

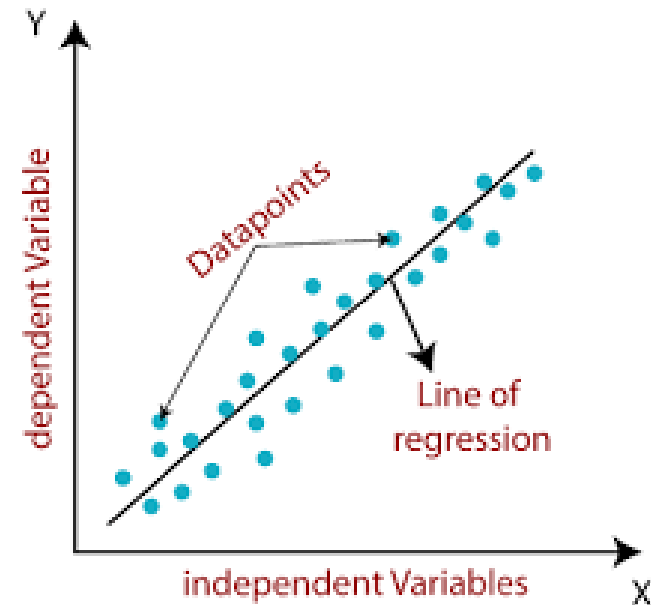


Logistic Regression

Dr. Sultan Alfarhood

Linear Regression

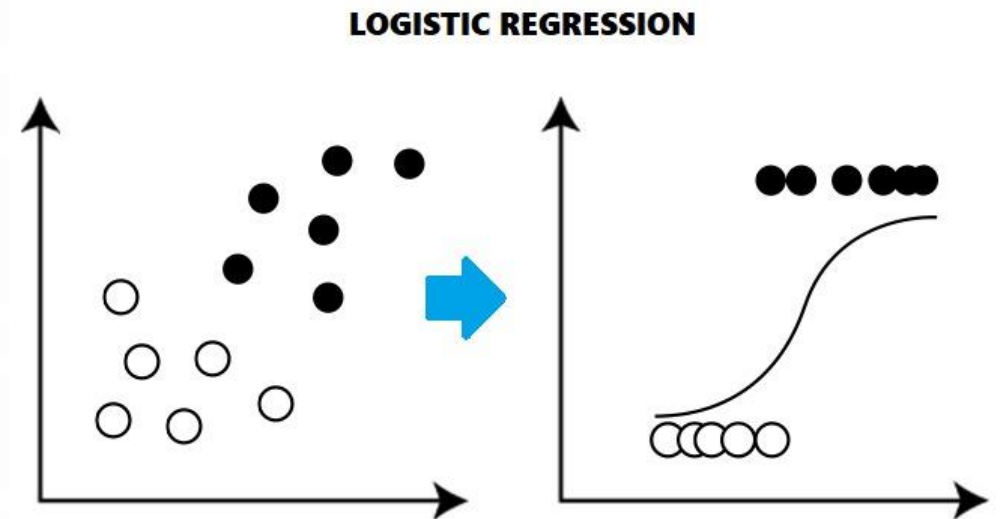
- Linear regression is a popular regression learning algorithm that learns a model which is a linear combination of features of the input example.
- The hyperplane in linear regression is chosen to be as close to all training examples as possible.



$$y = wx + b$$

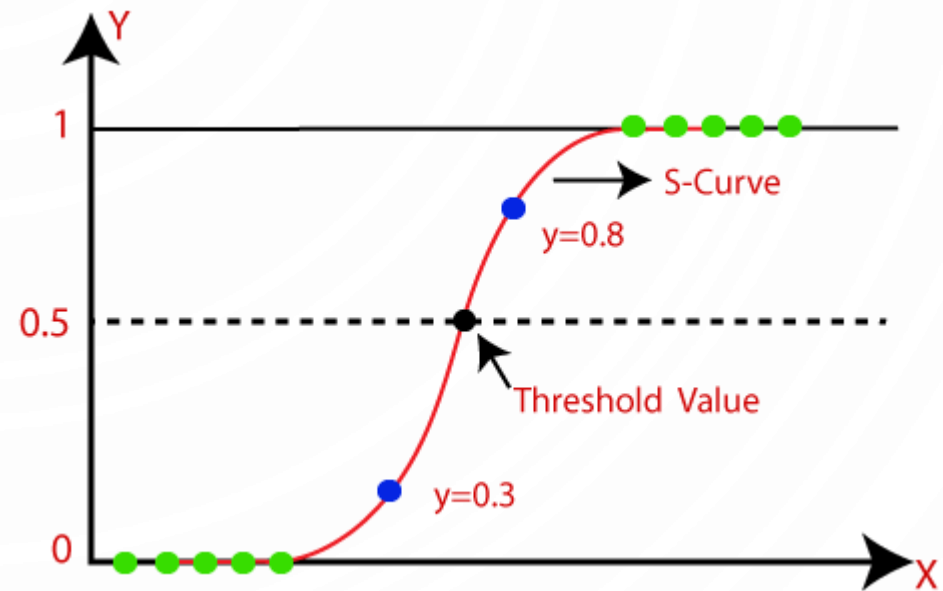
Logistic Regression

- Logistic regression predicts the output of a **categorical** dependent variable.
- Therefore, the outcome must be a categorical or discrete value.
 - Yes or No
 - 0 or 1
 - True or False
 - etc.
- Instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic regression is used for solving the **classification** problems.



