Linear and Logistic Regression

Linear Regression:

Linear regression is a supervised machine learning algorithm that uses labeled data to learn patterns and fit the best possible straight line through the data points. This line is then used to make predictions on new data. It assumes that the relationship between input and output is linear, meaning the output changes at a steady rate as the input changes.

For example, if we want to predict a student's exam score based on study hours:

- Independent variable (input i.e X): Hours studied (the factor we control/observe)
- Dependent variable (output i.e y): Exam score (depends on study hours) i.e. more hours studied → higher exam scores and vice-versa.
- Linear regression finds the best-fit line to make predictions

We use the input (hours studied) to predict the output (exam score).

There are 2 types of Linear regression algorithms based upon number of predictor variables:

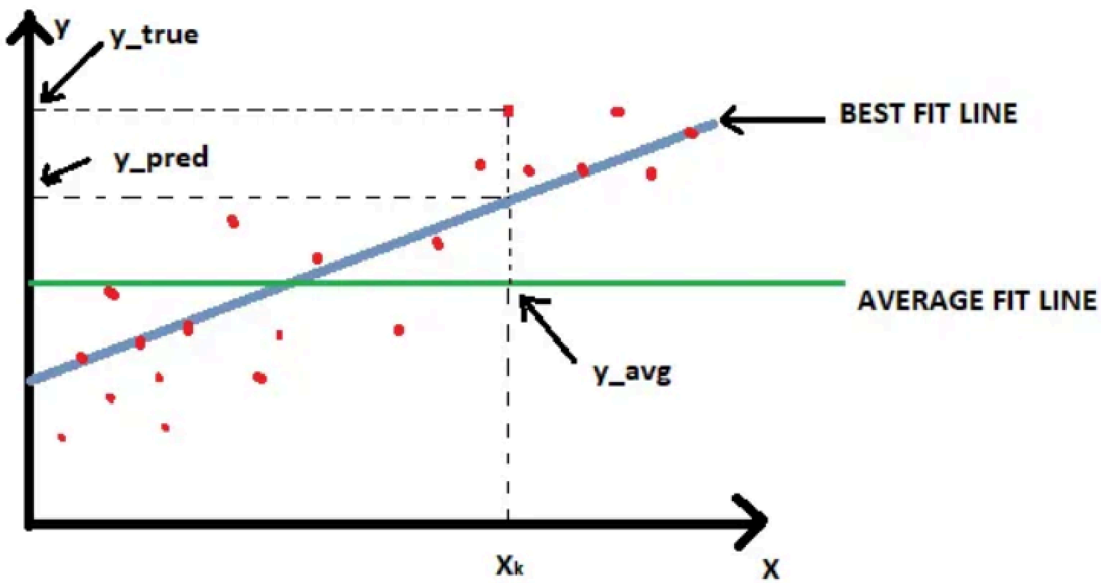
1. Simple Linear Regression: Only one predictor variable is used to predict the values of dependent variables.

Equation of the line: **y = mX + c.**

Where, y : dependent variable
X: predictor variable
m: slope of the line defining relationship between X and y, also called co-efficient of X (change in y with each unit change in X)
c: intercept (value of y when X=0)

2. Multiple Linear Regression: More than one predictor variable is used to predict the values of dependent variables.

Equation of the line: **y = c + m1x1+ m2x2 + m3x3 … + mixi**
(many predictor variables x1, x2, … xi).
(m1, m2,… mi are respective co-efficients)



Typical Linear Regression plot

This figure explains the main ideas of linear regression:

- Data Points (red dots): These are the actual observations in your dataset.
- Best Fit Line (blue line): This regression line minimizes the distance to all data points and shows the relationship between X and y.
- Average Line (green line): A horizontal line representing the mean of all y values (y_avg). This is what you'd predict if you ignored X.

Key Elements:
y_true: Actual observed values
y_pred: Predicted values from the regression line
y_avg: Average of all y values
Residuals (vertical dashed lines): The difference between predicted and actual values at a specific point (Xk), showing prediction errors.

In summary, the blue "best fit" line predicts y much better than simply using the average. The closer the points are to this line, the better the model. This figure shows how linear regression captures the true relationship between the variables.

Goal: Find the best-fit line for **X** and **y** that gives optimal values for slope (**m**) and intercept (**c**).

