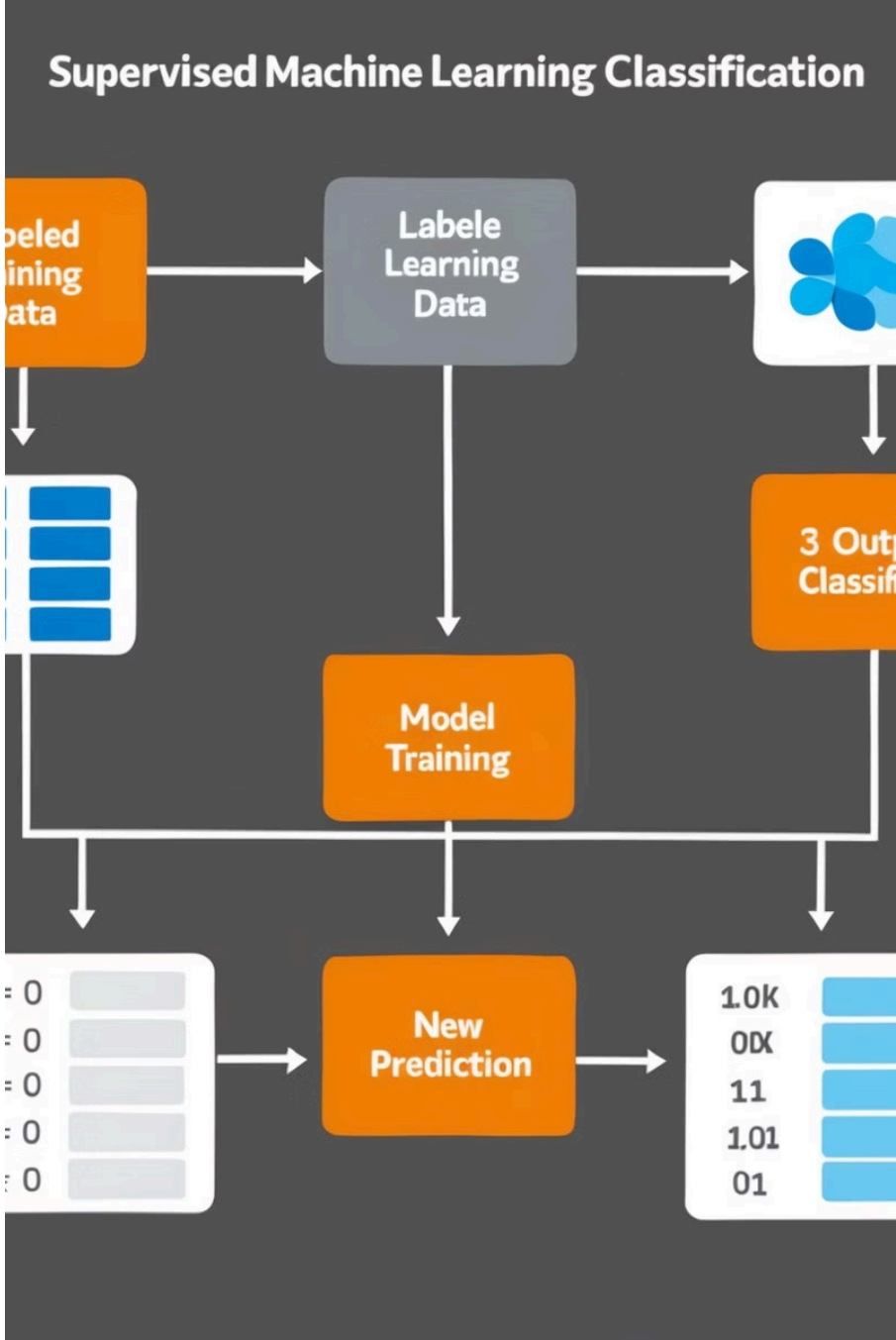
# Supervised Machine Learning – Classification



# Introduction to Supervised Learning

## Definition

Supervised Learning involves training on labeled data, where both features and target variables are provided. A common example is spam detection in emails.

## Types

- Regression: Predicting continuous values (e.g., house prices)
- Classification: Predicting discrete categories (e.g., spam or not spam)

# Classification Problem: Diabetes Prediction

## Problem Definition

**Binary Classification Task:** Predict diabetes status using plasma glucose levels as the primary feature.

Goal: Develop a machine learning model that can accurately classify patients as diabetic or non-diabetic based on their plasma glucose measurements.

