**Introduction to Machine Learning**

**What is learning?**

**Types of learning**

**Applications of machine learning**

**Introduction to Machine Learning**

**What is Machine Learning?**

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that enables systems to learn patterns from data and make decisions without being explicitly programmed. It focuses on developing algorithms that improve automatically through experience.

A widely accepted definition by Tom M. Mitchell states:  
*"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."*

For example, in email spam filtering:

* **T** = Classifying emails as spam or not spam
* **E** = Training on historical email data
* **P** = Accuracy of classification

**What is Learning?**

Learning in ML refers to the ability of an algorithm to identify patterns, adapt, and improve its performance on a task based on past data. It mimics human learning by analyzing data and making predictions or decisions accordingly.

A learning model consists of:

1. **Input Data (X):** The features used for learning
2. **Model:** The mathematical representation of the problem
3. **Output (Y):** The predicted or classified result
4. **Feedback:** Error analysis to improve the model

**Types of Learning**

Machine learning is broadly classified into three main types:

**1. Supervised Learning**

* The model is trained using labeled data, where input-output pairs are known.
* The goal is to learn a mapping function from input to output.
* Examples:
  + **Classification:** Predicting categories (e.g., spam detection)
  + **Regression:** Predicting continuous values (e.g., stock price prediction)

🔹 *Algorithms:* Linear Regression, Logistic Regression, Decision Trees, Support Vector Machines (SVM), Neural Networks

**2. Unsupervised Learning**

* The model is trained on **unlabeled data** without predefined outputs.
* It identifies hidden structures or patterns in data.
* Examples:
  + **Clustering:** Grouping similar data points (e.g., customer segmentation)
  + **Association:** Finding relationships (e.g., market basket analysis)

🔹 *Algorithms:* K-Means, Hierarchical Clustering, Apriori, Principal Component Analysis (PCA)

**3. Reinforcement Learning**

* The model learns by interacting with an environment and receiving rewards or penalties.
* It aims to maximize cumulative rewards over time.
* Used in decision-making tasks where sequential actions matter.
* Examples:
  + Self-driving cars
  + Game-playing AI (e.g., AlphaGo)

🔹 *Algorithms:* Q-Learning, Deep Q Networks (DQN), Policy Gradient Methods

**Applications of Machine Learning**

ML is widely used in various fields, revolutionizing industries with automation and predictive capabilities.

**1. Healthcare**

* Disease detection and diagnosis (e.g., cancer detection using deep learning)
* Drug discovery and personalized medicine
* Medical image analysis

**2. Finance**

* Fraud detection in banking transactions
* Stock market prediction
* Credit scoring and risk assessment

**3. E-commerce & Retail**

* Recommendation systems (e.g., Amazon, Netflix)
* Customer segmentation for targeted marketing
* Demand forecasting

**4. Autonomous Systems**

* Self-driving cars (Tesla, Waymo)
* Robotics and industrial automation

**5. Natural Language Processing (NLP)**

* Chatbots and virtual assistants (Siri, Google Assistant)
* Machine translation (Google Translate)
* Sentiment analysis (social media monitoring)

**6. Cybersecurity**

* Intrusion detection and prevention
* Malware analysis
* Threat detection and response

**7. Smart Agriculture**

* Crop yield prediction
* Pest detection
* Precision farming

**Conclusion**

Machine Learning is a rapidly evolving field that enables computers to learn and make decisions from data without explicit programming. Understanding different learning types and their applications is crucial for leveraging ML in real-world problems.

**Supervised Learning**

**Classification**

**k-NN, Naive Bayesian, Decision Trees, Support Vector Machines, Logistic**

**Regression**

**Introduction to Artificial Neural Networks – Percentron, MLP**

**Evaluation measures**

**Supervised Learning**

**What is Supervised Learning?**

Supervised learning is a type of machine learning where the model learns from labeled data. It maps input features (X) to output labels (Y) using a function f(X)=Y.

* **Training Phase:** The model learns patterns from labeled data.
* **Testing Phase:** The model is evaluated on new, unseen data.

**Example Use Cases**

* Email spam detection (Spam or Not Spam)
* Handwritten digit recognition (0-9 classification)
* Medical diagnosis (Disease or No Disease)

**Classification in Supervised Learning**

Classification is a supervised learning task where the goal is to assign an input to one of several predefined categories. It deals with **categorical output** (e.g., Yes/No, Fraud/Not Fraud).

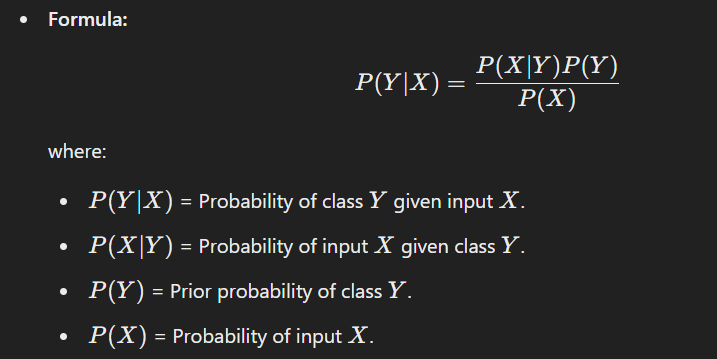
**Key Classification Algorithms**

**1. k-Nearest Neighbors (k-NN)**

* **Concept:** Classifies a data point based on the majority class of its k nearest neighbors.
* **Working:**
  1. Choose a value for k.
  2. Compute the distance (e.g., Euclidean) between the new data point and all training points.
  3. Select the k closest neighbors.
  4. Assign the most common class among these neighbors.
* **Pros:** Simple, effective for small datasets.
* **Cons:** Slow for large datasets, sensitive to irrelevant features.

