GLOW 2024

Fakultas Informatika
School of Computing
Telkom University

Universiti Tenaga Nasional
The Energy University

CCI
College of Computing and Informatics

SUPERVISED LEARNING

Classification

Nur Laila Ab Ghani
Department of informatics
College of Computing and Informatics
Universiti Tenaga Nasional
Laila@uniten.edu.my

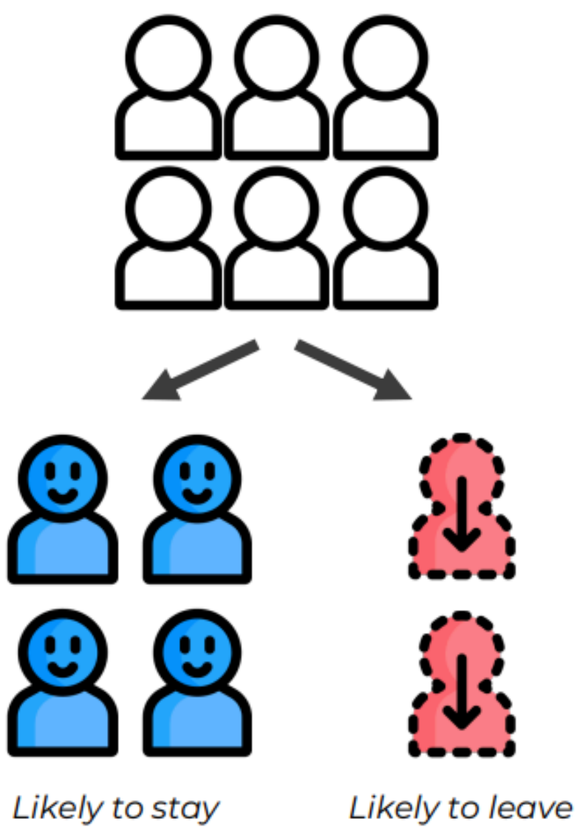
GLOW 2024

Outline

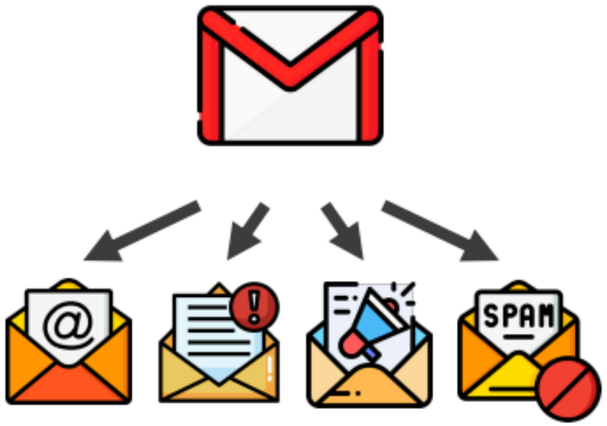
- Supervised Learning
- Classification Methods: K-Nearest Neighbor & Naive Bayes
- Evaluating Classification Results
- Implementing Classification Methods

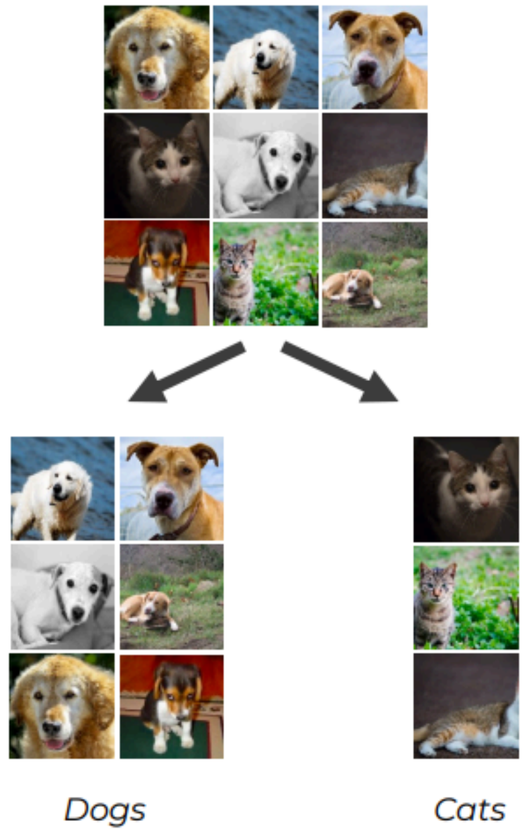
Adapted from Dr. Mahmud Dwi Sulistiyo Slides