Residual (Error):

Residual = ytrue − ypred

Residual Sum of Squares (RSS): y_true - y_pred gives us the error term associated with Xk, also called residual. These error terms can be positive or negative based upon the y_true value. So we take the square of these residual terms, in order to avoid negative signs. The sum of all such residuals is called Residual Sum of Squares (RSS). This forms our cost function. We need to minimize this cost function in order to get optimal values for slope(m) and intercept(c) for our linear regression line.

$$RSS = \sum_{i=1}^{n} (y_{pred} - y_{avg})2$$

Residual Sum of Squares equation

Explained Sum of Squares (ESS): Measures variation explained by the regression line compared to the average.

$$ESS = \sum_{i=1}^{n} (y_{pred} - y_{avg})2$$

Explained Sum of Squares equation

Total Sum of Squares (TSS): Measures total variation in observed values compared to the average.

$$TSS = \sum_{i=1}^{n} (y_{true} - y_{avg})2$$

Total Sum of Squares equation

From the above equations and the figure, we can verify that:

TSS=ESS+RSS

This shows that the total variation in the data is split into the part explained by the model (ESS) and the part left as error (RSS).

To find the best-fit line, we need to minimize the cost function (RSS).

Two common methods to do this:

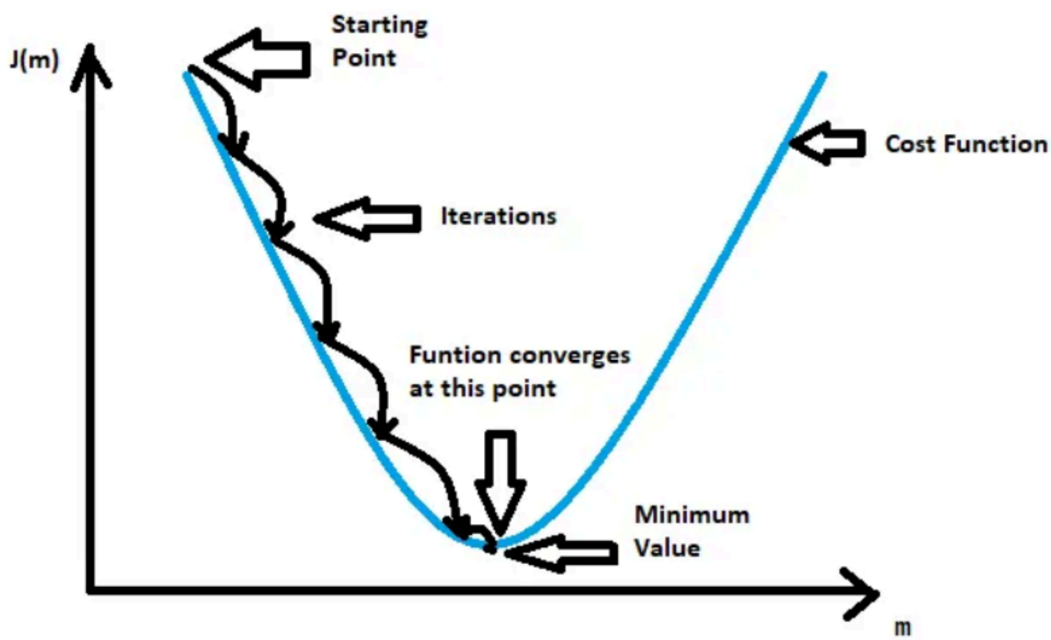
1. Closed-Form Solution:

- Take the derivative of the cost function, set it to 0 to find the minimum.
- Works well for small datasets.
- Becomes complex with multidimensional data.

2. Gradient Descent:

- Iterative approach to gradually reach the minimum of the cost function.
- Efficient and simpler for large datasets.
- Runs multiple iterations until the cost function is minimized.

Gradient Descent updates the slope (m) and intercept (c) step by step to reduce prediction errors, eventually finding the optimal best-fit line.



Gradient Descent method

Gradient Descent and Finding the Best-Fit Line

- The goal is to minimize the cost function J(m)J(m)J(m) (RSS) to find the optimal slope (m) and intercept (c).
- Iterations: Gradient Descent updates parameters step by step until the function converges (reaches minimum).
- Learning Rate: Controls the step size toward the minimum.
  Too large → may overshoot the minimum.
  Too small → slower convergence but safer
- Update Rule:

$$m_1 = m_0 - (\text{learning rate}) \cdot \frac{dJ}{dm}$$

  Where, m0 = current slope

  m1 = updated slope

- Cost function:

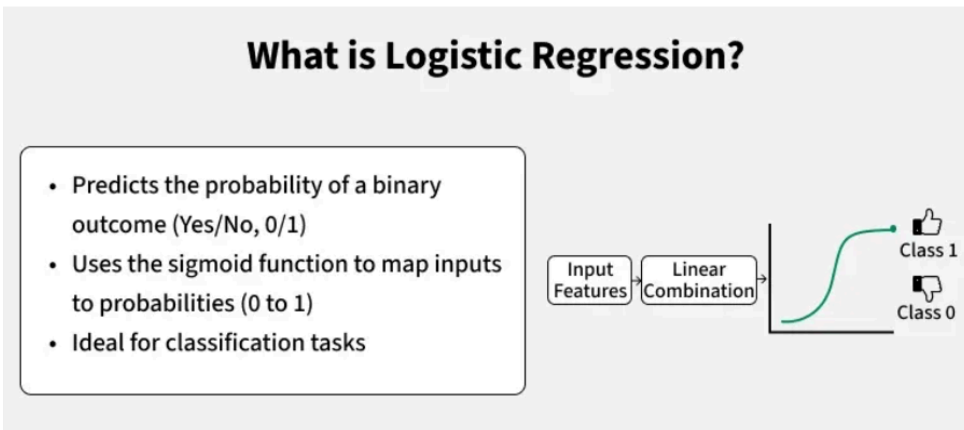
$$RSS = (y_{\text{pred}} - y_{\text{true}})^2$$

- With ypred = c + mx, we have two unknowns: m and c
- Take partial derivatives w.r.t m and c, set to 0 → solve for optimal values
- Outcome: When Gradient Descent converges, RSS is minimized
  The resulting line y = mx + c is the best-fit line.

Finally, linear regression involves some mathematics, but the main idea is to find the line that best fits the data by minimizing prediction errors.

Logistic Regression:

Logistic Regression is a supervised machine learning algorithm used for classification tasks. Unlike linear regression, which predicts continuous values, logistic regression predicts the probability that a given input belongs to a specific class. It is mainly used for binary classification, where the output can take one of two possible values, such as Yes/No, True/False, or 0/1. The algorithm applies the sigmoid function to convert input values into probabilities between 0 and 1. This allows us to classify inputs based on their likelihood of belonging to a particular category, making logistic regression a fundamental tool for understanding and solving classification problems.



Example: Predicting Whether a Student Passes an Exam

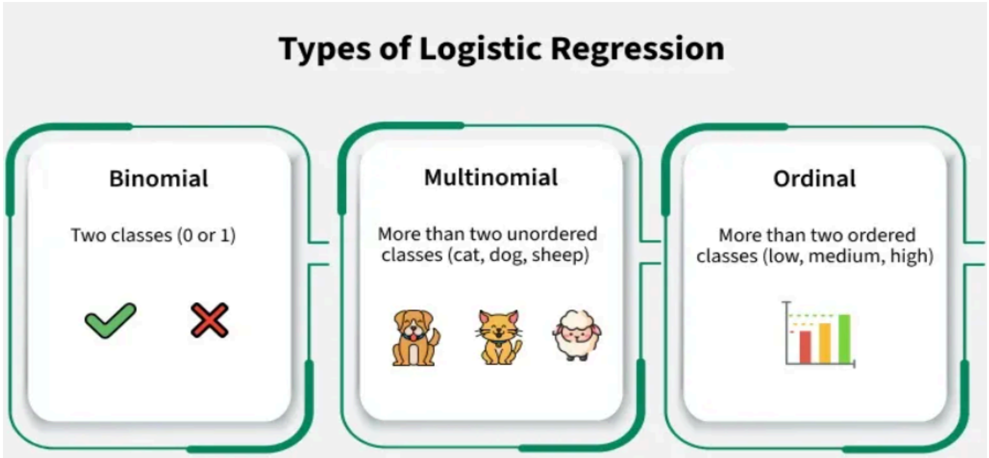
- Problem: We want to predict if a student passes (1) or fails (0) an exam based on the number of hours they studied.
- Input (Independent Variable X): Hours studied
- Output (Dependent Variable y): Pass (1) / Fail (0)

There are three main types of logistic regression based on the dependent variable:

1. Binomial Logistic Regression: Used when the dependent variable has two categories (e.g., Yes/No, Pass/Fail). This is the most common form.

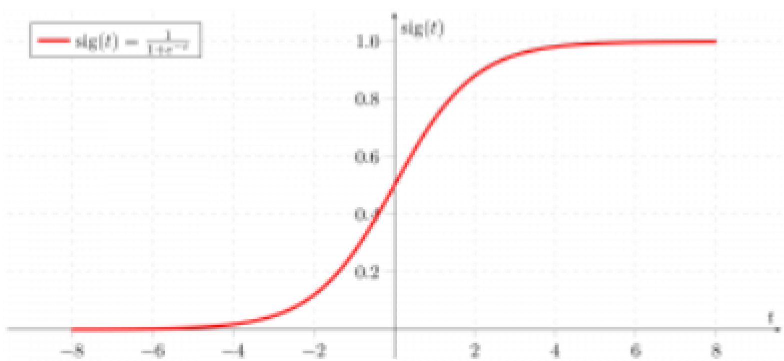
2. Multinomial Logistic Regression: Used when the dependent variable has three or more categories without a natural order (e.g., classifying animals as "cat," "dog," or "sheep").

