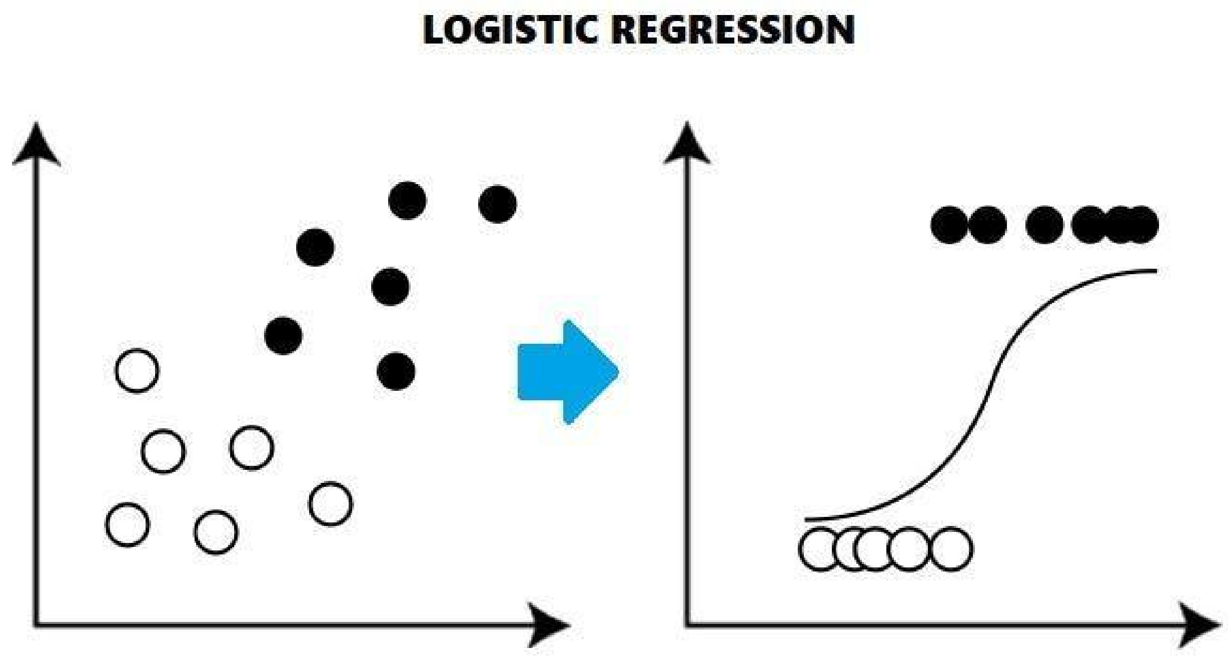
3- Supervised Classification Models - Logistic Regression

By: eng. Esraa Madhi

What is Classification?

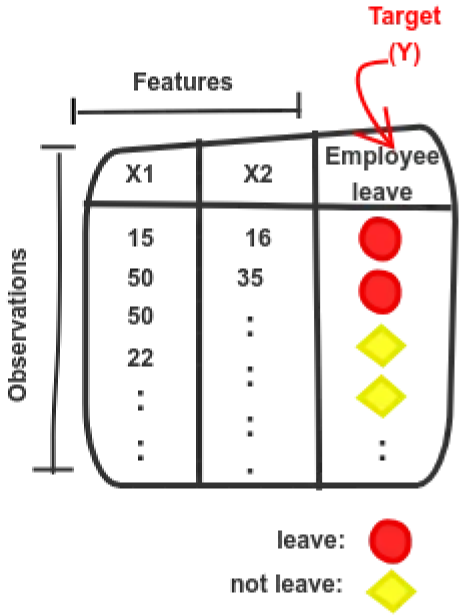
It is a type of supervised learning approach to predict the categorical labels or classes of data points.