## Sample Data

| Plasma Glucose Score | Diabetic Status |
|---|---|
| 90 | No |
| 120 | Yes |
| 100 | No |
| 98 | No |
| 130 | Yes |
| 170 | Yes |
| 101 | No |
| 110 | No |

# Probability vs. Odds

## Probability

Ratio of favorable outcomes to total outcomes.

Range: 0 to 1 (0% to 100%).

Example: 70% chance of rain = 0.7 probability.

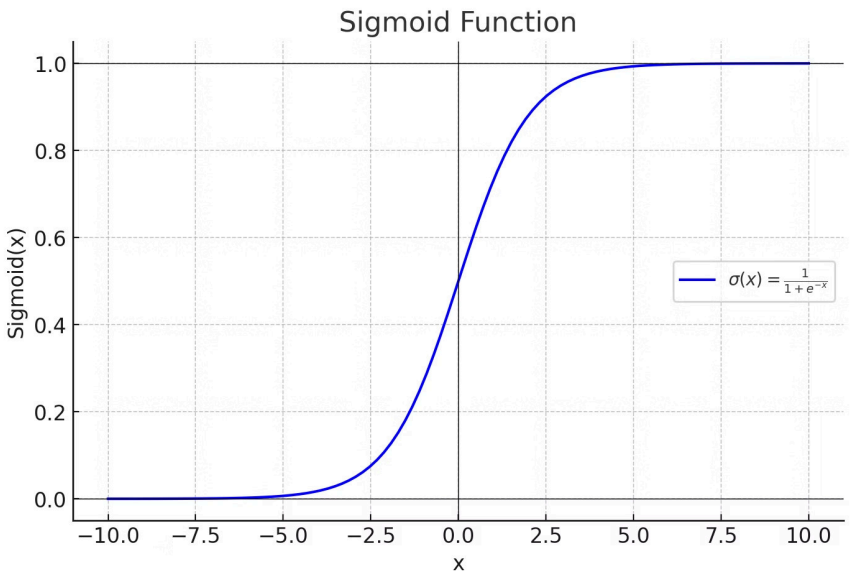
## Odds

Compares success likelihood to failure likelihood.

Range: 0 to ∞.

Example: 70% rain chance = 2.33 odds (2.33 to 1).

# What is a sigmoid function?




The equation of the sigmoid function is:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

# Logistic regression

- Logistic regression is a binary classification algorithm
- It predicts the probability of occurrence of a class label. Based on these probabilities the data points are labelled
- A threshold (or cut-off; commonly a threshold of 0.5 is used) is fixed.

| | |
|---|---|
| Probability > Threshold | Is Diabetic |
| Probability < Threshold | Is not Diabetic |

# Maximum Likelihood Estimation (MLE) in Logistic Regression

**Likelihood** refers to the probability of observing a given set of data under a specific model or assumption. It is commonly used in **statistics** and **machine learning** for parameter estimation.

Unlike probability, which measures how likely an event is to occur, **likelihood** measures how well a given model explains the observed data.

Example: Coin Toss and Maximum Likelihood Estimation (MLE) Imagine you are trying to determine whether a coin is biased or fair. You don't know if the coin is fair (i.e., the probability of heads is 0.5), so you conduct an experiment by flipping the coin multiple times and recording the outcomes.

Let’s say you observe the following data:

10 flips: H, T, H, H, T, H, H, H, T, H This means 7 heads and 3 tails. Your goal is to estimate the probability of getting heads, denoted as θ.

## Step 1: Define the Likelihood Function

In probability, the chance of getting **k heads** in **n flips** follows a **binomial distribution**:

$$P(X = k) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

- $n = 10$ (total flips)
- $k = 7$ (heads count)
- $\theta$ = probability of heads (what we want to estimate)

The **likelihood function** is:

$$L(\theta) = P(X = 7|\theta) = \binom{10}{7} \theta^7 (1-\theta)^3$$

Since the binomial coefficient is a constant for given $n$ and $k$, we focus on the part that depends on $\theta$:

$$L(\theta) = \theta^7 (1-\theta)^3$$

## Step 2: Find the Maximum Likelihood Estimate (MLE)

To find the value of $\theta$ that **maximizes the likelihood**, we take the **derivative of the likelihood function** and set it to **zero**.

$$\frac{d}{d\theta}[\theta^7 (1-\theta)^3] = 0$$

Since working directly with likelihood functions can be complex, we often take the **log-likelihood**, which simplifies differentiation:

$$\log L(\theta) = 7\log(\theta) + 3\log(1-\theta)$$

Now, differentiate with respect to $\theta$:

$$\frac{d}{d\theta}[7\log(\theta) + 3\log(1-\theta)] = \frac{7}{\theta} - \frac{3}{1-\theta}$$

Setting this equal to 0:

$$\frac{7}{\theta} = \frac{3}{1-\theta}$$

Solving for $\theta$:

Thus, the **maximum likelihood estimate (MLE) for θ is 0.7**. This suggests that based on our observed data, the best estimate for the probability of getting heads is **70%**

# Logistic Regression Assumptions

1. **Independence of Error**

   Sample group outcomes are separate; no duplicate responses.

