Machine Learning Models

Supervised Learning: Learning from examples

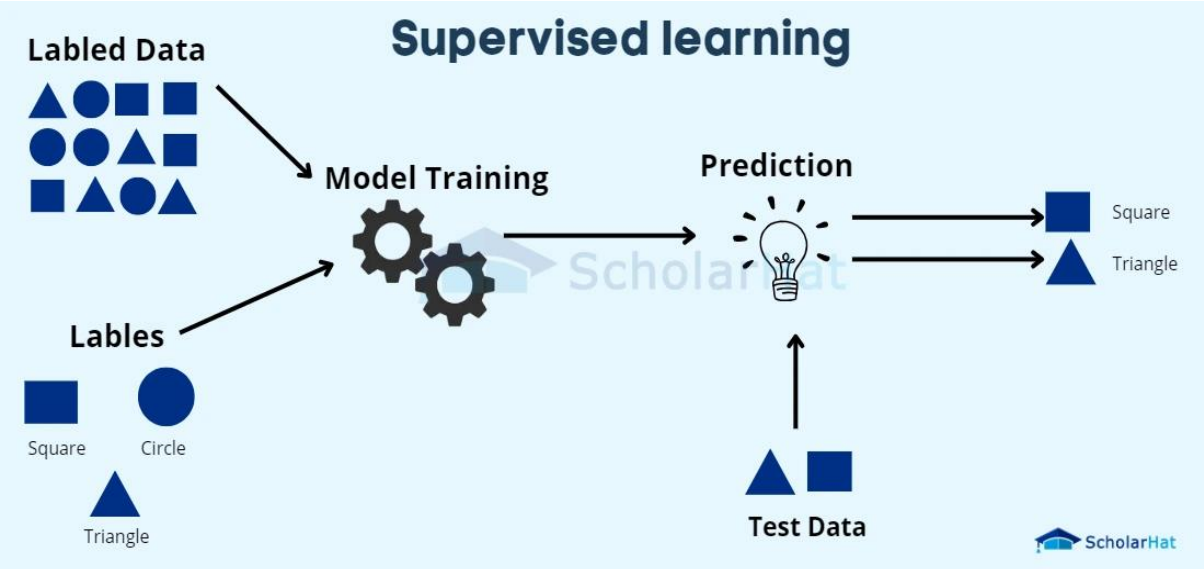
Progression:

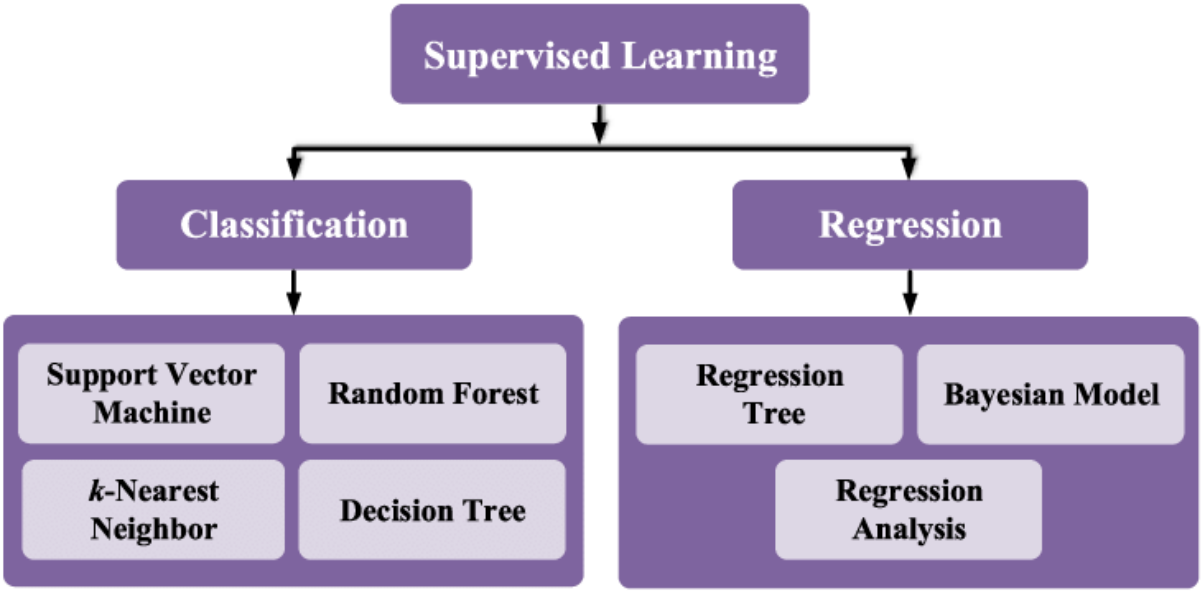
1. **Data Collection**: Gather labeled data (each input has a known output).
    - Example: Images of cats and dogs labeled as “cat” or “dog”.
2. **Data Preprocessing**: Clean, normalize, and split data into training and testing sets.
3. **Model Selection**: Choose an algorithm (e.g., Decision Tree, SVM, Neural Network).
4. **Training**: Feed the labeled data into the model so it learns the mapping from input to output.
5. **Evaluation**: Test the model on unseen data to measure accuracy.
6. **Prediction**: Use the trained model to predict outcomes for new inputs.

