Linear and Logistic Regression – Lab Work

What is regression?

Regression is the process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of **Market Trends**, prediction of House prices, etc.

The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).

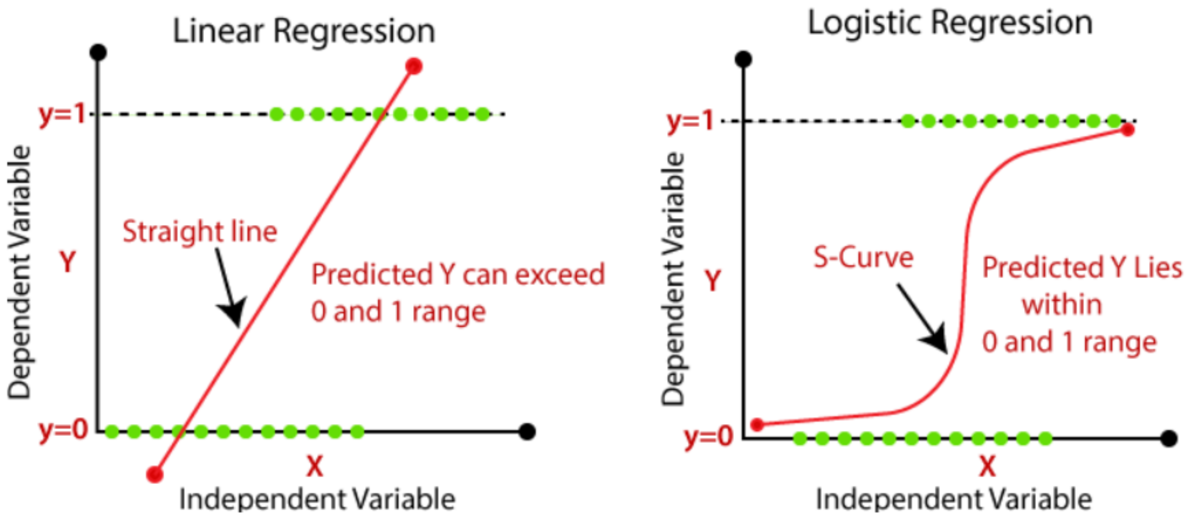
Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

Types of Regression Algorithm:

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression

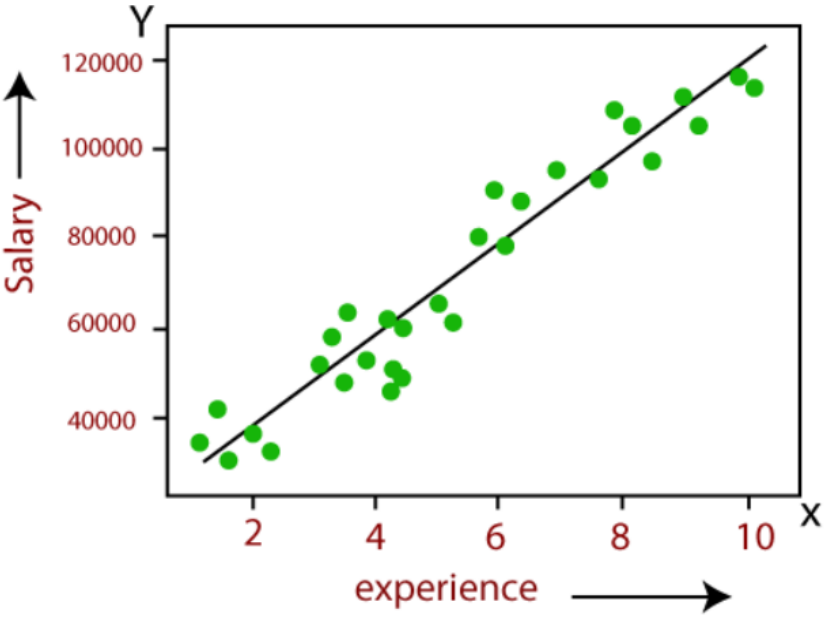
Linear Regression and Logistic Regression are the two famous Machine Learning Algorithms which come under supervised learning technique. Since both the algorithms are of supervised in nature hence these algorithms use labeled dataset to make the predictions. But the main difference between them is how 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.