**2. Naïve Bayes Classifier**

* **Concept:** A probabilistic classifier based on **Bayes' theorem** with the assumption of independence between features.

****

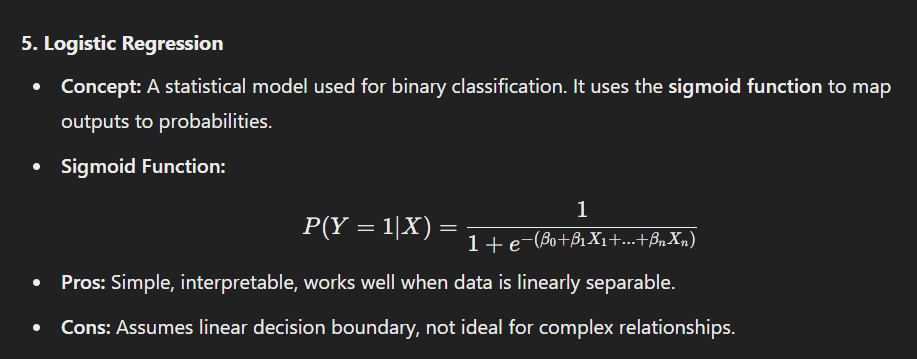
* **Types:**
  + Gaussian Naïve Bayes (for continuous data)
  + Multinomial Naïve Bayes (for text classification)
  + Bernoulli Naïve Bayes (for binary features)
* **Pros:** Fast, works well with high-dimensional data (e.g., text classification).
* **Cons:** Assumes feature independence, which may not hold in reality.

**3. Decision Trees**

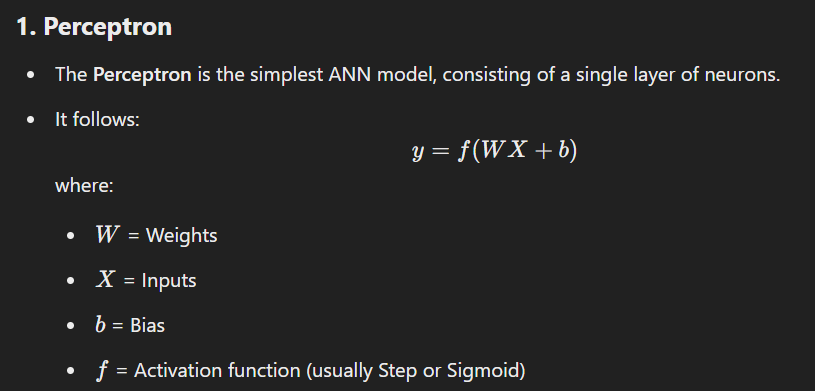
* **Concept:** A tree-like structure where each node represents a decision based on feature values.
* **Working:**
  1. Select the best feature using **Information Gain (IG)** or **Gini Index**.
  2. Split data based on the selected feature.
  3. Repeat until reaching a stopping condition (e.g., max depth).
* **Pros:** Easy to interpret, handles both numerical & categorical data.
* **Cons:** Prone to overfitting, sensitive to noisy data.

**4. Support Vector Machines (SVM)**

* **Concept:** Finds the best hyperplane that separates data points from different classes with the maximum margin.
* **Working:**
  + Uses **Kernel Trick** to handle non-linearly separable data (e.g., RBF, Polynomial).
  + **Soft margin SVM** allows some misclassification for better generalization.
* **Pros:** Works well for high-dimensional data, effective for classification.
* **Cons:** Computationally expensive for large datasets.

****

**Introduction to Artificial Neural Networks (ANNs)**

****

**Limitations of Perceptron**

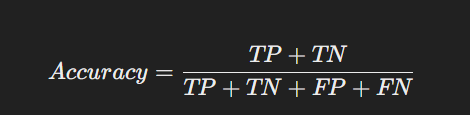
* Can only classify **linearly separable** data (e.g., AND, OR but not XOR).

**2. Multi-Layer Perceptron (MLP)**

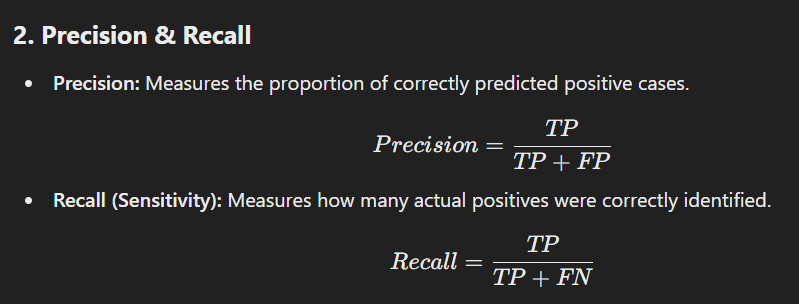
* A feedforward neural network with **hidden layers**.
* Uses activation functions like **ReLU, Sigmoid, Tanh** to introduce non-linearity.
* **Backpropagation Algorithm** is used for training:
  1. **Forward Propagation:** Compute output using weights.
  2. **Error Calculation:** Compare with actual output.
  3. **Backward Propagation:** Adjust weights using Gradient Descent.
* **Applications:** Image recognition, NLP, fraud detection.

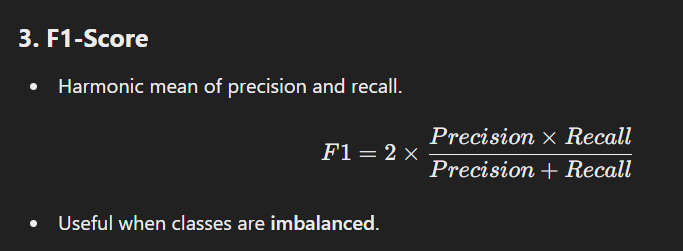
**Evaluation Measures for Classification Models**

**1. Accuracy**



* Measures correct predictions out of total predictions.
* Not ideal for **imbalanced datasets** (e.g., detecting rare diseases).