2. **Linearity in Logit**

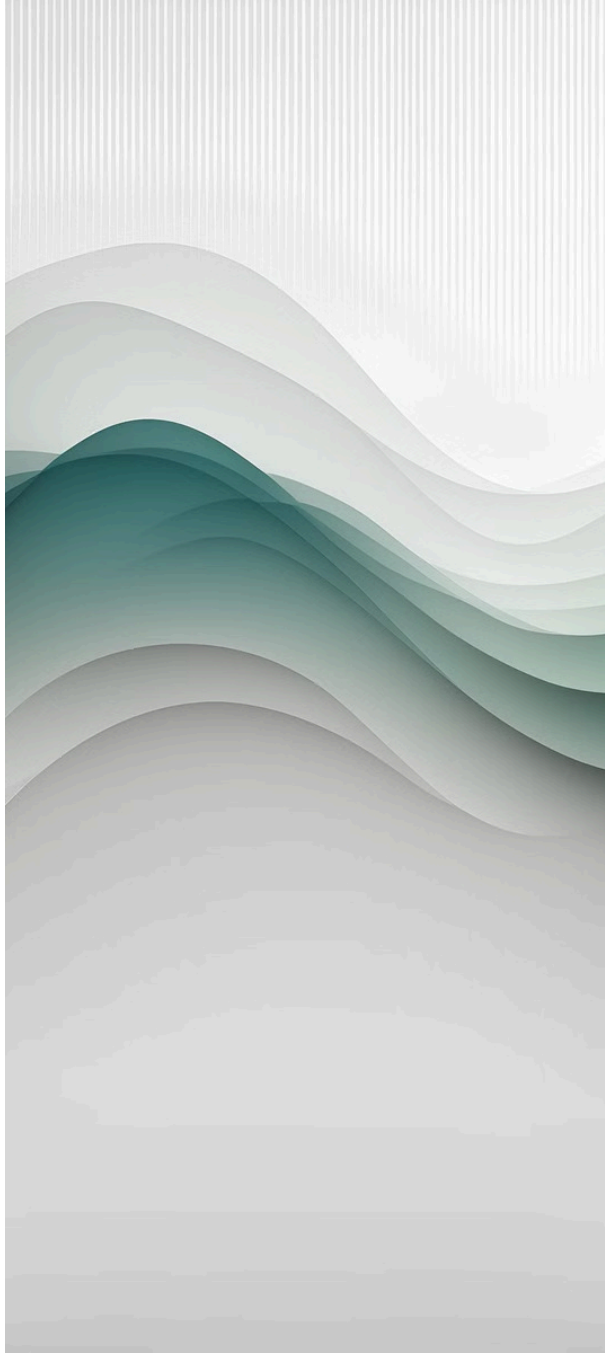
   Continuous independent variables have linearity in the logit.

3. **Absence of Multicollinearity**

   Multicollinearity should be absent.

4. **Lack of Influential Outliers**

   Strongly influential outliers should be absent.



# Model Evaluation– Pseudo $R^2$

- The non-pseudo $R^2$ or the $R^2$ in the linear regression framework is the explained variability and the correlation (for simple linear regression)
- An equivalent $R^2$ statistic does not exist in the logistic regression since the parameters are estimated by the method of maximum likelihood
- However, there are various pseudo $R^2$s developed which are similar on the scale, i.e., on [0,1], and work exactly the same with higher values indicating a better fit

# Confusion matrix

- Performance measure for classification problem
- It is a table used to compare predicted and actual values of the target variable

| | Actual values | |
|---|---|---|
| | **Positive (1)** | **Negative (0)** |
| **Predicted values** | | |
| **Positive (1)** | **True Positive:** Predicted value is positive and the actual value is also positive | **False Positive:** Predicted value is positive but the actual value is negative |
| **Negative (0)** | **False Negative:** Predicted value is negative but the actual value is positive | **True Negative:** Predicted value is negative and the actual value is also negative |