Classification: a Machine Learning technique to identify the category of new observations based on training data.



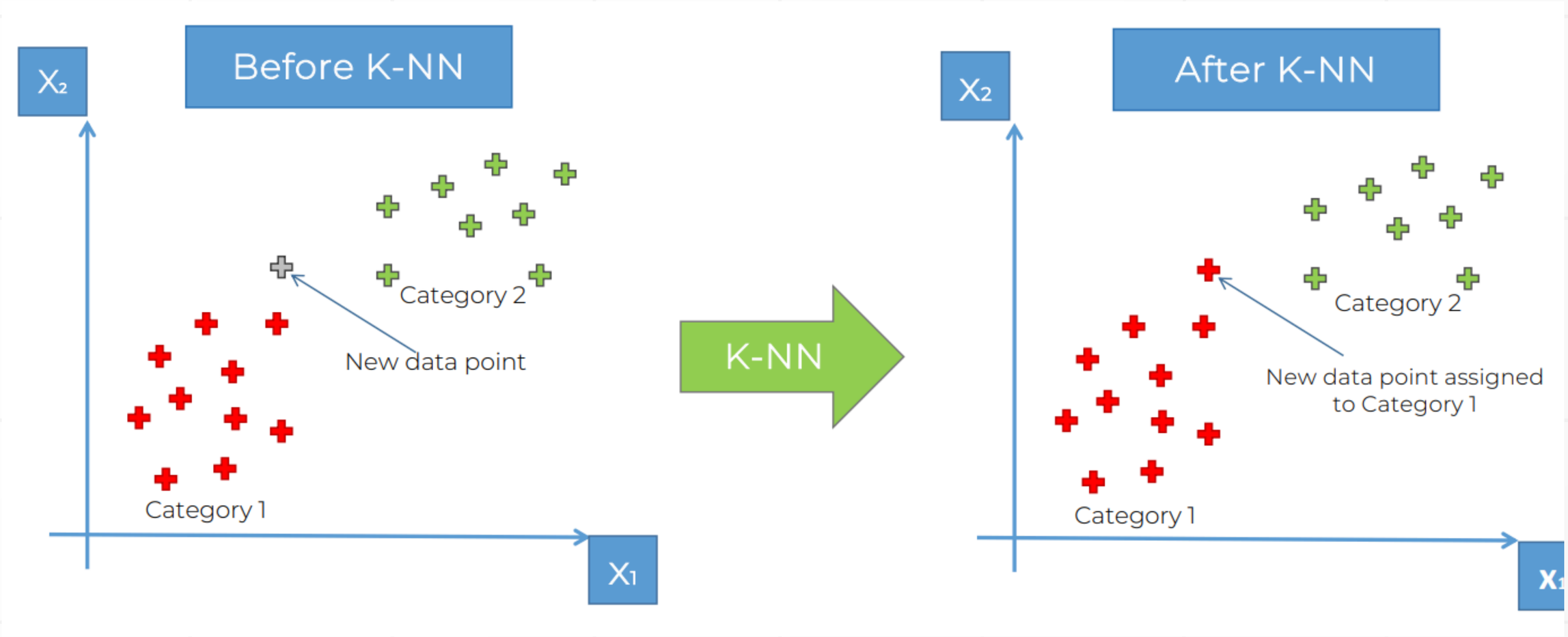
Likely to stay *Likely to leave*





Dogs *Cats*

K-Nearest Neighbor



How Does It Works?

