OTDM UNIT III

Dr Ravi Prakash Shahi

ravishahi71@gmail.com

Two things to remember in life:

Take care of your thoughts when you are alone, and take care of your words when you are with people.

Newton's Method in Optimization

Newton's Method (or the Newton–Raphson method) is a **second-order iterative optimization technique** used to find **stationary points** (minima, maxima, or saddle points) of a real-valued differentiable function $f(x)$.

It extends the 1D Newton–Raphson root-finding method to optimization problems by finding where the **gradient** (first derivative) becomes zero.

Objective

We want to find x^* such that:

$$\nabla f(x^*) = 0$$

where

- $\nabla f(x)$ = gradient vector of $f(x)$,
- $\nabla^2 f(x)$ = Hessian matrix (matrix of second derivatives).

Newton's Method Algorithm (Unconstrained Optimization)

1. **Initialize:** Choose a starting point x_0.
2. **Compute gradient:** $g_k = \nabla f(x_k)$.
3. **Compute Hessian:** $H_k = \nabla^2 f(x_k)$.
4. **Compute search direction:** $d_k = -H_k^{-1} g_k$.
5. **Update:** $x_{k+1} = x_k + d_k$.
6. **Check convergence:** If $||g_{k+1}|| < \epsilon$, stop; else repeat.

Example 1 on Newton's Method – Minimize f(x) = x^2 - 4x +4

Step 1: Compute derivatives

$$f'(x) = 2x - 4, \quad f''(x) = 2$$

Step 2: Newton's update

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{2x_k - 4}{2} = x_k - (x_k - 2) = 2$$

Step 3: Convergence

Regardless of the starting point, $x_{k+1} = 2$ immediately.

Hence, the minimum is at $x = 2$, and $f(2) = 0$.

Disadvantages of Newton’s Method

Disadvantage	Explanation
1. Requires Hessian computation	The Hessian ((n \times n) matrix) must be computed and inverted — expensive for large (n).
2. May not converge	If the Hessian is not positive definite (saddle point or maximum), the step can move away from minimum.
3. Sensitive to initial guess	Poor starting point can lead to divergence or convergence to the wrong stationary point.
4. High computational cost	Computing and inverting the Hessian costs (O(n^3)).
5. Not suitable for non-smooth functions	Requires continuous second derivatives.
6. Step may overshoot	If the step size is too large, the quadratic approximation fails — often a line search or damping factor is added.

Newton method

Question: Minimize $f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$ by taking the starting Point as $X_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$

Sol. To find X_2.

$$[J_1] = \begin{bmatrix} \frac{d^2f}{dx_1^2} & \frac{d^2f}{dx_1 dx_2} \\ \frac{d^2f}{dx_2 dx_1} & \frac{d^2f}{dx_2^2} \end{bmatrix}$$

$$\frac{df}{dx_1} = 1 + 4x_1 + 2x_2$$

$$\frac{d^2f}{dx_1^2} = 4$$

$$\frac{d^2f}{dx_1 dx_2} = 2$$

$$\frac{df}{dx_2} = -1 + 2x_1 + 2x_2$$

$$\frac{d^2f}{dx_2 dx_1} = 2$$

$$\frac{d^2f}{dx_2^2} = 2$$

$$[J_1] = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

$$[J_1]^{-1} = \frac{1}{4 \times 2 - 2 \times 2} \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix}$$

$$[J_1]^{-1} = \frac{1}{4} \begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cdot & \cdot \\ -2 & 4 \end{bmatrix}\begin{bmatrix} \cdot & \cdot \\ -1/2 & 1 \end{bmatrix}$$

$$g_1 = \begin{bmatrix} df/dx_1 \\ df/dx_2 \end{bmatrix}_{X_1} = \begin{Bmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{Bmatrix}\begin{matrix} 0 \to x_1 \\ 0 \to x_2 \end{matrix} = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$

$$\therefore \quad X_2 = X_1 - [J_1]^{-1} g_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} - \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1/2 \times 1 + (-1/2) \times (-1) \\ -1/2 \times 1 + 1 \times (-1) \end{bmatrix}$$

$$X_2 = \begin{Bmatrix} -1 \\ 3/2 \end{Bmatrix}$$

$$g_2 = \begin{Bmatrix} df/dx_1 \\ df/dx_2 \end{Bmatrix}_{X_2} = \begin{Bmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{Bmatrix}\begin{matrix} -1 \to x_1 \\ 3/2 \to x_2 \end{matrix} \Rightarrow g_2 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

$$X_3 = X_2 - [J_1]^{-1} g_2$$

$$[\;\cdot\;]\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$X_3 = X_2 - \begin{bmatrix} 0 \\ 0 \end{bmatrix}_{g_1 =}$$

Example: K-Means (Solved Problem)

Data points: (2,10), (2,5), (8,4), (5,8), (7,5), (6,4)

Let K=2 , initial centroids = (2,10) and (5,8)

Data points (2D):

$$P_1 = (2, 10),\ P_2 = (2, 5),\ P_3 = (8, 4),\ P_4 = (5, 8),\ P_5 = (7, 5),\ P_6 = (6, 4).$$

Number of clusters $K = 2$.

Initial centroids chosen:

$\mu_1^{(0)} = (2, 10)$ and $\mu_2^{(0)} = (5, 8)$.

K-Means repeats: (A) assign each point to nearest centroid, (B) recompute centroids as cluster means, until assignments stop changing.

Iteration 1 — Assignment step (distances to initial centroids)

We use **Euclidean distance** $d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$. (Only relative comparisons matter; I show squared distances to avoid unnecessary square roots.)