-

# Performance evaluation metrics

Confusion matrix can be used to calculate the following evaluation metrics for a model:

- Accuracy
- Precision
- Recall
- False Positive Rate
- Specificity
- $F_1$ score
- Kappa

# Accuracy

- **Definition**: Accuracy is the proportion of correctly predicted instances (both positives and negatives) out of the total instances.

$$\text{Accuracy} = \frac{\text{number of correctly predicted records}}{\text{Total number of records}}$$

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

- **Interpretation**: A high accuracy means the model is making more correct predictions overall. However, it can be misleading in imbalanced datasets (e.g., if 95% of the data belongs to one class, a model predicting only that class would have 95% accuracy but be useless).

## Precision (Positive Predictive Value)

- **Definition**: Precision measures how many of the predicted positive instances were actually positive.
- **Formula**:

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

- **Interpretation**: A high precision means fewer false positives. This is crucial in scenarios like spam detection (where predicting a legitimate email as spam is costly).

## Recall (Sensitivity or True Positive Rate)

- **Definition**: Recall measures how many actual positive instances were correctly predicted as positive.
- **Formula**:

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

- **Interpretation**: A high recall means fewer false negatives. This is important in medical diagnoses (e.g., missing a cancer diagnosis is far worse than falsely predicting cancer).

## False Positive Rate (FPR)

- **Definition**: FPR indicates how many actual negative instances were wrongly classified as positive.
- **Formula**:

$$\text{FPR} = \frac{FP}{FP + TN}$$

- **Interpretation**: A low FPR is desirable in scenarios like fraud detection (false fraud alerts should be minimized).

## Specificity (True Negative Rate)

- **Definition**: Specificity measures how many actual negative instances were correctly predicted as negative.
- **Formula**:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- **Interpretation**: A high specificity means fewer false positives. This is useful in medical tests where unnecessary treatment should be avoided.

## F1 Score

- **Definition**: The **F1 score** is the harmonic mean of **precision** and **recall**, balancing both metrics.
- **Formula**:

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

- **Interpretation**: A high F1 score means a good balance between precision and recall. It's useful when both false positives and false negatives are costly.

## Kappa (Cohen's Kappa)

- **Definition**: Cohen's Kappa measures how well the model performs compared to random guessing, considering agreement between predicted and actual values.
- **Formula**:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where:

- $P_o$ = observed agreement (accuracy)
- $P_e$ = expected agreement by chance

- **Interpretation**: Kappa values range from **0 (random predictions)** to **1 (perfect agreement)**. It is useful for handling imbalanced data.

# Kappa statistic

- **Kappa statistic** is a measure of inter-rater reliability or degree of agreement.
- **Kappa statistic** can take values from the range **[-1,1]**.

### Kappa Interpretation Table

| Kappa | Interpretation |
|---|---|
| < 0 | No agreement |
| 0 - 0.2 | Slight agreement |
| 0.2 - 0.4 | Fair agreement |
| 0.4 - 0.6 | Moderate agreement |
| 0.6 - 0.8 | Substantial agreement |
| 0.8 - 1 | Almost perfect agreement |

# KNN algorithm

### Basic Concept

The K - Nearest Neighbour (KNN) algorithm classifies the data based on the similarity measure

### The Role of K

K specifies the number of nearest Neighbours to be considered

### Training Requirements

**Does not require the data to be trained**

# KNN algorithm – Procedure

1 **Initial Setup**

Choose a distance measure and value of K

2 **Distance Calculation**

Compute the distance between the point whose label is to be identified (say x) and other data points

3 **Distance Sorting**

Sort the distances in ascending order

4 **Label Assignment**

Choose K data points which have the shortest distances and note their corresponding labels. Then the label which has the highest frequency will be assigned to the point x

# Example

- **Humidity:** (Independent variable) the percentage of humidity in the atmosphere
- **Temperature:** (Independent variable) the temperature—average temperature during precipitation
- **Rain:** (Target variable) indicates whether it rained or not; takes value **1** if rained and value **0** otherwise

| Observation | Humidity | Temperature | Rain |
|---|---|---|---|
| 1 | 58 | 19 | 0 |
| 2 | 62 | 26 | 0 |
| 3 | 40 | 30 | 0 |
| 4 | 36 | 35 | 0 |
| 5 | 87 | 19 | 1 |
| 6 | 93 | 18 | 0 |
| 7 | 79 | 16 | 1 |
| 8 | 69 | 17 | 1 |
| 9 | 62 | 33 | 0 |
| 10 | 71 | 15 | 0 |

# Example

Let us choose K = 5 and use the Euclidean distance.

