Introduction to Machine Learning

Machine Learning Workshop – Day 1

Erdener Emin Eker

May 10, 2025

TEDU AI Data Science & AFIT Workshop

About This Workshop

Purpose:

- Introduce core concepts and practical tools in machine learning
- Equip participants with hands-on skills to run ML workflows using Python and Colab

Workshop Schedule:

- **Day 1 – Introduction to ML:** Concepts, workflows, Python simple regression
- **Day 2 – Data Preprocessing:** Cleaning, feature engineering, scaling, CV
- **Day 3 – Classification Models:** Decision Trees, Random Forest
- **Day 4 – Time Series Forecasting:** Lagged features, stationarity, ML for forecasting

Today: Theory + practice using Google Colab — get comfortable with Python syntax, libraries, and running your first ML model.

What is Machine Learning?

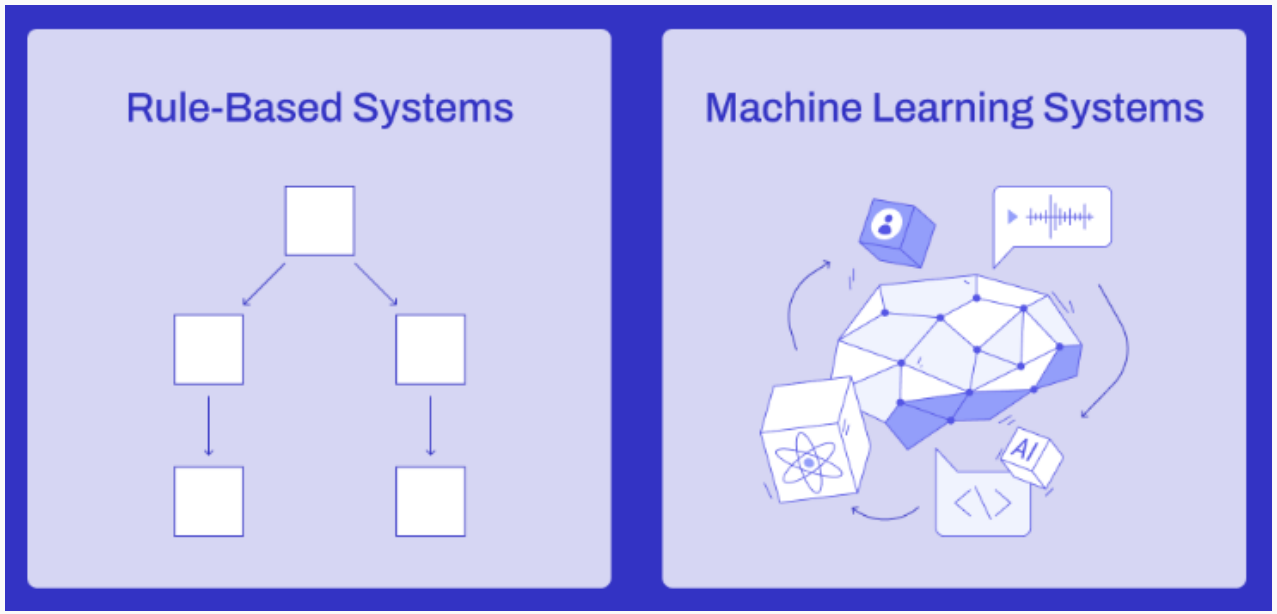
- A method of data analysis that automates analytical model building.
- Based on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention.
- Arthur Samuel (1959): "The field of study that gives computers the ability to learn without being explicitly programmed."

Machine Learning in Action

- Netflix recommends shows you might like
- Banks detect credit card fraud in real time
- Siri and Alexa understand your voice
- Google Translate learns new languages

Traditional Programming vs Machine Learning

- **Traditional Programming:** Humans write rules. The system applies these rules to data to generate output.
- **Machine Learning:** The system learns rules from examples — it infers patterns from data and known outputs.