Distances to $\mu_1^{(0)} = (2, 10)$

- $P_1 = (2,10)$: $d^2 = (2-2)^2 + (10-10)^2 = 0$
- $P_2 = (2,5)$: $d^2 = (2-2)^2 + (5-10)^2 = 25$
- $P_3 = (8,4)$: $d^2 = (8-2)^2 + (4-10)^2 = 36 + 36 = 72$
- $P_4 = (5,8)$: $d^2 = (5-2)^2 + (8-10)^2 = 9 + 4 = 13$
- $P_5 = (7,5)$: $d^2 = (7-2)^2 + (5-10)^2 = 25 + 25 = 50$
- $P_6 = (6,4)$: $d^2 = (6-2)^2 + (4-10)^2 = 16 + 36 = 52$

Distances to $\mu_2^{(0)} = (5, 8)$

- P_1: $d^2 = (2-5)^2 + (10-8)^2 = 9 + 4 = 13$
- P_2: $d^2 = (2-5)^2 + (5-8)^2 = 9 + 9 = 18$
- P_3: $d^2 = (8-5)^2 + (4-8)^2 = 9 + 16 = 25$
- P_4: $d^2 = (5-5)^2 + (8-8)^2 = 0$
- P_5: $d^2 = (7-5)^2 + (5-8)^2 = 4 + 9 = 13$
- P_6: $d^2 = (6-5)^2 + (4-8)^2 = 1 + 16 = 17$

Assign each point to the closer centroid (compare squared distances)

- P_1: to μ_1 (0 vs 13) → **Cluster 1**
- P_2: to μ_1 (25 vs 18) → **Cluster 2** (18 smaller)
- P_3: to μ_2 (72 vs 25) → **Cluster 2**
- P_4: to μ_2 (13 vs 0) → **Cluster 2**
- P_5: to μ_2 (50 vs 13) → **Cluster 2**
- P_6: to μ_2 (52 vs 17) → **Cluster 2**

Resulting clusters after Iteration 1:

- Cluster 1: $\{P_1\} = \{(2, 10)\}$
- Cluster 2: $\{P_2, P_3, P_4, P_5, P_6\} = \{(2, 5), (8, 4), (5, 8), (7, 5), (6, 4)\}$

Regression Analysis

- In machine learning, regression analysis is a statistical technique that predicts continuous numeric values based on the relationship between independent and dependent variables. The main goal of regression analysis is to plot a line or curve that best fit the data and to estimate how one variable affects another.
- Regression analysis is a fundamental concept in machine learning and it is used in many applications such as forecasting, predictive analytics, etc.
- In machine learning, **regression is a type of supervised learning**. The key objective of regression-based tasks is to predict output labels or responses, which are continuous numeric values, for the given input data. The output will be based on what the model has learned in the training phase.
- **Regression models** use the input data features (independent variables) and their corresponding continuous numeric output values (dependent or outcome variables) to learn specific associations between inputs and corresponding outputs.

Terms used in Regression Analysis

- **Independent Variables** – These variables are used to predict the value of the dependent variable. These are also called predictors. In dataset, these are represented as **features**.
- **Dependent Variables** – These are the variables whose values we want to predict. These are the main factors in regression analysis. In dataset, these are represented as **target variables**
- **Regression line** – It is a straight line or curve that a regressor plots to fit the data points best (Y = a + bX)
- **Overfitting and underfitting** – Overfitting is when the regression model works well with the training dataset but not with the testing dataset. It's also referred to as the problem of high variance. Underfitting is when the model doesn't work well with training datasets. It's also referred to as the problem of high bias.
- **Outliers** – These are data points that don't fit the pattern of the rest of the data. They are the extremely high or extremely low values in the data set.
- **Multicollinearity** – Multicollinearity occurs when independent variables (features) have dependency among them.

Types of Regression in ML

- Generally, the classification of **regression methods** is done based on the three metrics – the number of independent variables, type of dependent variables, and shape of the regression line.
- There are numerous regression techniques used in ML –
  1. Simple Linear Regression
  2. Multiple Linear Regression
  3. Logistic Regression
  4. Polynomial Regression
  5. Lasso Regression
  6. Ridge Regression
  7. Decision Tree Regression
  8. Random Forest Regression
  9. Support Vector Regression

Types of Regression(2)

- **Simple Linear Regression** - is one of the simplest and most widely used statistical models. This assumes that there is a linear relationship between the independent and dependent variables. This means that the change in the dependent variable is proportional to the change in the independent variables. For example predicting the price of a house based on its size.
- **Multiple Linear Regression-** extends simple linear regression by using multiple independent variables to predict target variable. For example predicting the price of a house based on multiple features such as size, location, number of rooms, etc.
- **Polynomial Regression-** is used to model with non-linear relationships between the dependent variable and the independent variables. It adds polynomial terms to the linear regression model to capture more complex relationships. Relationship is modelled as an n^{th} degree polynomial. For example when we want to predict a non-linear trend like population growth over time we use polynomial regression.
- **Logistic Regression** is a supervised machine learning algorithm used for classification problems. Unlike linear regression which predicts continuous values it predicts the probability that an input belongs to a specific class. It is used for binary classification where the output can be one of two possible categories such as Yes/No, True/False or 0/1. It uses sigmoid function to convert inputs into a probability value between 0 and 1.