Compute the Euclidean distance between the new data point (Humidity = 84, Temperature = 34) and each instance in the dataset.

For example, considering the first observation where: Humidity = 58 Temperature = 19

The Euclidean distance is calculated as:

$$\sqrt{(58-84)^2+(19-34)^2} = 31.623$$

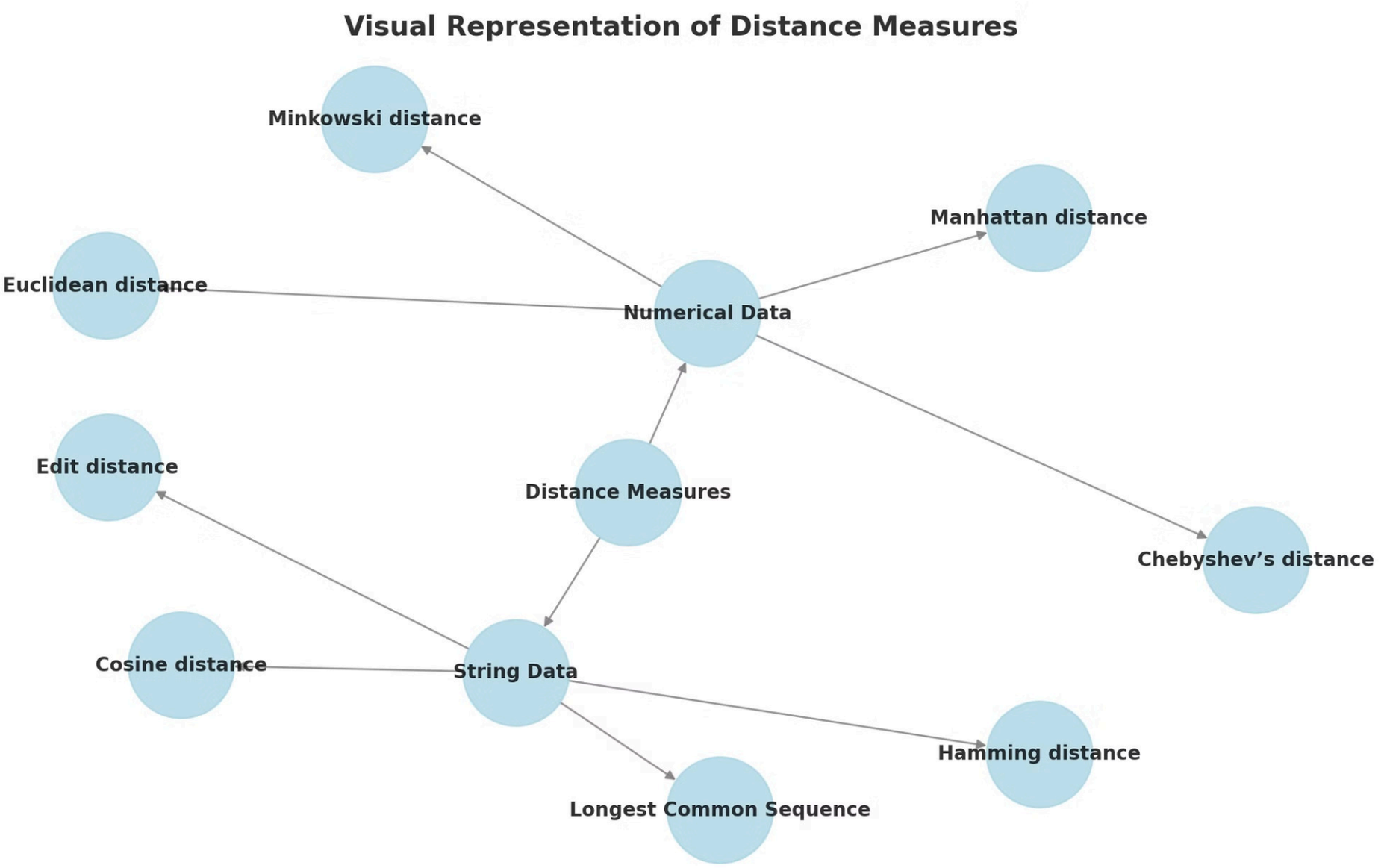
## Example

- Computed the **Euclidean distances** for each instance with the new data and **sorted** the data in ascending order with respect to the Euclidean distance.
- Sorted Euclidean Distances and Corresponding Rainfall Class Labels:

| Observation | Euclidean Distance (sorted) | Class Label (Rainfall) |
|---|---|---|
| 5 | 18.25 | 1 |
| 12 | 18.97 | 1 |
| 6 | 21.02 | 1 |
| 7 | 21.69 | 1 |
| 9 | 23.00 | 0 |
| 2 | 24.10 | 0 |
| 8 | 24.16 | 1 |
| 10 | 24.67 | 1 |