Key Traits:

- Requires **labeled data**
- Goal is **accurate prediction**
- Performance measured by metrics like accuracy, precision, recall

Example: Predicting whether an email is spam based on its content.





Classification Models:

- **Support Vector Machine (SVM)**: Finds the optimal boundary that separates classes with maximum margin.
- **Random Forest**: Combines multiple decision trees to improve accuracy and reduce overfitting.
- **K-Nearest Neighbours (KNN)**: Classifies based on the majority label among closest data points.
- **Decision Tree**: Splits data into branches based on feature thresholds to classify outcomes.

Regression Models:

- **Regression Tree:** Splits data into regions using decision rules to predict continuous outcomes within each region.
- **Regression Analysis:** Statistical method to model and analyze relationships between a dependent variable and one or more independent variables.
- **Bayesian Model:** Uses Bayes' theorem to update probabilities as new evidence is introduced, enabling probabilistic inference.

Unsupervised Learning: Discovering Hidden Patterns

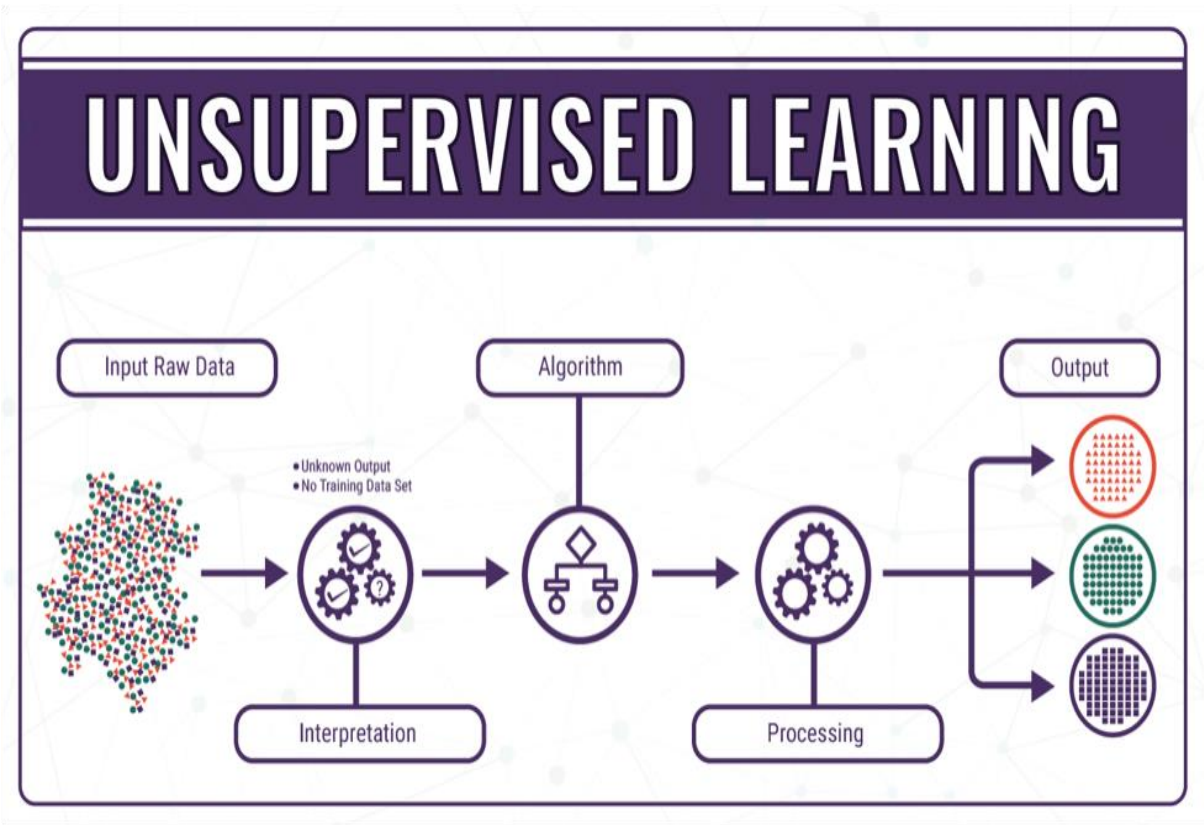
Progression:

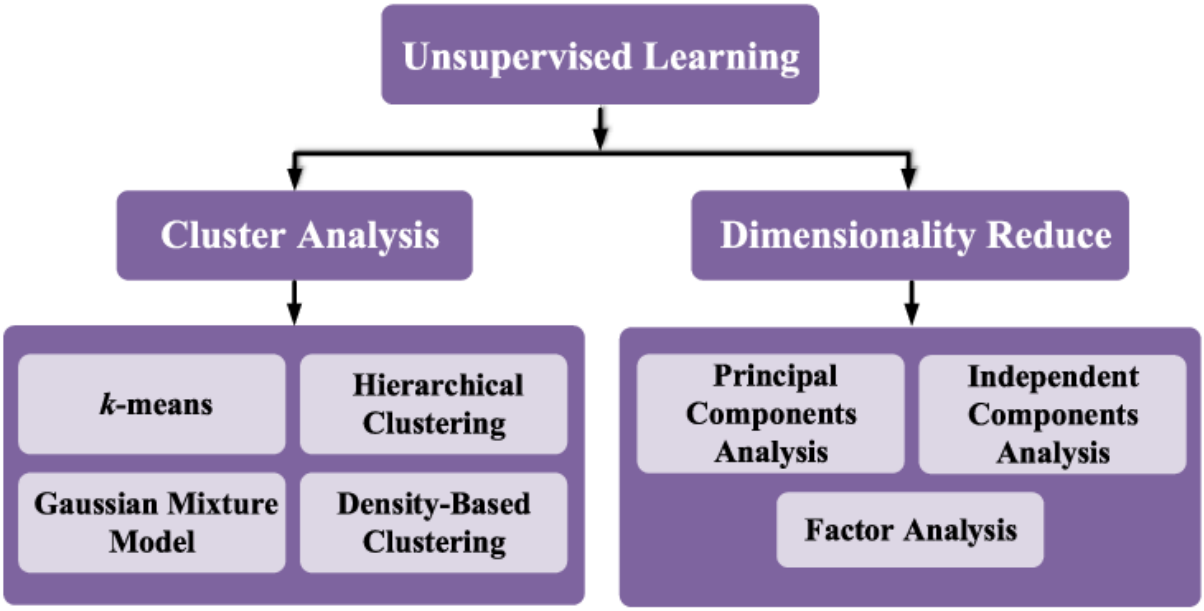
1. **Data Collection:** Gather unlabeled data (no predefined categories or outputs).
    - Example: Customer purchase histories.
2. **Data Preprocessing:** Normalize and clean data.
3. **Model Selection:** Choose an algorithm (e.g., K-Means, DBSCAN, PCA).
4. **Training:** The model analyzes data to find patterns, clusters, or structures.
5. **Evaluation:** Use metrics like silhouette score or visual inspection of clusters.
6. **Interpretation:** Understand the discovered patterns and apply insights.