Types of Regression(2)

- **Lasso Regression** – is is a regression method based on Least Absolute Shrinkage and Selection Operator and is used in regression analysis for variable selection and regularization. Also known as L1 regularization technique. It helps remove irrelevant data features and prevents overfitting. This allows features with weak influence to be clearly identified as the coefficients of less important variables are shrunk toward zero.
- **Ridge Regression,** also known as L2 regularization, is a technique used in linear regression to address the **problem of multicollinearity among predictor variables. Multicollinearity** occurs when independent variables in a regression model are highly correlated, which can lead to unreliable and unstable estimates of regression coefficients.
- **Decision Tree Regression** Uses a tree-like structure to make decisions where each branch of tree represents a decision and leaves represent outcomes. For example predicting customer behavior based on features like age, income, etc there we use decision tree regression.
- **Random Forest Regression** is is a ensemble method that builds multiple decision trees and each tree is trained on a different subset of the training data. The final prediction is made by averaging the predictions of all of the trees. For example customer churn or sales data using this.
- **Support Vector Regression (SVR)** is a type of regression algorithm that is based on the Support Vector Machine (SVM) algorithm. SVM is a type of algorithm that is used for classification tasks but it can also be used for regression tasks. SVR works by finding a hyperplane that minimizes the sum of the squared residuals between the predicted and actual values.

Applications of Regression

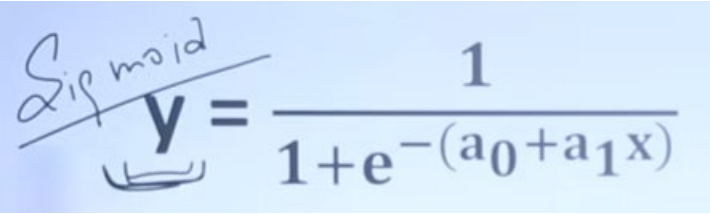
- **Forecasting or Predictive analysis** – One of the important uses of regression is forecasting or predictive analysis. For example, we can forecast GDP, oil prices, or, in simple words, the quantitative data that changes with the passage of time.
- **Optimization** – We can optimize business processes with the help of regression. For example, a store manager can create a statistical model to understand the peak time of coming customers.
- **Error correction** – In business, making correct decisions is equally important as optimizing the business process. Regression can help us to make correct decision as well as correct the already implemented decision.
- **Economics** – It is the most used tool in economics. We can use regression to predict supply, demand, consumption, inventory investment, etc.
- **Fintech Companies** – A FINTECH company is always interested in minimizing the risk portfolio and wants to know the factors that affect the customers. All these can be predicted with the help of a regression model.

Logistic Regression

- SUPERVISED LEARNING used in CLASSIFICATION MODEL
- Predict the classes e.g. Predict email is Spam(Y=1) or Not Spam(Y=0)
- **Dependent variable data which is to be precited is categorial and binary (o or 1) in nature.**
- Exam Result to be predicted (Pass Y=1) or Fail (Y = 0) based on the number of study hours.
- There can be one or more independent variables to predict the dependent var. (0 or 1).
- **Logistic Regression** is a **supervised learning** algorithm used for **classification**, not regression and it predicts **categorical outcomes**, usually **binary** (Yes/No, 0/1, Pass/Fail, Spam/Not Spam, etc.).
- Even though the name has "regression," it is actually a **classification algorithm** based on the **logistic (sigmoid) function**.
- Sigmoid function will give values in the range 0 to 1.

Independent variable **Dependent variable**

Study Hours	Exam Result
2	0 (Fail)
3	0
4	1 (Pass)
5	1
6	1
7	1
8	1



Logistic Regression

The **sigmoid** function maps any real number to a value between 0 and 1:

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

where

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Interpretation:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

- $P(Y = 1|X)$ is the **probability** that the outcome is 1 (for example, success, yes, etc.).
- The **decision boundary** is usually set at **0.5**:
  - If $P(Y = 1|X) \geq 0.5 \Rightarrow$ predict 1
  - Else predict 0

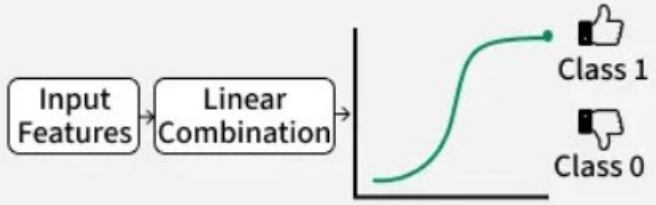
Logistic Regression

- SUPERVISED LEARNING used in CLASSIFICATION MODEL
- Algorithm used for **classification**, not regression and it predicts **categorical outcomes of the dependent variable Y**, usually **binary** (Yes/No, 0/1, Pass/Fail, Spam/Not Spam, etc.). Predict the classes e.g. Predict email is Spam(Y=1) or Not Spam(Y=0)
- Exam Result to be predicted (Pass Y=1) or Fail (Y = 0) based on the number of study hours. There can be one or more independent variables to predict the dependent var. (0 or 1).
- Even though the name has "regression," it is actually a **classification algorithm** based on the **logistic (sigmoid) function**. Sigmoid function will give values in the range 0 to 1. here a0 and a1 are based on the MLE (Maximum Likelihood Estimation) method.