## Prediction using KNN (K = 5)

- The **5 closest neighbors** (smallest Euclidean distances) are:
  a. Observation **5** → **Rain = 1**
  b. Observation **12** → **Rain = 1**
  c. Observation **6** → **Rain = 1**
  d. Observation **7** → **Rain = 1**
  e. Observation **9** → **Rain = 0**
- **Class Label Decision:**
  - **Rain (1) appears 4 times**, **No Rain (0) appears 1 time**.
  - Since the majority class is **1**, the KNN algorithm **predicts that it will rain**.


Visual Representation of Distance Measures
Minkowski distance
Manhattan distance
Euclidean distance
Numerical Data
Edit distance
Distance Measures
Chebyshev’s distance
Cosine distance
String Data
Hamming distance
Longest Common Sequence

# Mathematical Formulas for Distance Measures

Euclidean distance: $d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$

Manhattan distance: $d(p,q) = \sum_{i=1}^{n}|q_i - p_i|$

Minkowski distance: $d(p,q) = \left(\sum_{i=1}^{n}|q_i - p_i|^p\right)^{\frac{1}{p}}$

Chebyshev's distance: $d(p,q) = \max_i|q_i - p_i|$

Cosine distance: $d(p,q) = 1 - \frac{\sum p_i q_i}{\sqrt{\sum p_i^2} \cdot \sqrt{\sum q_i^2}}$

Hamming distance: $d(p,q) = \sum[p_i \neq q_i]$

# Probability Concepts

**Probability**

- Probability is how likely an event is to occur.

- Probability Formula:

$$P = \frac{Favorable\ Outcomes}{Total\ Possible\ Outcomes}$$

- The probability of an event always lies between 0 and 1.

- 0 indicates the impossibility of the event, and 1 indicates certainty.

# Conditional Probability Concept

**Conditional Probability**

- The probability of event A occurring given that B has already occurred.

- Denoted by P(A | B).

- Conditional Probability Formula:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

# Example: Conditional Probability

**Example: Conditional Probability in a Company**

- A company has **100 employees**.

- **40 employees** are in the **Marketing** department.

- **25 of these Marketing employees** have **MBA degrees**.

- **Total employees with an MBA** = 50.

- **Find**: Probability that an employee has an MBA **given** they are in Marketing.

**Solution Using Conditional Probability Formula:**

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A|B) = \frac{25/100}{40/100} = \frac{25}{40} = 0.625$$

**- Final Answer:** 62.5% probability.

# Problems

- **Card Drawing from a Deck**
  A standard deck of **52 cards** contains **4 Aces**. Suppose you randomly draw a card, and it turns out to be a **face card (King, Queen, or Jack)**.Given that the drawn card is a face card, what is the probability that it is a **King**?
- **Dice Rolling and Sums**
  Two fair **six-sided dice** are rolled. The sum of the numbers showing on both dice is found to be **8**.Given this information, what is the probability that at least one of the dice shows a **6**?

# Multiplication Theorem

**Multiplication Theorem**

- From conditional probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- Multiplying both sides by P(B):

$$P(A \cap B) = P(A|B) \cdot P(B)$$

- Similarly, for P(B | A):

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

- Multiplying both sides by P(A):

$$P(A \cap B) = P(B|A) \cdot P(A)$$

**Thus, we conclude:**

$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

# Bayes Theorem - Explanation

## Bayes Theorem - Formula

- Used in Naïve Bayes classification.