STEP 1: Choose the number K of neighbors

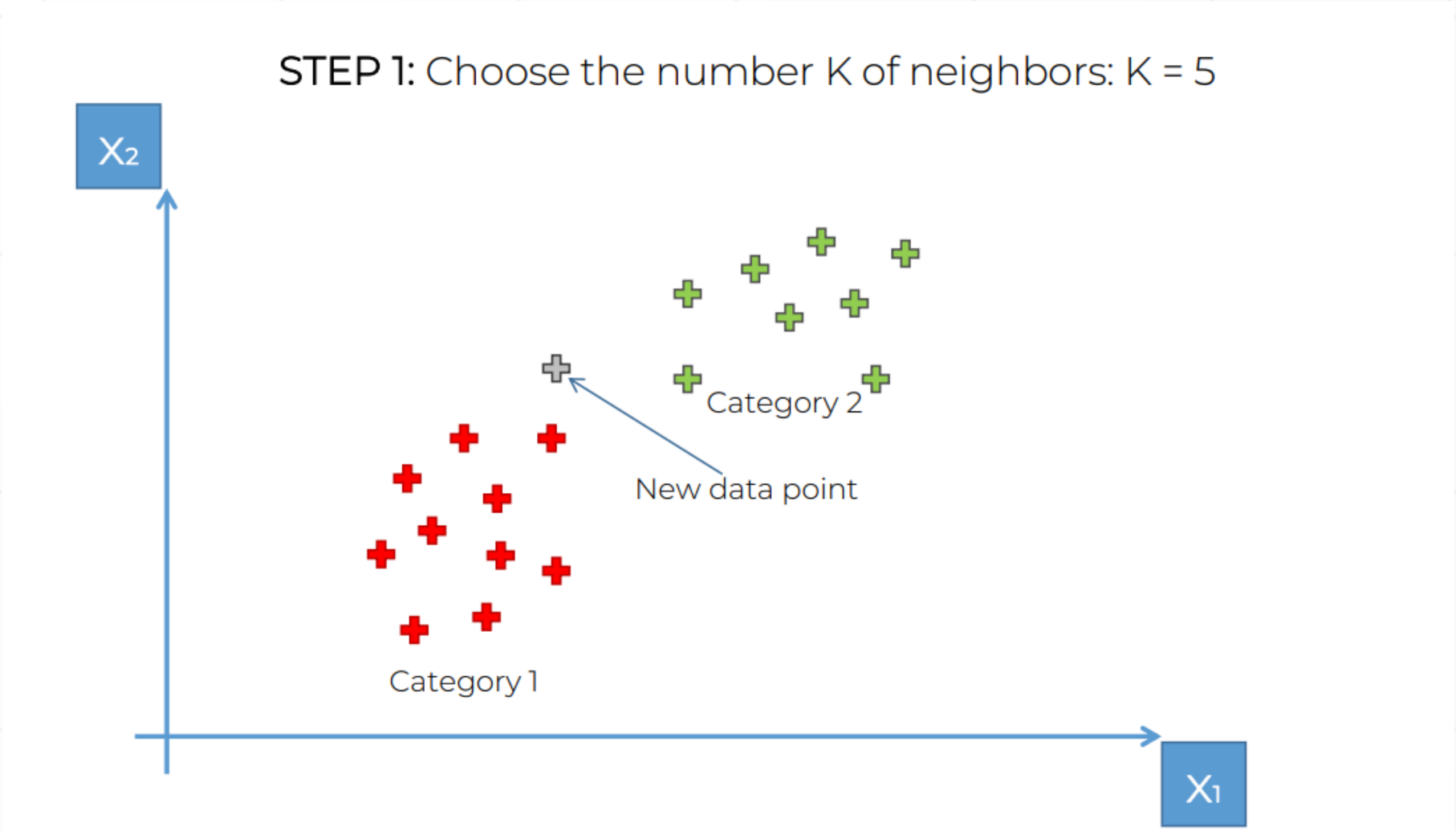
STEP 2: Take the K nearest neighbors of the new data point, according to the Euclidean distance