Linear Regression:

- Linear Regression is one of the most simple Machine learning algorithm that comes under Supervised Learning technique and used for solving regression problems.
- It is used for predicting the continuous dependent variable with the help of independent variables.
- The goal of the Linear regression is to find the best fit line that can accurately predict the output for the continuous dependent variable.
- If single independent variable is used for prediction then it is called Simple Linear Regression and if there are more than two independent variables then such regression is called as Multiple Linear Regression.
- By finding the best fit line, algorithm establish the relationship between dependent variable and independent variable. And the relationship should be of linear nature.
- We have several errors that can be calculated for the given predictions made by this regression model – including RelMSE, Absolute error, etc. **Conduct a survey on the multiple types of errors, their formulae, relationships and write about the same in your notebook.**
- The output for Linear regression should only be the continuous values such as price, age, salary, etc. The relationship between the dependent variable and independent variable can be shown in below image:



In the above image the dependent variable is on Y-axis (salary) and independent variable is on x-axis(experience). The regression line can be written as:

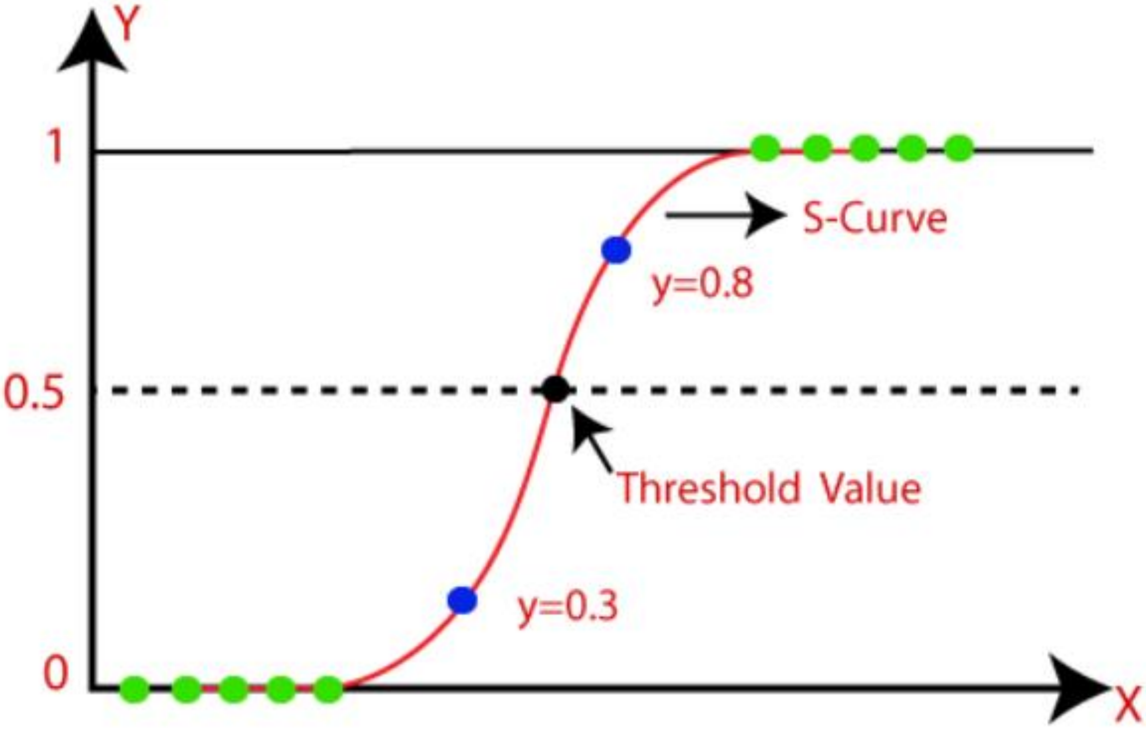
$$y= a_0 + a_1x + \varepsilon$$

Where, a_0 and a_1 are the coefficients and ε is the error term.

