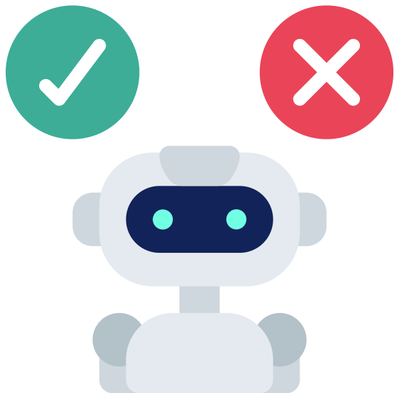
atomcamp

Introduction to Machine Learning & Classification

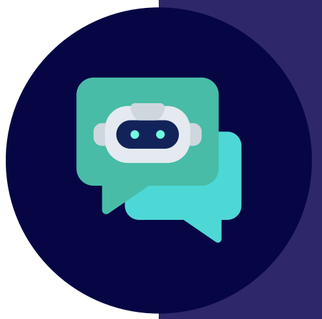
Understanding How Machines Learn From Data



Presented By:
Ms. Faryal Fatima Sabih

Introduction

What is Machine Learning?

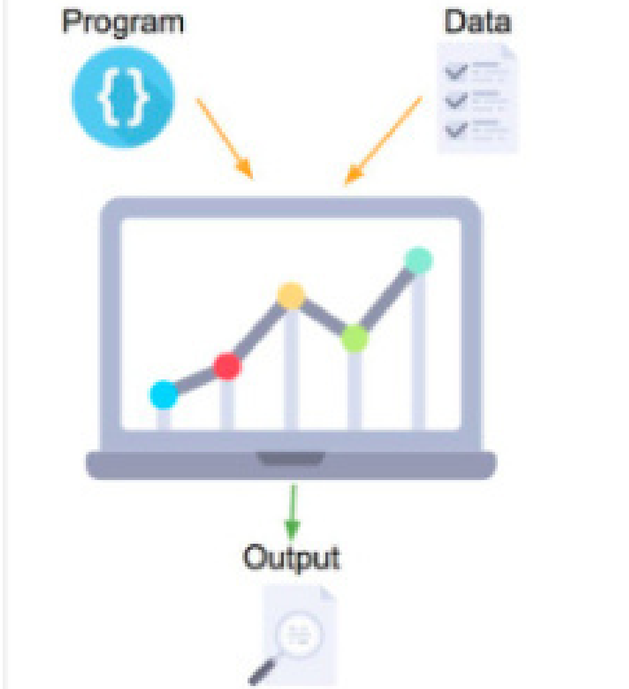


Machine Learning (ML) is a subfield of artificial intelligence in which computers learn from data to make predictions or decisions without being explicitly programmed.

Key Idea:
Instead of writing code to solve a problem, we provide data and let the machine *learn the patterns.*

"Learning is any process by which a system improves performance from experience." – Herbert Simon

Traditional Programming



VS

Machine Learning

Real-world Applications of ML

- Spam filtering
- Face Detection
- Netflix recommendations
- Self-driving cars
- Virtual assistants
- Stock price prediction
- Medical diagnosis
- Sentiment analysis

Categories of Machine Learning

Category	Key Characteristics
Supervised Learning	• Trained on labeled datasets (input-output pairs) • Learns to map inputs to known outputs • Used for classification and regression tasks • Requires annotated data for training
Unsupervised Learning	• Trained on unlabeled data • Identifies hidden patterns, structures, or groupings • Common for clustering and dimensionality reduction • No predefined outputs provided
Reinforcement Learning	• Learns by interacting with an environment • Receives rewards or penalties for actions • Aims to learn optimal decision-making policies • Suitable for sequential, goal-oriented tasks

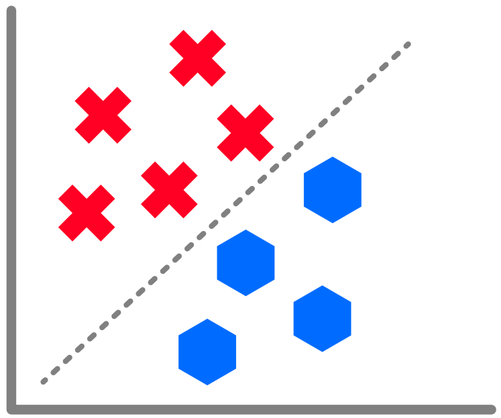
Categories of ML

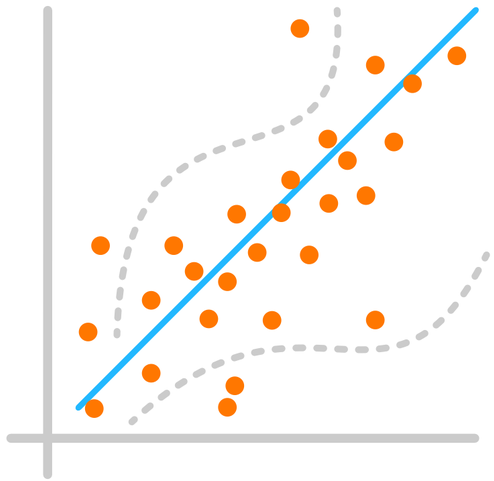
Supervised Learning

Supervised Learning is a type of machine learning where the model is trained on labeled data, meaning the input data is paired with the correct output.

