TASK_3

1. What is Machine learning?

Machine Learning (ML) is a branch of Artificial Intelligence that enables computers to learn patterns from data and make predictions or decisions **without being explicitly programmed**.

Instead of writing step-by-step rules, we train a model using historical data. The model learns relationships within the data and applies that knowledge to new, unseen data.

How Machine Learning Works

1. **Input Data (Features)** → Independent variables
2. **Model** → Mathematical algorithm
3. **Training** → Learn patterns from data
4. **Prediction** → Apply learned pattern to new data
5. **Evaluation** → Measure performance using metrics

Example

Suppose we want to predict house prices.

- Input features: size, location, number of rooms
- Output: price
- The ML model learns how these inputs affect price using past data
- Then predicts price for new houses

Real-World Applications

- Spam email detection
- Face recognition
- Recommendation systems (Netflix, Amazon)
- Medical diagnosis
- Fraud detection
- Stock market prediction

Traditional Programming vs Machine Learning

Traditional-Programming
Input + Rules → Output

Machine-Learning
Input + Output → Model (learns rules automatically)

---

2. Explain different types of Machine learning

1. Supervised Learning

Definition:

Supervised Learning is a type of Machine Learning where the model is trained on **labeled data**, meaning both input and correct output are provided.

The model learns the mapping between input features (X) and target variable (Y).

Types of Supervised Learning:

1. **Regression** → Predicts continuous values
    - Example: House price prediction
2. **Classification** → Predicts categorical values
    - Example: Spam or Not Spam

Common Algorithms:

- Linear Regression
- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)

Applications:

- Disease prediction
- Sales forecasting
- Credit scoring

2. Unsupervised Learning

Definition:

Unsupervised Learning works with **unlabeled data**. The model tries to identify hidden patterns or structures in the data.

There is no target variable.

Types:

1. Clustering

Groups similar data points together.
Example: Customer segmentation.

2. Dimensionality Reduction

Reduces number of features while preserving information.
Example: Principal Component Analysis (PCA).

Common Algorithms:

- K-Means
- Hierarchical Clustering
- DBSCAN
- PCA

Applications:

- Market segmentation
- Anomaly detection
- Pattern discovery

3. Semi-Supervised Learning

Definition:

Semi-Supervised Learning uses a **small amount of labeled data** and a **large amount of unlabeled data**.

It is useful when labeling data is expensive or time-consuming.

Example:

Medical image classification where only few images are labeled.

Advantage:

Improves accuracy compared to purely supervised learning when labeled data is limited.

4. Reinforcement Learning

Definition:

Reinforcement Learning is a type of learning where an **agent interacts with an environment** and learns by receiving rewards or penalties.

The goal is to maximize cumulative reward.

Key Components:

- Agent
- Environment
- Reward
- Policy

Example:

- Game playing (Chess, Go)
- Self-driving cars
- Robotics

Type	Labeled Data	Main Task	Example
Supervised	Yes	Prediction	Price prediction
Unsupervised	No	Pattern discovery	Customer grouping
Semi-Supervised	Partially	Improve learning	Image classification
Reinforcement	Reward-based	Decision making	Game AI

---

3. Difference between Regression and classification.

Regression

Definition:

Regression is a supervised learning technique used to predict **continuous numerical values**.

Output:

- Real numbers
- Continuous range (e.g., 10.5, 25000, 98.6)

Examples:

- Predicting house prices
- Predicting temperature
- Predicting sales revenue

Common Algorithms:

- Linear Regression
- Ridge & Lasso Regression

- Polynomial Regression

Evaluation Metrics:

- MAE
- MSE
- RMSE
- R^2 Score

Classification

Definition:

Classification is a supervised learning technique used to predict **categorical class labels**.