STEP 3: Among these K neighbors, count the number of data points in each category

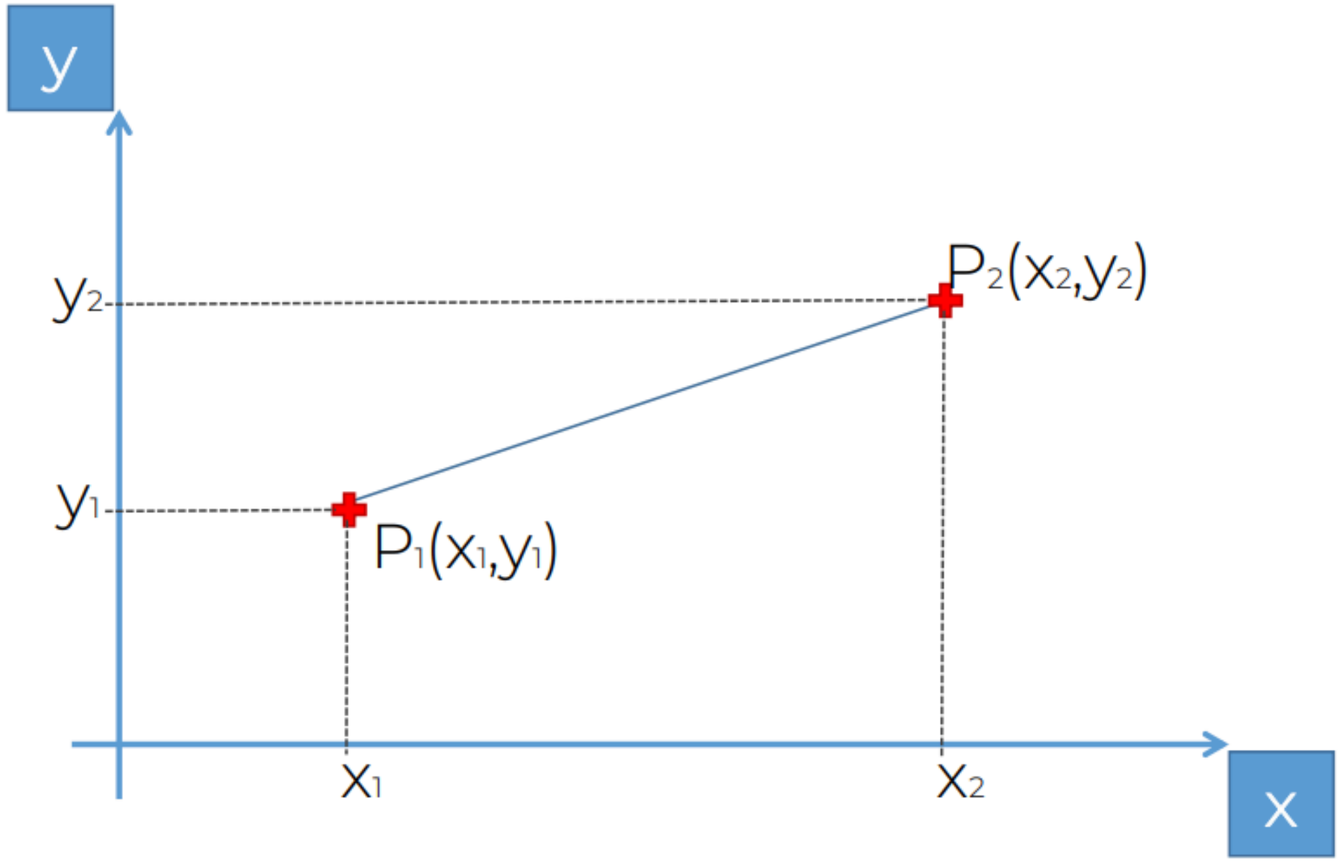
STEP 4: Assign the new data point to the category where you counted the most neighbors