The goal is to learn a function that maps inputs to outputs accurately, so the model can predict the output for new, unseen inputs.

Classification	Regression
• Predicts a discrete label or category • Output is from a fixed set of classes • Binary (Yes/No) or Multi-class (Cat/Dog/Fish) • Common in tasks involving decision-making	• Predicts a continuous numerical value • Output is a real number • Often used when outcome varies over a range • Suitable for estimating quantitative trends



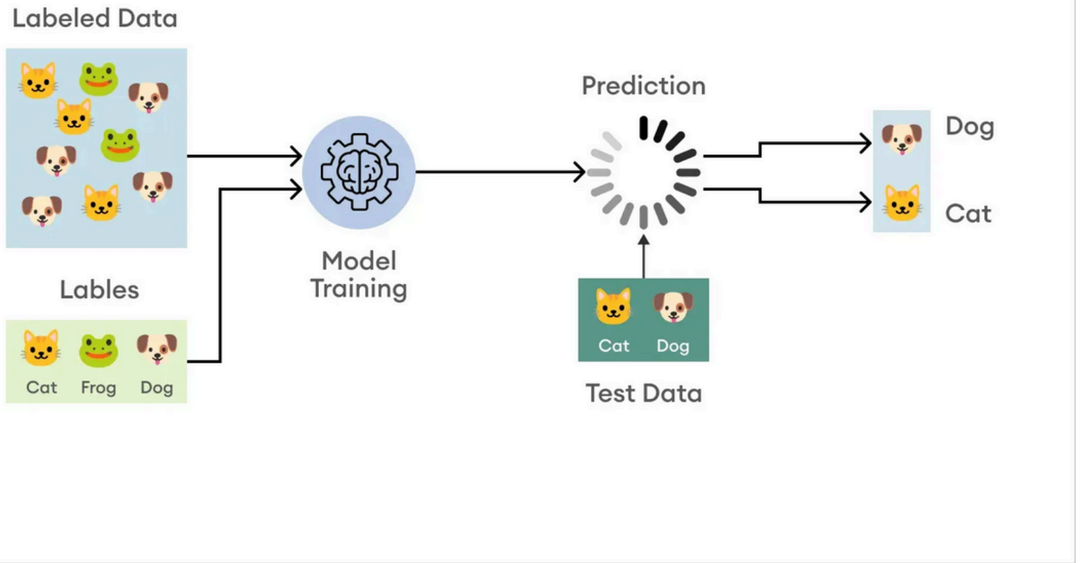


Categories of ML

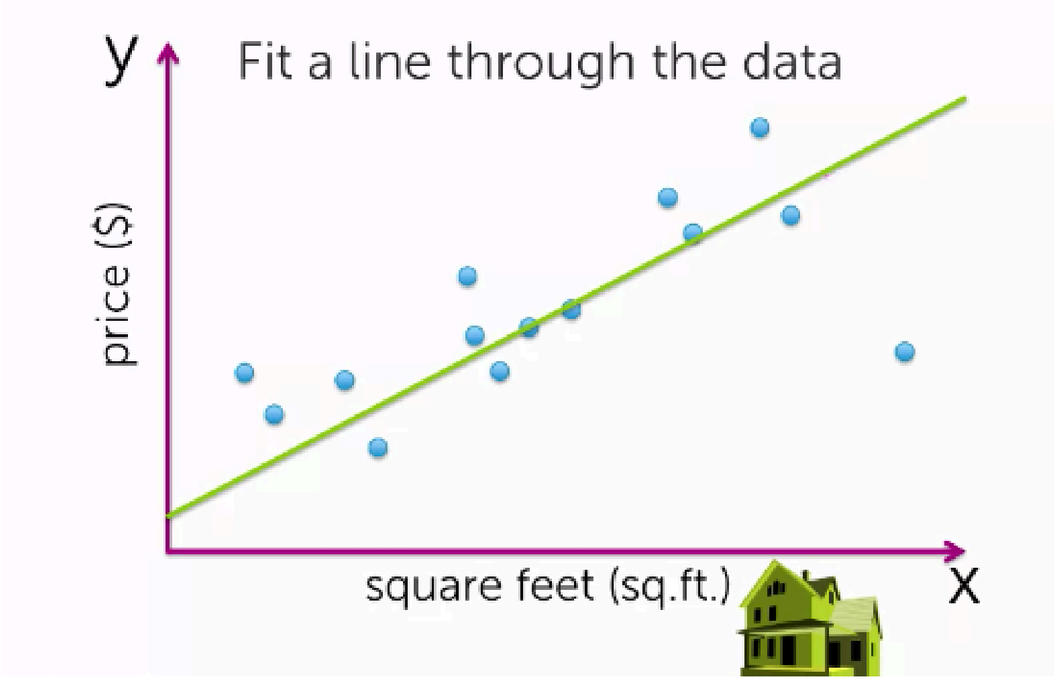
Supervised Learning

Examples

Image (Classification)



House Price Prediction (Regression)



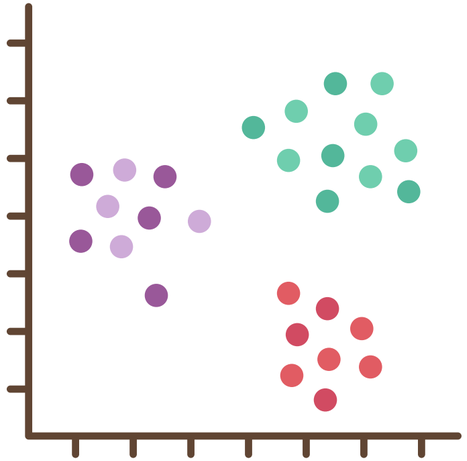
Categories of ML

Unsupervised Learning

Unsupervised Learning involves training a model on unlabeled data - the algorithm receives input data without predefined categories or answers. The goal is to identify patterns, structures, or relationships within the data on its own.