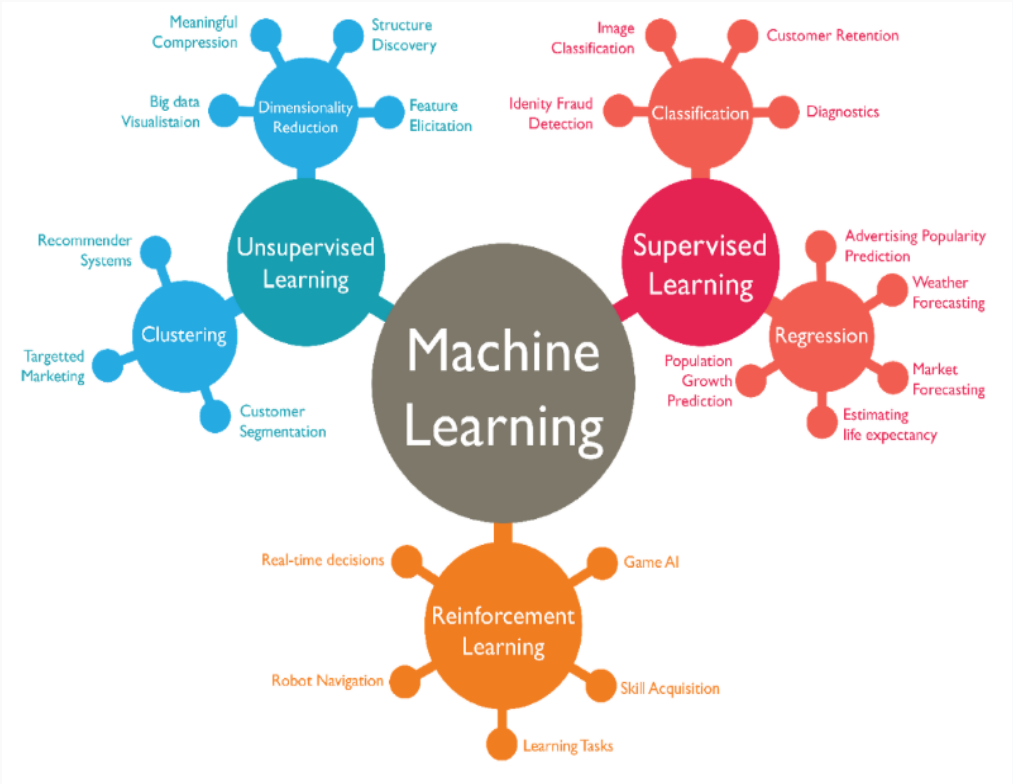


Why Machine Learning is Booming Now

- **Explosion of Data:** Massive digital footprints from social media, sensors, transactions
- **Computational Power:** GPUs, TPUs, and cloud computing make large-scale training feasible
- **Improved Algorithms:** Advances in neural networks, ensemble models, and optimization
- **Open Ecosystem:** Python, scikit-learn, TensorFlow, Colab — anyone can start learning

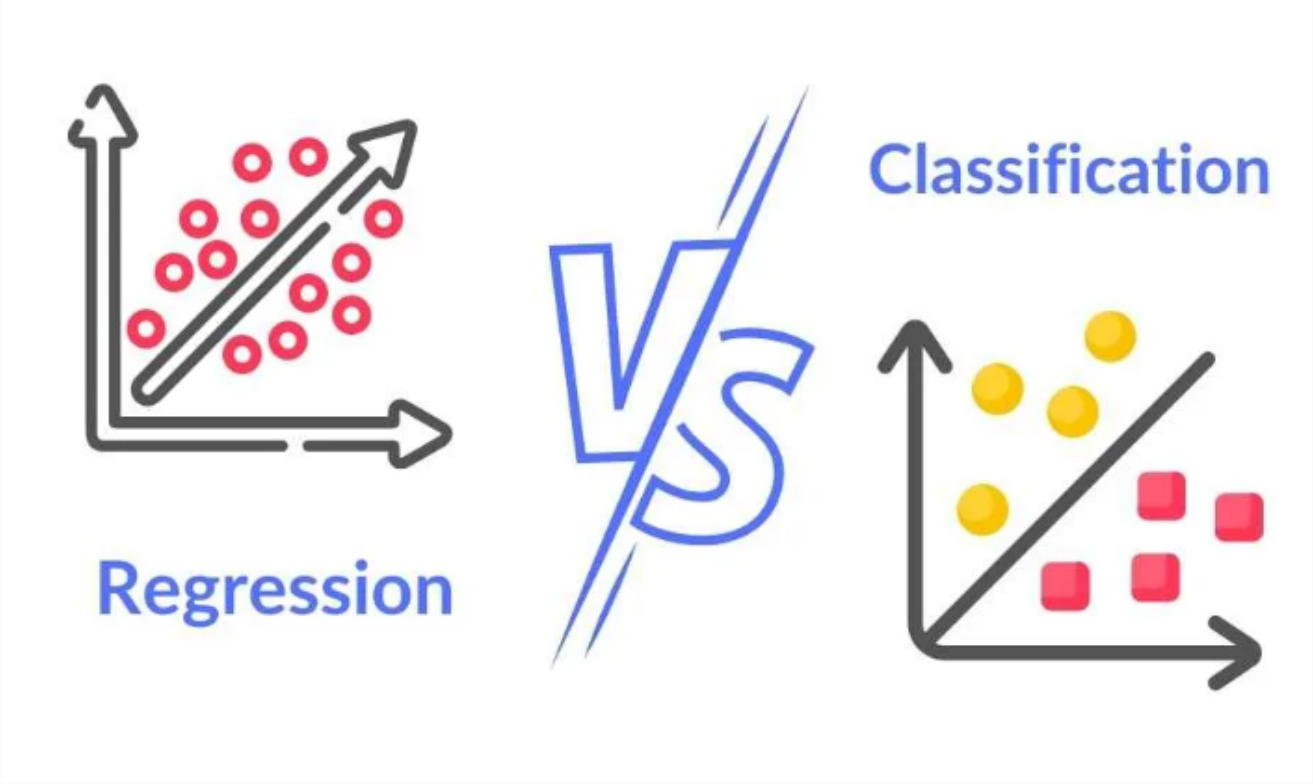
Types of Machine Learning

- **Supervised Learning:** Learn from labeled data e.g., predicting house prices
- **Unsupervised Learning:** Find hidden patterns in unlabeled data e.g., customer segmentation
- **Reinforcement Learning:** Learn by trial and error, with rewards e.g., AlphaGo, robotics