Output:

- Discrete categories
- Example: Yes/No, 0/1, Spam/Not Spam

Types:

- Binary Classification (2 classes)
- Multi-class Classification (More than 2 classes)
- Multi-label Classification

Common Algorithms:

- Logistic Regression
- Decision Tree
- Random Forest
- SVM
- KNN

Evaluation Metrics:

- Accuracy
- Precision
- Recall
- F1-Score
- ROC-AUC

Aspect	Regression	Classification
Type of Output	Continuous values	Categorical values
Example	House price prediction	Email spam detection
Goal	Estimate quantity	Assign class label
Algorithms	Linear Regression	Logistic Regression
Evaluation Metrics	RMSE, R^2	Accuracy, F1-score
Decision Boundary	Not required	Required

---

4. **What are the metrics used to evaluate Linear Regression and explain them?**

1. Mean Absolute Error (MAE)

- Formula:

$$MAE = \frac{1}{n}\sum |y_i - \hat{y}_i|$$

- Explanation:

  - It calculates the **average absolute difference** between actual and predicted values.
  - It does not square the errors.
  - All errors contribute equally.

- Advantages:

  - Easy to interpret.
  - Less sensitive to outliers compared to MSE.

- Limitation:

  - Does not penalize large errors strongly.

2. Mean Squared Error (MSE)

- Formula:

$$MSE = \frac{1}{n}\sum (y_i - \hat{y}_i)^2$$

- Explanation:

  - Squares the error before averaging.
  - Large errors get more weight.

- Advantages:

  - Penalizes large errors heavily.
  - Useful in optimization (differentiable).

- Limitation:

  - Unit is squared (not same as target variable).
  - Highly sensitive to outliers.

3. Root Mean Squared Error (RMSE)

- Formula:

$$RMSE = \sqrt{MSE}$$

- Explanation:

  - Square root of MSE.
  - Brings error back to original unit of target variable.

- Advantages:

  - Most commonly used metric.
  - Easy to interpret.

- Limitation:

  - Still sensitive to outliers.

4. R^2 Score (Coefficient of Determination)

- Formula:

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

Where:

- SSresSS_{res}SSres = Sum of squared residuals
- SStotSS_{tot}SStot = Total sum of squares

- Explanation:

  - Measures how much variance in the dependent variable is explained by the model.
  - Range: 0 to 1 (can be negative in rare cases).

- Interpretation:

  - $R^2 = 0$ → Model explains nothing.
  - $R^2 = 1$ → Perfect prediction.
  - $R^2 = 0.8$ → Model explains 80% of variance.

5. Adjusted R^2

- Formula:

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1}$$

Where:

- n = number of observations
- k = number of predictors

- Explanation:

- Adjusts R^2 based on number of features.
- Prevents overfitting.
- Only increases if new feature improves model.

Metric	Sensitive to Outliers	Same Unit as Target	Best Used When
MAE	Low	Yes	Robust evaluation
MSE	High	No	Penalizing large errors
RMSE	High	Yes	General purpose
R^2	No	No	Variance explanation
Adjusted R^2	No	No	Multiple regression

---

5. **Difference between r2 score and adjusted r2 score.**

1. R^2 Score (Coefficient of Determination)

- Definition:

R^2 measures the proportion of variance in the dependent variable that is explained by the independent variables.

- Formula:

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

Where:

- SSresSS_{res}SSres = Sum of Squared Residuals
- SStotSS_{tot}SStot = Total Sum of Squares

- Range:

- Typically between 0 and 1

- Can be negative if model performs worse than baseline

◇ Important Property:

R^2 **always increases or remains the same** when new features are added, even if the feature is irrelevant.

This may lead to **overfitting**.

2. Adjusted R^2 Score

◇ Definition:

Adjusted R^2 modifies R^2 by adjusting for the number of predictors in the model.

◇ Formula:

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1}$$

Where:

- nnn = Number of observations
- kkk = Number of independent variables

◇ Important Property:

Adjusted R^2 **increases only if the new feature improves the model performance significantly**. It decreases if the added feature is irrelevant.

Thus, it helps prevent overfitting.

