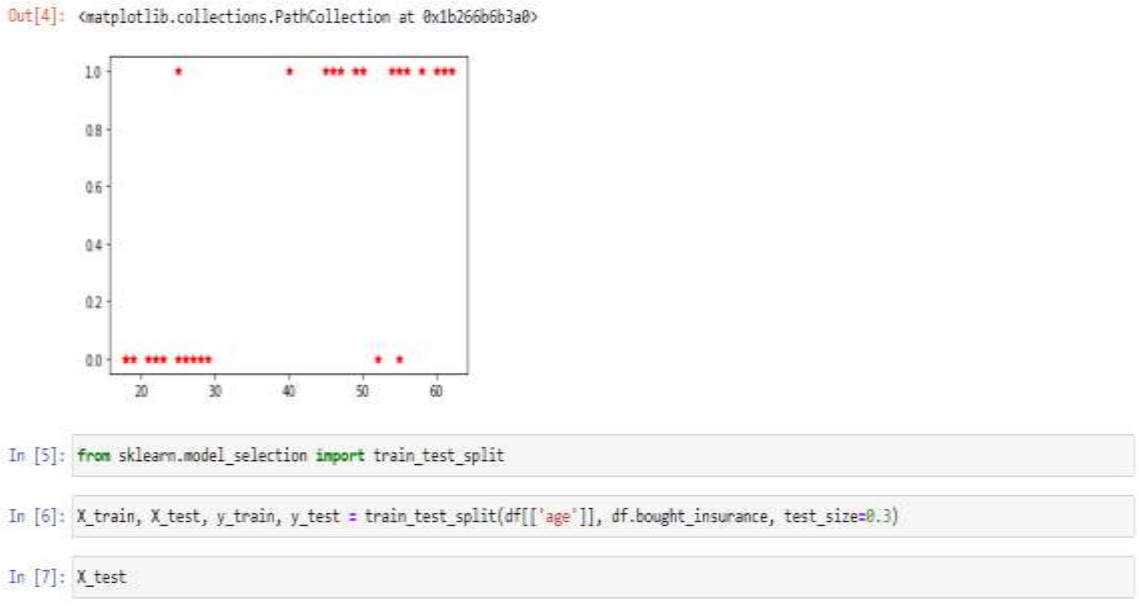
```
In [2]: df = pd.read_csv("insurance_data.csv")
        df.head()
```

Out[2]:

	age	bought_insurance
0	22	0
1	25	0
2	47	1
3	52	0
4	46	1

```
In [3]: df.shape
Out[3]: (27, 2)
```

```
In [4]: plt.scatter(df.age, df.bought_insurance, marker = '*', color = 'red')
```