- Formula:

$$P(t|x) = \frac{P(t) \cdot P(x|t)}{P(x)}$$

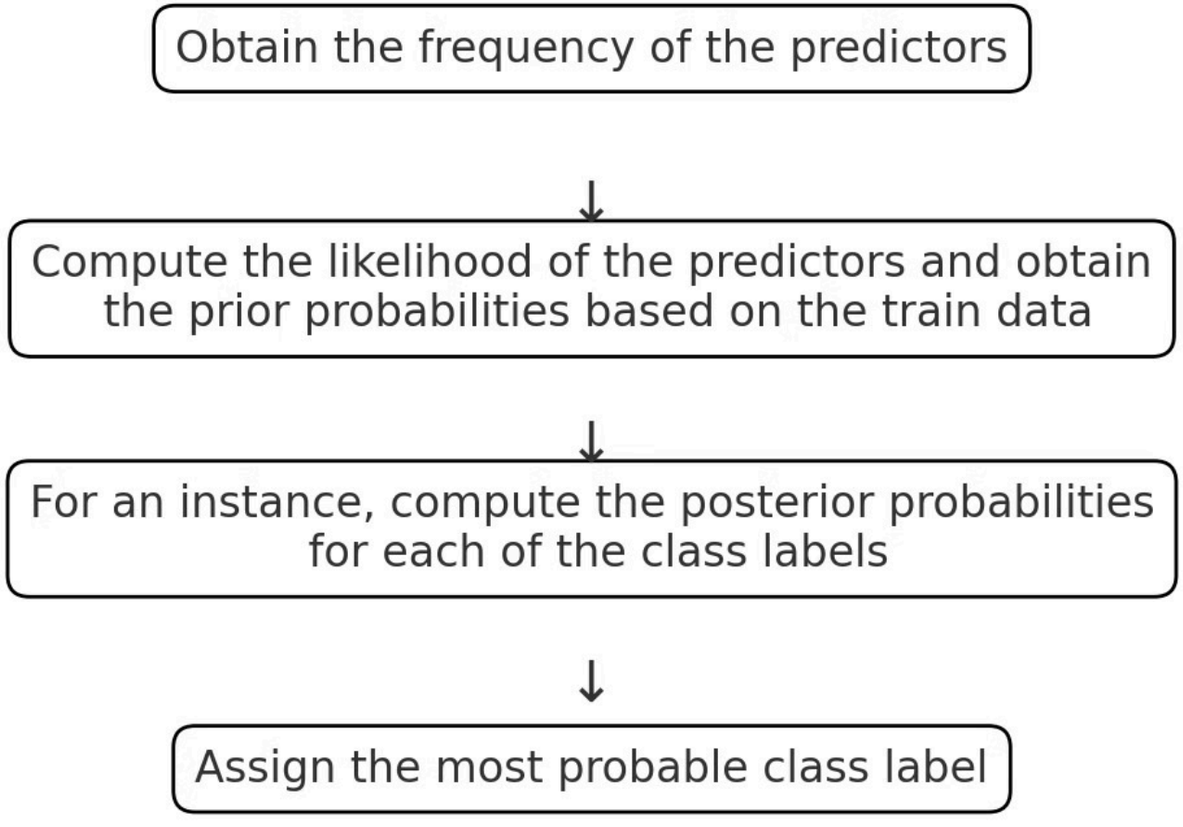
- P(t | x): Posterior Probability - Probability of class t given predictor x.

- P(t): Prior Probability - Probability of the class label before observing x.

- P(x | t): Likelihood - Probability of x given class t.

- P(x): Evidence - Probability of the predictor variable x.

# Naïve Bayes: Procedure

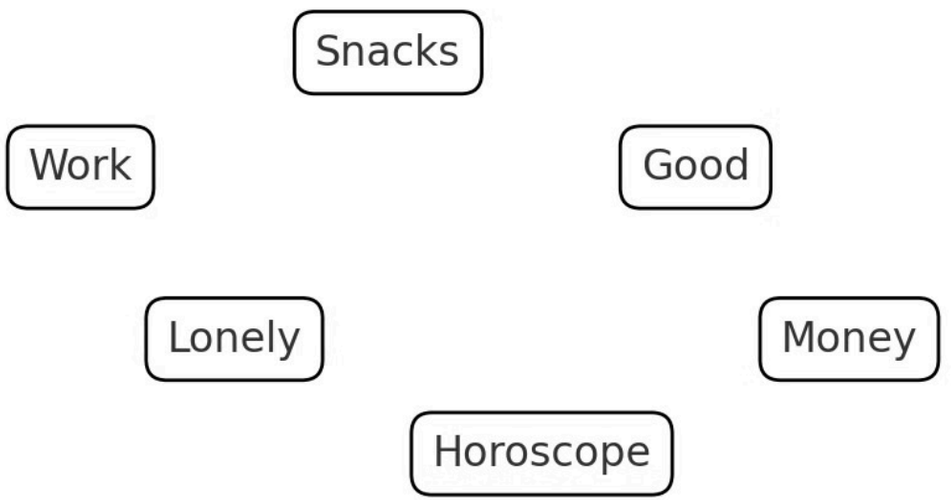


# Spam or Ham Email Classification

Business Problem: Label the Email as Spam or Ham

- We shall consider the problem of labeling received emails as spam or ham.