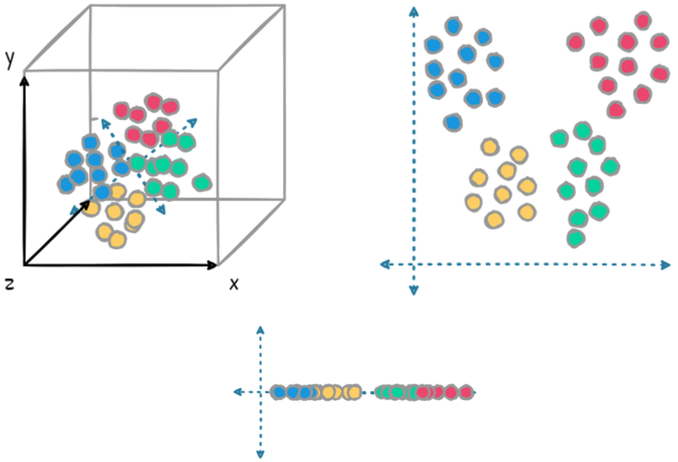
Clustering

- Groups similar data points based on shared characteristics
- Helps identify hidden structures or natural groupings in data
- Commonly used for market segmentation, user profiling, etc.



Dimensionality Reduction

- Simplifies high-dimensional data into fewer, meaningful features
- Useful for visualizing complex data and reducing noise
- Often used as a preprocessing step before further analysis

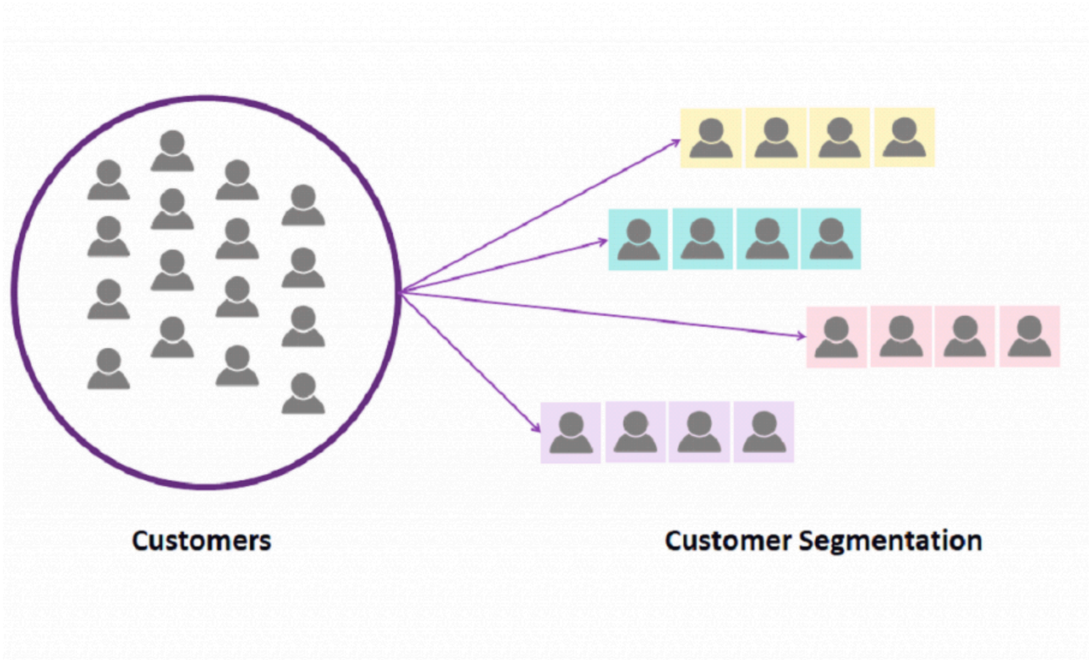


Categories of ML

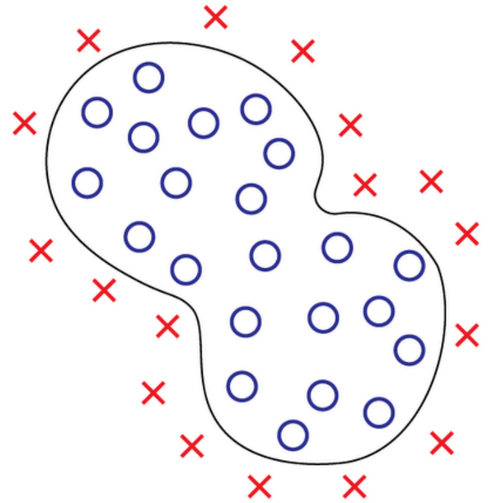
Unsupervised Learning

Examples

Market Segmentation



Anomaly Detection



Anomaly Detection

Reinforcement Learning

Reinforcement Learning is a type of machine learning where an agent learns by interacting with an environment, making decisions, and improving its behavior based on rewards or penalties.