Independent variable **Dependent variable**

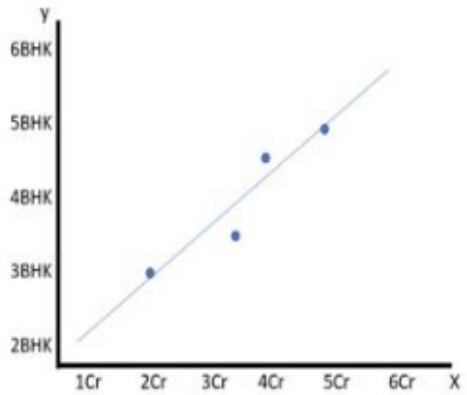
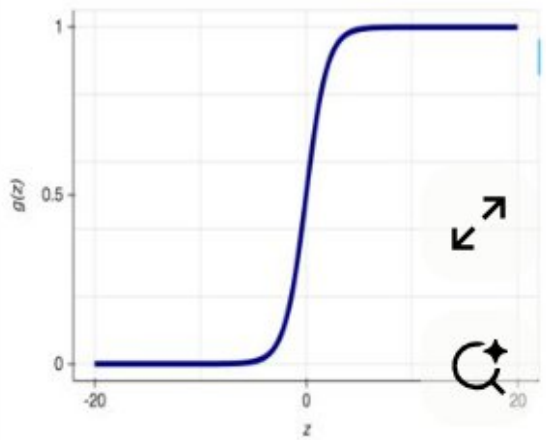
Study Hours	Exam Result
2	0 (Fail)
3	0
4	1 (Pass)
5	1
6	1
7	1
8	1

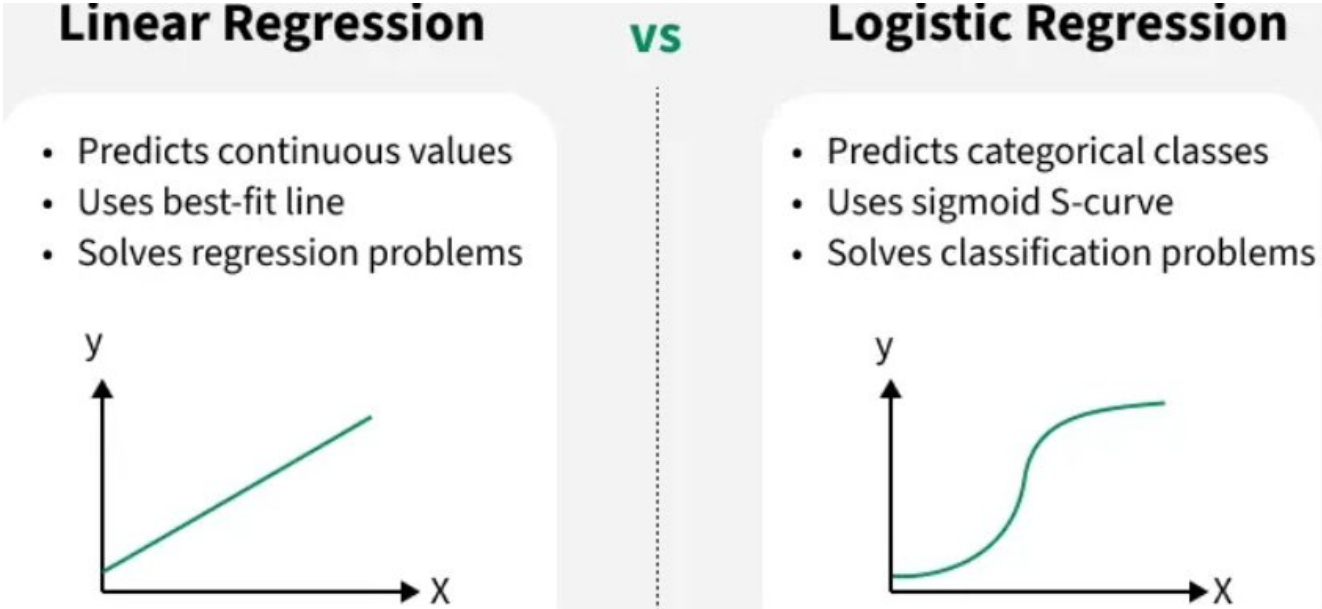
- Predicts the probability of a binary outcome (Yes/No, 0/1)
- Uses the sigmoid function to map inputs to probabilities (0 to 1)
- Ideal for classification tasks



Sigmoid

$$y = \frac{1}{1+e^{-(a_0+a_1x)}}$$

Linear Regression	Logistic Regression
Target is an interval variable	Target is discrete (binary or ordinal) variable
Predicted values are the mean of the target variable at the given values of the input variable	Predicted values are the probability of the particular levels of the given values of the input variable
Solve regression problems	Solve classification problems
Example : What is the Temperature?	Example : Will it rain or not?
Graph is straight line	Graph is S-curve
	



Formula for Logistic Regression Model is :

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

- $P(Y = 1|X)$ is the **probability** that the outcome is 1 (for example, success, yes, etc.).
- The **decision boundary** is usually set at **0.5**:
  - If $P(Y = 1|X) \geq 0.5 \Rightarrow \text{predict } 1$
  - Else $\text{predict } 0$

Log-Odds (Logit) formula for Logistic Regression

We can rewrite the logistic model in terms of **log-odds** (or **logit**):

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

This shows that logistic regression models a **linear relationship between the independent variable(s) and the log-odds of the outcome**.

Problem 1

Suppose the logistic regression model is:

$$p = \frac{1}{1 + e^{-(-4+0.8x)}}$$

where x represents **hours studied**, and p is the **probability of passing an exam.**

Find: a) The probability that a student who studied 5 hours passes the exam.

b) The decision (Pass or Fail) if the threshold = 0.5.