- Choose a few words you find in emails:



# Spam-ham example

1 Consider the frequency of these words used in spam and ham emails as shown below

| Word | Spam | Ham |
|---|---|---|
| Good | 2 | 10 |
| Lonely | 2 | 1 |
| Horoscope | 20 | 5 |
| Work | 5 | 12 |
| Snacks | 0 | 5 |
| Money | 21 | 7 |

| Words | Spam | Ham |
|---|---|---|
| Good | 2 | 10 |
| Lonely | 2 | 1 |
| Horoscope | 20 | 5 |
| Work | 5 | 12 |
| Snacks | 0 | 5 |
| Money | 21 | 7 |
| Total | 50 | 40 |

obtain likelihoods

| Words | Spam | Ham |
|---|---|---|
| Good | 2/50 = 0.04 | 10/40=0.25 |
| Lonely | 2/50=0.04 | 1/40=0.025 |
| Horoscope | 20/50=0.4 | 5/40=0.125 |
| Work | 5/50=0.1 | 12/40=0.3 |
| Snacks | 0/50 = 0 | 5/40=0.125 |
| Money | 21/50=0.42 | 7/40=0.175 |

# Obtain the prior probability

From the data we have 15% of the emails are spam and the remaining are ham Thus the prior probabilities are P(Spam) = 0.15 and P(Ham) = 0.85

# Likelihood

The probability that the word Good appears in a spam email,

ie P(Good | Ham) is 0.25.

This is the Likelihood.

# Compute the posterior probabilities for each the class labels – Spam or Ham

Spam-ham example

Compute the posterior probabilities for each the class labels - Spam or Ham

For Ham, P(Ham| Good, Work) = P(Ham) . P(Good | Ham). P (Work | Ham) = (0.85) . (0.25) . (0.30) = 0.063

For Spam, P(Spam| Good, Work) = P(Spam) . P(Good | Spam). P(Work | Spam) = (0.15) . (0.04) . (0.1) = 0.0006

From posterior probabilities for Ham > Spam so Good ,Work is Ham mail.

# Laplace smoothing method

To solve the zero-probability problem, we use the Laplace smoothing method

Add $\alpha$ to every count so; the count is never zero

$\alpha > 0$. Generally, $\alpha = 1$

Consider the $\alpha$ for the divisor as well

| Words | Spam | Ham |
|---|---|---|
| Good | 2 | 10 |
| Lonely | 2 | 1 |
| Horoscope | 20 | 5 |
| Work | 5 | 12 |
| Snacks | 0 | 5 |
| Money | 21 | 7 |
| Total | 50 | 40 |

Add alpha = 1

| Words | Spam | Ham |
|---|---|---|
| Good | 3 | 11 |
| Lonely | 3 | 2 |
| Horoscope | 21 | 6 |
| Work | 6 | 13 |
| Snacks | 1 | 6 |
| Money | 22 | 8 |
| Total | 56 | 46 |

| Words | Spam | Ham |
|---|---|---|
| Good | 3 | 11 |
| Lonely | 3 | 2 |
| Horoscope | 21 | 6 |
| Work | 6 | 13 |
| Snacks | 1 | 6 |
| Money | 22 | 8 |
| Total | 56 | 46 |

obtain new likelihoods

| Words | Spam | Ham |
|---|---|---|
| Good | 3/56 = 0.05 | 11/46=0.24 |
| Lonely | 3/56=0.05 | 2/46=0.04 |
| Horoscope | 21/56=0.37 | 6/46=0.13 |
| Work | 6/56=0.11 | 13/46=0.28 |
| Snacks | 1/56 = 0.02 | 6/46=0.13 |
| Money | 22/56=0.40 | 8/46=0.18 |

# Applications of Naïve Bayes

- Spam Filtering
- Sentiment Analysis
- Recommendation System

# Naïve Bayes: advantages

- Easy to implement in the case of text analytics problems
- Used for multiple class prediction problems
- Performs better for categorical data than numeric data

# Naïve Bayes: disadvantages

- Fails to find relationships among features
- May not perform well when the data has a large number of predictors
- The assumption of independence among features may not always hold true