Logistic Regression:

- Logistic regression is one of the most popular Machine learning algorithm that comes under Supervised Learning techniques.
- It can be used for Classification as well as for Regression problems, but mainly used for Classification problems.
- Logistic regression is used to predict the categorical dependent variable with the help of independent variables.
- The output of Logistic Regression problem can be only between the 0 and 1.
- Logistic regression can be used where the probabilities between two classes is required. Such as whether it will rain today or not, either 0 or 1, true or false etc.
- Logistic regression is based on the concept of Maximum Likelihood estimation. According to this estimation, the observed data should be most probable.

- In logistic regression, we pass the weighted sum of inputs through an activation function that can map values in between 0 and 1. Such activation function is known as **sigmoid function** and the curve obtained is called as sigmoid curve or S-curve. Consider the below image:



- The equation for logistic regression is:

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The differences between linear and logistic regression are-

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.

Simple linear regression:

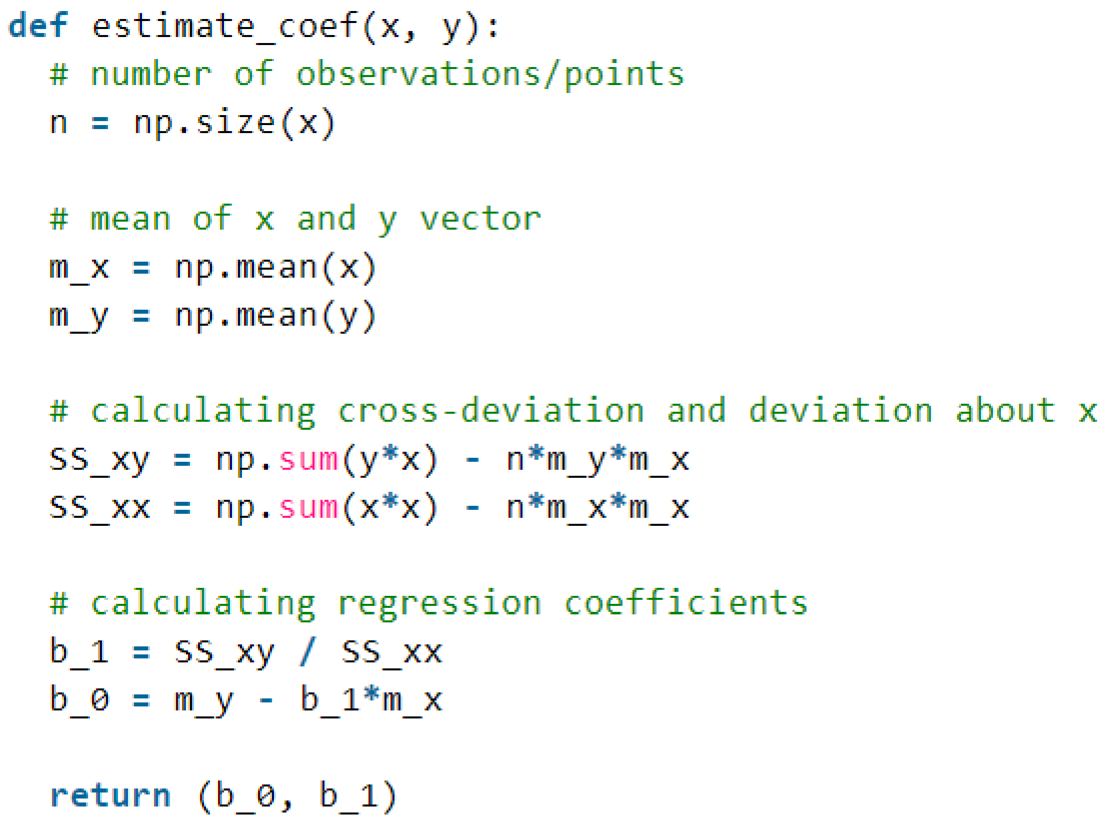
Find below code snippets to find the coefficients of the linear regression equation, and functions to plot the regression line on Python.