****

****

**4. ROC Curve & AUC**

* **Receiver Operating Characteristic (ROC) Curve** plots **True Positive Rate (TPR) vs. False Positive Rate (FPR)**.
* **AUC (Area Under Curve):** Measures the classifier’s ability to distinguish between classes. Higher values indicate better performance.

**Conclusion**

Supervised learning is a fundamental ML approach that uses labeled data to train models. Classification techniques like k-NN, Naïve Bayes, Decision Trees, SVMs, and Logistic Regression are widely used in various domains. Artificial Neural Networks (ANNs) add more power by handling complex patterns. Evaluating models with accuracy, precision, recall, F1-score, and ROC helps in selecting the best approach.

**Supervised Learning**

**Regression**

**Linear regression, Ridge, Lasso**

**Evaluation measures**

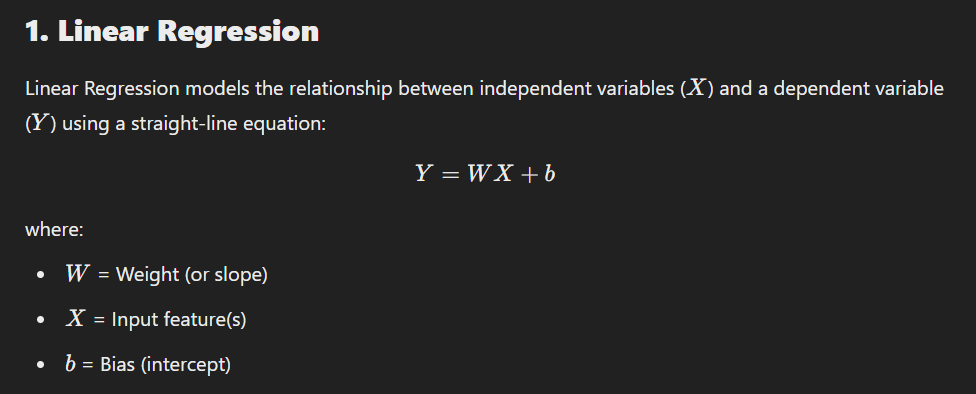
**Supervised Learning: Regression**

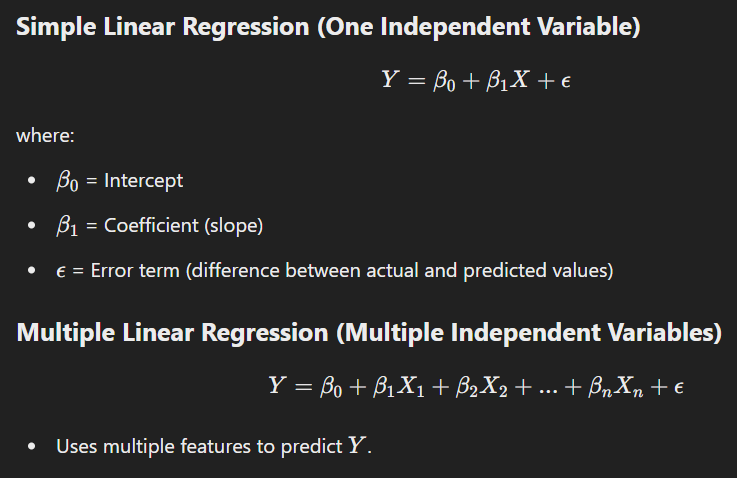
**What is Regression?**

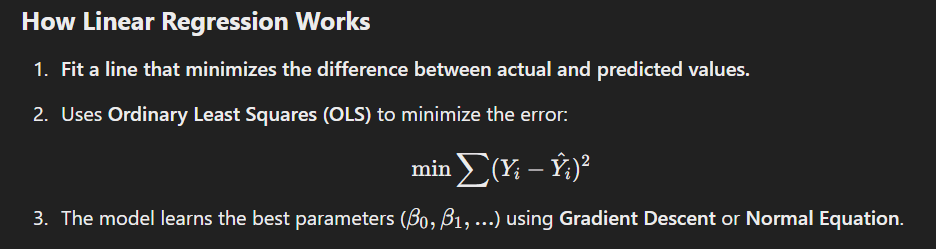
Regression is a type of supervised learning where the goal is to predict a **continuous numerical value** based on input features. Unlike classification, which assigns labels, regression models estimate relationships between variables and make predictions within a continuous range.

**Examples of Regression Applications:**

* **Stock Market Prediction:** Predicting future stock prices.
* **House Price Estimation:** Predicting house prices based on area, location, and features.
* **Weather Forecasting:** Predicting temperature, humidity, or rainfall.