Key Concepts

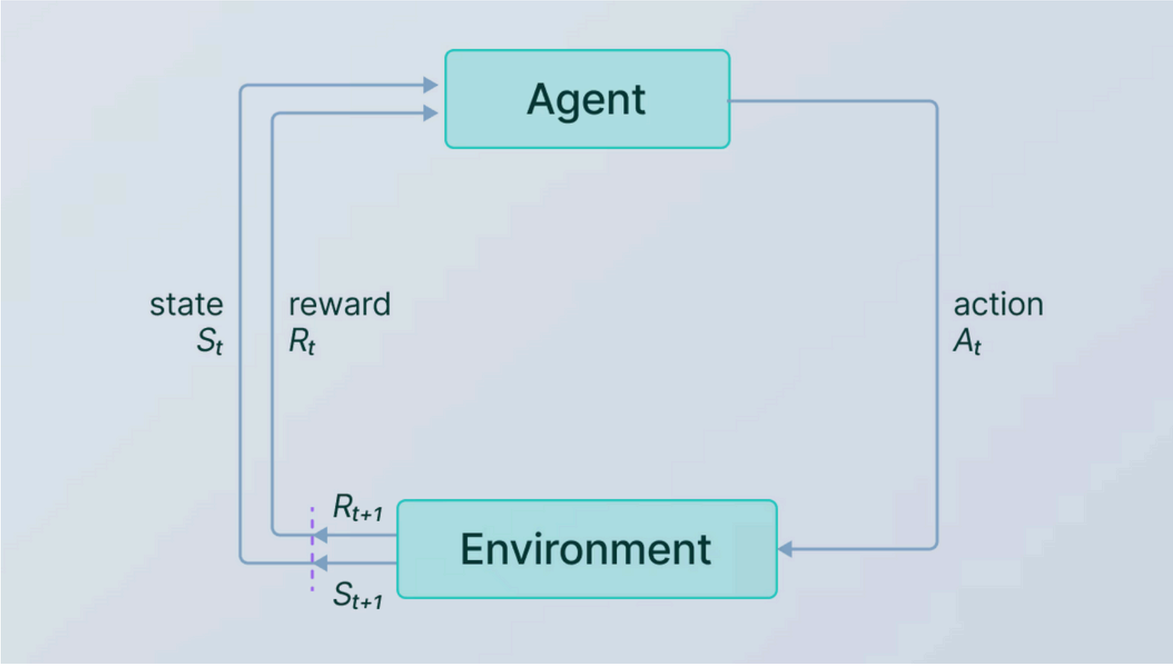
State: The current situation or context the agent is in

Action: A choice or move the agent can take

Reward: Numeric feedback received after an action

Policy: The agent's strategy that maps states to actions

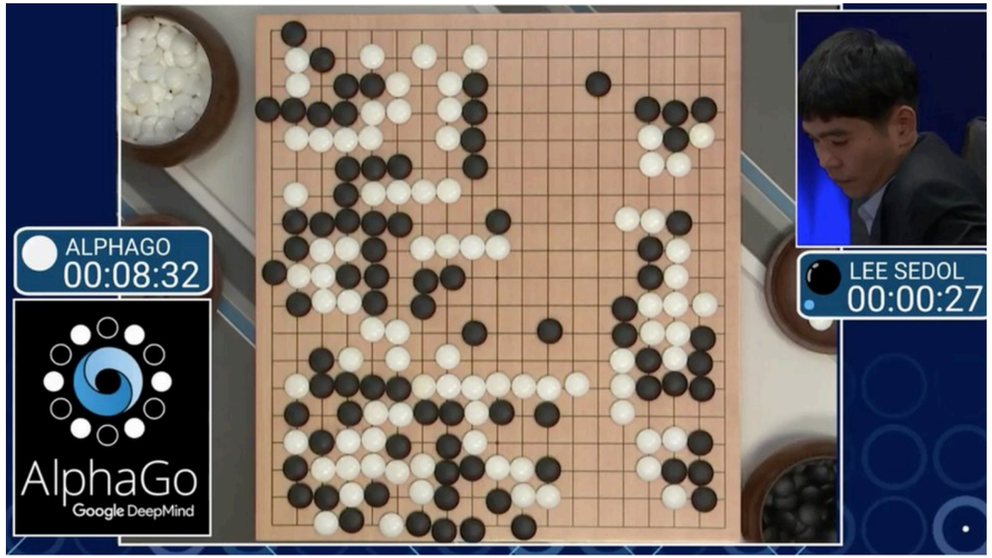
Exploration vs Exploitation: Balancing trying new actions vs choosing known good ones