Aspect	R^2 Score	Adjusted R^2
Considers number of features	No	Yes
Increases when new feature added	Always	Only if useful
Risk of overfitting	High	Lower
Best used for	Simple regression	Multiple regression
Penalizes unnecessary features	No	Yes

---

6. What are the different methods used for scaling?

1. Min-Max Scaling (Normalization)

- Formula:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Range:

Usually scaled between **0 and 1**

- Characteristics:

  - Preserves shape of distribution
  - Sensitive to outliers
  - Useful when data does not follow normal distribution

- When to Use:

  - Neural Networks
  - Image processing
  - When bounded range is required

2. Standardization (Z-Score Scaling)

- Formula:

$$X' = \frac{X - \mu}{\sigma}$$

Where:

- μ = Mean
- σ = Standard Deviation

- Properties:

  - Mean becomes 0
  - Standard deviation becomes 1
  - Output range is not fixed

- Advantages:

  - Less affected by outliers compared to Min-Max
  - Works well when data follows normal distribution

- When to Use:

  - Linear Regression
  - Logistic Regression

- SVM
- PCA

3. Robust Scaling

- Formula:

$$X' = \frac{X - Median}{IQR}$$

Where:

- IQR = Q3 - Q1

- Characteristics:

  - Uses Median and Interquartile Range
  - Not affected much by outliers

- When to Use:

  - When dataset contains many outliers

4. MaxAbs Scaling

- Formula:

$$X' = \frac{X}{|X_{max}|}$$

- Characteristics:
  - Scales data between -1 and 1
  - Preserves sparsity
  - Does not shift/center data

- When to Use:

  - Sparse datasets (like text data)

5. Unit Vector Scaling (Normalization to Unit Norm)

- Concept:

Scales feature vector so that its length (magnitude) becomes 1.

$$X' = \frac{X}{||X||}$$

- When to Use:
  - Text classification
  - Cosine similarity models
  - KNN

Method	Outlier Sensitive	Output Range	Best For
Min-Max	Yes	0 to 1	Neural Networks
Standardization	Moderate	No fixed range	Regression, SVM
Robust Scaling	No	No fixed range	Outlier-heavy data
MaxAbs	Yes	-1 to 1	Sparse data
Unit Norm	No	Unit length	Similarity models

---

7. **Explain different encoding techniques?**

   Encoding is the process of converting **categorical data (text labels)** into **numerical format** so that Machine Learning algorithms can process them.

   Most ML algorithms work with numbers, not text.
   Example:
   Gender → Male, Female → must be converted into numbers.

Different Encoding Techniques

1. Label Encoding

- Concept:

Assigns a unique integer to each category.

Example:

Category	Encoded
Red	0
Blue	1
Green	2

- Advantages:

  - Simple and fast
  - Works well for ordinal data

- Disadvantages:

  - Creates artificial order in nominal data
  - Model may assume Blue > Red

- Best Used:
  - Ordinal categories (Low, Medium, High)

2. One-Hot Encoding

- Concept:

Creates separate binary (0/1) columns for each category.

Example:

Color	Red	Blue	Green
Red	1	0	0
Blue	0	1	0

- Advantages:
  - No false ordering
  - Works well for nominal data
- Disadvantages:
  - Increases dimensionality
  - Not suitable for high-cardinality features
- Best Used:
  - Nominal categorical variables
  - Small number of categories

3. Ordinal Encoding

- Concept:

Assigns numbers based on meaningful order.

Example:

Education	Encoded
High School	1
Bachelor	2
Master	3
PhD	4

- Best Used:
  - When categories have natural ranking

4. Target Encoding (Mean Encoding)

- Concept:

Replace category with mean of target variable for that category.

Example:
If average salary for:

- IT = 50000
- HR = 30000

Then IT → 50000, HR → 30000

◇ Advantages:

- Handles high-cardinality data
- Reduces dimensionality

◇ Disadvantages:

- Risk of data leakage
- Needs careful cross-validation

5. Frequency Encoding

◇ Concept:

Replace category with its frequency in dataset.

