



WHAt is ml

what is dl

how is ml different from dl

data mining

data mininng vs ml

bias

variance

b-v tradeoff

what is r2

significance of r2

how r2 is estimated

how loss function is defined

loss func vs cost func

how gradient descent works

convex vs non convex

batch gradient

stochastic gradient

normal gradient

how do you determine gradient

what is L1

what is L2

L1 vs L2

cost functions of L1 and L2 regularizations

what is feature selection

how feature selection is different in ML and DL

ml vs dl

ml is a part of ai justify

ai vs dl

nn

ml vs dl vs nn

How cost is estimated

multiple linear regression

independent and dependent variables

These are the viva questoins that will be asked

Here is a comprehensive study guide formatted for your viva, organizing the questions into logical sections with clear, concise answers and comparison tables.

1. Core Definitions & Hierarchy

What is ML (Machine Learning)?

Machine Learning is a subset of AI that enables systems to learn from data and improve their performance without being explicitly programmed for every rule. It focuses on developing algorithms that can identify patterns and make predictions.

What is DL (Deep Learning)?

Deep Learning is a specialized subset of ML inspired by the structure of the human brain. It uses multi-layered artificial neural networks (ANNs) to model complex patterns in data. It excels at processing unstructured data like images, audio, and text.

ML vs. DL vs. NN (Neural Networks)

- **NN (Neural Network):** The fundamental building block of DL. A computational model containing nodes (neurons) and edges (weights) that processes information.
- **Relationship:** $\text{AI} \supset \text{ML} \supset \text{NN} \supset \text{DL}$.
 - **Justification:** Deep Learning is simply "deep" neural networks (many layers). Neural networks are a *type* of machine learning algorithm. Machine learning is a *type* of artificial intelligence.

ML vs. DL (Viva Difference Table)

Feature	Machine Learning (ML)	Deep Learning (DL)
Data Dependency	Can work with smaller datasets.	Requires massive amounts of data to perform well.
Feature Engineering	Manual: Humans must identify and extract features (e.g., "length", "pixel intensity").	Automatic: The network learns features itself (e.g., raw pixels → edges → shapes).
Hardware	Can run on standard CPUs.	Requires GPUs/TPUs due to matrix operations.
Execution Time	Faster to train (minutes/hours).	Very slow to train (days/weeks), but fast to test.
Interpretability	Easier to interpret (White Box).	"Black Box" (hard to explain <i>why</i> it made a decision).

Justify: "ML is a part of AI"

Artificial Intelligence (AI) is the broad science of mimicking human abilities (logic, learning, perception). Machine Learning (ML) is specifically the *subset* of AI that achieves this mimicry by learning from data rather than following static, hard-coded rules. Therefore, all ML is AI, but not all AI is ML (e.g., rule-based expert systems are AI but not ML).

Data Mining vs. ML

- **Data Mining:** Focuses on discovering previously unknown patterns or rules in existing databases (Descriptive). *Goal: Explain the past.*
- **Machine Learning:** Focuses on using those patterns to train models that can predict future outcomes (Predictive). *Goal: Predict the future.*

2. Model Performance & Evaluation

Bias vs. Variance

- **Bias (Underfitting):** Error introduced by approximating a real-world problem with a too-simple model. High bias causes the model to miss relevant relations (e.g., using a straight line for curved data).
- **Variance (Overfitting):** Error introduced by the model being too sensitive to noise in the training data. High variance causes the model to model the random noise rather than the intended outputs.

Bias-Variance Tradeoff

The conflict where minimizing bias usually increases variance and vice versa.

- **Goal:** Find the "Sweet Spot" (low bias, low variance) where the total error is minimized.
- Total Error = $Bias^2 + Variance + IrreducibleError$.

What is R² (R-Squared)?

Also called the **Coefficient of Determination**. It represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

- **Range:** 0 to 1 (0% to 100%).
- **Significance:** An R^2 of 0.8 means 80% of the variation in the output is explained by your input variables. Higher is generally better.

How R² is Estimated

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Where:

- SS_{res} (Sum of Squared Residuals): The error of your model ($\sum(y_{actual} - y_{predicted})^2$).
- SS_{tot} (Total Sum of Squares): The variance in the data ($\sum(y_{actual} - y_{mean})^2$).