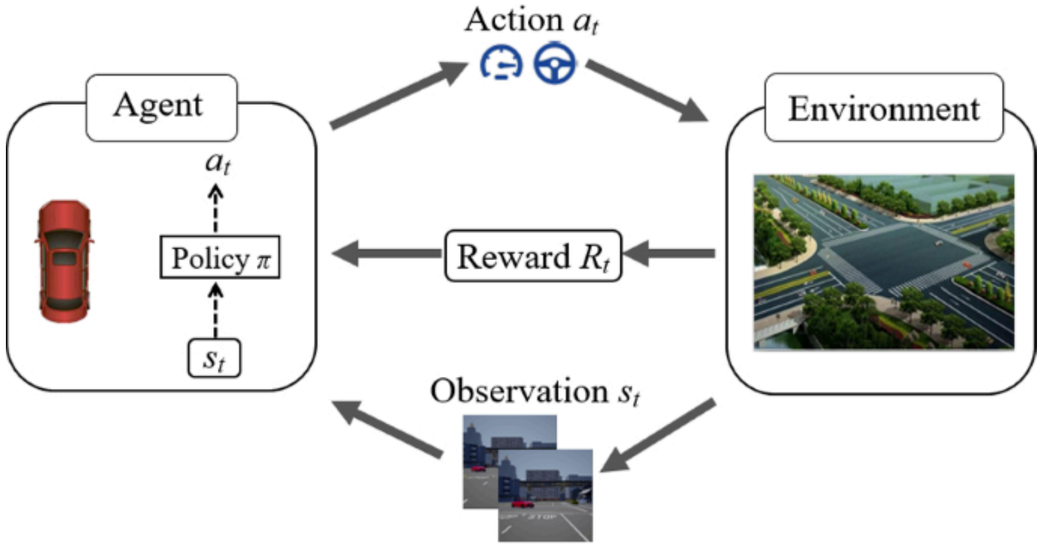
Reinforcement Learning

Examples

AlphaGO

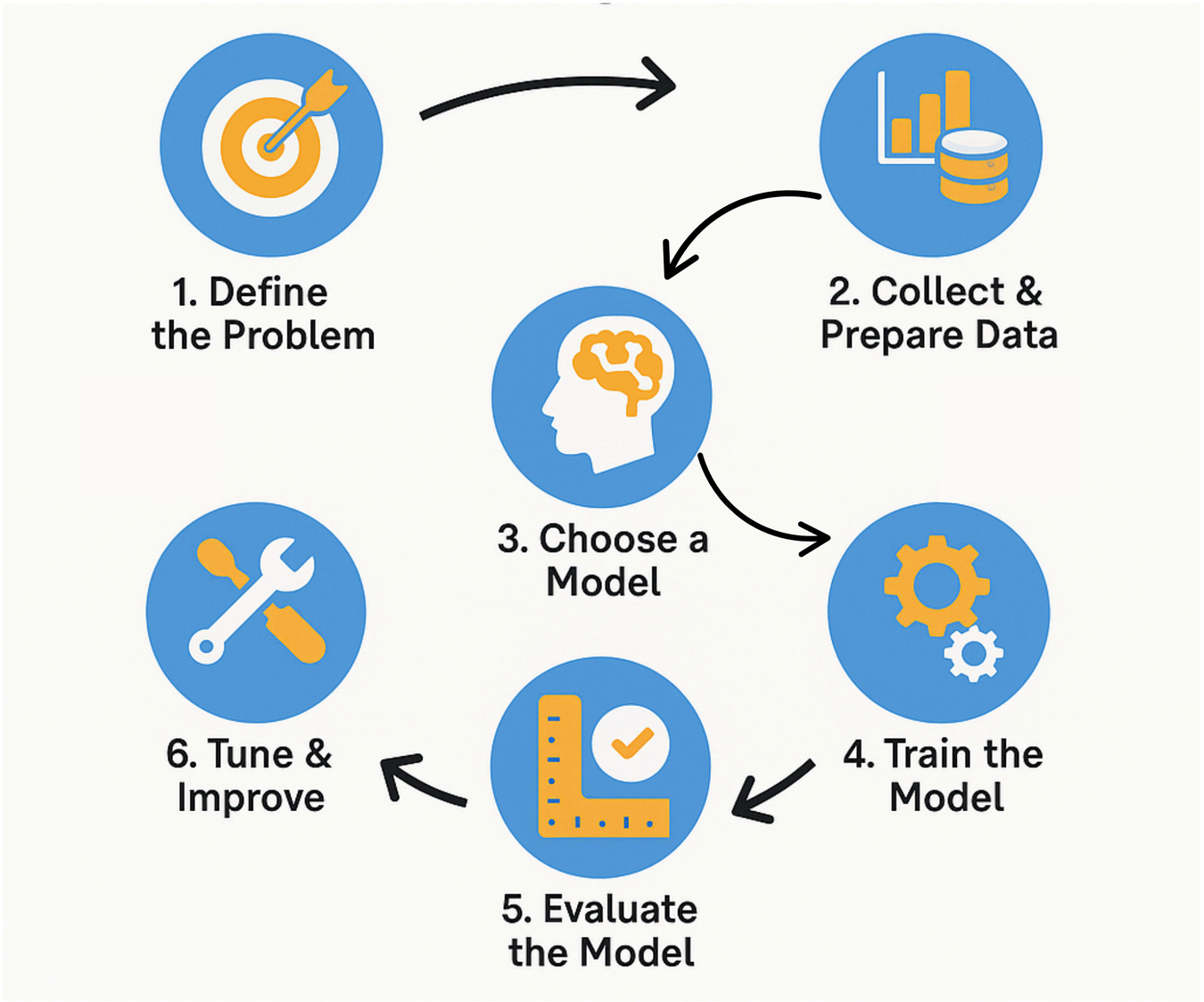


Autonomous Driving



How To Train Your Model?

