**🧮 1. Linear Regression**

**Definition**

Linear Regression is a **supervised learning algorithm** used to find the relationship between a **dependent variable (Y)** and one or more **independent variables (X)** using a straight line (best fit line).

Where:

* **m** = slope of the line (regression coefficient)
* **c** = intercept
* **Y** = predicted output

**Mathematical Formula**

**Steps**

1. Collect data points (X, Y).
2. Calculate mean of X and Y.
3. Compute slope (m) and intercept (c).
4. Form the regression equation .
5. Predict Y for any new X value.

**Advantages**

* Simple to implement and interpret.
* Works well for linearly related data.

**Limitations**

* Only captures linear relationships.
* Sensitive to outliers.

**Applications**

* Predicting house prices, sales, temperature, etc.

**📊 2. Multivariate Linear Regression**

**Definition**

Multivariate Linear Regression extends simple linear regression to multiple predictors.

**Mathematical Model**

where **B** is the vector of coefficients.

**Steps**

1. Prepare matrix X (features) and Y (target).
2. Add bias term (column of 1s).
3. Compute coefficients using the Normal Equation.
4. Predict Y using .

**Advantages**

* Handles multiple input variables.
* Provides stronger prediction power.

**Limitations**

* Assumes linearity and independence.
* Sensitive to multicollinearity.

**Applications**

* Economic forecasting, business analytics.

**🧠 3. Logistic Regression**

**Definition**

Logistic Regression is a **classification algorithm** that predicts categorical outcomes (0 or 1) using a sigmoid function.

**Concept**

* Converts linear regression output into probability between 0 and 1.
* Decision rule:
  + If
  + Else

**Steps**

1. Initialize weights and bias.
2. Compute prediction using sigmoid function.
3. Calculate error.
4. Update weights using Gradient Descent.
5. Repeat until convergence.

**Advantages**

* Simple and efficient for binary classification.
* Outputs probabilities.

**Limitations**

* Works only for linear decision boundaries.

**Applications**

* Email spam detection, disease prediction, churn analysis.

**🌳 4. CART (Classification and Regression Tree)**

**Definition**

CART builds a **decision tree** to split the data into subsets based on the best feature, creating branches until the stopping condition is met.

**Concepts**

* **Impurity measures:**
  + **Gini Index:**
  + **Entropy:**
* For regression, splits minimize **variance**.

**Steps**

1. Calculate impurity for all features.
2. Split data at the feature giving maximum information gain.
3. Repeat recursively.
4. Stop when nodes are pure or max depth reached.

**Advantages**

* Easy to interpret.
* Handles both classification & regression.

**Limitations**

* Overfitting if tree is too deep.

**Applications**

* Credit risk analysis, medical diagnosis, customer segmentation.

**🧩 5. Bagging (Bootstrap Aggregating)**

**Definition**

Bagging is an **ensemble method** that trains multiple models on random subsets of data (sampled with replacement) and averages their predictions.

**Concept**

* Reduces **variance** and improves **stability**.
* Common example: **Random Forest** (bagging of decision trees).

**Steps**

1. Create multiple bootstrap samples.
2. Train separate models on each sample.
3. Aggregate outputs (average or vote).

**Advantages**

* Reduces overfitting.
* Improves model accuracy.

**Limitations**

* Increases computational cost.

**Applications**

* Random Forests, financial risk models.

**⚡ 6. Boosting**

**Definition**

Boosting is a **sequential ensemble technique** where each new model focuses on errors made by previous models.

**Concept**

* Combines weak learners (usually decision trees) into a strong learner.
* Each model is weighted based on its accuracy.

**Steps**

1. Initialize equal weights for all samples.
2. Train weak learner.
3. Increase weights of misclassified samples.
4. Repeat and combine models with weighted votes.

**Advantages**

* High accuracy.
* Works well with weak models.

**Limitations**

* Sensitive to noise.
* Slower than bagging.

**Applications**

* AdaBoost, Gradient Boosting, XGBoost.