Example:

City	Frequency
Delhi	100
Mumbai	80

◇ Advantage:

- Works well for high-cardinality features

6. Binary Encoding

◇ Concept:

- First apply Label Encoding
- Then convert number to binary form
- Create separate binary columns

Reduces dimensionality compared to One-Hot.

7. Hash Encoding

◇ Concept:

Uses hash function to map categories into fixed number of columns.

◇ Advantages:

- Efficient for very large datasets
- Fixed number of features

- Disadvantages:
  - Possible collisions

Encoding	Creates Order?	Dimensionality	Best For
Label	Yes	Low	Ordinal
One-Hot	No	High	Nominal
Ordinal	Yes	Low	Ranked data
Target	No	Low	High-cardinality
Frequency	No	Low	Large categories
Binary	No	Medium	Large categories
Hash	No	Fixed	Very large datasets

---

8. Explain different methods to handle Outliers?

Outliers are extreme data points that significantly differ from other observations in a dataset.
They may occur due to:

- Data entry errors
- Measurement errors
- Experimental errors
- Genuine rare events

Outliers can distort statistical measures (mean, variance) and negatively affect Machine Learning models, especially regression models.

Different Methods to Handle Outliers

Handling outliers depends on whether they are errors or genuine observations.

1. Removing Outliers

- Concept:

Delete rows containing extreme values.

- When to Use:
  - When outliers are due to data entry or measurement errors.
  - When dataset is large enough.
- Limitation:
  - Risk of losing important information.

2. Z-Score Method

- Formula:

$$Z = \frac{X - \mu}{\sigma}$$

  - μ = Mean
  - σ = Standard deviation

- Rule:

If |Z| > 3 → Considered outlier.

- Best For:

  - Normally distributed data.

- Limitation:

  - Sensitive to extreme values.

3. IQR (Interquartile Range) Method

- Steps:

  1. Calculate Q1 (25th percentile)
  2. Calculate Q3 (75th percentile)
  3. Compute IQR = Q3 - Q1

- Rule:

Lower Bound = Q1 − 1.5 × IQR
Upper Bound = Q3 + 1.5 × IQR

Values outside this range are outliers.

- Advantage:

  - Robust to non-normal data.
  - Less sensitive than Z-score.

4. Winsorization (Capping)

- Concept:

Instead of removing outliers, cap them at a threshold.

Example:
If 99th percentile = 100,
Replace all values above 100 with 100.

- Advantage:

  - Preserves data size.
  - Reduces impact of extreme values.

5. Log Transformation

- Concept:

Apply log function to reduce skewness.

$$X' = \log(X)$$

- Best For:

  - Right-skewed distributions.
  - Financial or income data.

6. Square Root or Box-Cox Transformation

Used to normalize skewed data and reduce extreme effects.

7. Using Robust Statistical Methods

Instead of removing outliers:

- Use **Median** instead of Mean
- Use **Robust Scaler**
- Use algorithms less sensitive to outliers (Decision Trees, Random Forest)

8. Model-Based Detection

- Isolation Forest
- Local Outlier Factor (LOF)
- DBSCAN

Used for anomaly detection in large datasets.

Method	Removes Data?	Sensitive to Distribution	Best For
Remove	Yes	No	Error values
Z-score	Yes	Yes (Normal)	Normal data
IQR	Yes	No	Skewed data
Winsorization	No	No	Preserve dataset
Log Transform	No	Skewed data	Right-skewed
Robust Models	No	No	ML pipelines

---

9. What is Logistic Regression?

Logistic Regression is a **supervised machine learning algorithm** used for **classification problems**, especially **binary classification** (two classes).

Despite its name, it is **not a regression algorithm** in the traditional sense. It is used to predict the **probability** that a given input belongs to a particular class.

◇ Why is it Called "Regression"?