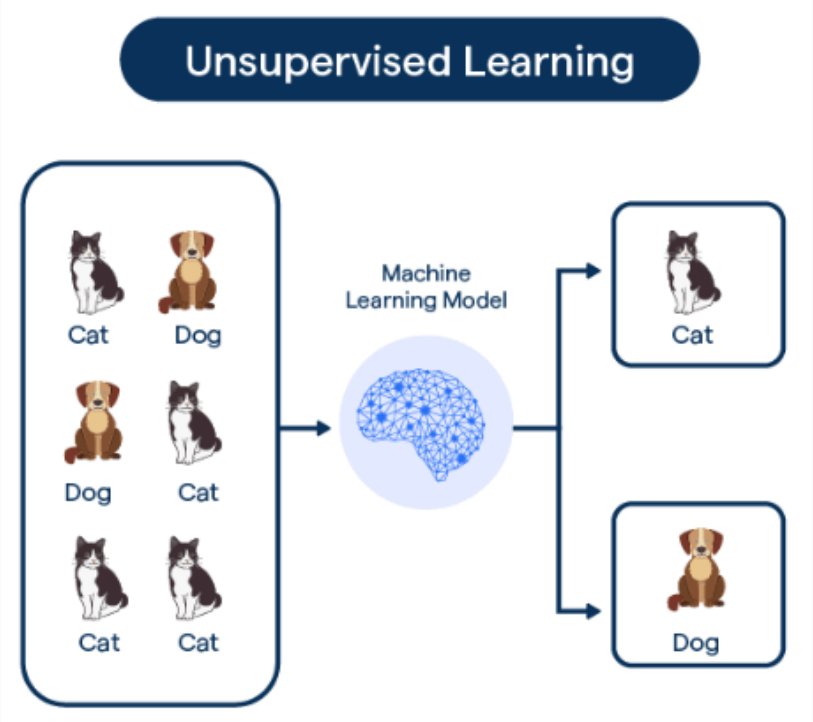
Supervised Learning

- The model learns from **labeled data**, where both inputs and correct outputs are provided.
- Goal: Find a mapping from input features X to output labels y.
- Common tasks:
  - **Regression:** Predict continuous values (e.g., house prices)
  - **Classification:** Predict categories (e.g., spam detection)



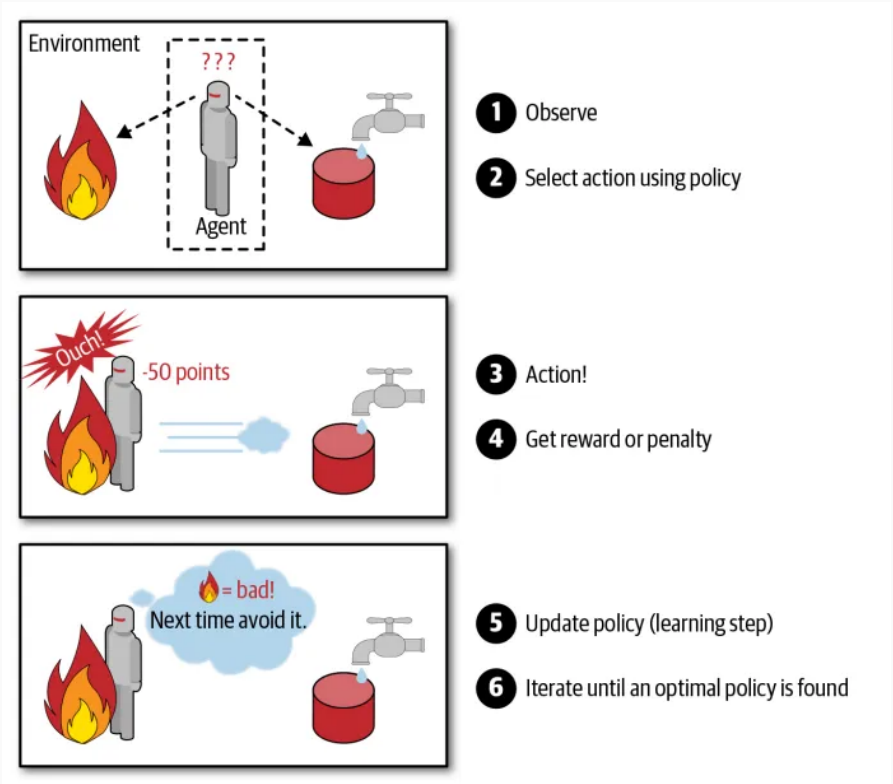
Unsupervised Learning

- No labeled outputs — the algorithm tries to find structure in the data.
- Goal: Discover hidden patterns or groupings in data.
- Common tasks:
  - **Clustering:** Group similar data points (e.g., customer segmentation)
  - **Dimensionality Reduction:** Simplify data while preserving relationships (e.g., PCA)