SOLUTION : First write the model and in next step substitute x = 5 in the model

$$p = \frac{1}{1 + e^{-(-4+0.8x)}}$$

$$z = -4 + 0.8(5) = -4 + 4 = 0$$

$$p = \frac{1}{1 + e^{-0}} = \frac{1}{1+1} = 0.5$$

Interpretation (Step 3) : The probability of passing when studying 5 hours is **0.5.**

Since the threshold = 0.5, we predict “Pass” if we take $p \geq 0.5$

Step 4 : Check another case (for another value of x)

If $x = 7$:

$$z = -4 + 0.8(7) = -4 + 5.6 = 1.6$$

$$p = \frac{1}{1 + e^{-1.6}} \approx \frac{1}{1 + 0.201} = 0.832$$

So, the probability of passing when studying 7 hours is **0.83** 🎭 **Predict Pass.**

If $x = 2$:

$$z = -4 + 0.8(2) = -4 + 1.6 = -2.4$$

$$p = \frac{1}{1 + e^{2.4}} \approx \frac{1}{1 + 11.02} = 0.083$$

Probability of passing is **0.08** 🎭 **Predict Fail.**

Hours Studied (x)	z	p (Probability of Pass)	Prediction
2	-2.4	0.083	Fail
5	0	0.5	Pass (boundary)
7	1.6	0.832	Pass

Concept	Description
Output	Probability between 0 and 1
Decision rule	Usually threshold = 0.5
Link function	Logit = ln(p / (1 - p))
Estimation	Coefficients (β) are found using **Maximum Likelihood Estimation (MLE)**
Use cases	Binary classification: spam detection, disease prediction, churn prediction, etc.

Multivariate Logistic Regression

When we have **more than one independent variable**, logistic regression generalizes easily.
Model Definition in this case is as follows:

For n features $x_1, x_2, \ldots, x_n$, the logistic regression model is:

$$p = P(Y = 1|X) = \frac{1}{1 + e^{-z}}$$

where

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Equivalently,

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Example - A company wants to predict whether a customer will buy a product (**Y = 1**) or not (**Y = 0**) based on:

Variable	Description
x_1	Age (in years)
x_2	Monthly Income (in ₹ thousands)

The fitted logistic regression model is:

$$p = \frac{1}{1 + e^{-(-6+0.04x_1+0.3x_2)}}$$

Calculate: a) The probability of purchase for a 30-year-old earning ₹25,000/month.

b) The decision (Buy / Not Buy) at a 0.5 threshold.

Step 1: Compute the value of z

$$z = -6 + 0.04(30) + 0.3(25)$$

$$z = -6 + 1.2 + 7.5 = 2.7$$

4. Step 2: Compute Probability

$$p = \frac{1}{1 + e^{-2.7}} = \frac{1}{1 + 0.067} = 0.937$$

So, **probability = 0.937 (93.7%)** that the customer will buy the product.

5. Step 3: Decision

Since $p = 0.937 > 0.5$,

Prediction: Customer will buy the product (Y = 1).

6. Step 4: Try another case

Customer B: $x_1 = 22$ years, $x_2 = 10$ (₹10,000/month)

$$z = -6 + 0.04(22) + 0.3(10) = -6 + 0.88 + 3 = -2.12$$

$$p = \frac{1}{1 + e^{2.12}} = \frac{1}{1 + 8.33} = 0.107$$

So, $p = 0.107$ → **Prediction: Will not buy (Y = 0).**

7. Step 5: Interpret Coefficients

Coefficient	Meaning
$\beta_0 = -6$	Base log-odds when all predictors = 0.
$\beta_1 = 0.04$	For each **extra year of age**, log-odds of buying increase by 0.04.
$\beta_2 = 0.3$	For each **₹1000 increase in monthly income**, log-odds of buying increase by 0.3.

Thus, **income** has a stronger effect on purchase probability than **age**.

8. Step 6: Decision Boundary

The decision boundary is found when $p = 0.5$, i.e., $z = 0$:

$$-6 + 0.04x_1 + 0.3x_2 = 0$$

$$0.3x_2 = 6 - 0.04x_1$$

$$x_2 = 20 - 0.133x_1$$

So the decision boundary is a **straight line** in the (x_1, x_2) plane dividing "Buy" and "Not Buy" regions.

9. Step 7: Summary Table

Case	Age (x_1)	Income (x_2)	z	p	Decision
A	30	25	2.7	0.937	Buy
B	22	10	-2.12	0.107	Not Buy
C	28	15	-6 + 1.12 + 4.5 = -0.38	0.406	Not Buy
D	35	20	-6 + 1.4 + 6 = 1.4	0.802	Buy

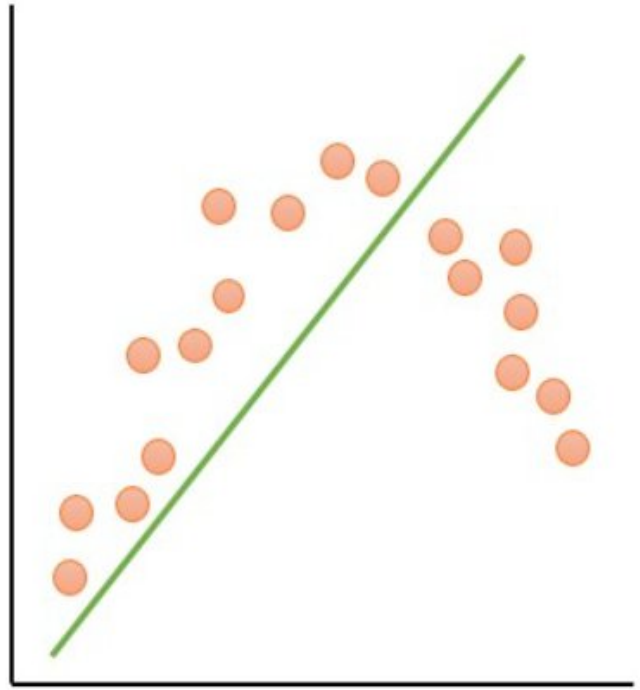
Key Interpretations

- Logistic regression models the **probability** that $Y = 1$as a function of predictors.
- The coefficients affect the **log-odds**, not directly the probability.
- The model creates a **linear decision boundary** between classes.