It is called regression because:

- It uses a **linear equation** similar to Linear Regression.
- But instead of predicting a continuous value, it predicts a **probability**.

How Classification is Done

After calculating probability:

- If probability $\geq 0.5 \rightarrow$ Class = 1
- If probability $< 0.5 \rightarrow$ Class = 0

The threshold can be changed depending on the problem.

◇ Output Interpretation

If model outputs:

- 0.90 → 90% probability of class 1
- 0.20 → 20% probability of class 1

◇ Types of Logistic Regression

1**. Binary Logistic Regression**

- Two classes (Yes/No)

2**. Multinomial Logistic Regression**

- More than two classes

3**. Ordinal Logistic Regression**

- Ordered categories

◇ Assumptions

- Dependent variable is categorical
- Observations are independent
- No strong multicollinearity
- Linear relationship between log-odds and features

◇ Advantages

- Simple and interpretable
- Outputs probability

- Works well for linearly separable data
- Fast to train

◇ **Limitations**

- Cannot handle complex nonlinear relationships
- Sensitive to outliers
- Requires feature scaling for best performance

◇ **Real-World Applications**

- Spam detection
- Disease prediction
- Credit risk analysis
- Customer churn prediction

---

10. What is the role of Sigmoid in Logistic?

Role of Sigmoid Function in Logistic Regression

The **Sigmoid function** plays a central role in **Logistic Regression** because it converts the linear output of the model into a **probability value between 0 and 1**.

1. The Problem Without Sigmoid

Logistic Regression first calculates a linear equation:

$$z = w_1x_1 + w_2x_2 + ... + b$$

The value of zzz can range from $-\infty$ **to** $+\infty$.

But in classification, we need:

- A probability value
- Between 0 and 1

This is where the **Sigmoid function** is used.

2. Sigmoid Function Formula

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

3. What Sigmoid Does

- Converts any real number into a value between **0 and 1**
- Produces an **S-shaped curve**

- Helps interpret output as probability

Example:

z value	Sigmoid Output
-10	~0
0	0.5
+10	~1

4. Why Sigmoid is Important

1. Converts Linear Output into Probability

Ensures output is meaningful for classification.

2. Enables Binary Classification

After sigmoid:

- If probability $\geq 0.5 \rightarrow$ Class 1
- If probability $< 0.5 \rightarrow$ Class 0

3. Smooth & Differentiable

- Makes gradient descent optimization possible.
- Important for minimizing loss function.

4. Models Log-Odds Relationship

Logistic Regression models the **log-odds** of probability:

$$\log\left(\frac{p}{1-p}\right) = z$$

Sigmoid converts log-odds back to probability.

5. Intuition

Think of sigmoid as a function that:

- Takes any input
- Compresses it
- Produces a probability

It acts as a **bridge between linear regression and classification**.

---

11. What are the metrics used to evaluate classification problems?

Metrics Used to Evaluate Classification Problems

In classification problems, evaluation metrics measure how well a model predicts class labels. The choice of metric depends on class balance and business requirements.

Most classification metrics are derived from the **Confusion Matrix**.

1. Confusion Matrix (Foundation of All Metrics)

For Binary Classification:

	Predicted Positive	**Predicted Negative**
Actual Positive	TP (True Positive)	FN (False Negative)
Actual Negative	FP (False Positive)	TN (True Negative)

Where:

- **TP** → Correctly predicted positive
- **TN** → Correctly predicted negative
- **FP** → Incorrectly predicted positive
- **FN** → Incorrectly predicted negative

2 Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Meaning:

Percentage of correct predictions.

Limitation:

Not reliable for **imbalanced datasets**.

3. Precision

$$Precision = \frac{TP}{TP + FP}$$

Meaning:

Out of all predicted positives, how many were correct?

Important When:

False positives are costly (e.g., spam detection).

4. Recall (Sensitivity or True Positive Rate)

$$Recall = \frac{TP}{TP + FN}$$

Meaning:

Out of actual positives, how many were correctly predicted?