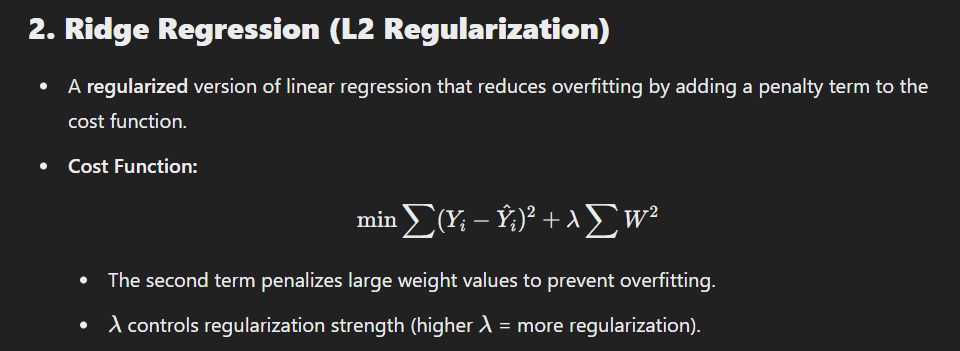
****

****

****

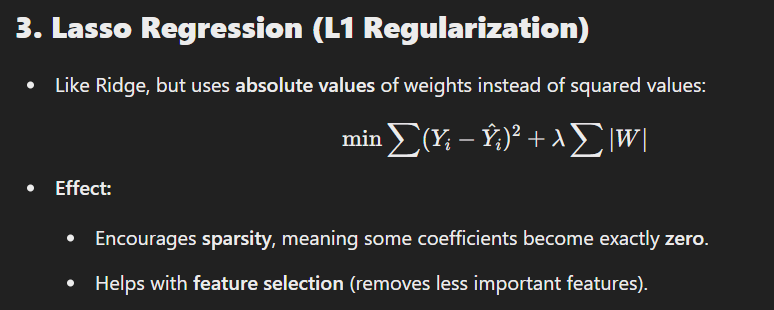
**Limitations of Linear Regression**

* Assumes a **linear** relationship between variables.
* Sensitive to **outliers**.
* Cannot handle **multicollinearity** (high correlation between independent variables).

****

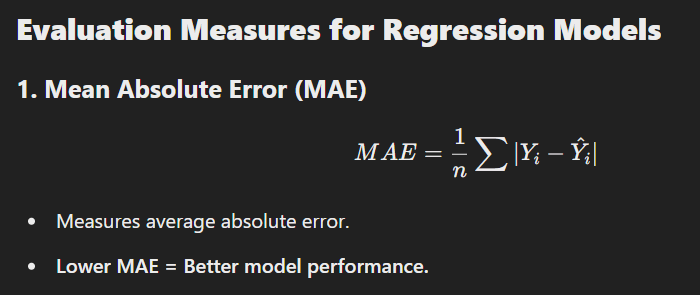
**Advantages:**  
✅ Prevents overfitting  
✅ Works well with multicollinearity  
✅ Suitable for complex datasets

**Disadvantages:**  
❌ Cannot shrink coefficients to exactly zero (unlike Lasso)

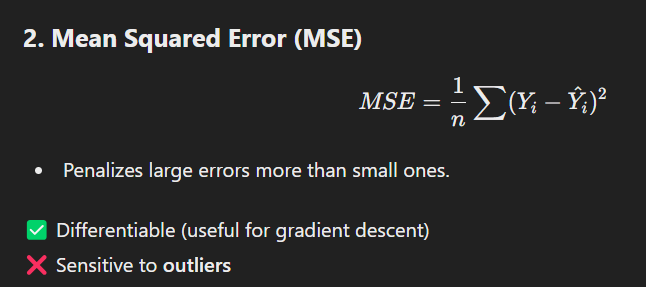
****

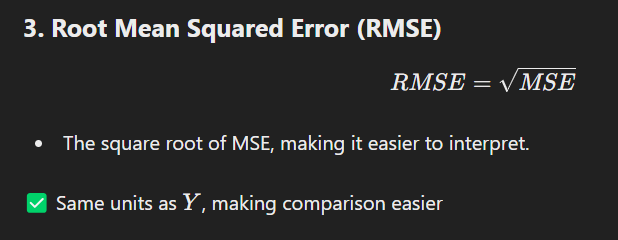
**Advantages:**  
✅ Reduces overfitting  
✅ Performs automatic **feature selection**

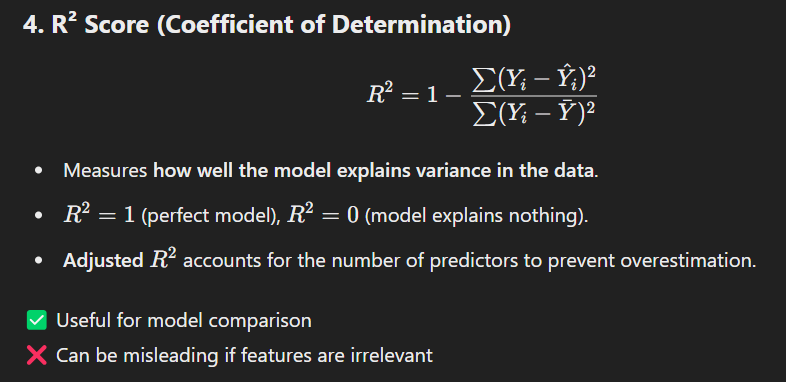
**Disadvantages:**  
❌ May discard useful variables if λ is too large

****

✅ Interpretable in real-world units  
❌ Does not penalize large errors significantly

****

****

****

**Conclusion**

Regression is a powerful technique for predicting numerical values.

* **Linear Regression** is simple and interpretable but suffers from overfitting.
* **Ridge & Lasso Regression** help reduce overfitting and improve model performance.
* **Evaluation metrics** like MAE, MSE, RMSE, and R2 help assess model accuracy.

**Unsupervised Learning**

**Clustering**

**Partion-based (k-Means, k-Medoid, k-Mode, PAM)**

**Hierarchical Clustering (AGNES, DIANA)**

**Density based Clustering**

**Evaluation measures**

**Unsupervised Learning: Clustering**

**What is Unsupervised Learning?**

Unsupervised learning is a type of machine learning where the model learns patterns from **unlabeled** data. Unlike supervised learning, where the algorithm has predefined labels, unsupervised learning identifies hidden structures in data without explicit supervision.