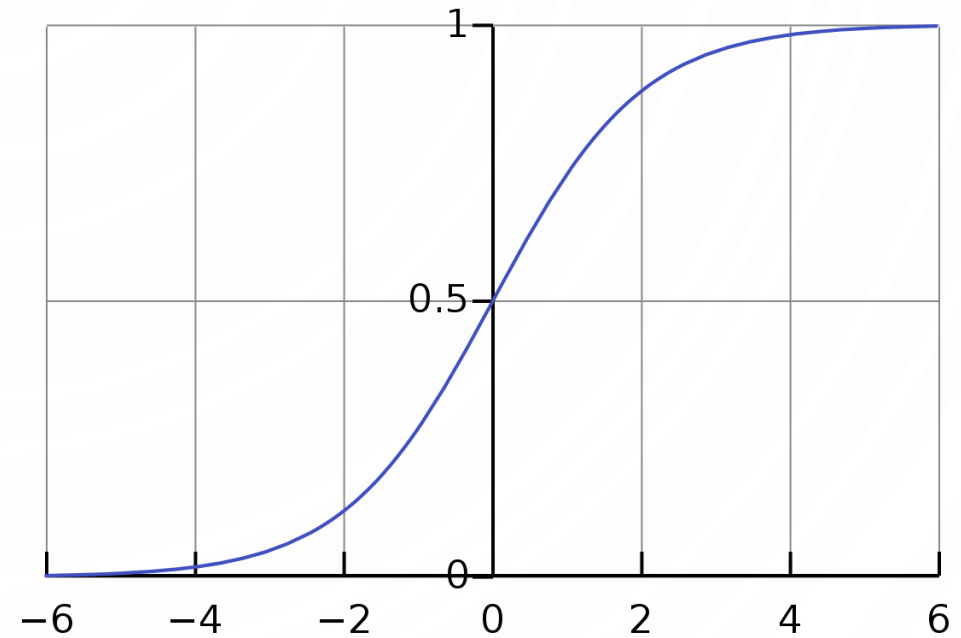
Logistic Regression

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function
 - 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 Function (Sigmoid Function)

$$f(x) = \frac{1}{1 + e^{-(x)}}$$



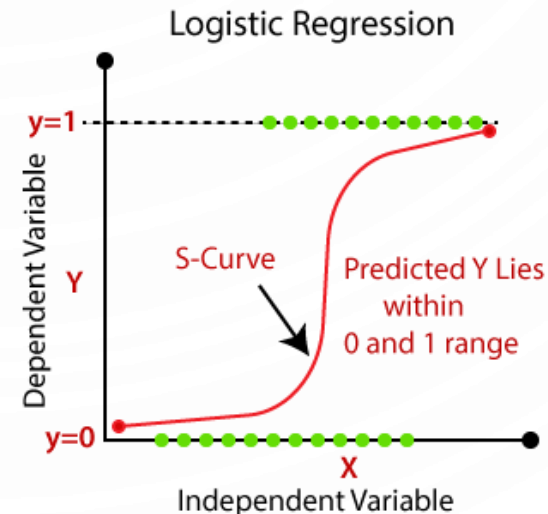
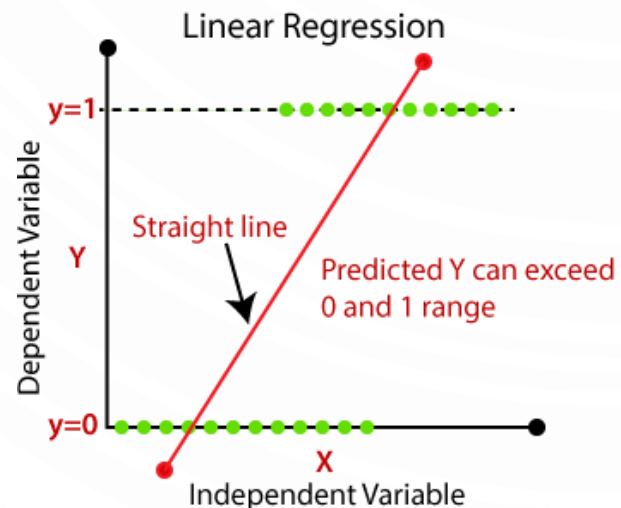
It maps any real value into another value within a range of 0 and 1

Logistic Regression Model

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

Logistic vs Linear Regression

- Both utilize a linear equation to arrive at predictions.
- In Linear regression, the result is continuous.
- In Logistic Regression, the outcome is a continuous number between the values of 0 and 1.