Hours Studied (x)	Is Passed
1	0
2	1
3	1
4	0
5	1

Types of Classification

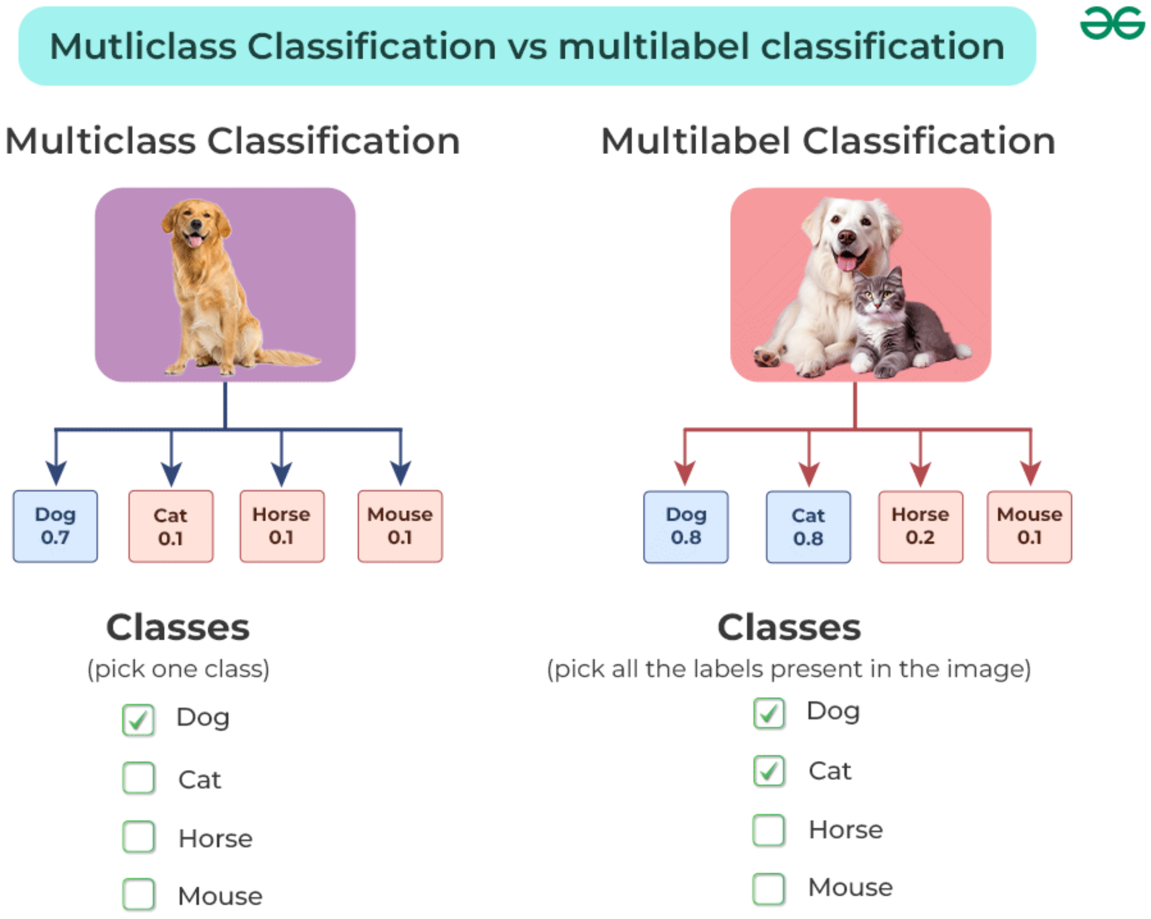
- **Binary Classification:** Involves two classes to differentiate between (e.g., yes/no, true/false).
- **Multiclass Classification:** Involves more than two classes (e.g., predicting types of fruits—apples, oranges, bananas).
- **Multilabel Classification**: Each data point can be assigned multiple classes (e.g., a news article that can be categorized into politics, economy, and education simultaneously).



X1	X2	Employee leave
15	16	●
50	35	●
50	:	◆
22	:	◆
:	:	:
:	.	

leave: ●

not leave: ◆


Mutliclass Classification vs multilabel classification
Multiclass Classification
Multilabel Classification
Dog 0.7
Cat 0.1
Horse 0.1
Mouse 0.1
Dog 0.8
Cat 0.8
Horse 0.2
Mouse 0.1
Classes
(pick one class)
☑ Dog
□ Cat
□ Horse
□ Mouse
Classes
(pick all the labels present in the image)
☑ Dog
☑ Cat
□ Horse
□ Mouse

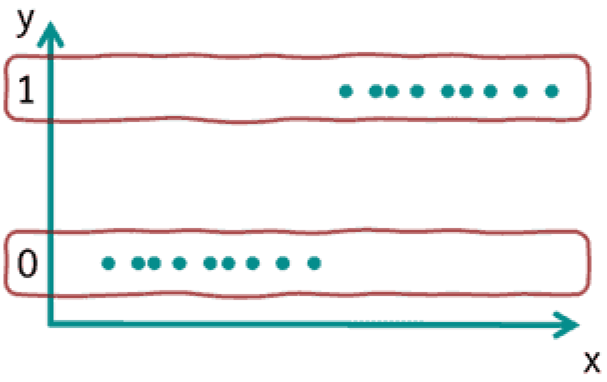
What is Logistic regression?