Important When:

Missing positive cases is dangerous (e.g., disease detection).

5. F1-Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Meaning:

Harmonic mean of Precision and Recall.

Important When:

You need balance between Precision and Recall.

6. Specificity (True Negative Rate)

$$Specificity = \frac{TN}{TN + FP}$$

Meaning:

How well model identifies negative class.

7. ROC Curve (Receiver Operating Characteristic)

- Plots **True Positive Rate (Recall)** vs **False Positive Rate**
- Shows performance at different thresholds.

8. AUC (Area Under Curve)

- Measures area under ROC curve.
- Range: 0 to 1
- Higher value → Better model

Interpretation:

- 0.5 → Random model
- 1.0 → Perfect model

9. Log Loss (Cross-Entropy Loss)

$$LogLoss = -\frac{1}{n}\sum[y\log(p) + (1-y)\log(1-p)]$$

Meaning:

Measures how confident the model is in its predictions.

Lower log loss → Better model.

10. Matthews Correlation Coefficient (MCC)

- Balanced metric even for imbalanced data.
- Range: -1 to +1
- +1 → Perfect prediction

Metric	Best Used When	Sensitive to Imbalance
Accuracy	Balanced dataset	Yes
Precision	FP costly	Moderate
Recall	FN costly	Moderate
F1-score	Need balance	Less
Specificity	Negative class focus	Moderate
ROC-AUC	Threshold independent	No
Log Loss	Probabilities important	No
MCC	Highly imbalanced data	No

---

12. Explain all classification metrics.

1. Confusion Matrix (Foundation of All Metrics)

For Binary Classification:

	Predicted Positive	Predicted Negative
Actual Positive	TP (True Positive)	FN (False Negative)
Actual Negative	FP (False Positive)	TN (True Negative)

Where:

- **TP** → Correctly predicted positive
- **TN** → Correctly predicted negative
- **FP** → Incorrectly predicted positive
- **FN** → Incorrectly predicted negative

2 Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Meaning:

Percentage of correct predictions.

Limitation:

Not reliable for **imbalanced datasets**.

3. Precision

$$Precision = \frac{TP}{TP + FP}$$

Meaning:

Out of all predicted positives, how many were correct?

Important When:

False positives are costly (e.g., spam detection).

4. Recall (Sensitivity or True Positive Rate)

$$Recall = \frac{TP}{TP + FN}$$

Meaning:

Out of actual positives, how many were correctly predicted?

Important When:

Missing positive cases is dangerous (e.g., disease detection).

5. F1-Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Meaning:

Harmonic mean of Precision and Recall.

Important When:

You need balance between Precision and Recall.

6. Specificity (True Negative Rate)

$$Specificity = \frac{TN}{TN + FP}$$

Meaning:

How well model identifies negative class.

7. ROC Curve (Receiver Operating Characteristic)

- Plots **True Positive Rate (Recall)** vs **False Positive Rate**
- Shows performance at different thresholds.

8. AUC (Area Under Curve)

- Measures area under ROC curve.
- Range: 0 to 1
- Higher value → Better model

Interpretation:

- 0.5 → Random model
- 1.0 → Perfect model

9. Log Loss (Cross-Entropy Loss)

$$LogLoss = -\frac{1}{n}\sum[y\log(p) + (1-y)\log(1-p)]$$

Meaning:

Measures how confident the model is in its predictions.

Lower log loss → Better model.

10. Matthews Correlation Coefficient (MCC)

- Balanced metric even for imbalanced data.
- Range: -1 to +1
- +1 → Perfect prediction

11. Fβ-Score

$$F_\beta = (1+\beta^2)\frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}$$

- $\beta > 1$ → More weight to Recall
- $\beta < 1$ → More weight to Precision

12. False Positive Rate (FPR)

$$FPR = \frac{FP}{FP + TN}$$

Probability of incorrectly predicting positive.

13.Precision-Recall Curve

- Plots Precision vs Recall.
- More useful for **imbalanced datasets**.