Reinforcement Learning

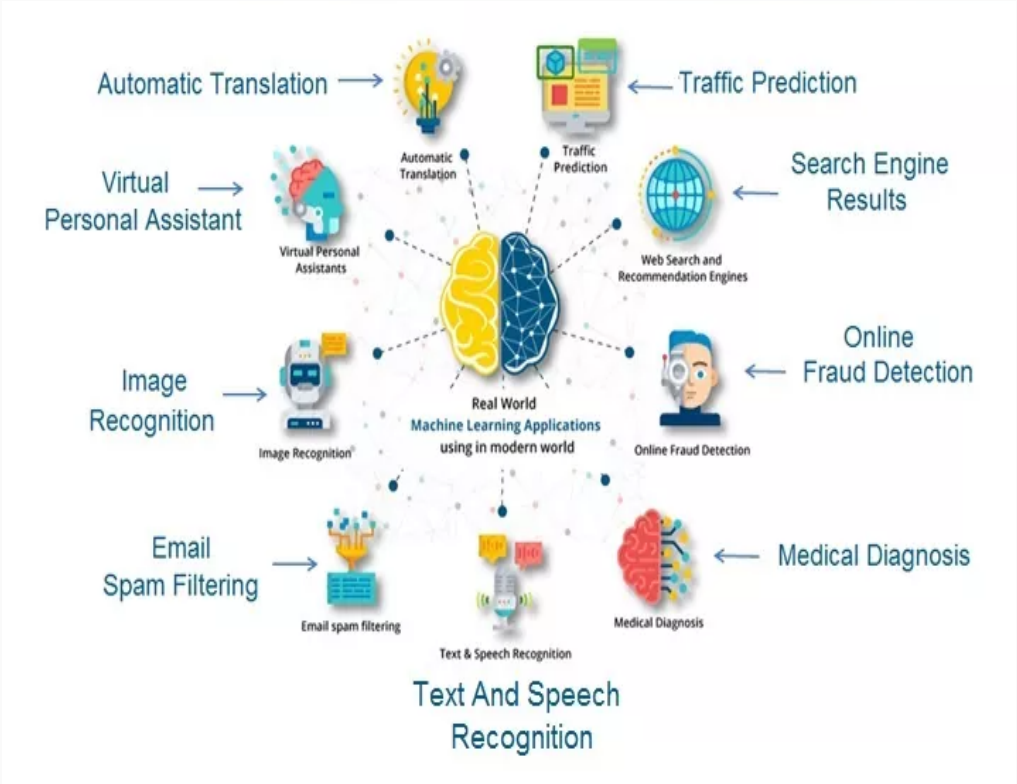
- The agent learns by interacting with an environment.
- It takes actions, receives feedback (rewards or penalties), and adjusts its behavior.
- Goal: Learn a strategy (policy) that maximizes cumulative reward.
- Used in games, robotics, and decision-making tasks.



Applications of Machine Learning

Where is ML used?

- **Finance:** Fraud detection, credit scoring, algorithmic trading
- **Healthcare:** Disease prediction, medical imaging
- **Marketing:** Recommendation engines, customer segmentation
- **Transportation:** Route optimization, autonomous vehicles
- **Natural Language:** Chatbots, translation, sentiment analysis