<u>Estimating Coefficients Function</u>

This function, estimate_coef(), takes the input data x (independent variable) and y (dependent variable) and estimates the coefficients of the linear regression line using the least squares method.

- **Calculating Number of Observations:** n = np.size(x) determines the number of data points.
- **Calculating Means:** m_x = np.mean(x) and m_y = np.mean(y) compute the mean values of x and y, respectively.

- **Calculating Cross-Deviation and Deviation about x**: SS_xy = np.sum(y*x) - n*m_y*m_x and SS_xx = np.sum(x*x) - n*m_x*m_x calculate the sum of squared deviations between x and y and the sum of squared deviations of x about its mean, respectively.
- **Calculating Regression Coefficients:** b_1 = SS_xy / SS_xx and b_0 = m_y - b_1*m_x determine the slope (b_1) and intercept (b_0) of the regression line using the least squares method.
- **Returning Coefficients:** The function returns the estimated coefficients as a tuple (b_0, b_1).



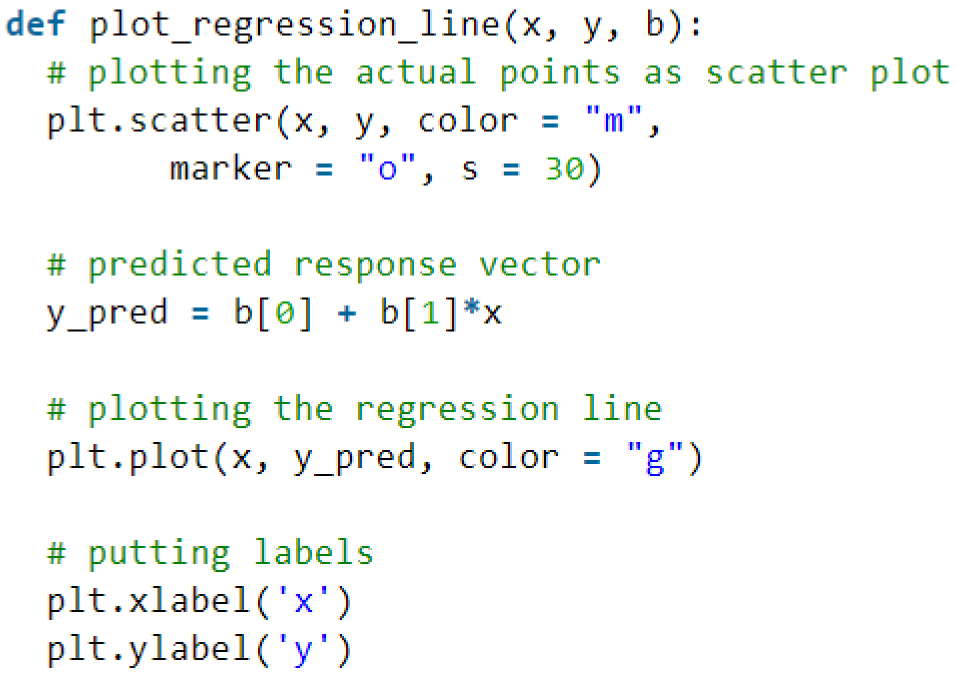
```
def estimate_coef(x, y):
  # number of observations/points
  n = np.size(x)

  # mean of x and y vector
  m_x = np.mean(x)
  m_y = np.mean(y)

  # calculating cross-deviation and deviation about x
  SS_xy = np.sum(y*x) - n*m_y*m_x
  SS_xx = np.sum(x*x) - n*m_x*m_x

  # calculating regression coefficients
  b_1 = SS_xy / SS_xx
  b_0 = m_y - b_1*m_x

  return (b_0, b_1)
```

Plotting Regression Line Function

This function, named plot_regression_line(), takes the input data from data, i.e. x (independent variable), y (dependent variable), and the estimated coefficients b to plot the regression line and the data points.