**What is Clustering?**

Clustering is an **unsupervised learning technique** used to group similar data points together based on their characteristics. It is useful for:

* Customer segmentation in marketing
* Anomaly detection in fraud detection
* Image compression and pattern recognition

**Types of Clustering**

**1. Partition-Based Clustering**

Partition-based clustering divides the dataset into **k** clusters, where each data point belongs to the cluster with the nearest centroid.

**1.1 k-Means Clustering**

* **Concept:** Assigns n data points into k clusters by minimizing the variance within each cluster.
* **Algorithm:**
  1. Choose k cluster centroids randomly.
  2. Assign each data point to the nearest centroid.
  3. Compute new centroids as the mean of assigned points.
  4. Repeat until centroids stabilize (convergence).
* **Advantages:**  
  ✅ Fast and scalable  
  ✅ Works well for large datasets
* **Disadvantages:**  
  ❌ Requires predefining k  
  ❌ Sensitive to outliers

**1.2 k-Medoid Clustering (PAM - Partitioning Around Medoids)**

* **Concept:** Similar to k-Means, but instead of using the mean, it selects actual data points as cluster representatives (medoids).
* **Algorithm:**
  1. Select k medoids randomly.
  2. Assign each data point to the closest medoid.
  3. Swap medoids with non-medoids to minimize the total distance within clusters.
* **Advantages:**  
  ✅ More robust to outliers than k-Means
* **Disadvantages:**  
  ❌ Computationally expensive for large datasets

**1.3 k-Mode Clustering**

* **Concept:** Extension of k-Means for **categorical data**.
* **Algorithm:** Uses **modes** (most frequent category) instead of means for cluster centroids.
* **Advantage:** Works for categorical data like customer survey analysis.

**2. Hierarchical Clustering**

Hierarchical clustering builds a **tree-like structure** (dendrogram) to represent nested clusters.

**2.1 AGNES (Agglomerative Nesting)**

* **Concept:** A **bottom-up approach** where each point starts as its own cluster, and clusters are merged iteratively.
* **Linkage Criteria:**
  + **Single-linkage:** Merge clusters with the smallest minimum distance.
  + **Complete-linkage:** Merge clusters with the smallest maximum distance.
  + **Average-linkage:** Merge clusters based on average distances.
* **Advantages:**  
  ✅ No need to predefine k.
* **Disadvantages:**  
  ❌ Computationally expensive for large datasets.

**2.2 DIANA (Divisive Analysis Clustering)**

* **Concept:** A **top-down approach**, where all data points start in one cluster and are split recursively.
* **Advantage:**  
  ✅ More accurate than AGNES for some datasets.
* **Disadvantage:**  
  ❌ More computationally expensive.

**3. Density-Based Clustering**

Clusters are formed as dense regions separated by sparser areas.

**3.1 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

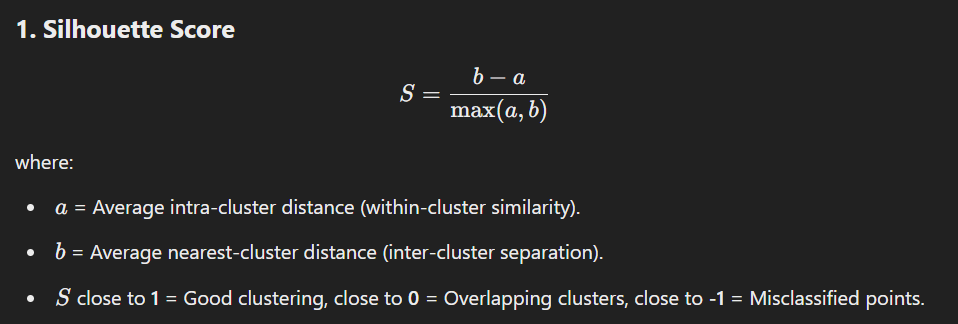
* **Concept:** Groups points based on **density** rather than distance.
* **Algorithm:**
  1. Select a random point and check if it has **minPts** neighbors within **ε-radius**.
  2. If yes, expand the cluster; otherwise, mark it as noise.
  3. Repeat for remaining points.
* **Advantages:**  
  ✅ Detects clusters of arbitrary shape.  
  ✅ Handles outliers well.
* **Disadvantages:**  
  ❌ Sensitive to **ε** and **minPts** values.

**3.2 OPTICS (Ordering Points to Identify Clustering Structure)**

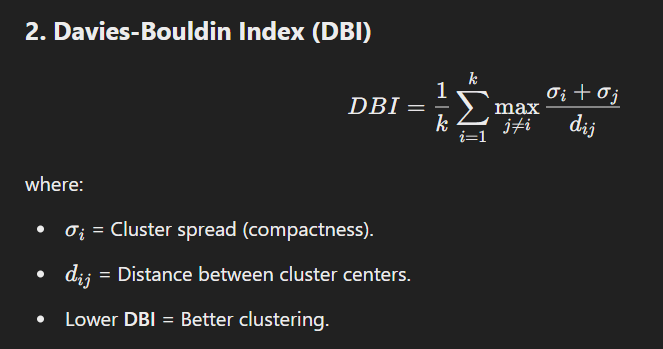
* **Concept:** Similar to DBSCAN but **adaptively changes ε** to detect clusters of varying densities.
* **Advantage:** Works better than DBSCAN when cluster densities vary.