Your Model is Ready

K-NN Algorithm

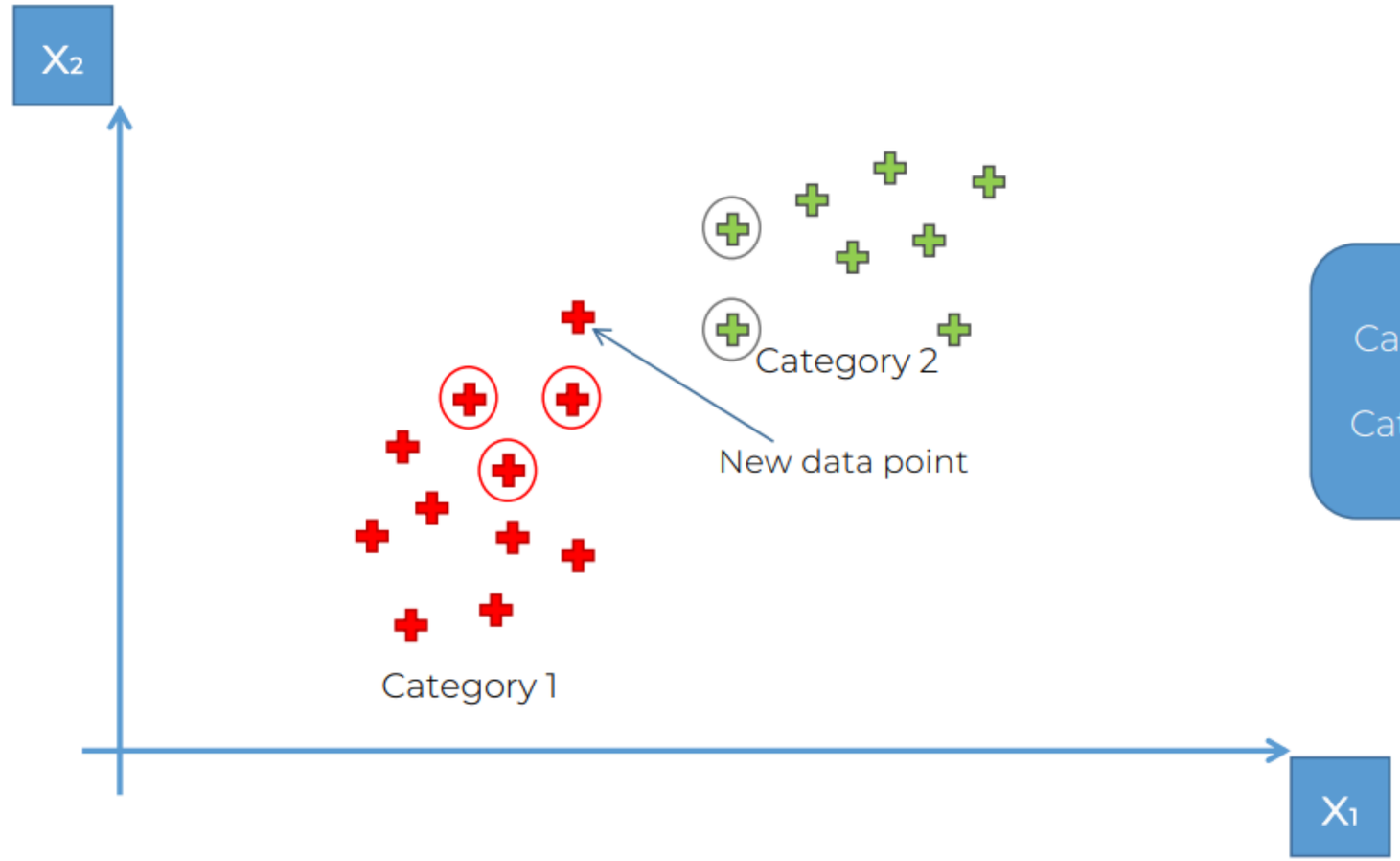


K-NN Algorithm



$$\text{Euclidean Distance between } P_1 \text{ and } P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