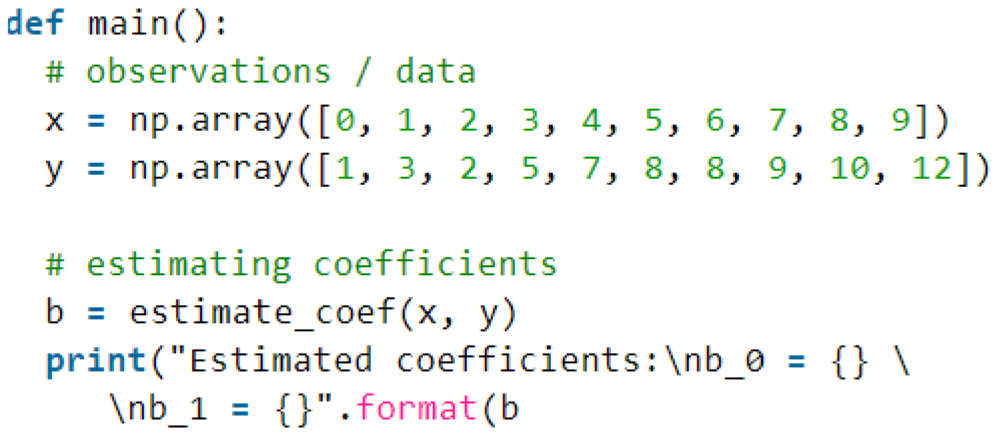
- **Plotting Scatter Plot:** plt.scatter(x, y, color = "m", marker = "o", s = 30) plots the original data points as a scatter plot with red markers.
- **Calculating Predicted Response Vector:** y_pred = b[0] + b[1]*x calculates the predicted values for y based on the estimated coefficients b.
- **Plotting Regression Line:** plt.plot(x, y_pred, color = "g") plots the regression line using the predicted values and the independent variable x.

- **Adding Labels:** plt.xlabel('x') and plt.ylabel('y') label the x-axis as 'x' and the y-axis as 'y', respectively.



```
def plot_regression_line(x, y, b):
  # plotting the actual points as scatter plot
  plt.scatter(x, y, color = "m",
        marker = "o", s = 30)

  # predicted response vector
  y_pred = b[0] + b[1]*x

  # plotting the regression line
  plt.plot(x, y_pred, color = "g")

  # putting labels
  plt.xlabel('x')
  plt.ylabel('y')
```