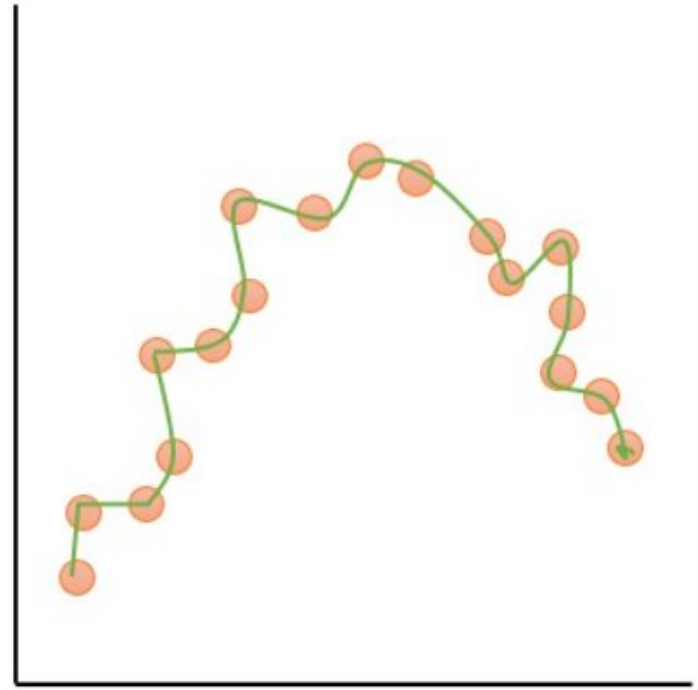
Bias Variance Dichotomy Model (Trade-off Model)

- **Bias** refers to the error that results from oversimplifying the underlying relationship between the input features and the output variable. At the same time, **variance** refers to the error that results from being too sensitive to fluctuations in the training data.
- In Optimization, we strive to **minimize both bias and variance** in order to build a model that can accurately predict on unseen data. A high-bias model may be too simplistic and underfit the training data. In contrast, a model with high variance may overfit the training data and fail to generalize to new data.
- Bias is calculated as the difference between average prediction and actual value. Bias (systematic error) occurs when a model makes incorrect assumptions about data. A model with high bias does not match well training data as well as test data. It leads to high errors in training and test data. While the model with low bias matches the training data well (high training accuracy or less error in training). It leads to low error in training data
- **High Bias** – High bias occurs due to erroneous assumptions in the machine learning model. Models with high bias cannot capture the hidden pattern in the training data. This leads to **underfitting**. Features of high bias are a highly simplified model, underfitting, and high error in training and test data.
- **Low Bias** – Models with low bias can capture the hidden pattern in the training data. Low bias leads to high variance and, eventually, **overfitting**. Low bias generally occurs due to the ML model being overly complex.

High Bias, Underfitting



Low Bias, Overfitting



Variance Concept in Bias Variance Dichotomy Model

- **Variance** is a measure of the spread or dispersion of numbers in a given set of observations with respect to the mean.
- In Optimization, Variance is how much a model's predictions change when it's trained on different data.
- It shows how much model prediction varies when there is a slight variation in data. If model accuracies on training and test data vary greatly, the model has high variance.
- A model with high variance can even fit noises on training data but lacks generalization to new, unseen data.
    - **High variance:** The model is too sensitive to small changes and may overfit.
    - **Low variance:** The model is more stable but might miss some patterns

Mathematical Formula for Bias and Variance

Bias

$$\text{Bias}^2 = \left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2$$

Where,

- $\hat{f}(x)$: predicted value by the model
- $f(x)$: true value
- $\mathbb{E}[\hat{f}(x)]$: expected prediction over different training sets

Variance

$$\text{Variance} = \mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}[\hat{f}(x)]\right)^2\right]$$

Where,

- $\hat{f}(x)$: predicted value by the model
- $\mathbb{E}[\hat{f}(x)]$: average prediction over multiple training sets

Types of Variance

High Variance – High variance models capture noise along with hidden pattern. It leads to **overfitting**. High variance models show high training accuracy but low test accuracy. Some features of a high variance model are an overly complex model, overfitting, low error on training data, and high error or test data.

Low Variance – A model with low variance is unable to capture the hidden pattern in the data. Low variance may occur when we have a very small amount of data or use a very simplified model. Low variance leads to **underfitting**.

Bias-Variance Tradeoff

Model Type	Bias	Variance	Result
Underfitting	High	Low	Poor training and test performance
Optimal	Moderate	Moderate	Best generalization
Overfitting	Low	High	Poor test performance

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

This decomposition helps us understand why models sometimes **underfit** or **overfit**.

Irreducible Error - This is the noise inherent in data that **no model** can explain.

$$\text{Irreducible Error} = Var(\varepsilon)$$

where ε is the random noise.

Total Expected Prediction Error Formula

The expected mean squared error (MSE) at a point x can be decomposed as:

$$E[(Y - \hat{f}(x))^2] = [\text{Bias}(\hat{f}(x))]^2 + \text{Variance}(\hat{f}(x)) + \sigma^2$$

Where:

- $Y = f(x) + \varepsilon$,
- σ^2 is the variance of noise (irreducible error).

This decomposition is known as the **Bias–Variance Trade-off**.

(a) Irreducible Error

- Comes from the random noise ε.
- Even a perfect model can't predict noise.
- Formally: $\mathrm{Var}(\varepsilon) = \sigma^2$

You **cannot reduce** this part — it's inherent in the data.

