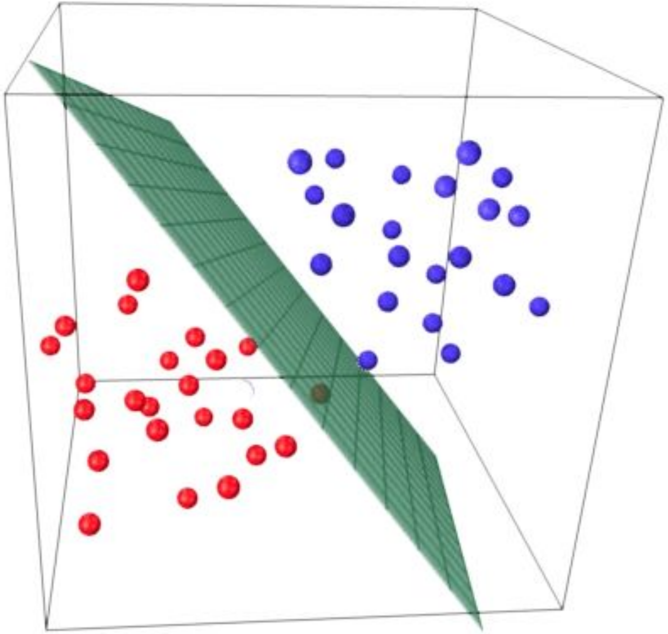
Logistic Regression

Quick Overview

Logistic Regression is a popular **classification** algorithm and is used to predict **binary outcomes.**

In contrast to Linear Regression, where it predicts continuous values, Logistic Regression predicts binary values **(0's or 1's)**.



-

Why Logistic Regression?

Understanding this algorithm is **important** because it’s used everywhere and:

- It’s fast
- Easy to interpret
- And allows you to predict probability outcomes!

Simple Example

Logistic Regression allows you to make:

- Soft Predictions (% Probability)
- Hard Predictions (0’s, 1’s)

Predicting Credit Default:

Name	Predicted Default Probability	
Bob	0.85	
Henry	0.76	
Mary	0.53	
Paul	0.32	Threshold: 0.15
Maria	0.14	
Sonny	0.12	
Ryan	0.07	
Howard	0.04	

Custom Threshold

Pre-Requisites

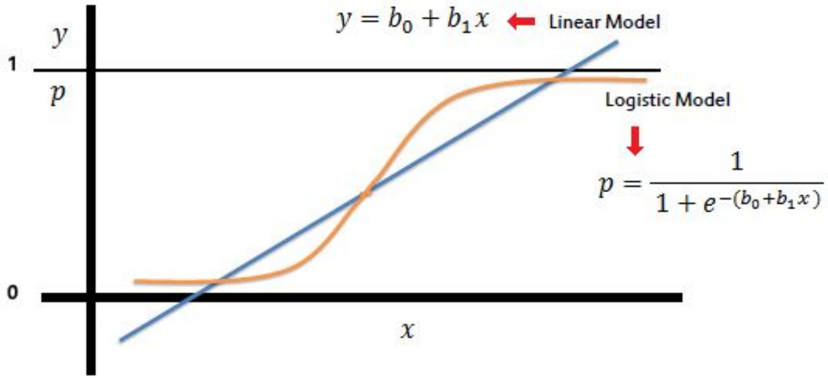
Before we dive in deep to understand how Logistic Regression works, here are some important concepts you should be familiar with:

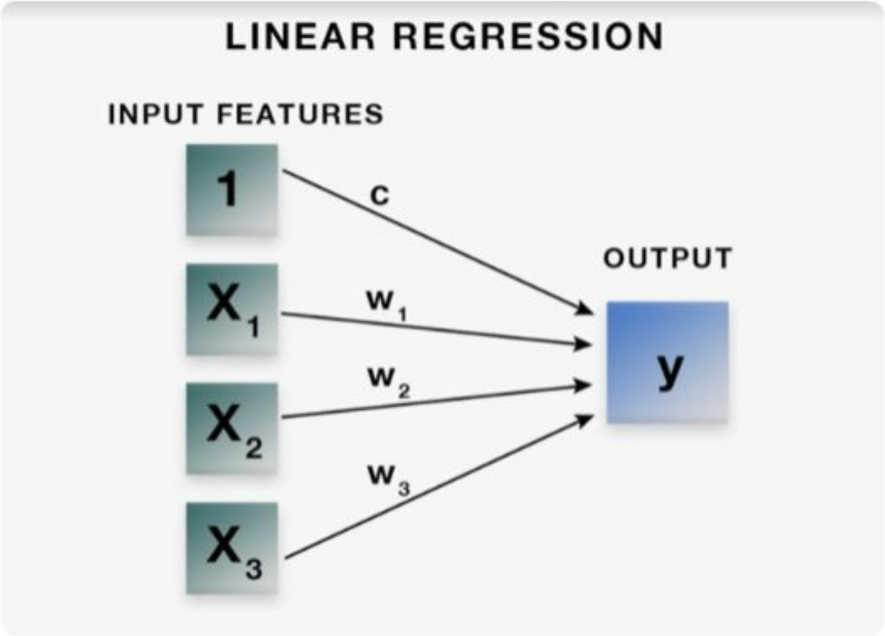
1. Probability
2. Odds
3. Logit
4. Log Odds
5. Log Loss
6. Maximum Likelihood Estimation (MLE)

