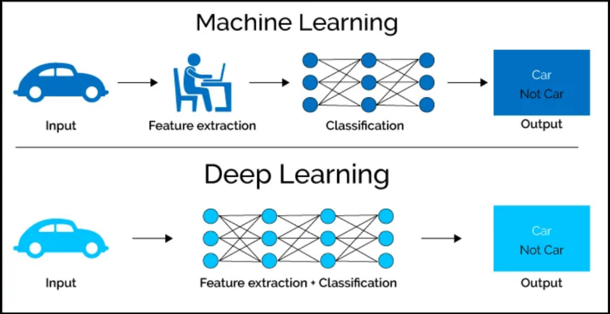
KEYSIGHT

# Week 02- Session 01 - Supervised Learning




Kelvin C. - PassDowns/Walkthroughs II

# Table Of Content

# What is Supervised Learning?

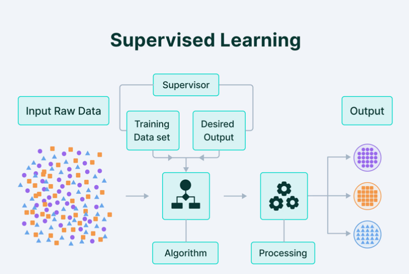



*Figure: Illustration of Supervised Learning.(Sourced from web).*

- A core type of machine learning where a model learns from labeled data.
- Goal: Learn a function that maps inputs (features) to outputs (labels).
- Examples:
  - Spam detection: Emails $\rightarrow$ `spam/not spam`
  - Image classification: Images $\rightarrow$ `cat/dog/etc.`

# Core Concepts

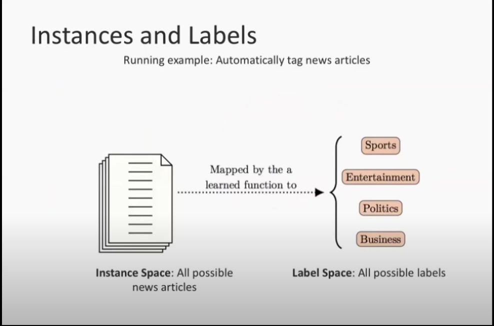



*Figure: Illustration of Variables in Stats.Sourced from Google.*

- **Instance Space** ($\mathcal{X}$): Set of all possible input examples (e.g., feature vectors).
- **Label Space** ($\mathcal{Y}$): Set of all possible outputs (e.g., classes or real values).
- **Hypothesis Space** ($\mathcal{H}$): Set of all models/functions we consider: $h : \mathcal{X} \to \mathcal{Y}$.