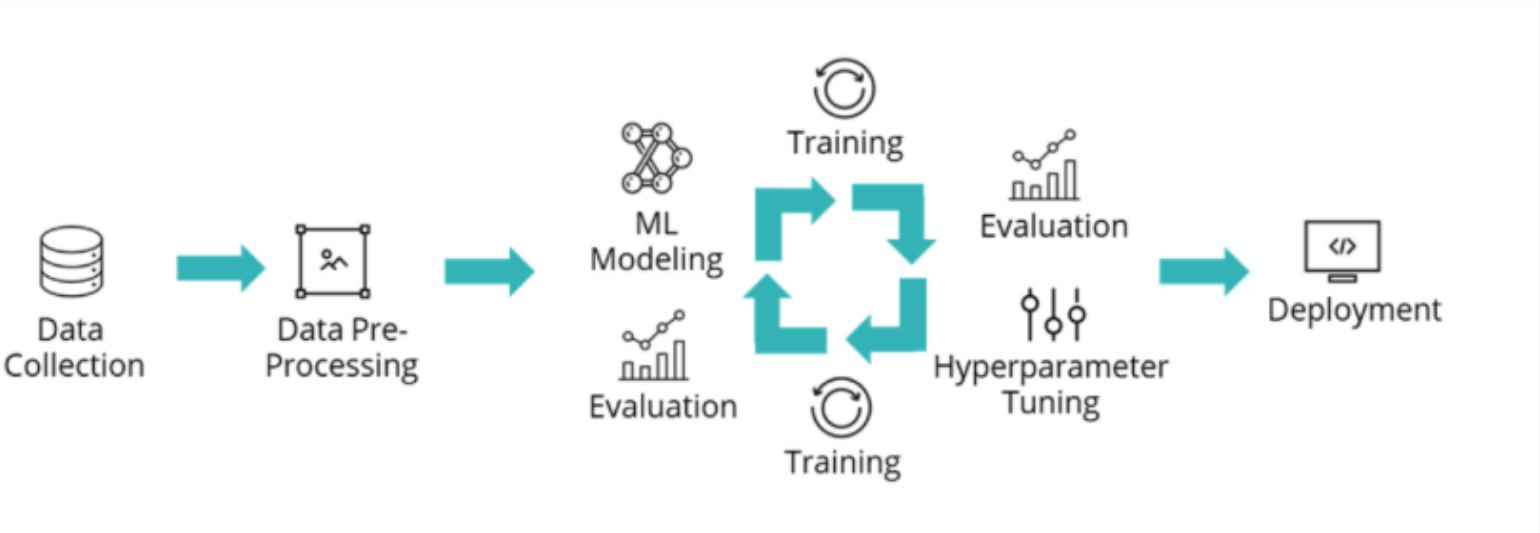


Common Tasks in Machine Learning

- **Regression:** Predict a continuous value Example: Forecasting stock prices, predicting electricity demand
- **Classification:** Predict discrete categories Example: Email spam detection, disease diagnosis (benign vs malignant), sentiment analysis
- **Clustering:** Group similar data points without labels Example: Customer segmentation, grouping news articles by topic
- **Dimensionality Reduction:** Reduce the number of input features while preserving structure Example: Visualizing high-dimensional image data using PCA or t-SNE
- **Anomaly Detection:** Identify rare or unusual observations Example: Credit card fraud, machine failure prediction, outlier detection in finance

The Machine Learning Workflow

- ML is more than just fitting models — it's an end-to-end process.
- Typical steps:
  1. Define the problem
  2. Collect the data
  3. Clean and preprocess
  4. Engineer features
  5. Select and train a model
  6. Evaluate performance
  7. Tune and improve
  8. Deploy and monitor



Step 1: Define the Problem

- Clearly state what you are trying to predict or understand.
- Choose a measurable goal tied to a business or research need.
- Decide whether it's a:
  - **Regression problem** (predicting numbers)?
  - **Classification problem** (predicting categories)?
  - **Clustering or ranking problem**?
- Avoid vague goals like "use AI on this data."
- A well-defined problem leads to relevant features, appropriate metrics, and successful models.

Step 2: Collect the Data

- Data is the foundation of any ML project.
- Choose data that is relevant, representative, and timely.
- Common data sources:
  - Internal databases (e.g., customer transactions)
  - Public datasets (e.g. Kaggle, World Bank)
  - APIs and web scraping
  - Sensors, logs, user activity
- Understand the size, format, and quality of your data early on.

Step 3: Data Preprocessing

- Clean and transform raw data into usable form
- Typical tasks:
  - Handle missing values (drop, impute)
  - Remove duplicates and outliers
  - Convert categories into numbers (encoding)
  - Normalize or scale features
- Good preprocessing improves model accuracy significantly

Before vs After Data Cleaning

Before Cleaning

Name	Age	Income	Gender
john smith		55K	Male
JANE DOE	twenty-five	$50,000	F
Robert Jr.	42	50000	male

After Cleaning

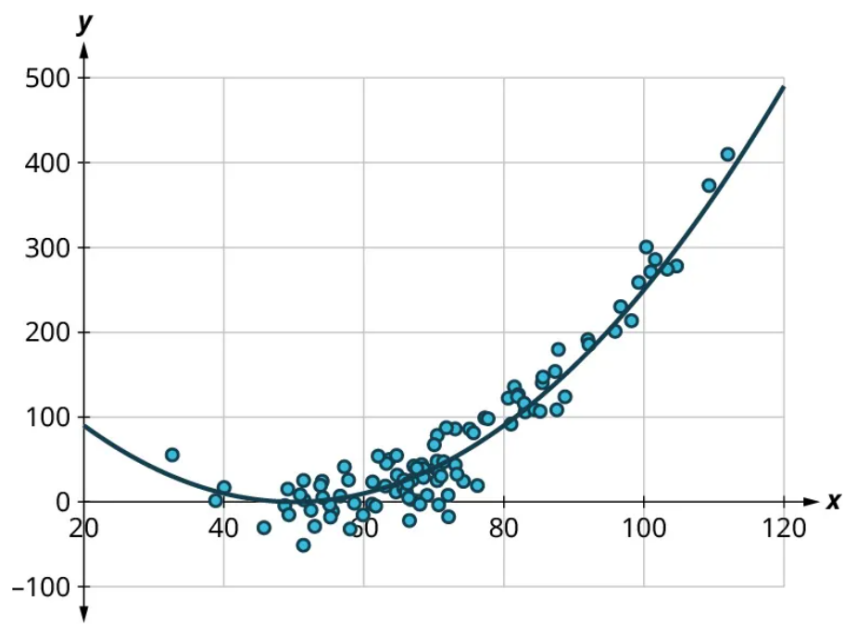
Name	Age	Income	Gender
John Smith	30	55000	1
Jane Doe	25	50000	0
Robert Junior	42	50000	1

Step 4: Feature Engineering

- Creating or transforming input variables to improve model performance
- Combines domain knowledge with data intuition
- Examples:
  - From "date" $\rightarrow$ extract hour, day of week, is weekend?
  - From raw text $\rightarrow$ word counts or sentiment scores
  - From financial data $\rightarrow$ debt-to-income ratio, rolling averages
- Choose features that are predictive, interpretable, and generalizable
- "Better data beats fancier algorithms"