3. Ordinal Logistic Regression: Used when the dependent variable has three or more ordered categories (e.g., ratings like "low," "medium," and "high"), taking the order into account in the model.



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.

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Sigmoid function

1. Logistic Regression Model

- Logistic regression predicts the probability that an input X belongs to a class (0 or 1).
- Instead of predicting Y directly like linear regression, it predicts probability P(Y = 1 | X)

$$P(Y = 1|X) = \sigma(z)$$

Where, b0 = intercept, b1, b2, ..., bn = weights
σ(z) = sigmoid function

2. Sigmoid Function: The sigmoid function maps any real number to a value between 0 and 1.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Output close to 1 → high probability of class 1
Output close to 0 → high probability of class 0

- $\sigma(z)$ tends towards 1 as $z \to \infty$
- $\sigma(z)$ tends towards 0 as $z \to -\infty$
- $\sigma(z)$ is always bounded between 0 and 1

where the probability of being a class can be measured as:

$P(y = 1) = \sigma(z)$

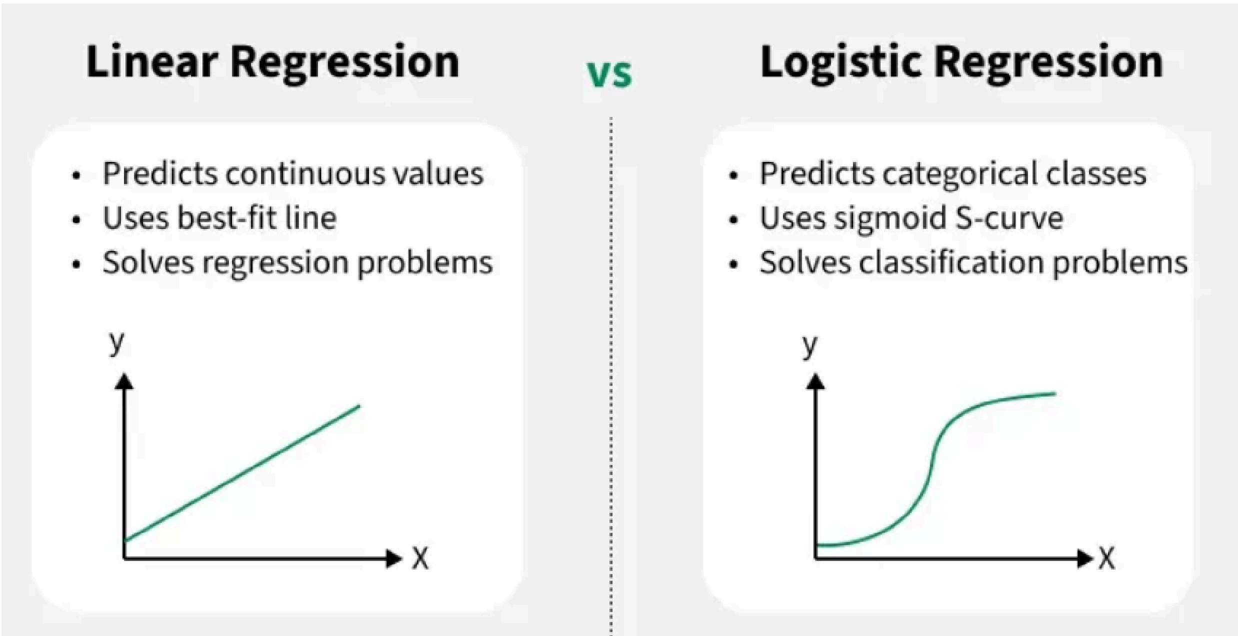
$P(y = 0) = 1 - \sigma(z)$

Optimization:

- Parameters are learned using Gradient Descent or other optimization methods.
- Gradients are calculated using partial derivatives of the cost function with respect to each parameter.
- Iteratively update weights until the cost function reaches minimum.

In summary, logistic regression maps inputs to probabilities using the sigmoid function, models log-odds linearly, and finds optimal weights by minimizing the log loss.

Comparison between Linear and Logistic Regression:



Linear regression is used to predict a continuous dependent variable based on a set of independent variables, making it suitable for regression problems. It estimates values such as price, age, or salary and relies on a linear relationship between the dependent and independent variables. The goal is to find the best-fit line, and accuracy is usually estimated using the least squares method. Collinearity among independent variables is possible.

On the other hand, logistic regression is used to predict a categorical dependent variable, making it ideal for classification problems. It predicts outcomes such as 0/1, Yes/No, or other categories, using the sigmoid (S-curve) function. Logistic regression does not require a linear relationship between variables, and its accuracy is estimated using maximum likelihood estimation. Additionally, it works best when there is little to no collinearity among independent variables.