(b) Bias

- Bias measures the **systematic error** in your model's assumptions.
- It is the **difference between the true function** $f(x)$ and the **expected prediction** $E[\hat{f}(x)]$ of your model.

$$\mathrm{Bias}(x) = E[\hat{f}(x)] - f(x)$$

and

$$\mathrm{Bias}^2 = [E[\hat{f}(x)] - f(x)]^2$$

High Bias → Model makes strong assumptions, oversimplifies relationships.

Example: Linear regression used for a nonlinear relationship.

(c) Variance

- Variance measures how much $\hat{f}(x)$ would vary if we trained it on different datasets.
- High variance means the model is **too sensitive to training data** — small changes in data cause big changes in prediction.

Formally:

$$\text{Variance}(x) = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

High Variance → Model memorizes training data instead of generalizing.

Example: Deep decision trees or k-NN with $k = 1$.

3. Total Error Decomposition

Putting it together:

$$E[(y - \hat{f}(x))^2] = \underbrace{[\text{Bias}(x)]^2}_{\text{Systematic error}} + \underbrace{\text{Variance}(x)}_{\text{Model sensitivity}} + \underbrace{\sigma^2}_{\text{Irreducible noise}}$$

Interpretation

Model Complexity	Bias	Variance	Total Error
Very Simple (Underfit)	High	Low	High
Optimal (Balanced)	Medium	Medium	**Lowest**
Very Complex (Overfit)	Low	High	High

Goal of Model

The learning algorithm aims to **find a balance**:

$$\text{Minimize } (\text{Bias}^2 + \text{Variance})$$

because both extremes lead to high error.

This trade-off guides:

- Model complexity choice
- Regularization techniques (L1, L2)
- Cross-validation strategies
- Ensemble learning methods (bagging reduces variance, boosting reduces bias)

Practical Insight

Situation	Cause	Remedy
High Bias	Model too simple, underfitting	Use more features, increase model capacity
High Variance	Model too complex, overfitting	Regularize, collect more data, use cross-validation

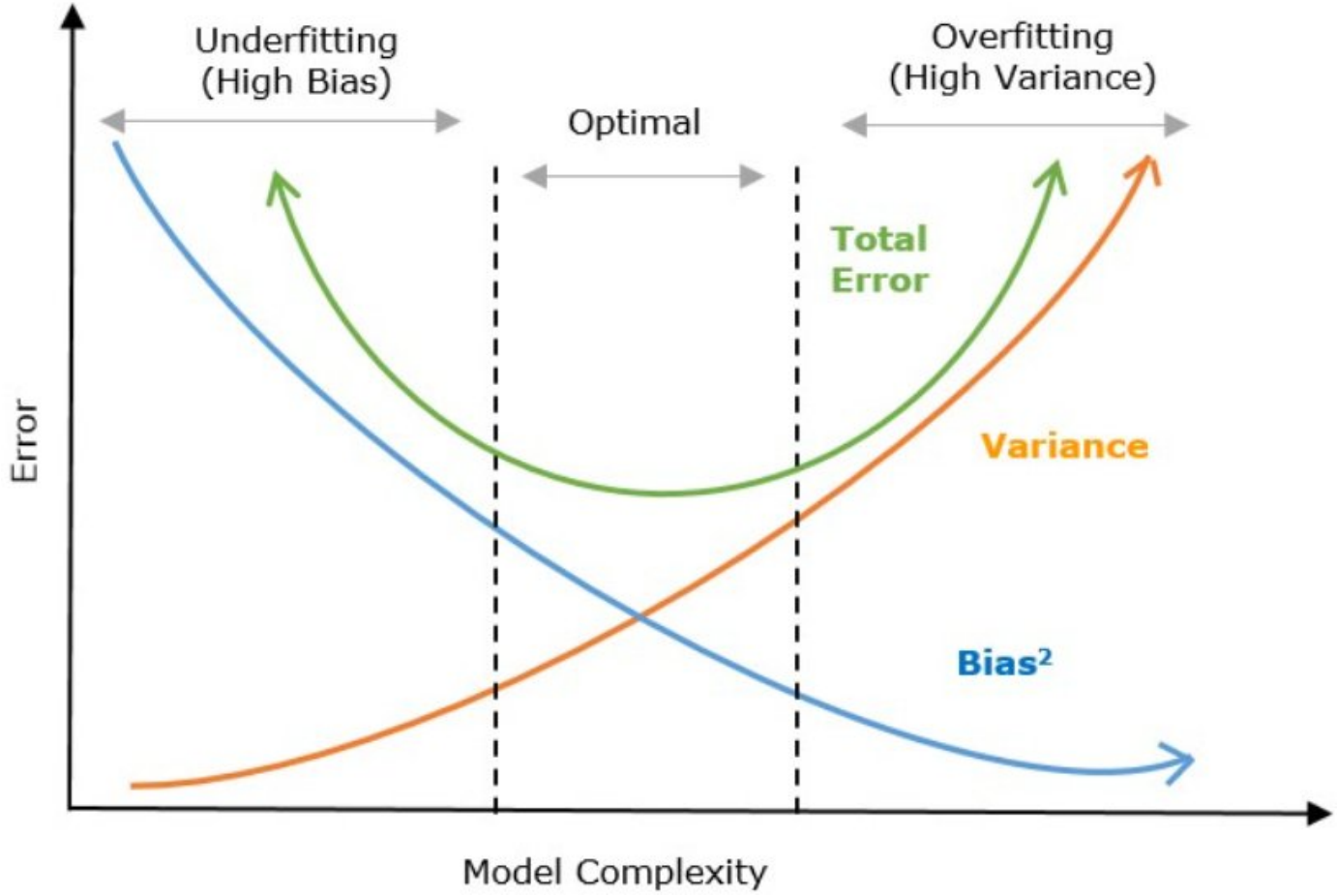
Summary

Concept	Description
Bias	Error from wrong assumptions
Variance	Error from sensitivity to training data
Irreducible Error	Random noise not explainable by model
Goal	Find sweet spot minimizing both Bias^2 and Variance
Techniques to Control Bias/Variance	Regularization (Lasso/Ridge), Cross-validation, Pruning, Bagging/Boosting

The Tradeoff

Model Type	Bias	Variance	Behavior
Simple Model (Linear)	High	Low	Underfits
Complex Model (High-degree polynomial)	Low	High	Overfits

The **goal** is to find an optimal model complexity where the **sum of bias² + variance** is minimized.