Linear Regression V.S. Logistic Regression

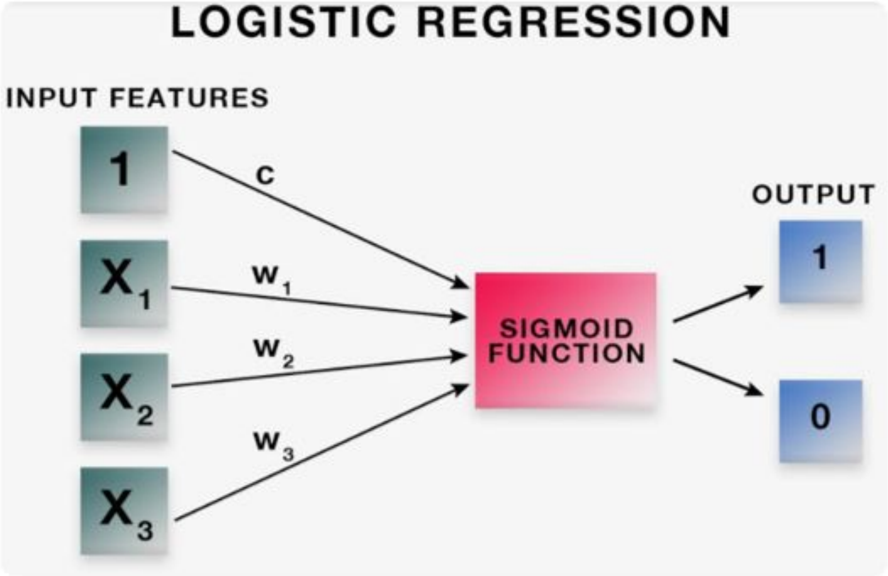
- Uses **Logit Transformation** on DV to fit a regression model
- DV has two (binary) outcomes

$$Ln\left(\frac{P}{1-P}\right)=\beta_0+\beta_1X_1+\beta_2X_2+\ldots+\beta_kX_k$$





$$y = c + x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + \ldots\ldots + x_n * w_n$$



$y = \text{logistic}\,(c + x_1{*}w_1 + x_2{*}w_2 + x_3{*}w_3 + \ldots\ldots.. + x_n{*}w_n)$

$y = 1 / 1 + e\,[-\,(c + x_1{*}w_1 + x_2{*}w_2 + x_3{*}w_3 + \ldots\ldots.. + x_n{*}w_n)]$

Understanding the Algorithm

- Models the **PROBABILITY** of an event (rather than a measure)
- Probabilities range from (0’s to 1’s)
- Requires a **LOGIT** transformation on DV (hence the name Logistic Regression)
- Still considered a Linear Model because of Input Parameters

The Logistic Function

$$\text{Log}\left[\frac{Y}{(1-Y)}\right] = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_nX_n$$

Log(Likelihood)

diet score (0-15)

age group (0/1)

sex (0/1)

Log Odds

$$Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

Logistic Regression models the **Log Odds** of an event

- **Odds**: p / 1-p, where p is the probability of positive class

Logit Regression Equation:

- Log(Odds) = B0 + B1X1 + B2X2 +... + BkXk
- Similar to Linear Regression

Why Not: P = B0 + B1X1 + B2X2 +... + BkXk?

- P ranges from 0 to 1, and Log Odds ranges from -inf to inf

$$\ln\left(\frac{P}{1-P}\right) = a + bX$$

$$\frac{P}{1-P} = e^{a+bX}$$

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Log Loss Function

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Objective: Minimize the Log Loss Error

Logistic Regression uses the Log Loss Function (as opposed to OLS) to find optimal parameters

- **y(i)**: the DV (0 or 1)
- **h0(xi)**: predicted probability

When y=0 and probability is **LOW** → then low error

When y=1 and probability is **HIGH** → then low error

Strengths and Limitations

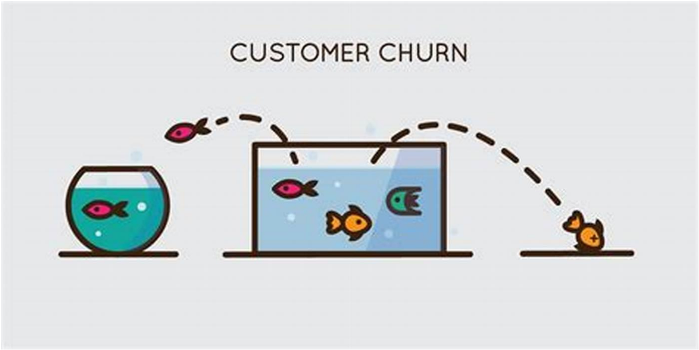
Strengths:

- Highly interpretable
- Fast training & predictions
- No model tuning required (except for regularization)
- No need to scale features
- Outputs probability scores
- Good baseline model

Limitations:

- Assumes a linear relationship
- Performance is not as great compared to other models
- Sensitive to outliers
- Can’t automatically learn feature interactions like tree-based methods

Common Use Case



Practical Considerations

- Definitely use Logistic Regression as your first baseline model for model comparison

- Make sure the assumptions are met when trying to interpret the coefficients

-