Table of Contents

---

1. Introduction to Machine Learning

Machine Learning is all about teaching computers to *learn* from data and make decisions based on patterns—without us having to manually tell it what to do. Instead of hardcoding rules, we let data and math do the talking. 🧠

Imagine you want your app to suggest a new outfit based on your mood and the weather. Instead of programming thousands of combinations, you'd feed past choices and outcomes to a model that learns what works. Voilà—ML in action.

---

2. Mathematical Foundations (The Core Mindset)

If ML is a house, math is the foundation. Don't worry—we'll break it down intuitively with relatable examples. ➕➖➗✖️

2.1 Linear Algebra

- **Vectors**: Think of these like one-dimensional lists of numbers. In ML, features (like age, height, GPA) are often stored as vectors.
- **Matrices**: A grid of numbers (like your timetable). Datasets are usually matrices.
- **Dot Product**: Measures similarity between vectors.
    - Example: Comparing two Spotify playlists—more overlap in songs → higher dot product.
- **Matrix Multiplication**: Used in linear transformations, like projecting your data into a new space for analysis.
- **Eigenvalues & Eigenvectors**: Help in PCA (Principal Component Analysis), which reduces data dimensions (like summarizing 10 essays into 2 main ideas).

2.2 Probability & Statistics

- **Random Variable**: An outcome you can't predict with certainty but can model.
- **Probability Distribution**: How likely each outcome is.
- **Mean, Median, Mode**: Central tendencies.
- **Variance & Standard Deviation**: Measure of spread in your data.
- **Bayes' Theorem**: [P(A|B) = \frac{P(B|A)P(A)}{P(B)} \ ]
    - Super useful in models like Naive Bayes for spam classification.

2.3 Calculus for Optimization
ML models often learn by minimizing a cost function.

- **Derivatives**: Measure the rate of change. Like: How fast your GPA changes with every extra hour of study.
- **Gradient Descent**: A step-by-step algorithm to find the minimum of a function.
  - Like slowly finding the lowest point in a hilly area, by always walking downhill.
  - Formula:

2.4 Evaluation Metrics
For Classification:

- **Accuracy** = Correct predictions / Total predictions
- **Precision** = TP / (TP + FP)
- **Recall** = TP / (TP + FN)
- **F1-Score** = 2 * (Precision * Recall) / (Precision + Recall)
- **Confusion Matrix** = Table showing TP, FP, FN, TN.
  Imagine testing whether an email is spam:
    - True Positive = Correctly predicted spam
    - False Positive = Predicted spam but wasn't
- **ROC-AUC Curve**: Shows how well your model distinguishes between classes.

For Regression:

- **Mean Absolute Error (MAE)** = Average of absolute errors
- **Mean Squared Error (MSE)** = Average of squared errors
- **Root Mean Squared Error (RMSE)** = Square root of MSE
  MAE is like your average mistake, MSE punishes big mistakes harder.

3. Core ML Terminologies

- **Feature**: An input variable (like 'age', 'income').
- **Label**: The output we want to predict (like 'will buy' or 'won't buy').
- **Training Set**: Data used to train the model.
- **Test Set**: Data used to evaluate the trained model.
- **Overfitting**: The model memorizes training data too well and fails on new data.
- **Underfitting**: The model is too simple to capture the patterns.

---

4. Categories of ML Algorithms

Supervised Learning
Here, the model learns from labeled data. It's like preparing for an exam with an answer key.
Examples:

- Email Spam Detection (Label: spam or not)
- Movie Recommendation (Label: liked or not)

Types:

- **Classification**: Output is a category (e.g., cat vs dog).
- **Regression**: Output is continuous (e.g., predicting salary).

Unsupervised Learning
No labels! The model finds hidden patterns.
Examples:

- Grouping customers by behavior
- Organizing your selfies into albums based on vibes

Types:

- **Clustering**: Group similar data points
- **Dimensionality Reduction**: Reduce data complexity

---

5. Supervised Learning Algorithms

5.1 Linear Regression

Used for: Predicting continuous values.
Equation:

- values are the model coefficients.
- is the error term.

Goal: Minimize the cost function:
Real-World Example: Predicting the price of a college apartment based on area, distance from campus, and number of roommates.