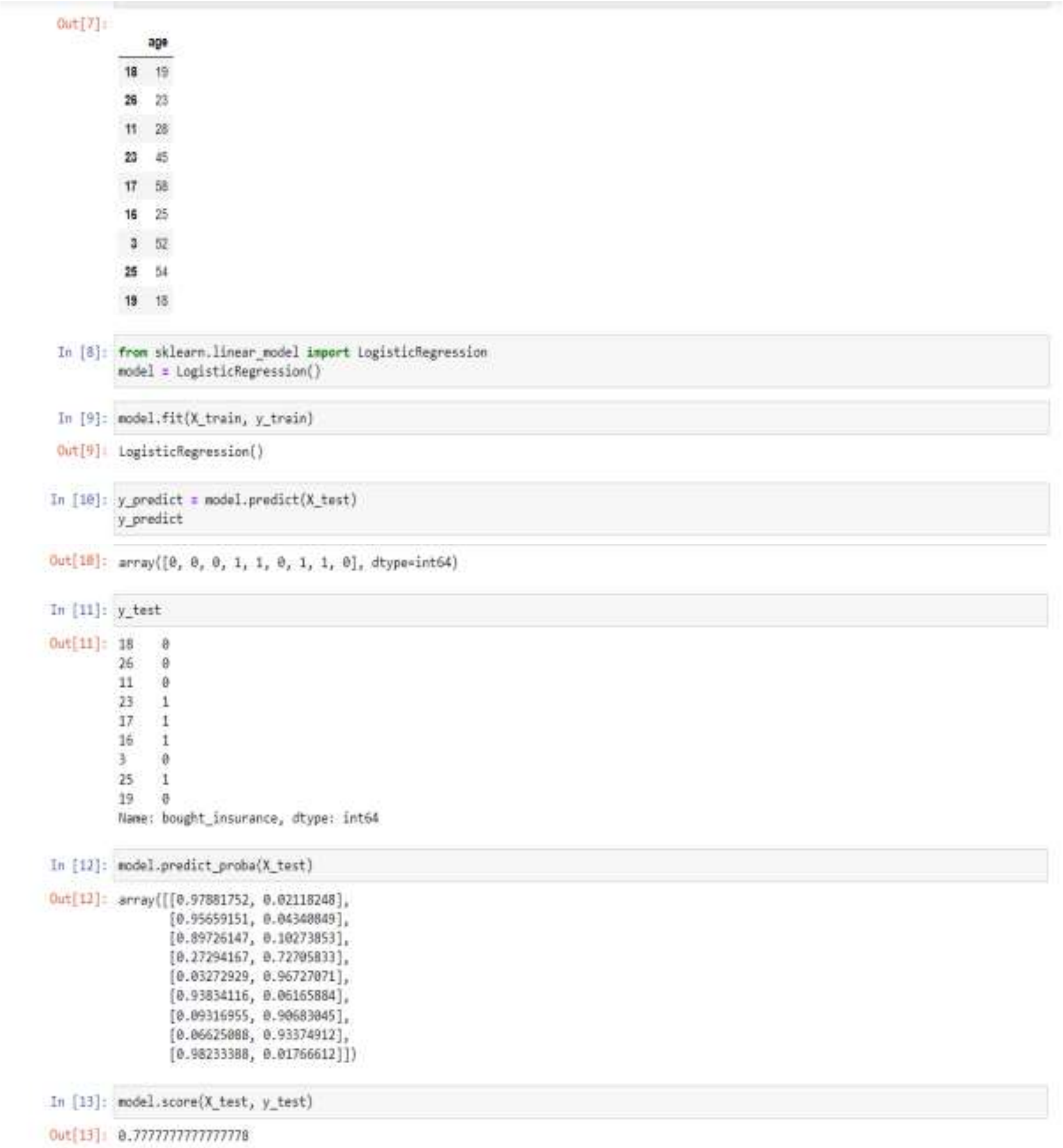


```
Out[4]: <matplotlib.collections.PathCollection at 0x1b266b6b3a0>
```

```
In [5]: from sklearn.model_selection import train_test_split
```

```
In [6]: X_train, X_test, y_train, y_test = train_test_split(df[['age']], df.bought_insurance, test_size=0.3)
```

```
In [7]: X_test
```



```
Out[7]:
        age
   18   19
   26   23
   11   28
   23   45
   17   58
   16   25
    3   52
   25   54
   19   18
```

```
In [8]: from sklearn.linear_model import LogisticRegression
        model = LogisticRegression()

In [9]: model.fit(X_train, y_train)

Out[9]: LogisticRegression()

In [10]: y_predict = model.predict(X_test)
         y_predict

Out[10]: array([0, 0, 0, 1, 1, 0, 1, 1, 0], dtype=int64)

In [11]: y_test

Out[11]: 18    0
         26    0
         11    0
         23    1
         17    1
         16    1
         3     0
         25    1
         19    0
         Name: bought_insurance, dtype: int64

In [12]: model.predict_proba(X_test)

Out[12]: array([[0.97881752, 0.02118248],
                [0.95659151, 0.04340849],
                [0.89726147, 0.10273853],
                [0.27294167, 0.72705833],
                [0.03272929, 0.96727071],
                [0.93834116, 0.06165884],
                [0.09316955, 0.90683045],
                [0.06625088, 0.93374912],
                [0.98233388, 0.01766612]])

In [13]: model.score(X_test, y_test)

Out[13]: 0.7777777777777778
```

Linear Regression vs Logistic Regression

Linear Regression and Logistic Regression are the two famous Machine Learning Algorithms which come under supervised learning technique. Since both the algorithms are of supervised in nature hence these algorithms use labeled dataset to make the predictions. But the main difference between them is how they are being used. The Linear Regression is used for solving Regression problems whereas Logistic Regression is used for solving the Classification problems. The description of both the algorithms is given below along with difference table.