**Evaluation Measures for Clustering**

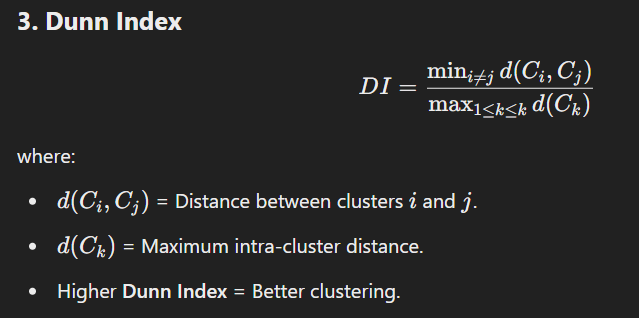
Since clustering is **unsupervised**, evaluation does not rely on predefined labels.

****

✅ Works for all clustering techniques

****

✅ Measures both compactness and separation  
❌ Works best for compact, well-separated clusters

****

✅ Good for compact and well-separated clusters  
❌ Sensitive to outliers

**4. Rand Index (RI)**

* Measures similarity between predicted and actual cluster assignments.
* RI=1 (perfect match), RI=0 (random clustering).

✅ Useful when ground truth is available

**Conclusion**

* **Partition-Based Clustering:** Fast but requires predefining k (k-Means, k-Medoid, k-Mode).
* **Hierarchical Clustering:** Builds a tree structure (AGNES, DIANA).
* **Density-Based Clustering:** Detects arbitrary-shaped clusters (DBSCAN, OPTICS).
* **Evaluation Metrics:** Silhouette Score, DBI, Dunn Index help assess cluster quality.

**Association Rule Mining**

**Association Rule Mining (ARM)**

**Introduction**

Association Rule Mining (ARM) is an **unsupervised learning** technique used to discover interesting relationships (or associations) between items in large datasets. It is widely used in **market basket analysis**, where businesses identify frequently bought item combinations to optimize sales strategies.

**Example Use Cases:**

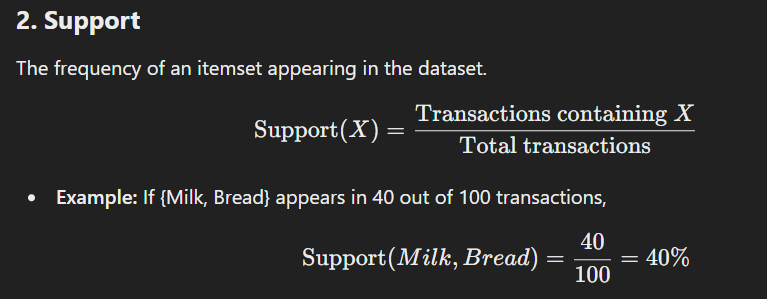
* **Retail & E-commerce:** "Customers who buy bread often buy butter"
* **Medical Diagnosis:** "Patients with symptom A and B are likely to have disease C"
* **Web Usage Mining:** "Users visiting page X often visit page Y next"

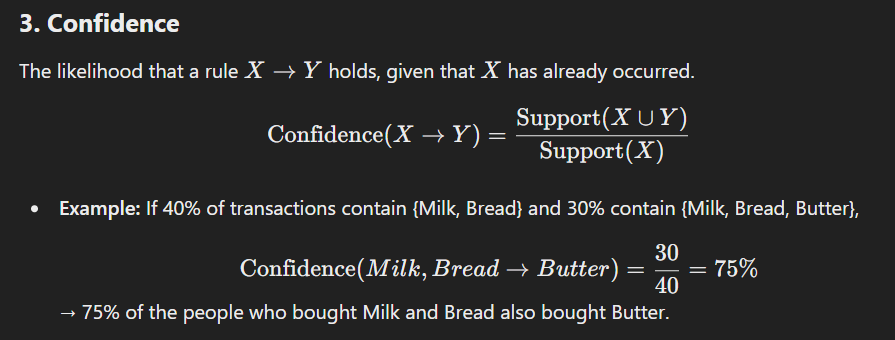
**Key Terminologies in ARM**

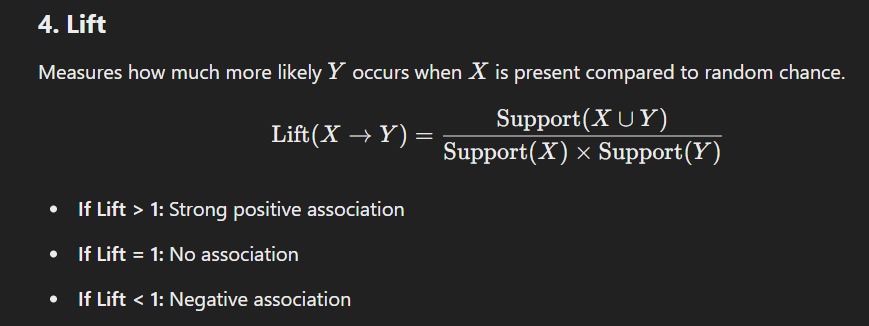
**1. Itemset**

A group of items appearing together in a transaction.

* **Example:** {Milk, Bread, Butter}

****

****

****

**Association Rule Mining Algorithms**

**1. Apriori Algorithm**

* Based on the **Apriori property**: "If an itemset is frequent, all its subsets must also be frequent."
* Works by generating frequent itemsets and then forming association rules.

**Algorithm Steps:**

1. **Generate Frequent Itemsets:**
   * Find all itemsets whose support is above a given threshold.
2. **Rule Generation:**
   * Generate association rules from frequent itemsets that meet the confidence threshold.