Leveraging Machine Learning in Daily Life



The Machine Learning Workflow

From Data to Decisions

Step 1: Data Collection

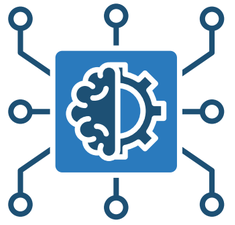
- Gather raw data (structured: CSV, databases; unstructured: images, text, audio).
- Ensure data is representative of the real-world scenario.

Step 2: Data Preprocessing

- Cleaning (handle missing values, duplicates, outliers).
- Feature engineering (derive new meaningful features).
- Normalization/standardization.
- Train-test split (to fairly evaluate later).

Step 3: Model Training

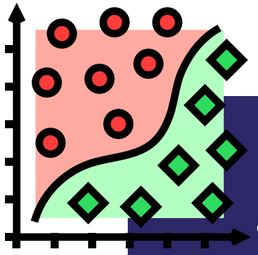
- Choose algorithm (Logistic Regression, Decision Tree, KNN.).
- Fit model to training data.
- Tune hyperparameters (learning rate, regularization strength, etc.).



Step 4: Model Evaluation

- Measure performance with metrics (accuracy, F1, etc.).
- Validate on unseen test set.
- Iterate: improve features or algorithm choice.

Diving Deeper Into Classification



- Classification models learn decision boundaries that separate data points into classes, which can be simple straight lines or more complex non-linear shapes.
- Instead of only giving labels, many models output probabilities that indicate how likely an input belongs to each class, and we set a threshold (e.g., 0.5) to decide the final label.
- Classification can be binary (two classes like spam vs. not spam), multiclass (choosing one from many like cat, dog, or bird), or multilabel (an instance can belong to multiple categories, like an email marked urgent and work-related).
- Real-world datasets are often imbalanced, where one class dominates (e.g., 99% normal transactions vs. 1% fraud), making accuracy misleading and requiring better metrics like precision, recall, and F1-score.

Understanding TP, TN, FP, FN

- True Positive (TP): Model correctly predicts the positive class.

Example: Patient has COVID-19 → test predicts COVID-19 ✅

- True Negative (TN): Model correctly predicts the negative class.

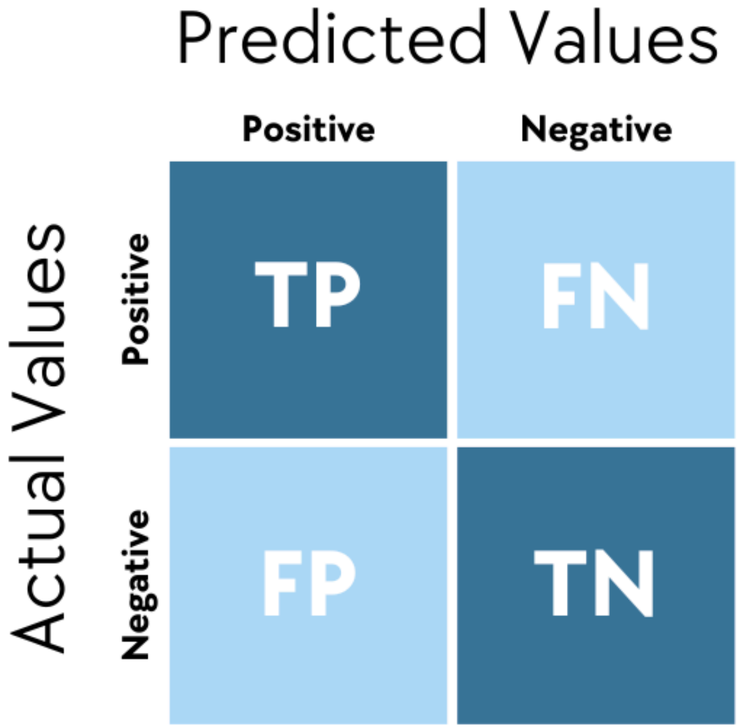
Example: Patient does not have COVID-19 → test predicts no COVID-19 ✅

- False Positive (FP): Model predicts positive when it's actually negative (false alarm).

Example: Patient does not have COVID-19 → test predicts COVID-19 🚨

- False Negative (FN): Model predicts negative when it's actually positive (missed case).

Example: Patient has COVID-19 → test predicts no COVID-19 ❌



- Positive Class: The class we are trying to detect (e.g., COVID-19 = positive).
- Negative Class: The opposite outcome (e.g., No COVID-19 = negative).

Evaluation Metrics

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

- **Definition:** Proportion of all predictions that are correct.
- **When it works well:** Balanced datasets where each class has roughly equal representation (e.g., digit recognition, sentiment analysis with equal positive/negative samples).

- **Limitations:** Misleading for imbalanced datasets — e.g., fraud detection with 99% safe vs 1% fraud → predicting everything as safe gives 99% accuracy but misses all fraud cases.
- **Takeaway:** Accuracy is a good starting metric but not enough in skewed datasets.

Evaluation Metrics

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

- **Definition:** Of all actual positives, how many did the model correctly identify.
- **When it works well:** Situations where false negatives are costly.

- Example: Medical diagnosis → better to catch all possible cancer cases, even if some are false alarms.
- Example: Security screening → better to flag all suspicious items than miss a real threat.

- **Limitations:** High recall may lead to many false positives → system becomes impractical (too many alarms).
- **Takeaway:** Use recall when missing positives is unacceptable, even if it means more false alarms.