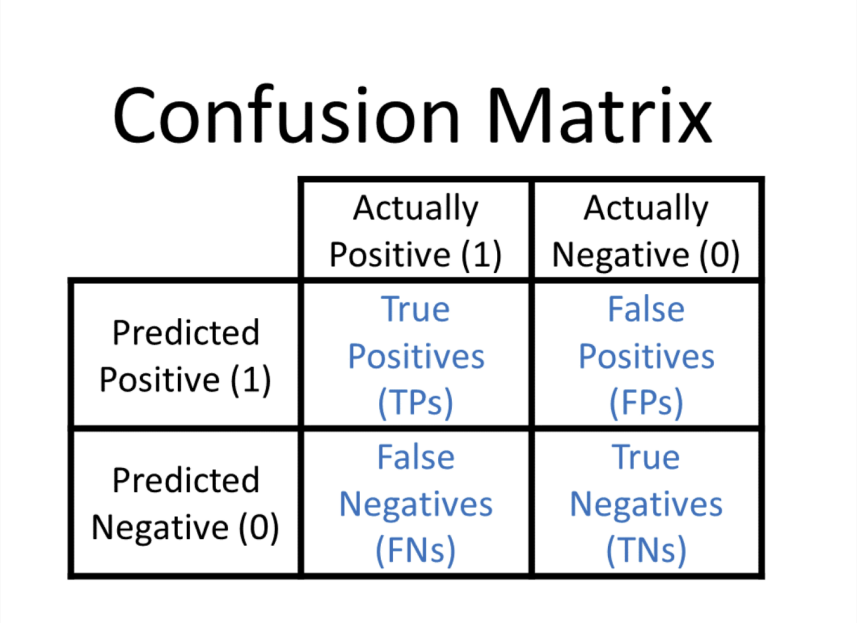
Step 5: Train the Model

- Use training data to teach the model how to map inputs to outputs
- Choose an algorithm based on task and data size:
  - Linear Regression, Decision Trees, k-NN, SVM, Random Forest, XGBoost, etc.
- The model adjusts its internal parameters to minimize prediction error
- Training is an optimization problem (e.g., minimizing loss function)



Step 6: Evaluate the Model

- Use a separate test set to measure generalization performance
- Choice of evaluation metric depends on the task:
  - **Regression:** Mean Squared Error (MSE), Mean Absolute Error (MAE), R^2
  - **Classification:** Accuracy, Precision, Recall, F1-Score
- Use confusion matrix to visualize performance in classification
- A good model performs well not only on training data but also on unseen data

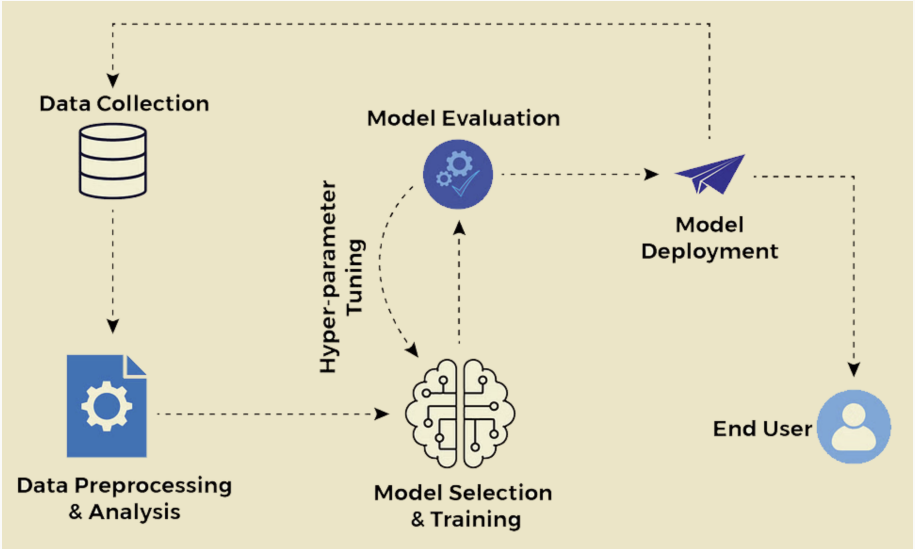


Step 7: Tuning and Optimization

- Even after training, models can be improved through tuning
- Most ML models have **hyperparameters** — settings not learned from data
  - Examples: tree depth, learning rate, number of neighbors
- Use cross-validation to evaluate model stability
- Common techniques:
  - Grid Search
  - Random Search
- Goal: Find the hyperparameter set that yields the best validation performance