**Advantages:**  
✅ Simple and widely used  
✅ Works well with small to medium-sized datasets

**Disadvantages:**  
❌ Computationally expensive for large datasets  
❌ Generates many unnecessary candidate itemsets

**2. FP-Growth (Frequent Pattern Growth)**

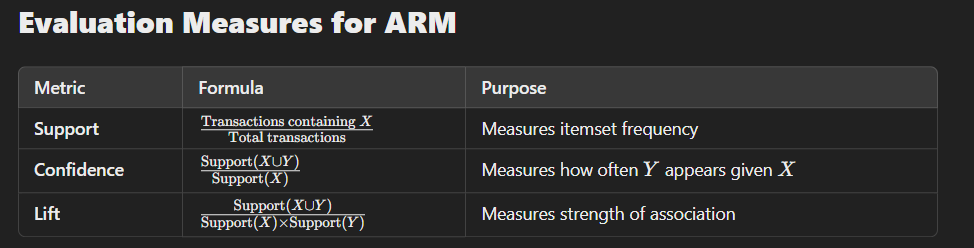
* Uses a **tree-based structure (FP-Tree)** instead of candidate generation like Apriori.

**Algorithm Steps:**

1. Build an **FP-Tree**, a compact data structure representing frequent patterns.
2. Extract frequent itemsets using recursive pattern mining.

**Advantages:**  
✅ Faster than Apriori for large datasets  
✅ Reduces computational complexity

**Disadvantages:**  
❌ More complex implementation  
❌ High memory usage for dense datasets

****

**Conclusion**

* **Apriori Algorithm** is simple but computationally expensive.
* **FP-Growth Algorithm** is faster and more efficient.
* **Evaluation Metrics** (Support, Confidence, Lift) help assess rule quality.

**Dimension Reduction**

**Principal Component Analysis**

**Least Discriminant Analysis**

**Dimensionality Reduction**

**Introduction**

Dimensionality reduction is the process of reducing the number of input variables (features) in a dataset while preserving as much relevant information as possible. It helps in:  
✅ Reducing computational complexity  
✅ Improving model performance by removing noise  
✅ Avoiding overfitting in high-dimensional data

**Why is Dimensionality Reduction Needed?**

* **Curse of Dimensionality**: As dimensions increase, data becomes sparse, making machine learning models less effective.
* **Feature Redundancy**: Many features may be correlated, adding unnecessary complexity.
* **Visualization**: Reducing dimensions helps in plotting and understanding data distributions.

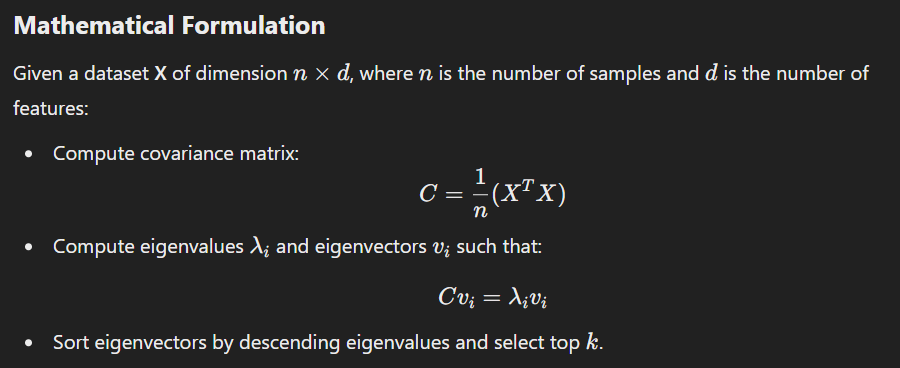
**1. Principal Component Analysis (PCA)**

**Concept**

PCA is a statistical technique used to **transform** high-dimensional data into a lower-dimensional space while retaining the most important variance (information).

**Steps in PCA Algorithm**

1. **Standardize the Data**: Ensure all features have a mean of 0 and variance of 1.
2. **Compute the Covariance Matrix**: Measures relationships between different features.
3. **Compute Eigenvalues and Eigenvectors**:
   * Eigenvectors represent **principal components** (new axes).
   * Eigenvalues measure the importance of each principal component.
4. **Select Principal Components**:
   * Choose top kkk eigenvectors based on highest eigenvalues.
5. **Transform the Data**:
   * Project original data onto the new subspace of selected principal components.

****

**Advantages of PCA**

✅ Reduces dimensionality while preserving variance  
✅ Removes correlated features  
✅ Improves computational efficiency

**Disadvantages of PCA**

❌ Loses interpretability of features  
❌ Assumes linear relationships (not ideal for non-linear data)

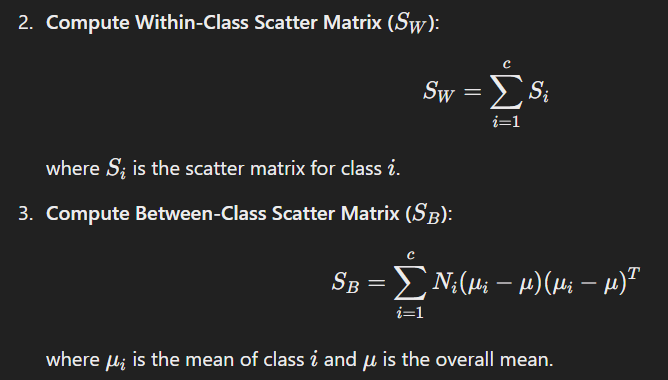
**2. Linear Discriminant Analysis (LDA)**

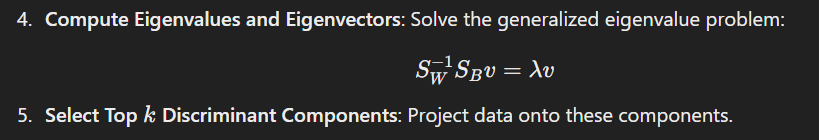
**Concept**

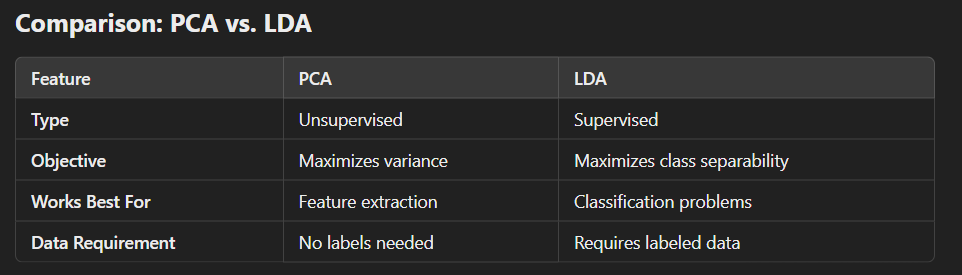
LDA is a **supervised** dimensionality reduction technique that **maximizes class separability**. Unlike PCA, which captures variance, LDA focuses on **finding the optimal feature space for classification**.