Likelihood Function

- In statistics, the likelihood function defines how likely the observation (an example) is according to our model.
- The optimization criterion in logistic regression is called **maximum likelihood**, we now maximize the likelihood of the training data according to our model:

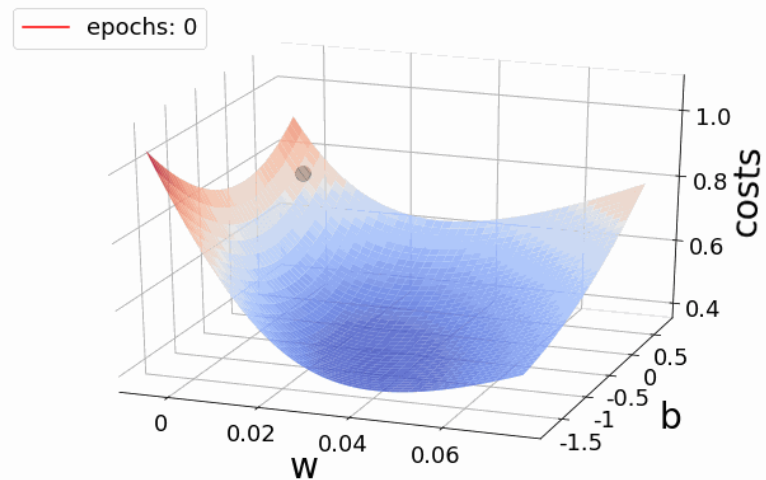
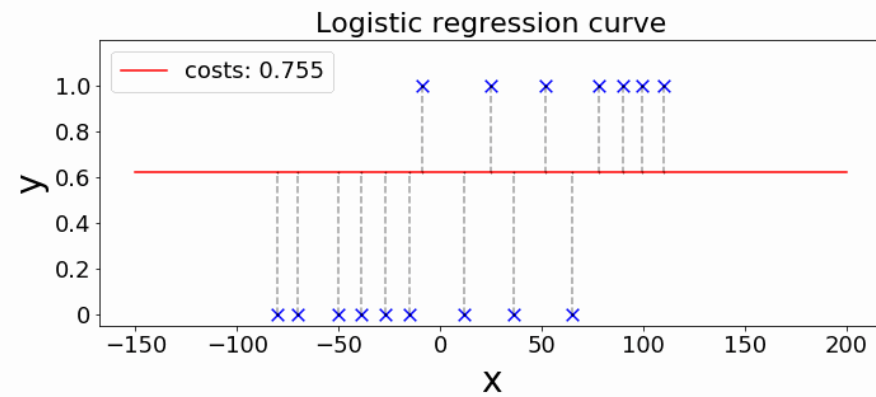
$$L_{w,b} = \prod_{i=1}^n \left(f_{w,b}(x_i) \right)^{y_i} \left(1 - f_{w,b}(x_i) \right)^{(1-y_i)}$$

$f_{w,b}(x_i)$ is the predicted likelihood
 y_i is the true value (1 or 0)

When $y_i=1$

When $y_i=0$

Parameters Learning



Classification Evaluation

- Many metrics can be used to evaluate the predictions for these problems
- Here are some:
 1. Classification Accuracy
 2. Confusion Matrix
 3. Precision, Recall, and F_1 score
 4. Area Under ROC Curve (AUC)





Classification Accuracy

- It is the number of **correct predictions made over all predictions made**
- This is only suitable when there is an equal number of observations in each class (balanced dataset) and all predictions and prediction errors are of equal importance
- The most common evaluation metric for classification problems



Logistic Regression

Python Cheatsheet

Data Preprocessing	Feature Engineering & EDA	Model Building	Model Evaluation
 <pre>import pandas as pd df.isnull() df.isnull().count() df.isnull().sum() df.drop() df.dropna() df.fillna()</pre>	 <pre>import matplotlib.pyplot df[<column>].plot() df[<column>].quantile(...) import LabelEncoder LabelEncoder().fit_transform() import seaborn df.corr() sns.heatmap()</pre>	 <pre>import train_test_split train_test_split(...) import LogisticRegression LogisticRegression(...) reg.fit(X_train, y_train) reg.predict(X_test) reg.predict_proba(X_test)</pre>	 <pre>import metrics metrics.plot_confusion_matrix() metrics.accuracy_score() metrics.roc_curve() metrics.roc_auc_score()</pre>

visit www.visual-design.net for step by step guide

Google Colab Example

- <https://colab.research.google.com/drive/1HweQRlgnm3SrO5TfZpumEjLm8xwaQAMw?usp=sharing>



Thank you