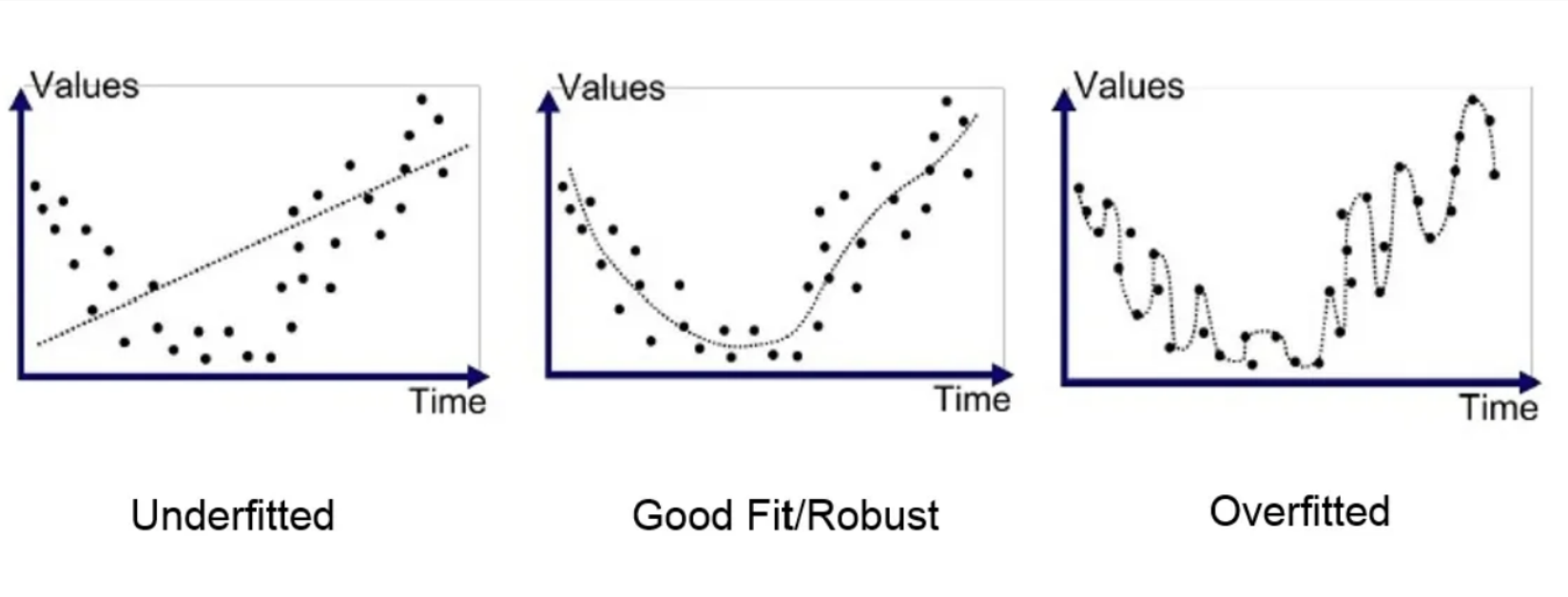
Step 8: Deploy the Model

- Once validated, the model can be integrated into real systems
- Common deployment targets:
  - Web applications (e.g., recommendation systems)
  - Mobile apps (e.g., on-device speech recognition)
  - APIs (e.g., prediction endpoints)
- Monitor for performance decay, drift, or bias
- Deployment is not the end — it's the beginning of iteration



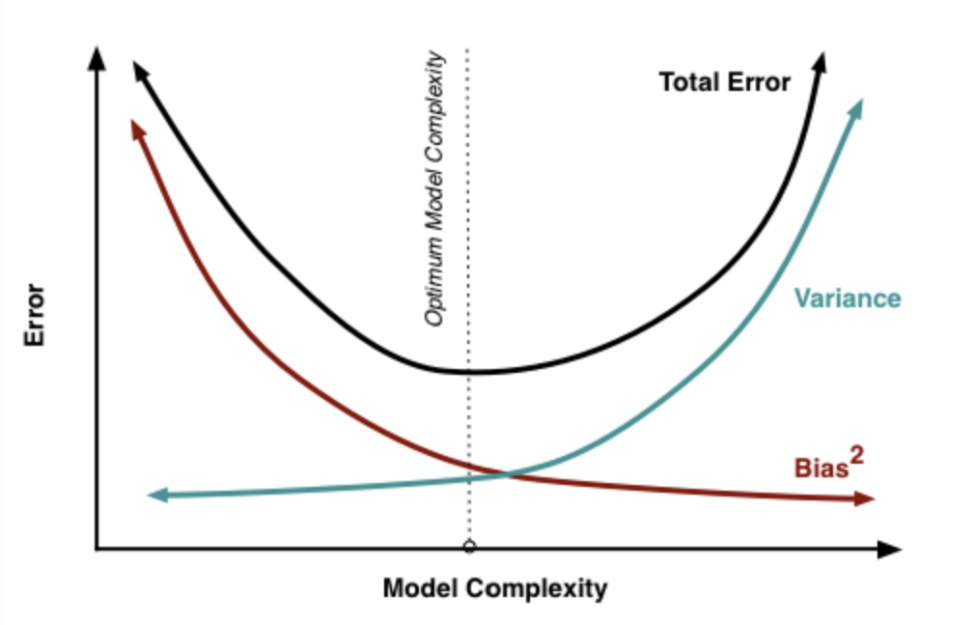
Overfitting vs Underfitting

- **Underfitting:** Model is too simple, fails to capture patterns (high bias)
- **Overfitting:** Model is too complex, memorizes noise (high variance)
- Goal: Generalize well to unseen data