**Hypothesis Space ($\mathcal{H}$)**: Set of all models/functions we consider: $h : \mathcal{X} \to \mathcal{Y}$.

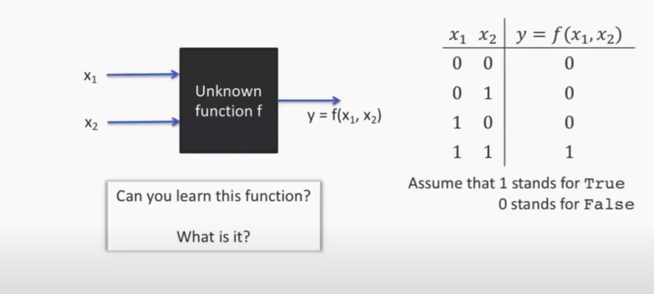



*Figure: Illustration of Hypothesis Space. Sourced from Google.*

# Classified Variables in Stats

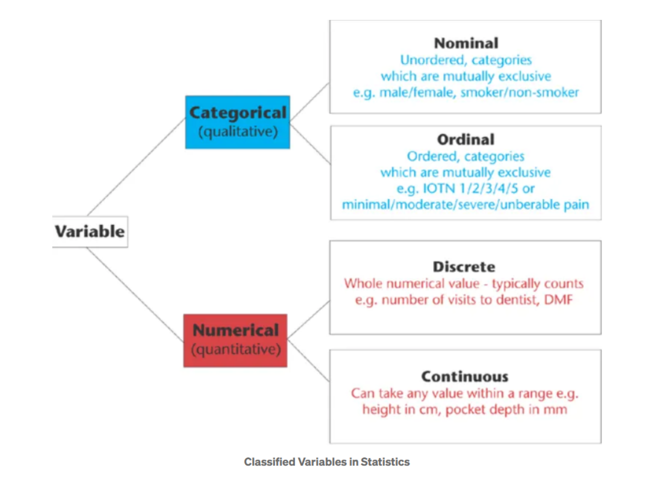



*Figure: Illustration of Variables in Stats.Sourced from Google.*

# Types of Variables (Features) in Supervised Learning

## Numerical Variables

- Represent quantities or measurements.
- Take continuous or discrete numeric values.
- Examples:
  - Age: 25, 30, 45 years
  - Height: 172.5 cm
  - Temperature: 36.6°C

## Categorical Variables

- Represent categories or labels.(Nominal or Ordinal)
- Take values from a limited set of discrete categories.
- Examples:
  - Color: Red, Blue, Green
  - Type of animal: Cat, Dog, Bird
  - Education Level: High School, Bachelor's Degree, Master's Degree, PhD.

**Why it matters?** Some algorithms handle numerical and categorical data differently, so understanding variable types helps in data preprocessing and model choice.

# Supervised Learning Setup

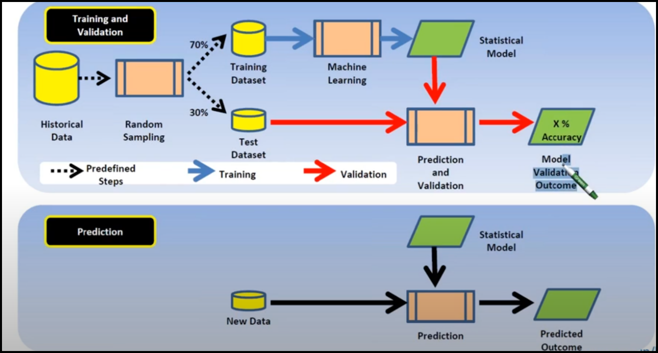



*Figure: Process/flow of Supervised Learning.(Sourced from web).*

# Parametric vs Non-Parametric Models in ML Algorithms

**How does a model learn? One key distinction:**

## Parametric Models

- Assume a fixed form (e.g., linear) for the function $h(x)$.
- Learning $=$ estimating a finite set of parameters.
- **Pros:** Simple, fast to train, less prone to overfitting.
- **Cons:** Limited flexibility.
- **Examples:** Linear Regression, Logistic Regression, Naive Bayes

## Non-Parametric Models

- Do not assume a fixed form. Complexity can grow with data.
- Learning = memorizing and generalizing from data directly.
- **Pros:** Flexible, can capture complex patterns.
- **Cons:** More data needed, slower, risk of overfitting.
- **Examples:** k-Nearest Neighbors (k-NN), Decision Trees, Random Forests

# Parametric vs Non-Parametric Models

**Two broad categories of supervised learning models:**

| **Aspect** | **Parametric Models** | **Non-Parametric Models** |
| --- | --- | --- |
| Model Structure | Assumes a fixed form or structure (e.g., linear) | No fixed form; adapts to the data |
| Number of Parameters | Fixed and finite | Grows with data size |
| Training Data Needs | Works well with less data | Typically needs more data |
| Computation Time | Fast training and inference | Slower, especially with large data |
| Flexibility | Less flexible, risk of underfitting | More flexible, risk of overfitting |