5.2 Logistic Regression

Used for: Binary classification (Yes/No, True/False).
Equation:

- Uses **Sigmoid function** to squash output between 0 and 1.

Real-World Example: Will I pass the midterm based on my study hours and attendance? 🤓

5.3 Decision Tree

Used for: Both classification and regression.
Key Concepts:

- **Node**: A decision point.
- **Leaf**: Final outcome.
- **Entropy**:
- **Information Gain**:
- **Gini Index**:
- **Split Criteria**: Choose the attribute that gives the highest information gain or lowest Gini.

Real-World Example: Classifying if a movie is worth watching based on trailer length, actors, and genre.

5.4 K-Nearest Neighbors (KNN)

Used for: Classification and regression.
How it works:

- Find the 'K' closest points to the input.
- Classification: majority vote.
- Regression: average of neighbors' outputs.
- **Distance Metric**:

Real-World Example: What outfit should I wear today? Look at what people with similar taste wore in similar weather.

5.5 Random Forest

Used for: Classification and regression.

- Ensemble of decision trees.
- Uses **bagging** (bootstrap aggregation).
- Random subsets of features and data improve accuracy.

Real-World Example: Asking multiple friends for fashion advice and choosing the majority opinion.

5.6 Naive Bayes
Used for: Classification problems.
Based on Bayes' Theorem:
Assumes features are **independent**.
Real-World Example: Predicting if an email is spam based on keywords, without worrying about how words relate to each other.

---

6. Unsupervised Learning Algorithms

6.1 K-Means Clustering
Goal: Group data into K clusters.
Steps:

1. Choose K initial centroids.
2. Assign each point to the nearest centroid.
3. Recalculate centroids.
4. Repeat until convergence.

Distance: Usually Euclidean.
Real-World Example: Organizing Spotify songs into mood-based playlists without knowing genres beforehand.
6.2 K-Medoids

- Similar to K-Means but uses actual data points (medoids) as centers.
- More robust to outliers.

Real-World Example: Forming friend groups by choosing actual people as group leaders, not made-up averages.

6.3 Hierarchical Clustering
Two types:

- **Agglomerative**: Bottom-up (each point starts as its own cluster).
- **Divisive**: Top-down (start with one cluster and split).

Distance Matrix + Dendrogram used to visualize clusters.
Real-World Example: Organizing your photo gallery into nested albums: "College → Events → Birthdays."

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

- Groups densely packed points.
- Doesn't require K.
- Can detect outliers (noise).

Key Terms:

- **Epsilon (ε)**: Radius for neighborhood.
- **MinPts**: Minimum points in the ε-neighborhood.

Real-World Example: Detecting groups of friends hanging out in a canteen (while ignoring loners).

6.5 Spectral Clustering

- Uses graph theory.
- Builds a similarity graph → computes Laplacian matrix → performs eigen-decomposition.

Steps:

1. Create a similarity matrix.
2. Compute the Laplacian.
3. Apply K-means to top eigenvectors.

Real-World Example: Grouping users on a dating app by shared interests + social connections.

GLOSSARY

- **Model**: A mathematical representation of a process based on data.
- **Training**: Teaching the model using data.
- **Prediction**: The model's output for unseen input.
- **Feature Vector**: A list of values that represent a data point.
- **Label**: The output or category the model is trying to predict.
- **Bias**: Error from assumptions in the learning algorithm.
- **Variance**: Error from sensitivity to small fluctuations in training data.
- **Underfitting**: Model is too simple to capture patterns.
- **Overfitting**: Model learns noise along with patterns.
- **Hyperparameter**: A parameter set before training (like K in KNN).
- **Parameter**: Learned from data (like weights in Linear Regression).
- **Normalization**: Scaling values to a range (e.g., 0 to 1).
- **Standardization**: Mean = 0, Standard Deviation = 1 transformation.
- **Confusion Matrix**: A table for evaluating classification models.
- **Precision**: How many selected items are relevant.
- **Recall**: How many relevant items are selected.
- **F1 Score**: Harmonic mean of Precision and Recall.
- **ROC Curve**: Plots TPR vs FPR.
- **AUC (Area Under Curve)**: The higher, the better the model.