Evaluation Metrics

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

- **Definition:** Of all items predicted as positive, how many are actually positive.
- **When it works well:** Situations where false positives are costly.

- Example: Email spam filter → you don't want important emails wrongly flagged as spam.
- Example: Fraud detection → avoid blocking genuine customer payments.

- **Limitations:** High precision can come at the cost of low recall → may miss many true positives.
- **Takeaway:** Use precision when it's more important to be confident about positive predictions.

Evaluation Metrics

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **Definition:** Harmonic mean of precision and recall; balances the two.
- **When it works well:** Datasets with imbalanced classes where both false positives and false negatives are important.

- Example: Fraud detection → balance catching fraud (recall) with avoiding blocking genuine users (precision).
- Example: Disease testing → balance detecting patients with avoiding too many false alarms.

- **Limitations:** Can hide trade-offs — two models with the same F1 may have very different precision/recall balances.
- **Takeaway:** Best when you need a single score to summarize both precision and recall.

Evaluation Metrics

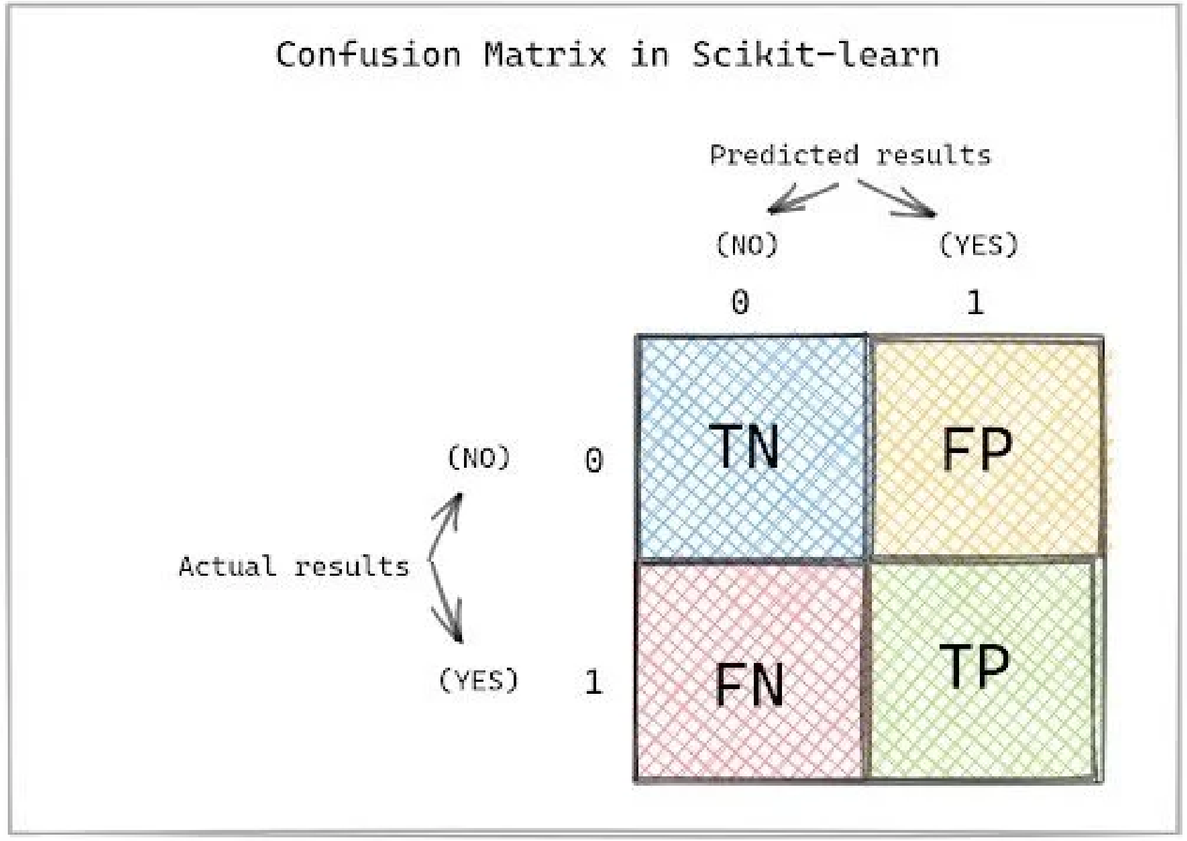
- **Definition:** Table that shows counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

- **Structure:** Rows = actual classes, Columns = predicted classes.

- **Why it's useful:** Lets you see what types of errors the model is making, not just how many.
- Example: In disease classification, model may confuse pneumonia with flu often, but rarely misclassify cancer.