Table: Comparison of Parametric and Non-Parametric Models

# Supervised Learning—Classification

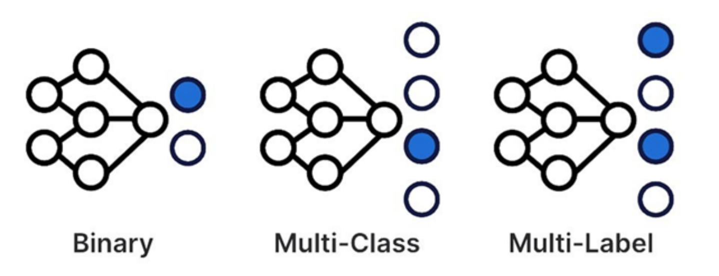



*Figure: Types of classification. Image sourced from Web.*

# Common Classification Algorithms

Here are some widely used machine learning algorithms for classification tasks:

## Decision Tree

- Splits data based on feature values to form a tree-like model.
- Easy to interpret, can overfit on small data.

## Random Forest

- An ensemble of decision trees.
- Reduces overfitting and improves accuracy.

## Support Vector Machine (SVM)

- Finds the optimal hyperplane that separates classes.
- Effective in high-dimensional spaces, works well with clear margins.

## Logistic Regression

- Models the probability of a class using a logistic function.
- Good baseline model for binary classification.

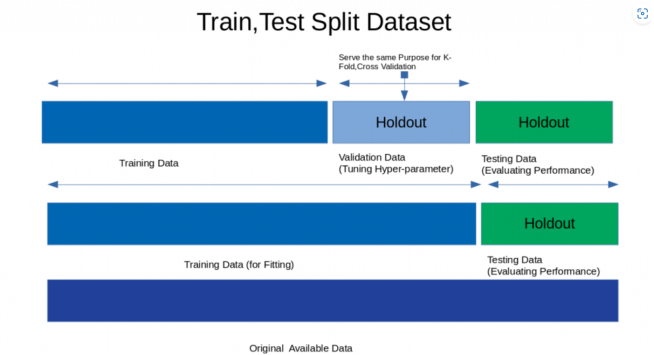
## Linear Discriminant Analysis (LDA)

- Projects data to maximize class separability.
- Assumes normal distribution and equal class covariances.

## Naive Bayes

- Probabilistic classifier based on Bayes' theorem.
- Assumes feature independence (naive assumption).
- Fast and works well for text classification.

# Dataset-Splitting




*Ratio of Training x Testing + Val in splitting Dataset. Image sourced from web.*

# Evaluation Metric in Classification

**How do we measure the performance of a classification model?**

# Accuracy

## Accuracy

The proportion of correctly predicted instances among all instances:

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

# Confusion Matrix

## Confusion Matrix

A table that summarizes prediction results:

| | **Predicted Positive** | **Predicted Negative** |
|---|---|---|
| **Actual Positive** | True Positive (TP) | False Negative (FN) |
| **Actual Negative** | False Positive (FP) | True Negative (TN) |

# Precision and Recall

## Precision

Out of all the positive predicted, what percentage is truly positive.:

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

## Recall (Sensitivity)

Out of the total positive, what percentage are predicted positive.:

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

# F1 Score

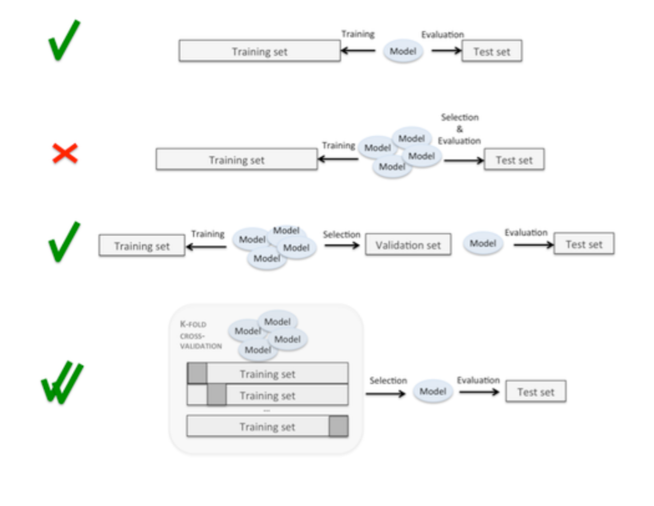
## F1 Score

Harmonic mean of precision and recall, used to balance the trade-off between them:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

