



## Fundamentals

### Key Concepts

**Supervised Learning:** Learning from labeled data to predict outcomes or classify new data points.

**Unsupervised Learning:** Discovering patterns and structures in unlabeled data.

**Reinforcement Learning:** Training an agent to make decisions in an environment to maximize a reward.

**Model:** A mathematical representation of a real-world process, used for prediction or understanding.

**Features:** Input variables used by a model to make predictions.

**Labels:** Output variables that a supervised learning model is trained to predict.

**Training Set:** Data used to train a machine learning model.

**Validation Set:** Data used to tune the hyperparameters of a model.

**Test Set:** Data used to evaluate the performance of a trained model.

### Bias-Variance Tradeoff

**Bias** The error due to oversimplification in a learning algorithm. High bias can cause an algorithm to miss relevant relations between features and target outputs (underfitting).

**Variance** The error due to the model's sensitivity to small fluctuations in the training data. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).

**Tradeoff** Finding the right balance between bias and variance is crucial for building accurate models. Reducing one often increases the other.

### Common Evaluation Metrics

**Accuracy:** The proportion of correctly classified instances.

**Precision:** The proportion of true positives among the instances predicted as positive.

**Recall:** The proportion of true positives among the actual positive instances.

**F1-score:** The harmonic mean of precision and recall.

**AUC-ROC:** Area under the Receiver Operating Characteristic curve, which measures the ability of a classifier to distinguish between classes.

**Mean Squared Error (MSE):** The average squared difference between predicted and actual values (for regression).

## Supervised Learning

### Regression Algorithms

<b>Linear Regression</b>	Models the relationship between variables using a linear equation. Sensitive to outliers. Assumes linearity.
<b>Polynomial Regression</b>	Extends linear regression by adding polynomial terms. Can model non-linear relationships.
<b>Ridge Regression</b>	Adds L2 regularization to linear regression to prevent overfitting.
<b>Lasso Regression</b>	Adds L1 regularization to linear regression, which can also perform feature selection.
<b>Support Vector Regression (SVR)</b>	Uses support vector machines for regression tasks. Effective in high dimensional spaces.

### Classification Algorithms

<b>Logistic Regression</b>	Predicts the probability of a binary outcome. Uses a sigmoid function. Output between 0 and 1.
<b>Support Vector Machine (SVM)</b>	Finds the optimal hyperplane to separate data points into different classes. Effective in high-dimensional spaces.
<b>Decision Tree</b>	Uses a tree-like structure to make decisions based on features. Prone to overfitting.
<b>Random Forest</b>	An ensemble of decision trees, which reduces overfitting and improves accuracy.
<b>Naive Bayes</b>	Applies Bayes' theorem with strong (naive) independence assumptions between features. Fast and simple.
<b>K-Nearest Neighbors (KNN)</b>	Classifies data points based on the majority class among its k nearest neighbors. Sensitive to feature scaling.

### Ensemble Methods

**Bagging:** Training multiple models on different subsets of the training data and averaging their predictions. Reduces variance.

**Boosting:** Training models sequentially, where each model tries to correct the errors of its predecessors. Reduces bias.

**Examples of Bagging:** Random Forest.

**Examples of Boosting:** AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost.

# Unsupervised Learning

## Clustering Algorithms

<b>K-Means</b>	Partitions data into k clusters, where each data point belongs to the cluster with the nearest mean (centroid). Requires specifying the number of clusters (k).
<b>Hierarchical Clustering</b>	Builds a hierarchy of clusters, either agglomerative (bottom-up) or divisive (top-down). Doesn't require specifying the number of clusters beforehand.
<b>DBSCAN (Density-Based Spatial Clustering of Applications with Noise)</b>	Groups together data points that are closely packed together, marking as outliers points that lie alone in low-density regions. Does not require specifying the number of clusters and is robust to outliers.
<b>Spectral Clustering</b>	Uses the eigenvalues of the similarity matrix of the data to reduce the dimensionality of the data before clustering in fewer dimensions.

## Reinforcement Learning

### Core Concepts

<b>Agent:</b> An entity that interacts with an environment to learn optimal behavior.
<b>Environment:</b> The world in which the agent operates.
<b>State:</b> A representation of the current situation of the environment.
<b>Action:</b> A choice made by the agent to interact with the environment.
<b>Reward:</b> A scalar value that measures the immediate feedback from the environment to the agent.
<b>Policy:</b> A strategy that the agent uses to determine which action to take in a given state.
<b>Value Function:</b> A function that estimates the expected cumulative reward from a given state following a particular policy.
<b>Q-Function:</b> A function that estimates the expected cumulative reward from taking a specific action in a given state and following a particular policy.

## Dimensionality Reduction

<b>Principal Component Analysis (PCA)</b>	Transforms data into a new coordinate system where the principal components capture the most variance in the data. Reduces the number of features while preserving important information.
<b>t-distributed Stochastic Neighbor Embedding (t-SNE)</b>	Reduces the dimensionality of data while preserving the local structure of the data points. Useful for visualizing high-dimensional data in lower dimensions.
<b>Autoencoders</b>	Neural networks that learn to encode and decode data, typically used for dimensionality reduction or feature learning. Can capture non-linear relationships.

## Association Rule Mining

<b>Apriori Algorithm:</b> An algorithm for frequent itemset mining and association rule learning over relational databases. Identifies frequent itemsets and generates association rules based on minimum support and confidence thresholds.
<b>Eclat Algorithm:</b> Uses a depth-first search approach to find frequent itemsets. Can be more efficient than Apriori for certain datasets.

### Algorithms

<b>Q-Learning</b>	An off-policy algorithm that learns the optimal Q-function by iteratively updating Q-values based on observed rewards. Uses the Bellman equation.
<b>SARSA (State-Action-Reward-State-Action)</b>	An on-policy algorithm that updates Q-values based on the current policy. More conservative than Q-learning.
<b>Deep Q-Network (DQN)</b>	Uses a deep neural network to approximate the Q-function. Combines Q-learning with deep learning to handle high-dimensional state spaces.
<b>Policy Gradients</b>	Directly optimizes the policy without using a value function. Suitable for continuous action spaces.
<b>Actor-Critic Methods</b>	Combines policy gradients with value-based methods. Uses an actor to learn the policy and a critic to estimate the value function.

### Exploration vs. Exploitation

<b>Exploration:</b> Trying out different actions to discover new states and rewards.
<b>Exploitation:</b> Choosing the action that is believed to yield the highest reward based on current knowledge.
<b>Tradeoff:</b> Balancing exploration and exploitation is crucial for learning optimal policies. Common strategies include $\epsilon$ -greedy and Upper Confidence Bound (UCB).