Logistic regression is a type of statistical analysis used for predicting the outcome of a categorical dependent variable based on one or more independent variables.

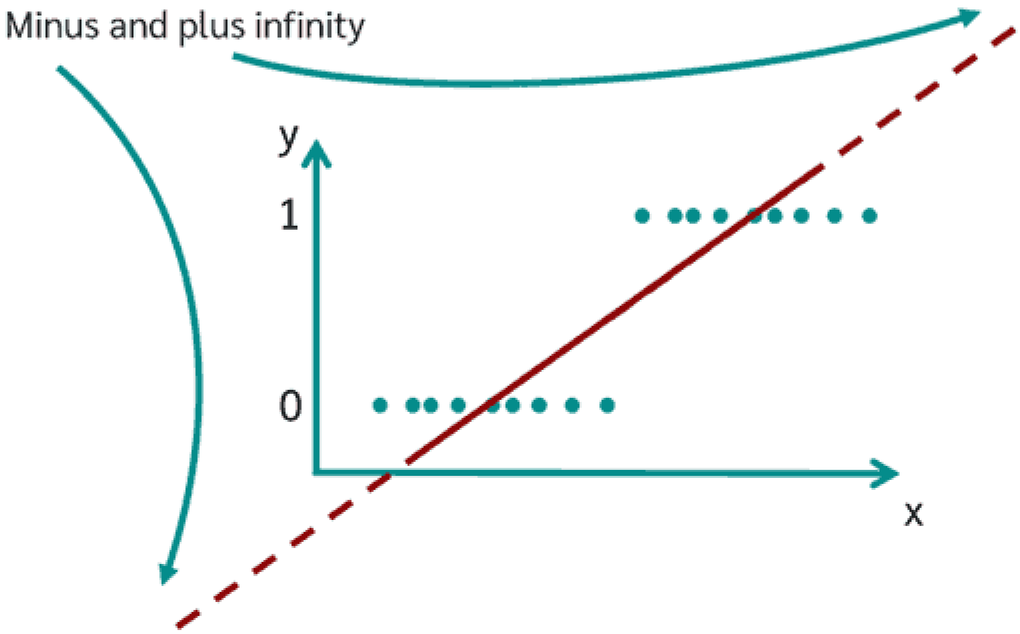
The goal of logistic regression is to find the best-fitting model (line) same as linear regression to describe the relationship between the dependent variable and the independent variable(s).

Hours Studied (x)	Is Passed
1	0
2	1

3	1
4	0
5	1



How to fit a line?



We don’t want continuous value as output value, we need the output variable is a probability (a value between 0 and 1) rather than a continuous value. Hence, the output is not just classes (e.g., 0 or 1), but the likelihood of belonging to a class.

The goal of logistic regression, however, is to **estimate the probability of occurrence and not the value of the variable itself.** Therefore, the **this the line equation must still be transformed.**

To do this, it is necessary to restrict the value range for the **prediction to the range between 0 and 1.** To ensure that only values between 0 and 1 are possible, wthe **logistic function *f*** is used.

So the model equation becomes:

$$y_i = b1x_i + b0$$

$$p(y = 1 \,|\, \mathbf{x}) = \frac{1}{1+e^{-y}}$$

$$p(y = 0 \,|\, \mathbf{x}) = 1 - p(y = 1 \,|\, \mathbf{x})$$

Where:

- $(p(y = 1 \,|\, \mathbf{x})$) is the probability that the dependent variable (y) equals 1 given predictors $(\mathbf{x})$.
- $(\mathbf{x})$ represents the vector of input features (predictors).
- $(\mathbf{b})$represents the vector of coefficients, including the intercept.
- (e) is the base of the natural logarithm.