*F1 Score is especially useful when dealing with imbalanced classes.*

# Model Evaluation



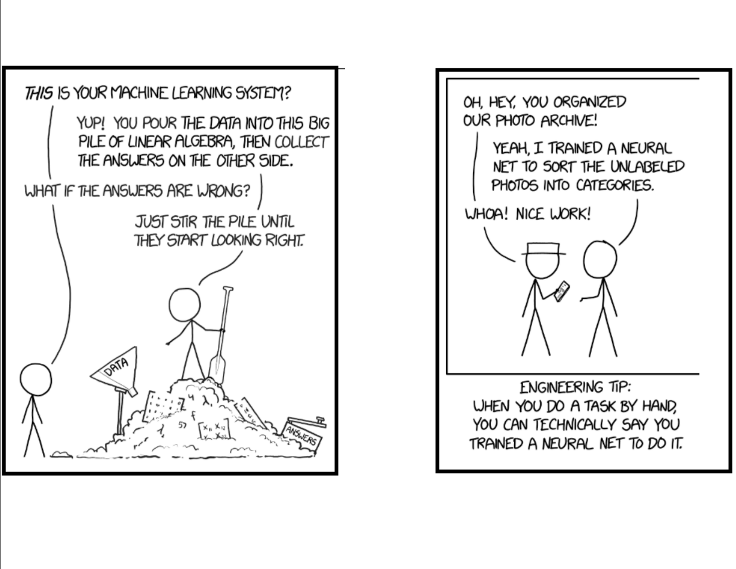

*Model Evaluation. Image sourced from web.*

# Useful Videos on YouTube

Check out these interesting YouTube video on Supervised Learning & EDA:

- Machine Learning YT Playlist-UOFU
- EDA Intro

# The End



THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS
OH, HEY, YOU ORGANIZED OUR PHOTO ARCHIVE!
YEAH, I TRAINED A NEURAL NET TO SORT THE UNLABELED PHOTOS INTO CATEGORIES.
WHOA! NICE WORK!
ENGINEERING TIP:
WHEN YOU DO A TASK BY HAND, YOU CAN TECHNICALLY SAY YOU TRAINED A NEURAL NET TO DO IT.

# Week 02- Session 02 - Linear Programming x Dynamic Programming


Kelvin C. - PassDowns/Walkthroughs II

**Linear Programming (LP):**

- Mathematical technique to optimize a linear objective function
- Subject to linear equality and inequality constraints
- Commonly used in resource allocation, scheduling, and operations research

**Dynamic Programming (DP):**

- Algorithmic technique for solving complex problems by breaking them into simpler subproblems
- Uses recursion and memoization or tabulation
- Widely used in optimization problems, decision making, and combinatorics

# Intro to Dynamic Programming

Check out these interesting YouTube video on DP:

- DP in Knapsack Problem

# Differences

| Feature | Linear Programming | Dynamic Programming |
|---|---|---|
| Nature | Mathematical optimization model | Algorithmic paradigm |
| Problem Type | Linear objective and constraints | Problems with overlapping subproblems |
| Approach | Solves using simplex or interior point methods | Breaks problem into subproblems recursively |
| Typical Applications | Resource allocation, production planning | Sequence alignment, shortest paths, knapsack |

Table: Comparison between Linear Programming and Dynamic Programming

# Whats Next? - Lets get your hands dirty with it =D


*Meme Sourced from web.*