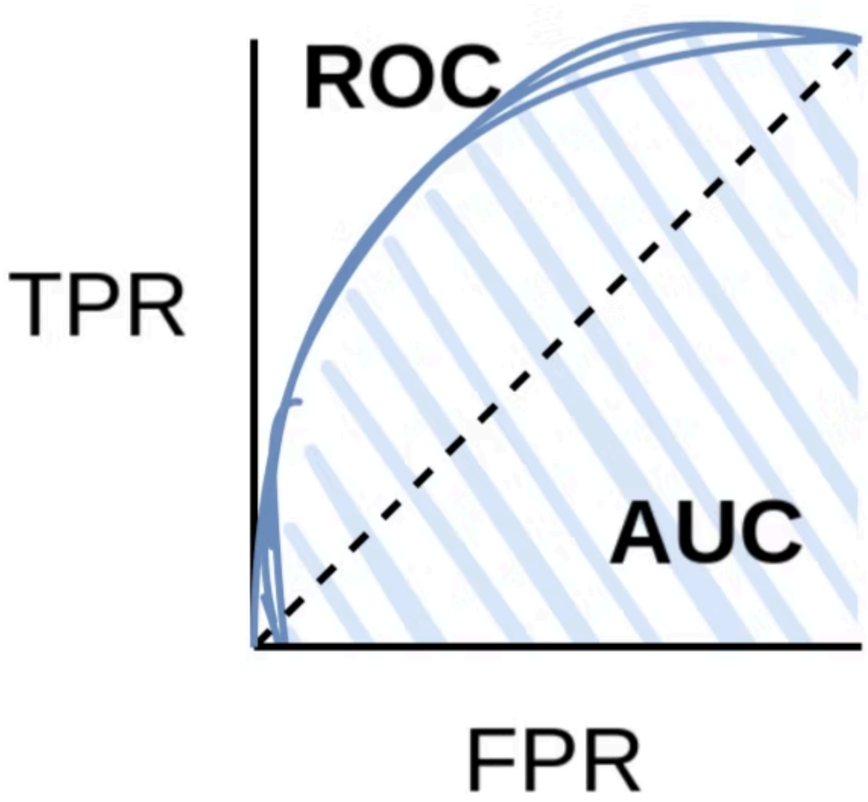
- **When it works well:** For detailed error analysis of multiclass classification tasks.

- **Limitation:** Hard to interpret quickly without summarizing into metrics like accuracy, precision, recall.
- **Takeaway:** The confusion matrix is the foundation — all metrics are derived from it.



Evaluation Metrics

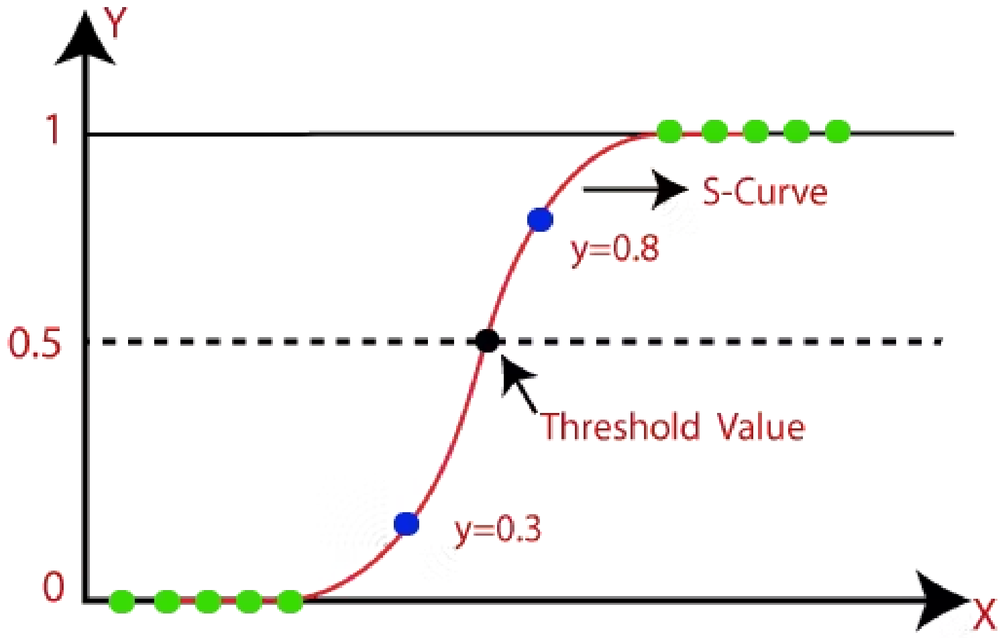
- **ROC Curve:** Plots True Positive Rate (Recall) vs. False Positive Rate across different thresholds.
- **AUC (Area Under Curve):** Measures how well the model separates classes (1 = perfect, 0.5 = random)
- **When it works well:** Comparing classifiers across thresholds.
- Example: Email spam filters → choose threshold balancing spam caught vs. genuine mails misclassified.
- **Limitations:** Can be misleading for imbalanced data — AUC may look good even if the model is failing on the minority class. In such cases, Precision-Recall curves are better.
- **Takeaway:** Use ROC-AUC for overall comparison; switch to Precision-Recall curves for rare-event problems like fraud or medical screening.



Diving Further Into Classification...

Logistic Regression

- A model for classification that predicts the **probability** of belonging to a specific class.
- Uses the sigmoid function **σ(z) = 1 / (1 + e^(−z))** to map any value into [0, 1].
- Probabilities are converted into labels by choosing a threshold (commonly 0.5).
- Finds a decision boundary (line in 2D, hyperplane in higher dimensions) that separates classes.
- Works well for tasks like spam detection, churn prediction, and medical diagnosis.
- Advantages: simple, interpretable, fast to train.
- Limitation: assumes a linear relationship between features and outcome; struggles with complex non-linear boundaries.



Food for Thought

“If machines can learn from data… what should we teach them?”

🧠 Simple Questions to Reflect On:

- Can machines make fair decisions?
- What if the data they learn from is wrong or biased?
- Should machines always replace human decisions?