3. Optimization & Training

Loss Function vs. Cost Function

- **Loss Function:** Calculates the error for a **single** training example (e.g., $(y - \hat{y})^2$).
- **Cost Function:** The **average** of the loss functions for the **entire** training dataset. It is the single number you try to minimize during training.

How Loss Function is Defined (Example)

For Linear Regression, the most common is **Mean Squared Error (MSE)**:

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$$

Gradient Descent (How it works)

An iterative optimization algorithm used to minimize the cost function.

1. Initialize weights randomly.
2. Calculate the gradient (slope) of the cost function.
3. Move weights in the opposite direction of the gradient (downhill) by a small step (learning rate).
4. Repeat until the cost (error) is minimized.

Convex vs. Non-Convex Functions

- **Convex:** Has only one global minimum (bowl shape). Gradient descent is guaranteed to find the best solution. (e.g., Linear Regression).
- **Non-Convex:** Has multiple peaks and valleys (local minima). Gradient descent might get stuck in a local minimum instead of the global best. (e.g., Neural Networks).

Types of Gradient Descent

- **Batch Gradient Descent:** Uses the **entire dataset** to calculate the gradient for one step.
Slow but stable.
- **Stochastic Gradient Descent (SGD):** Uses **one random sample** to calculate the gradient.
Fast but noisy/erratic.
- **"Normal" Gradient:** Usually refers to Batch Gradient Descent (calculating the true gradient).

How do you determine gradient?

Mathematically, by taking the **partial derivative** of the cost function with respect to each weight (parameter). This tells you how much the error changes if you change that specific weight slightly.

4. Regularization (L1 vs. L2)

Regularization adds a penalty to the cost function to prevent overfitting (high variance).

What is L1 (Lasso)?

Adds the **Absolute Value** of the weights to the cost function.

- **Key Feature:** Can shrink weights to exactly **zero**. This performs **feature selection** (removes useless features).

What is L2 (Ridge)?

Adds the **Squared Value** of the weights to the cost function.

- **Key Feature:** Shrinks weights *near* zero but never exactly zero. Good for handling correlated features and preventing large weights.

Cost Functions of L1 and L2

- **L1 Cost:** $Cost = \text{Original Loss} + \lambda \sum |w|$
- **L2 Cost:** $Cost = \text{Original Loss} + \lambda \sum w^2$
(Where λ is the regularization strength)

Feature Selection: ML vs. DL

- **In ML:** Feature selection is often a manual step (removing columns) or done via L1 regularization. You choose what matters.
- **In DL:** Feature selection is **implicit and automatic**. The hidden layers of the neural network learn to ignore irrelevant parts of the input (e.g., background noise in an image) by assigning them near-zero weights during training.

5. Regression Basics

Multiple Linear Regression

An extension of simple linear regression that uses **two or more independent variables** to predict a dependent variable.

- Equation: $y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$

Independent vs. Dependent Variables

- **Independent Variable (Input/Feature/X):** The variable you control or observe (e.g., "Hours Studied").
- **Dependent Variable (Output/Target/Y):** The variable you are trying to predict (e.g., "Exam Score").

How Cost is Estimated

In regression, cost is estimated by summing the squared difference between what the model predicted and what the actual value was, averaged over all data points (MSE). Ideally, you want this number to be as close to 0 as possible.

**

1. <https://www.geeksforgeeks.org/machine-learning/ml-bias-variance-trade-off/>
2. <https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization>
3. <https://aws.amazon.com/compare/the-difference-between-machine-learning-and-deep-learning/>
4. <https://www.datacamp.com/tutorial/loss-function-in-machine-learning>
5. <https://www.bmc.com/blogs/bias-variance-machine-learning/>
6. https://www.youtube.com/watch?v=k2P_pHQDlp0
7. <https://www.youtube.com/watch?v=3tyTbjpetWA>
8. <https://pmc.ncbi.nlm.nih.gov/articles/PMC8433983/>
9. <https://www.simplilearn.com/data-mining-vs-machine-learning-article>
10. <https://www.datacamp.com/tutorial/adjusted-r-squared>