Key Traits:

- **No labels** required
- Goal is **pattern discovery**
- Often used for **clustering, dimensionality reduction, or anomaly detection**

Example: Segmenting customers into groups based on buying behavior.





Cluster Analysis:

- **k-means:** Partitions data into k clusters by minimizing intra-cluster variance.
- **Gaussian Mixture Model:** Models data as a mixture of multiple Gaussian distributions for soft clustering.
- **Hierarchical Clustering:** Builds nested clusters by either merging or splitting groups based on distance.
- **Density-Based Clustering:** Groups data based on areas of high density, identifying noise and outliers.

Dimensionality Reduction:

- **Principal Components Analysis (PCA):** Transforms data into fewer dimensions by maximizing variance.
- **Independent Components Analysis (ICA):** Separates mixed signals into statistically independent components.
- **Factor Analysis:** Identifies underlying latent variables that explain observed correlations.

Reinforcement Learning: Learning by Trial and Error

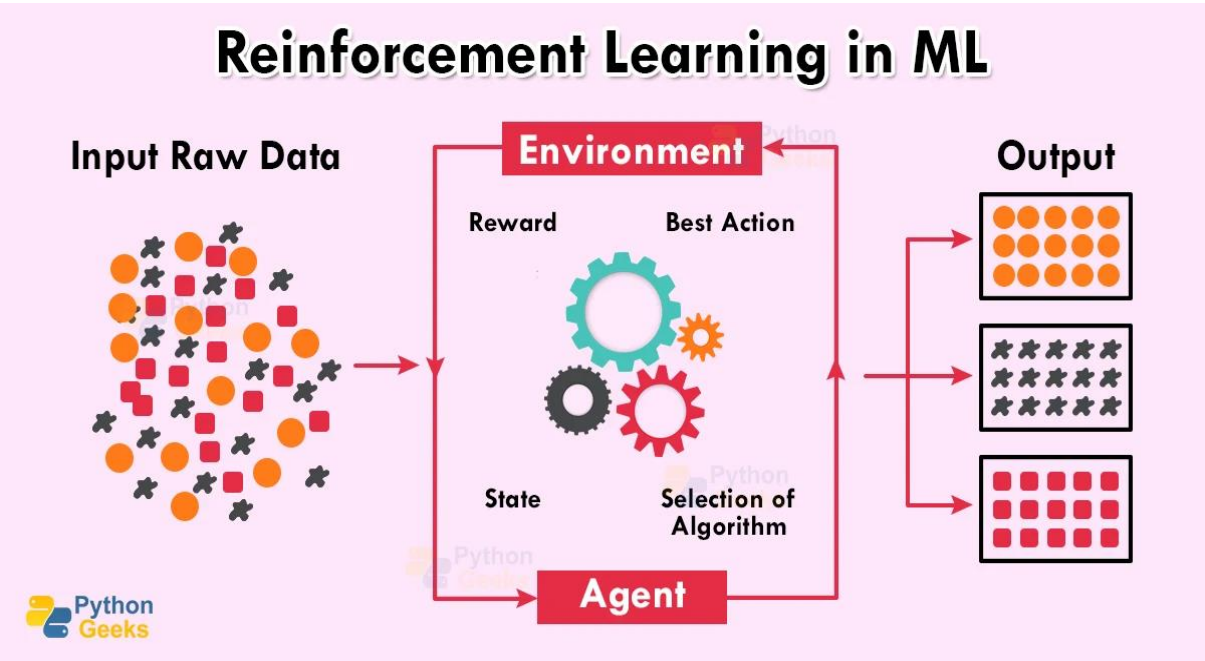
Progression:

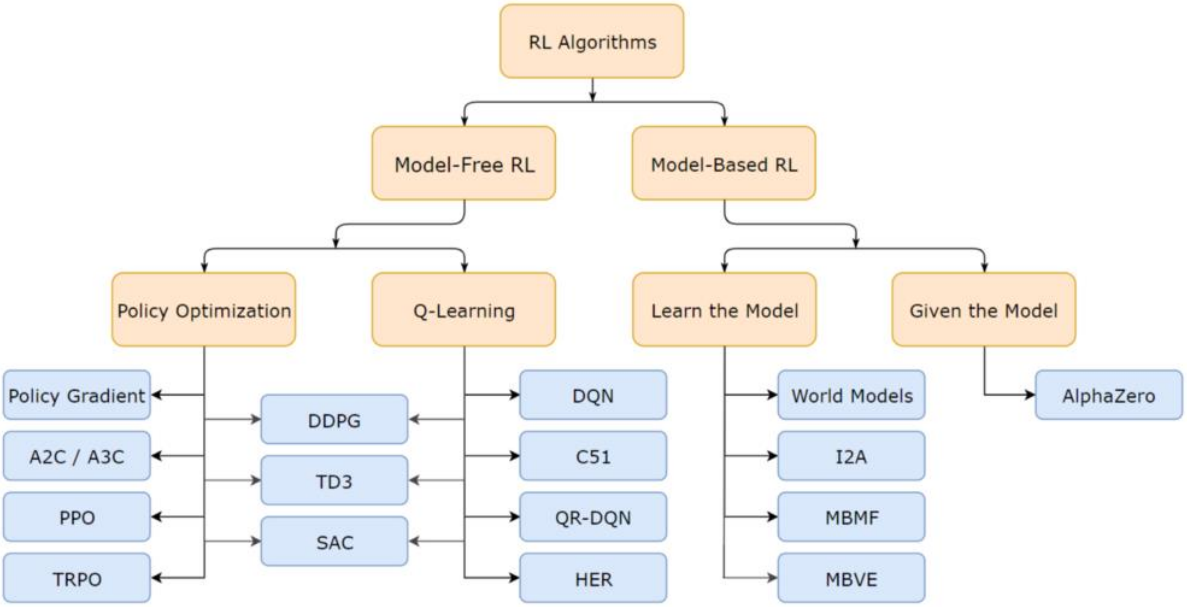
1. **Define Environment:** Set up the world where the agent operates.
    - Example: A maze, a video game, or a robotic arm.
2. **Define Agent and Actions:** Specify what the agent can do.
3. **Set Rewards and Penalties:** Define what earns points and what loses them.
4. **Training Loop:**
    - Agent takes an action.
    - Environment responds with a new state and reward.
    - Agent updates its strategy (policy) to maximize future rewards.
5. **Policy Optimization:** Improve the agent's decision-making over time.
6. **Deployment:** Use the trained agent in real-world or simulated tasks.

Key Traits:

- Learns from **interaction**
- Goal is **maximizing cumulative reward**
- Often uses **exploration vs. exploitation strategies**

Example: A robot learning to walk by trying different movements and adjusting based on success.





Model-Free RL:

- **Policy Optimization:** Improves the agent's decision-making strategy by directly adjusting the policy to maximize expected rewards.
- **Q-Learning:** Estimates the value of action-state pairs to learn the best action policy using temporal difference updates.

Model-Based RL:

- **Learn the Model:** The agent tries to approximate the environment's transition and reward functions from experience.
- **Given the Model:** The agent is provided with the environment's dynamics and uses planning algorithms to derive optimal policies.