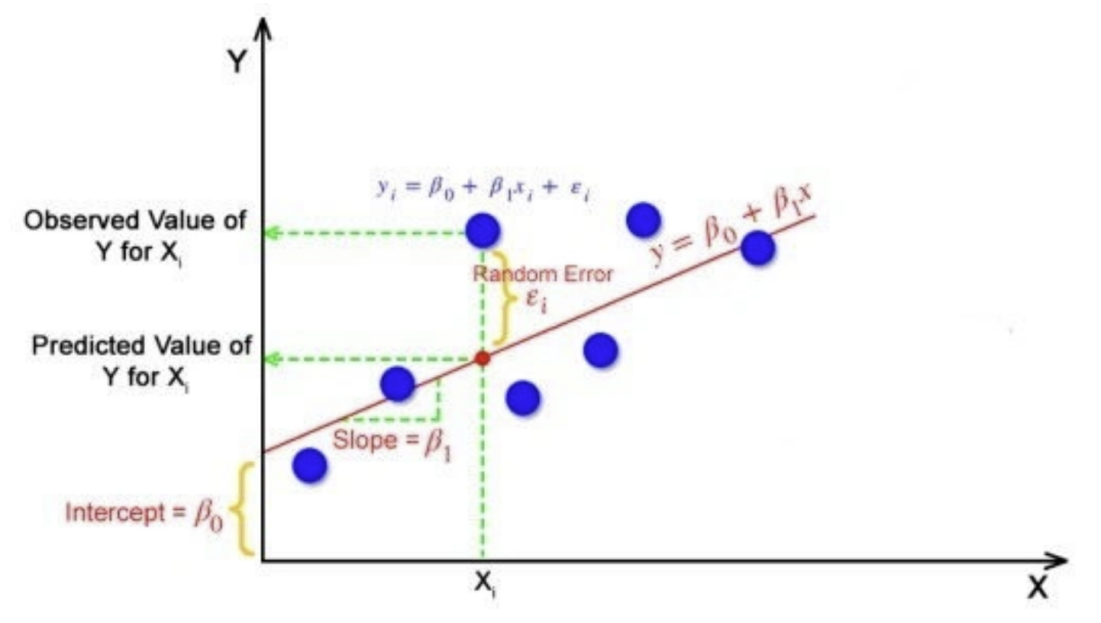
Bias-Variance Tradeoff

- A fundamental tradeoff in machine learning model design:
  - **Bias:** Error from incorrect assumptions (underfitting)
  - **Variance:** Error from sensitivity to small fluctuations in training data (overfitting)
- **Goal:** Minimize total error by balancing bias and variance
- Model complexity affects this tradeoff — more complex always better



Linear Regression Intuition

- A fundamental algorithm used for predicting continuous values
- It finds the line that best fits the data by minimizing the squared error
- Equation: $y = \beta_0 + \beta_1 x$
- Parameters β_0 (intercept) and β_1 (slope) are learned from data
- Example: Predicting house price based on square meters



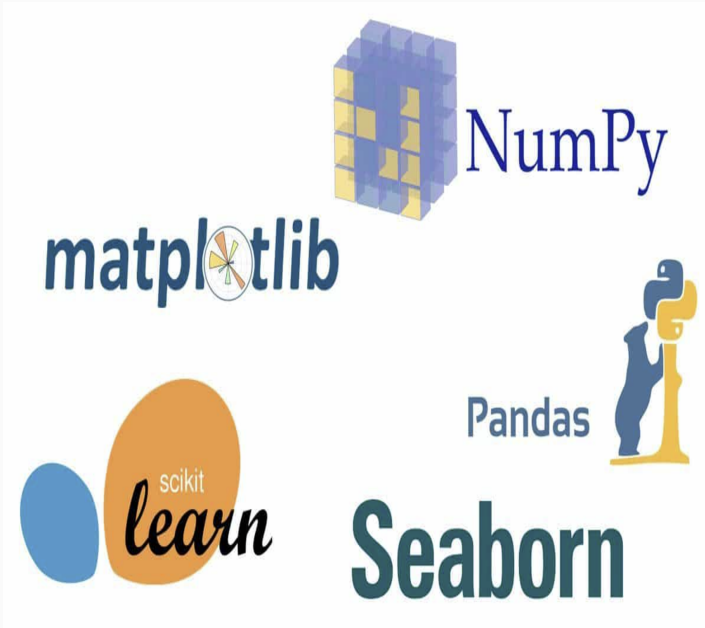
Tools and Ecosystem for Machine Learning

Popular Languages

- Python (most widely used)
- R (especially in statistics)

Core Libraries

- **scikit-learn:** Classical ML
- **Pandas & NumPy:** Data handling
- **Matplotlib & Seaborn:** Visualization
- **TensorFlow, PyTorch:** Deep learning



Getting Started in Google Colab

- Google Colab is a free, cloud-based environment for running Python notebooks
- No setup required — runs entirely in your browser
- You can:
  - Write and run Python code
  - Upload and manipulate data
  - Visualize results with plots
  - Train and test ML models
- Today, we'll walk through:
  - Loading and exploring data
  - Writing your first regression model
  - Understanding how model training works

→ Let's switch to Colab and get hands-on!