**📈 7. Support Vector Machine (SVM)**

**Definition**

SVM finds an **optimal hyperplane** that separates data points of different classes with **maximum margin**.

**Mathematical Model**

Margin = distance between hyperplane and closest points (support vectors).

**Concept**

* Maximizes separation margin.
* Uses **kernel trick** for non-linear data (e.g., RBF, polynomial).

**Advantages**

* Effective in high-dimensional space.
* Works well for clear margins.

**Limitations**

* Computationally expensive for large datasets.

**Applications**

* Image recognition, text classification, handwriting detection.

**🌐 8. Graph-Based Clustering**

**Definition**

Graph-based clustering represents data as a **graph** where nodes = data points and edges = similarities.

**Concept**

* Clusters are found by removing weak edges or finding **connected components**.
* Example: **Spectral Clustering** uses eigenvalues of Laplacian matrix.

**Steps**

1. Compute similarity (adjacency) matrix.
2. Build graph.
3. Apply partitioning (e.g., K-means on eigenvectors).
4. Extract clusters.

**Advantages**

* Works for non-convex shapes.
* Can detect complex cluster structures.

**Limitations**

* Requires similarity threshold.
* Computationally expensive.

**Applications**

* Image segmentation, social network analysis.

**📍 9. DBSCAN**

**Definition**

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clusters points based on **density connectivity**.

**Concept**

* **Core point:** Has ≥ MinPts within ε distance.
* **Border point:** Fewer neighbors but close to a core.
* **Noise:** Not reachable from any core.

**Steps**

1. Choose ε and MinPts.
2. Visit each point:
   * If unvisited → find neighbors within ε.
   * If neighbors ≥ MinPts → form cluster.
3. Expand cluster recursively.

**Advantages**

* Can find clusters of any shape.
* Detects noise/outliers.

**Limitations**

* Sensitive to ε and MinPts.
* Not ideal for varying density datasets.

**Applications**

* GPS data clustering, anomaly detection.

**📉 10. PCA (Principal Component Analysis)**

**Definition**

PCA is a **dimensionality reduction** technique that transforms correlated features into uncorrelated **principal components**.

**Concept**

* Finds directions (eigenvectors) capturing maximum variance.
* First component explains most variance.

where W = eigenvectors of covariance matrix.

**Steps**

1. Standardize data.
2. Compute covariance matrix.
3. Find eigenvalues & eigenvectors.
4. Sort by descending eigenvalues.
5. Project data onto top k components.

**Advantages**

* Reduces dimensions while preserving information.
* Removes multicollinearity.

**Limitations**

* Loses interpretability.
* Assumes linear relationships.

**Applications**

* Image compression, noise reduction, data visualization.

**🧾 11. LDA (Linear Discriminant Analysis)**

**Definition**

LDA is a **supervised dimensionality reduction** technique that maximizes class separability.

**Concept**

* Maximizes **between-class variance** and minimizes **within-class variance**.

is used to compute discriminant directions.

**Steps**

1. Compute class means.
2. Compute Within-class scatter (SW).
3. Compute Between-class scatter (SB).
4. Solve eigenvalue problem .
5. Select top eigenvectors and project data.

**Advantages**

* Good for classification.
* Works even with small datasets.

**Limitations**

* Assumes normal distribution.
* Requires labeled data.

**Applications**

* Face recognition, medical diagnostics.

**🔢 12. SVD (Singular Value Decomposition)**

**Definition**

SVD decomposes a matrix A into three components:

Where:

* **U** → Left singular vectors
* **Σ** → Diagonal matrix of singular values
* **V^T** → Right singular vectors

**Concept**

* Captures essential information with fewer dimensions.
* Used in data compression, noise reduction, and latent factor analysis.

**Steps**

1. Compute .
2. Keep top k singular values.
3. Reconstruct approximation .

**Advantages**

* Robust and stable.
* Reveals hidden structure in data.

**Limitations**

* Computationally expensive for large matrices.

**Applications**

* Image compression, recommender systems, NLP (Latent Semantic Analysis).