14.Matthews Correlation Coefficient (MCC)

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Range:

- +1 → Perfect prediction
- 0 → Random
- -1 → Totally wrong

Best for highly imbalanced data.

14. Balanced Accuracy

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$$

Useful when dataset is imbalanced.

15. Cohen's Kappa

Measures agreement between predicted and actual classes while correcting for chance agreement.

Metric	Focus	Best For
Accuracy	Overall correctness	Balanced data
Precision	FP control	Spam detection
Recall	FN control	Medical diagnosis
F1	Balance of P & R	Imbalanced data
Specificity	TN accuracy	Screening tests

ROC-AUC	Threshold independent	General evaluation
PR Curve	Precision vs Recall	Imbalanced datasets
Log Loss	Probability confidence	Probabilistic models
MCC	Balanced evaluation	Highly imbalanced
Balanced Accuracy	Class balance	Imbalanced data

---

13. What are the ways to handle missing values?

What Are Missing Values?

Missing values occur when no data value is stored for a variable in an observation.
They may appear as:

- NULL
- NaN
- Blank cells
- Special symbols (?, -999)

Missing data can reduce model accuracy and cause bias if not handled properly.

◇ **Ways to Handle Missing Values**

Handling missing values depends on:

- Percentage of missing data
- Data type (Numerical / Categorical)
- Business context
- Model being used

1. Deletion Methods

◇ **A) Remove Rows (Listwise Deletion)**

Remove records that contain missing values.

When to Use:

- Missing data percentage is small (<5%)
- Large dataset available

Disadvantage:

- Loss of valuable data

◇ **B) Remove Columns**

Drop entire feature if it has too many missing values (e.g., >40–50%).

When to Use:

- Feature is not important
- Too many missing entries

2. Imputation Methods (Replacing Missing Values)

- **A) Mean Imputation (Numerical Data)**

Replace missing values with mean of column.

Advantage:

- Simple and fast

Disadvantage:

- Affected by outliers

- **B) Median Imputation (Numerical Data)**

Replace missing values with median.

Advantage:

- Robust to outliers
- Best for skewed data

- **C) Mode Imputation (Categorical Data)**

Replace missing values with most frequent category.

3. Advanced Imputation Methods

- **A) KNN Imputation**

- Finds nearest neighbors
- Uses average of similar samples

Advantage:

- More accurate than mean/median

Disadvantage:

- Computationally expensive

- **B) Regression Imputation**

Predict missing values using regression model.

Example:
Use other features to predict missing salary.

- **C) Iterative Imputation (MICE)**

  - Predicts missing values multiple times
  - More sophisticated approach

4. Forward Fill / Backward Fill (Time Series Data)

- **Forward Fill:**

Replace missing value with previous value.

- **Backward Fill:**

Replace with next available value.

Best for:

- Time-series datasets

5. Constant Value Imputation

Replace missing values with a constant like:

- 0
- "Unknown"
- -1

Used when:

- Missing itself carries meaning

6. Using Algorithms That Handle Missing Values

Some models can handle missing data directly:

- Decision Trees
- Random Forest
- XGBoost

Method	Data Type	Best Used When
Remove Rows	Any	Very few missing values
Remove Columns	Any	Too many missing values
Mean	Numerical	Normal distribution
Median	Numerical	Skewed data
Mode	Categorical	Nominal data
KNN	Numerical	Pattern-based filling
Regression	Numerical	Strong feature relationship
Forward Fill	Time series	Sequential data
Constant	Any	Missing has meaning

14. What are Outliers ?

Outliers are data points that are significantly different from the majority of observations in a dataset. They lie far away from other values and may represent abnormal behavior, rare events, or errors.

◇ Simple Definition

An outlier is an observation that deviates so much from other observations that it raises suspicion of being generated by a different process.

◇ Example

Consider the following salary data (in lakhs):

5, 6, 7, 8, 6, 7, **120**

Here, **120** is an outlier because it is much larger than the rest of the values.