A sample program run using the functions defined is-



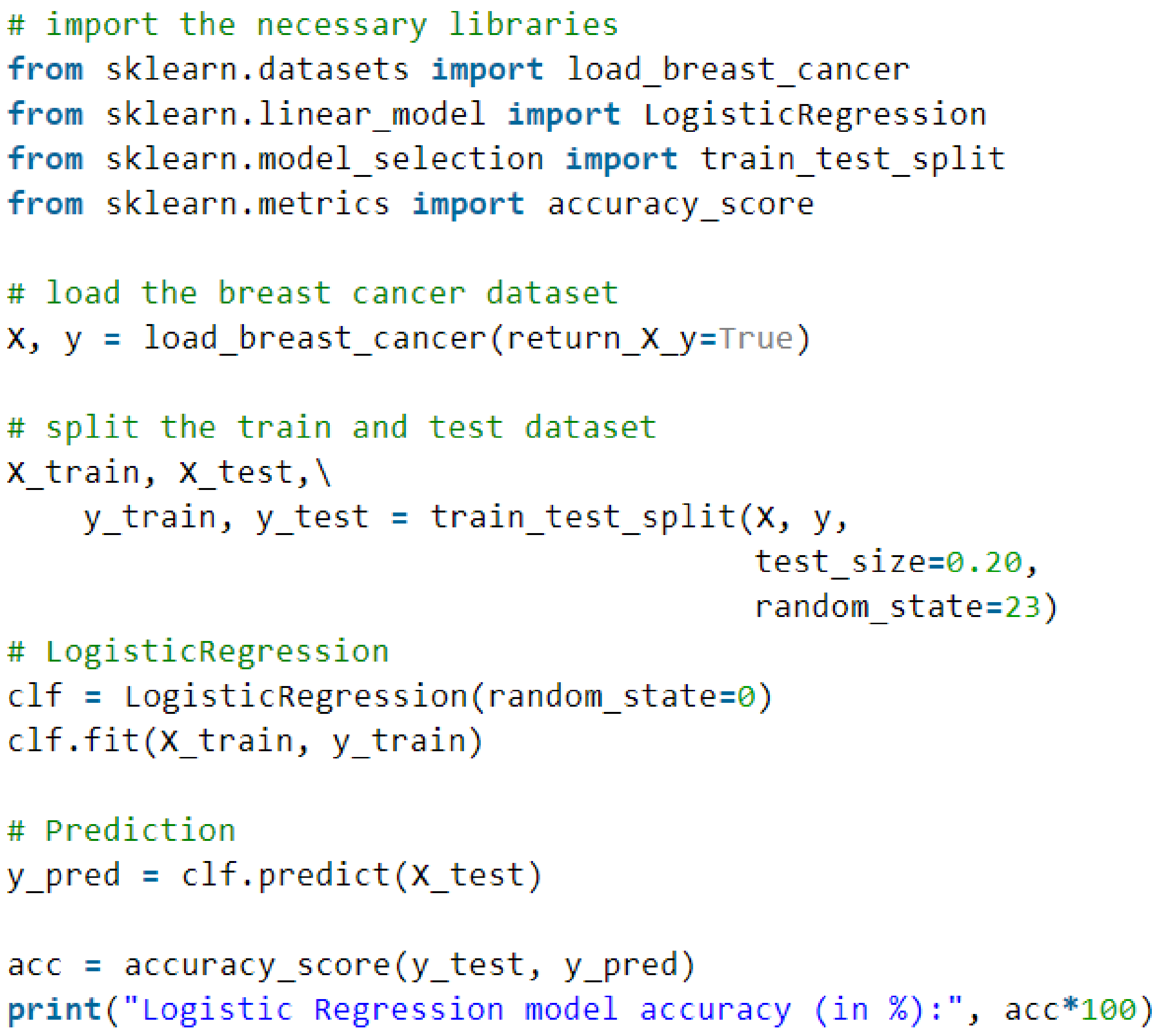
```
def main():
  # observations / data
  x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
  y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])

  # estimating coefficients
  b = estimate_coef(x, y)
  print("Estimated coefficients:\nb_0 = {} \
     \nb_1 = {}".format(b
```

Code implementation for logistic regression

The logistic regression model transforms the linear regression function continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1. This function is known as the logistic function.

Target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc., in this case, sigmoid functions are used.

Importing necessary libraries based on the requirement of model. This Python code shows how to use the breast cancer dataset to implement a Logistic Regression model for classification, using built in libraries. **Try and implement the same by using the mathematical formulae instead of the built-in model. Refer to the links after the code to understand the mathematical functions in logistic regression.**



```
# import the necessary libraries
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# load the breast cancer dataset
X, y = load_breast_cancer(return_X_y=True)

# split the train and test dataset
X_train, X_test,\
    y_train, y_test = train_test_split(X, y,
                                       test_size=0.20,
                                       random_state=23)
# LogisticRegression
clf = LogisticRegression(random_state=0)
clf.fit(X_train, y_train)

# Prediction
y_pred = clf.predict(X_test)

acc = accuracy_score(y_test, y_pred)
print("Logistic Regression model accuracy (in %):", acc*100)
```

Other references:

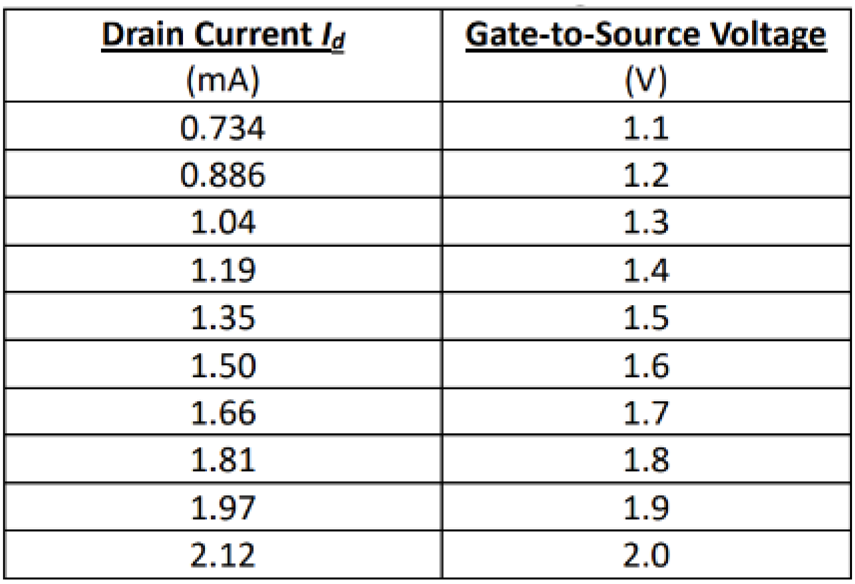
https://www.knowledgehut.com/blog/data-science/linear-regression-for-machine-learning#advantages-of%C2%A0linear-regression%C2%A0

https://www.youtube.com/watch?v=2C8IqOLO1os (Logistic Regression)

https://www.geeksforgeeks.org/understanding-logistic-regression/

Execution Questions

1. Shown below is a simple model to express drain current as a function of Gate-to-Source Voltage for a MOS transistor. The measurements made were tabulated, as shown below.



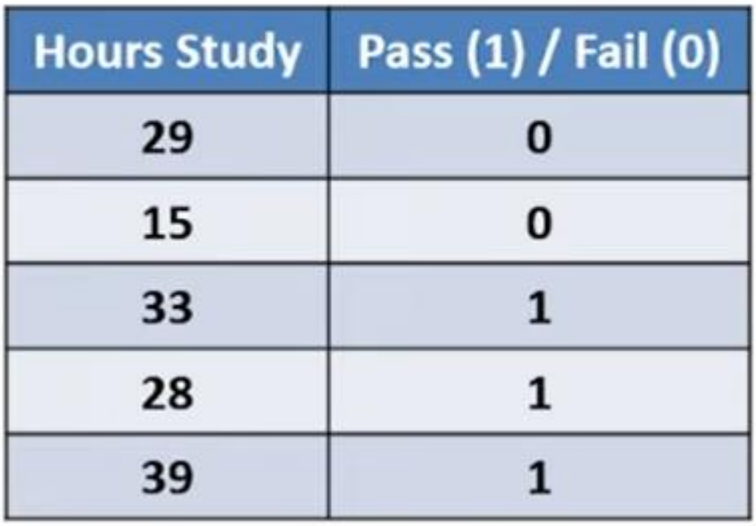
Drain Current I_d (mA)	**Gate-to-Source Voltage** (V)
0.734	1.1
0.886	1.2
1.04	1.3
1.19	1.4
1.35	1.5
1.50	1.6
1.66	1.7
1.81	1.8
1.97	1.9
2.12	2.0

(a) Develop the linear regression equation for the above data.

(b) Plot all the points, and compare how well they fit on the linear regression equation line.

(c) Find the error (for all entries in table) between predicted values obtained from linear regression equation, and actual values seen in the table.

2. The dataset of 5 students who passed/failed in an exam is given. Use logistic regression as a classifier to answer the following questions:



Hours Study	Pass (1) / Fail (0)
29	0
15	0
33	1
28	1
39	1

i. Calculate probability of pass for the student who studied 33 hours.

ii. Atleast how many hours should a student study such that his/her probability of passing is more than 95%

Assume the model suggested by the optimizer for odds of passing the course is,

log(odds)= -64 + 2*hours