In logistic regression, the probability of a certain class or event existing is modeled using a logistic function. The logistic function, also called the sigmoid function. Sigmoid function: It is an S-shaped curve function that maps any input to a value between 0 and 1.

To answer Check this example

https://youtu.be/EKm0spFxFG4

The logistic regression algorithm starts by **assuming a random set of weights** for the independent variables and then uses a method called maximum likelihood estimation to find the set of weights that best explain the data, which seeks to find the coefficients ((β)) that maximize the likelihood of the observed sample. This involves using optimization algorithms like gradient descent.

The log likelihood (Log Loss) (Cross-Entropy Loss)is defined as the probability of the data given the parameters (weights) of the model.

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} [y_i \log p(y_i = 1 \mid x_i) + (1 - y_i) \log(1 - p(y_i = 1 \mid x_i))]$$

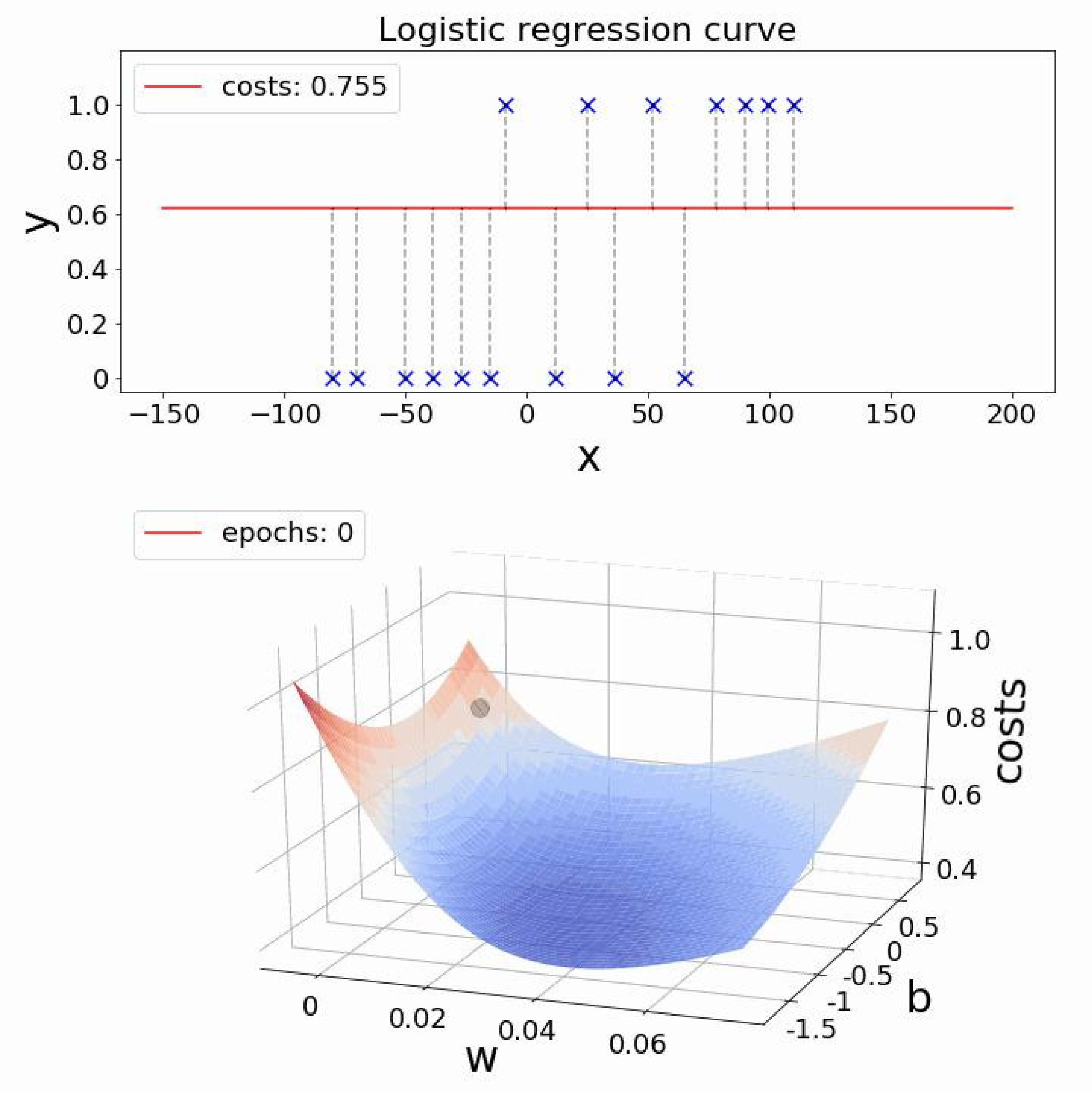
where :