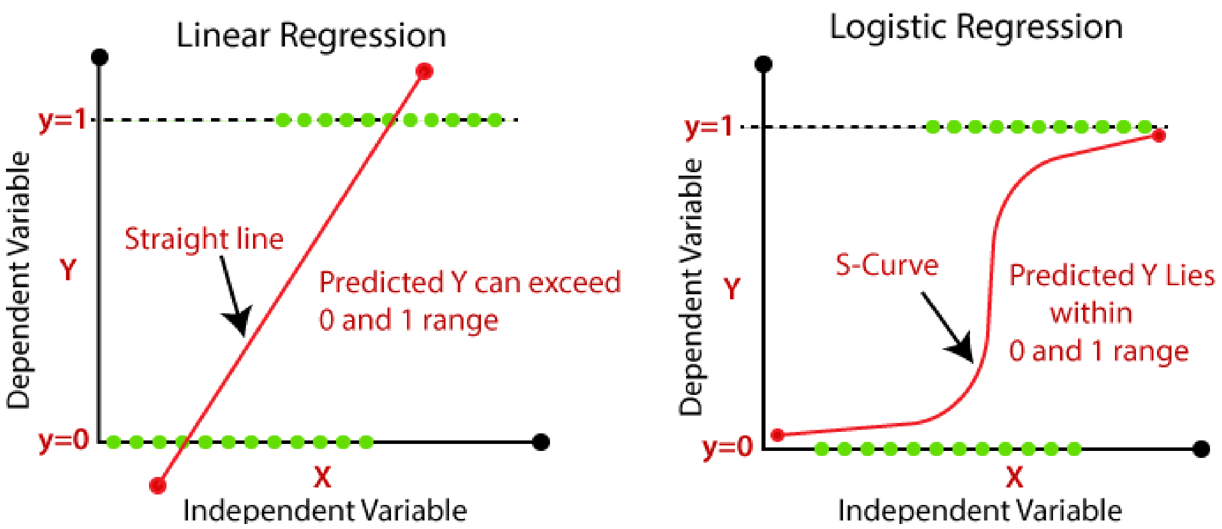


Figure: Linear Regression vs Logistic Regression

Linear Regression	Logistic Regression
Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables.
Linear Regression is used for solving Regression problem.	Logistic regression is used for solving Classification problems.
In Linear regression, we predict the value of continuous variables.	In logistic Regression, we predict the values of categorical variables.
In linear regression, we find the best fit line, by which we can easily predict the output.	In Logistic Regression, we find the S-curve by which we can classify the samples.
Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for estimation of accuracy.
The output for Linear Regression must be a continuous value, such as price, age, etc.	The output of Logistic Regression must be a Categorical value such as 0 or 1, Yes or No, etc.
In Linear regression, it is required that relationship between dependent variable and independent variable must be linear.	In Logistic regression, it is not required to have the linear relationship between the dependent and independent variable.
In linear regression, there may be collinearity between the independent variables.	In logistic regression, there should not be collinearity between the independent variable.

Results

We learn **Linear Regression and Logistic Regression** after completing the lab task and lab activity. We also learn how to add data, read data, solve problem using **Linear Regression and Logistic Regression**. We are now able to produce shows on these subjects.

Discussion

Python is a general-purpose, versatile and popular programming language. It’s great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer’s stack as it can be used for everything from web development to software development and data science applications.

This lab task is a great introduction to both fundamental programming concepts and the Python programming language. Python 3 is the most up-to-date version of the language with many improvements made to increase the efficiency and simplicity of the code that we write. Day by-day, python new version are realising and all the new versions have new features and these new features are better than previous one. So, finally I can say that python will be more user friendly in future.