Example

Suppose we are estimating a function $f(x) = x^2$ using a model trained multiple times on random data.

From several experiments, we observe:

Quantity	Symbol	Value
True value at $x = 2$	$f(2)$	4
Average predicted value $E[\hat{f}(2)]$	3.5	
Average squared prediction $E[\hat{f}(2)^2]$	13.25	

1. **Bias**^{2}
2. **Variance**
3. **Total Expected Error** (assuming noise variance $\sigma^2 = 0.5$)

Step 1: Compute Bias

$$\text{Bias}(2) = E[\hat{f}(2)] - f(2) = 3.5 - 4 = -0.5$$

$$\text{Bias}^2 = (-0.5)^2 = 0.25$$

Step 2: Compute Variance

$$\text{Variance} = E[\hat{f}(2)^2] - (E[\hat{f}(2)])^2 = 13.25 - (3.5)^2 = 13.25 - 12.25 = 1.0$$

Step 3: Compute Total Expected Error

$$\text{Expected Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

$$= 0.25 + 1.0 + 0.5 = 1.75$$

Interpretation

- **Bias^2 (0.25)** is small → model's predictions are close to the true function.
- **Variance (1.0)** is significant → model predictions vary across datasets.
- **Total error (1.75)** indicates that reducing variance (via regularization or ensemble) could improve model stability.

Summary

Term	Meaning	Desirable?
Bias	Systematic error	Low
Variance	Sensitivity to training data	Low
Irreducible Error	Noise	Unavoidable
Tradeoff	Balance between bias² and variance	Optimal complexity minimizes total error

> A model has a training error of 1% and a test error of 25%.
> What does this suggest in terms of bias and variance?

Answer:

Low training error → low bias.

High test error → high variance.

Hence, the model **overfits** the training data.

1. Linear Regression

- It is the most commonly used regression model in machine learning. It may be defined as the statistical model that analyzes the linear relationship between a dependent variable with a given set of independent variables.
- A linear relationship between variables means that when the value of one or more independent variables changes (increase or decrease), the value of the dependent variable will also change accordingly (increase or decrease).
- Linear regression is further divided into two subcategories: simple linear regression and multiple linear regression (also known as multivariate linear regression).
- In simple linear regression, a single independent variable (or predictor) is used to predict the dependent variable. Mathematically, the simple linear regression can be represented as follows- $Y = a + bX$ where,
  - Y is the dependent variable we are trying to predict.
  - X is the independent variable we are using to make predictions
  - *b* is the slope of the regression line, which represents the effect X has on Y.
  - *a* is a constant known as the Y-intercept. If X = 0, Y would be equal to a.
- In multi-linear regression, multiple independent variables are used to predict the dependent variables.

Multiple Linear Regression Model

- Multiple Linear Regression extends this concept by modelling the relationship between a dependent variable and two or more independent variables. This technique allows us to understand how multiple features collectively affect the outcomes.
- Steps to perform this are similar to that of simple linear Regression but difference comes in the evaluation process. We can use it to find out which factor has the highest influence on the predicted output and how different variables are related to each other. Assumptions of this Model are:
    1. **Linearity**: Relationship between dependent and independent variables should be linear.
    2. **Homoscedasticity**: Variance of errors should remain constant across all levels of independent variables.
    3. **Multivariate Normality**: Residuals should follow a normal distribution.
    4. **No Multicollinearity**: Independent variables should not be highly correlated
- Equation for multiple linear regression is:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Where:

- $y$is the dependent variable
- $X_1, X_2, \cdots X_n$ are the independent variables
- β_0 is the intercept
- $\beta_1, \beta_2, \cdots \beta_n$ are the slopes

Multicollinearity in Regression Analysis

- Multicollinearity occurs when two or more independent variables in a regression model are highly correlated with each other. So, **multicollinearity e**xists when there are linear relationships among the independent variables, this causes issues in regression analysis because it does not follow the assumption of independence among predictors.
- **Causes of Multicollinearity in Regression Analysis**
    1. **Correlation Among Predictor Variables**: Multicollinearity often occurs when predictor variables in a regression model exhibit a **high correlation** with one another. This situation arises when one predictor variable can be accurately predicted from the others, complicating the estimation of individual predictor effects within the model.
    2. **Overparameterization of the Model**: Introducing too many predictor variables closer to the number of observations can also lead to multicollinearity. More predictors can cause redundancy and increase the variance of the coefficient estimates.
    3. **Data Collection Issues**: Problems in the data collection process can also introduce multicollinearity. For instance, if certain variables are measured with exceptional precision or are inherently interconnected, it can lead to multicollinearity in the regression model.
- **To detect multicollinearity we can use:**
    1. **Correlation Matrix:** A correlation matrix helps to find relationships between independent variables. High correlations (close to 1 or -1) suggest multicollinearity.
    2. **VIF (Variance Inflation Factor):** VIF quantifies how much the variance of a regression coefficient increases if predictors are correlated. A high VIF typically above 10 indicates multicollinearity.

This Concludes Today's Presentation

Thank you for your attention