- y is the true dependent variable
- p(x) is the probability of y = 1 given x.

https://mlu-explain.github.io/logistic-regression/

Logistic Regression • mlu-explain.github.io

the gradient descent algorithm is often used to find the local optimal solution.



The algorithm iteratively adjusts the weights to maximize the likelihood of the data. Once the model is trained, it can be used to predict the probability of a certain class or event for new data. **The predicted probability can then be transformed into a binary prediction (e.g. class 1 or class 0) by applying a threshold value, usually 0.5.**

Example: Studied Hours vs. Test Result

Hours Studied	Passed Test
1	0
2	0
3	0
4	1
5	1
6	1
7	1
8	1

Step-by-Step Logistic Regression:

1. Fit the logistic regression model to the data. Assume coefficients were found to be ($\beta_0 = -4$) and ($\beta_1 = 1$).
   - Line equation :

$$y_i = \beta_0 + \beta_1 \times x_1$$

$$y_i = \beta_0 + \beta_1 \times \text{Hours Studied}$$

   - Model:

$$\text{Probability of Passing} = \frac{1}{1+e^{-y_i}}$$

$$\text{Probability of Passing} = \frac{1}{1+e^{-(\beta_0+\beta_1 \times \text{Hours Studied})}}$$

   a. For a student who studies 4 hours, the probability of passing is:

$$p(4) = \frac{1}{1+e^{-(-4+1\times 4)}} = \frac{1}{1+e^{0}} = 0.5$$

This model suggests that studying for 4 hours leads to a 50% chance of passing, depending on the fitted model coefficients.

$(p(y = 1 | x = 1) = \frac{1}{1+e^{-(-3)}} \approx 0.0474)$
$(p(y = 1 | x = 2) = \frac{1}{1+e^{-(-2)}} \approx 0.1192)$
$(p(y = 1 | x = 3) = \frac{1}{1+e^{-(-1)}} \approx 0.2689)$
$(p(y = 1 | x = 4) = \frac{1}{1+e^{0}} \approx 0.5)$
$(p(y = 1 | x = 5) = \frac{1}{1+e^{1}} \approx 0.7311)$
$(p(y = 1 | x = 6) = \frac{1}{1+e^{2}} \approx 0.8808)$
$(p(y = 1 | x = 7) = \frac{1}{1+e^{3}} \approx 0.9526)$
$(p(y = 1 | x = 8) = \frac{1}{1+e^{4}} \approx 0.9820)$

3. Gradient descent is an optimization algorithm commonly used to find the minimum (or maximum, in the case of logistic regression where we typically maximize the likelihood) of a function. In the context of logistic regression, the function we want to maximize is the likelihood function, or more commonly, the log-likelihood function because it simplifies calculations and numerically stabilizes them due to the properties of logarithms.

   $$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} [y_i \log p(y_i = 1 | x_i) + (1 - y_i) \log(1 - p(y_i = 1 | x_i))]$$

   For each parameter (β_j), the gradient component is:

   $$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} [y_i - p(y_i = 1 | x_i)] x_{ij}$$

   Where$(x_{ij})$is the value of feature (j) for observation (i).

   For our simple model with one predictor and an intercept:

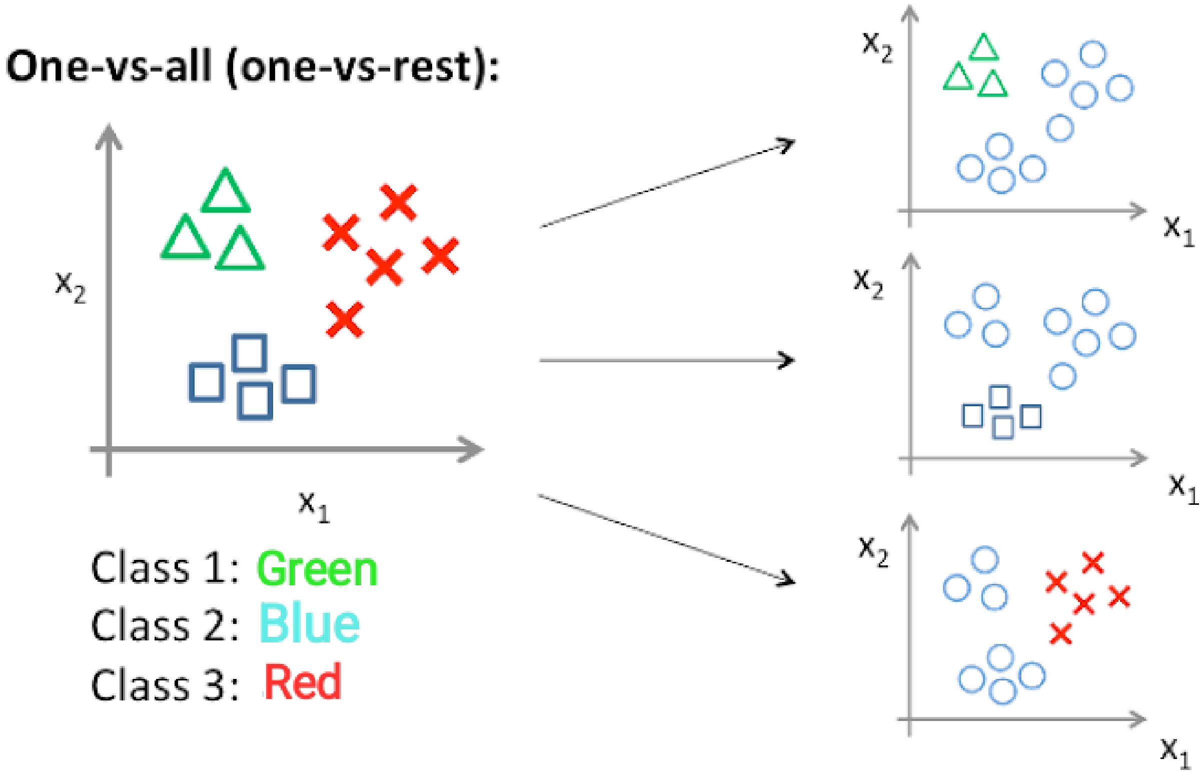
   - $(\frac{\partial \ell}{\partial \beta_0} = \sum_{i=1}^{n} [y_i - p(y_i = 1 | x_i)])$
   - $(\frac{\partial \ell}{\partial \beta_1} = \sum_{i=1}^{n} [y_i - p(y_i = 1 | x_i)] x_i)$

5. **Repeat** steps 2-3 until convergence, i.e., until the changes in the parameters are below a certain small threshold, or for a set number of iterations.

Logistic Regression for Multi-Class Classification

Logistic regression is traditionally used for binary classification problems, but it can be extended to handle multi-class classification scenarios using techniques like "one-vs-rest" (OvR) and "multinomial logistic regression" (also known as "softmax regression"). Let's explore how logistic regression can be adapted to these situations:

- In the one-vs-rest (OvR) approach, a separate binary logistic regression classifier is trained for each class to predict the probability of that class versus all other classes. Here's how it works:
    1. **Binary Classifiers**: Suppose you have three classes: A, B, and C. You would create three binary logistic regression models:
        - Model 1 (Model A): Class A vs. Classes B and C
        - Model 2 (Model B) : Class B vs. Classes A and C
        - Model 3 (Model C) : Class C vs. Classes A and B
    2. **Training:** Each model is trained independently on all the data, where the target variable for each model is whether each instance belongs to the respective class (positive) or not (negative).
    3. **Prediction**: For a new instance, each model gives a probability that the instance belongs to its respective class. The final class prediction is the class whose model gives the highest probability.



- Multinomial Logistic Regression (Softmax Regression)

Multinomial logistic regression is a direct extension of binary logistic regression to multi-class problems, without needing to train multiple separate models. It uses the softmax function to handle multiple classes.

Resources:

- https://www.natasshaselvaraj.com/logistic-regression-explained-in-7-minutes/
- https://medium.com/@satyarepala/understanding-logistic-regression-a-step-by-step-explanation-9a404344964b
- https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/