**Steps in LDA Algorithm**

1. **Compute the Mean Vectors**:
   * Compute the mean of each class in the dataset.

****

****

****

**Advantages of LDA**

✅ Enhances class separability for classification  
✅ Reduces dimensionality without losing important class information

**Disadvantages of LDA**

❌ Requires labeled data  
❌ Assumes normally distributed classes with equal variance

**Conclusion**

* **PCA** is useful for general dimensionality reduction when labels are not available.
* **LDA** is effective for classification tasks where maximizing class separability is important.

**Ensemble Learning**

**Bagging**

**Boosting**

**Random Forests**

**Ensemble Learning**

**Introduction**

Ensemble learning is a powerful technique in machine learning where multiple models (often called **weak learners**) are combined to create a stronger, more accurate model. The goal is to improve predictive performance by reducing **bias**, **variance**, or both.

**Why Use Ensemble Learning?**

✅ Improves accuracy and robustness  
✅ Reduces overfitting compared to individual models  
✅ Works well for both classification and regression tasks

**Types of Ensemble Learning**

1. **Bagging (Bootstrap Aggregating)**
2. **Boosting (Sequential Learning Approach)**
3. **Random Forests (An extension of Bagging)**

**1. Bagging (Bootstrap Aggregating)**

**Concept**

Bagging is an ensemble technique that trains multiple models independently on different **random subsets** of the training data and then **aggregates** their predictions (e.g., majority vote for classification, averaging for regression).

**Algorithm Steps**

1. Generate multiple training sets by randomly sampling with replacement (**bootstrap sampling**).
2. Train a separate **weak learner** (usually Decision Trees) on each subset.
3. Aggregate predictions using:
   * **Classification:** Majority voting
   * **Regression:** Averaging

**Example: Bagging with Decision Trees**

* If we train **100 Decision Trees** using Bagging, each tree is trained on a different bootstrap sample.
* During prediction, each tree votes, and the majority vote is taken as the final result.

**Advantages of Bagging**

✅ Reduces overfitting (lowers variance)  
✅ Works well with high variance models (e.g., Decision Trees)  
✅ Improves stability

**Disadvantages of Bagging**

❌ Less effective if base models are already stable  
❌ Computationally expensive due to training multiple models

**2. Boosting**

**Concept**

Boosting is an ensemble method where weak learners are trained **sequentially**, with each model focusing on the mistakes of the previous ones. It **assigns higher weights to misclassified instances** so that the next model pays more attention to them.

**Popular Boosting Algorithms**

1. **AdaBoost (Adaptive Boosting)**
   * Assigns weights to each training sample
   * Misclassified samples get **higher** weights in the next iteration
   * Final prediction is a weighted sum of all models
2. **Gradient Boosting**
   * Fits new models to the **residual errors** of previous models
   * Uses **gradient descent** to minimize loss
   * More flexible than AdaBoost
3. **XGBoost (Extreme Gradient Boosting)**
   * Optimized version of Gradient Boosting
   * Faster and more efficient
   * Handles missing values and large datasets better

**Advantages of Boosting**

✅ High accuracy and performance  
✅ Reduces bias (effective with weak learners)  
✅ Works well for complex relationships

**Disadvantages of Boosting**

❌ More prone to overfitting compared to Bagging  
❌ Computationally expensive

**3. Random Forest (Extension of Bagging)**

**Concept**

Random Forest is an **ensemble of Decision Trees**, where:

* Each tree is trained on a **random subset** of the data (Bagging).
* At each split in a tree, only a **random subset of features** is considered.

**Algorithm Steps**

1. Generate multiple bootstrap samples.
2. Train Decision Trees on each subset.
3. At each node, choose the **best split** from a random subset of features.
4. Aggregate results from all trees (majority vote for classification, averaging for regression).

**Advantages of Random Forest**

✅ Handles both classification and regression tasks  
✅ Less prone to overfitting compared to a single Decision Tree  
✅ Handles missing values well  
✅ Works well with large datasets

**Disadvantages of Random Forest**

❌ Can be slow for very large datasets  
❌ Less interpretable than a single Decision Tree

**Comparison: Bagging vs. Boosting vs. Random Forest**

| **Feature** | **Bagging** | **Boosting** | **Random Forest** |
| --- | --- | --- | --- |
| **Goal** | Reduce variance | Reduce bias | Improve accuracy |
| **Training** | Independent models | Sequential learning | Uses Bagging with feature randomness |
| **Overfitting Risk** | Low | High (if too deep) | Low |
| **Computation** | Moderate | High | Moderate |
| **Example Models** | Bagged Decision Trees | AdaBoost, Gradient Boosting, XGBoost | Ensemble of Decision Trees |

**Conclusion**

* **Bagging** helps reduce variance and works well for high variance models.
* **Boosting** reduces bias by focusing on hard-to-classify examples but can overfit.
* **Random Forest** is a practical, efficient, and widely used ensemble technique.