◇ Causes of Outliers

1. Data entry errors
2. Measurement errors
3. Experimental errors
4. Genuine rare events
5. Natural variability

◇ Types of Outliers

1. Global Outliers

Very far from entire dataset.

2. Contextual Outliers

Abnormal in specific context (e.g., high temperature in winter).

3. Collective Outliers

Group of values that behave unusually together.

◇ How to Detect Outliers

1. Statistical Methods

- **Z-score method**
- **IQR (Interquartile Range)** method

2. Visualization Methods

- Boxplot
- Scatter plot
- Histogram

3. Machine Learning Methods

- Isolation Forest
- DBSCAN
- Local Outlier Factor

◇ **Why Are Outliers Important?**

Negative Impact:

- Distort mean and standard deviation
- Affect regression line
- Reduce model accuracy

Positive Impact:

- May indicate fraud
- May show rare but important patterns
- Can reveal valuable business insights

◇ **Effect on Machine Learning Models**

Model Type	Impact of Outliers
Linear Regression	Highly affected
KNN	Affected
SVM	Affected
Decision Tree	Less affected
Random Forest	Less affected

---

15. What are the steps that we have to take while building regression projects?

Steps to Follow While Building a Regression Project

Building a regression project follows a structured Machine Learning lifecycle. Each step ensures that the model is accurate, reliable, and production-ready.

1. Problem Definition

- Clearly define the business problem.
- Identify the **target variable (dependent variable)**.
- Understand what needs to be predicted (e.g., price, sales, demand).
- Define success criteria (RMSE, R^2 threshold).

2. Data Collection

- Gather relevant data from databases, APIs, CSV files, etc.
- Ensure sufficient quantity and quality of data.
- Verify data relevance to the problem.

3. Data Understanding (Exploratory Data Analysis - EDA)

- Understand feature types (numerical/categorical).

- Analyze distributions.
- Check correlations.
- Identify patterns and trends.
- Detect missing values and outliers.

4. Data Cleaning

- Remove duplicate records.
- Fix incorrect data types.
- Standardize inconsistent formats.
- Handle noisy data.

5. Handle Missing Values

- Remove rows/columns (if minimal).
- Impute using mean/median/mode.
- Use advanced methods like KNN imputation.

6. Handle Outliers

- Detect using Z-score or IQR.
- Remove, cap (Winsorize), or transform.
- Decide based on business context.

7. Feature Engineering

- Create new meaningful features.
- Combine or transform variables.
- Apply log transformation if required.
- Encode categorical variables.

8. Feature Scaling

- Apply Standardization or Min-Max scaling.
- Important for gradient-based models.

9. Train-Test Split

- Split dataset (e.g., 80% train, 20% test).
- Prevent data leakage.
- Optionally use cross-validation.

10. Model Selection

Choose suitable regression algorithms:

- Linear Regression
- Ridge / Lasso
- Decision Tree Regressor
- Random Forest
- Gradient Boosting

11. Model Training

- Fit model on training data.
- Optimize parameters.

12.Model Evaluation

Evaluate using:

- MAE
- MSE
- RMSE
- R^2
- Adjusted R^2

Compare training vs testing performance.

13.Hyperparameter Tuning

- Grid Search
- Random Search
- Cross-validation

Improve model performance.

14. Model Validation

- K-Fold Cross Validation
- Ensure model generalizes well.

15. Model Deployment

- Save model (Pickle/Joblib).
- Deploy using API or web app.
- Integrate into production system.

16. Monitoring & Maintenance

- Track model performance over time.
- Detect data drift.
- Retrain when necessary.

Summary Workflow

Problem → Data → EDA → Cleaning → Feature Engineering → Scaling → Train/Test Split → Model → Evaluation → Tuning → Deployment → Monitoring

Key Points to Remember

- Avoid data leakage.
- Always validate model.
- Choose evaluation metric carefully.
- Understand business context before removing data.