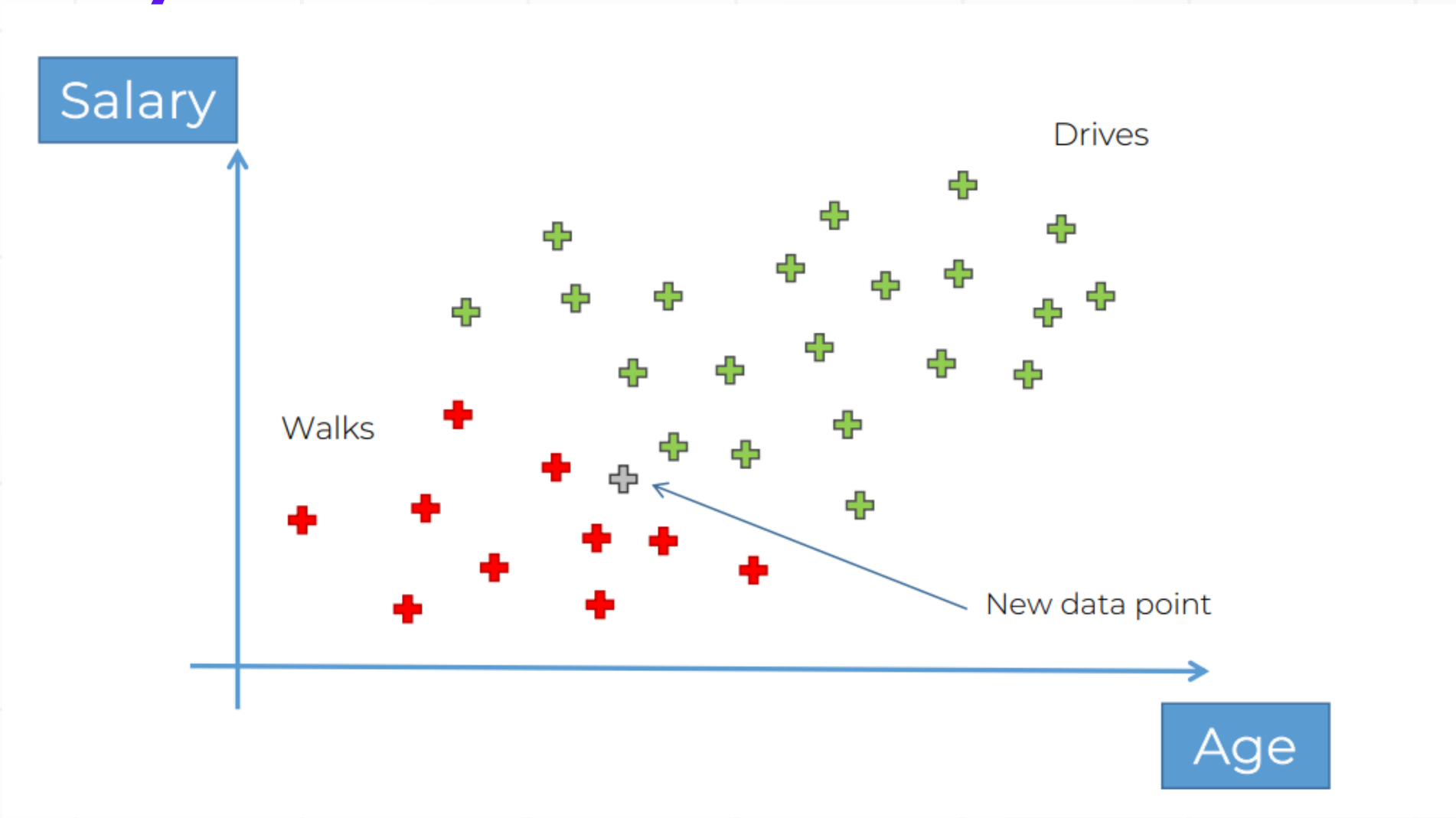
K-NN Algorithm



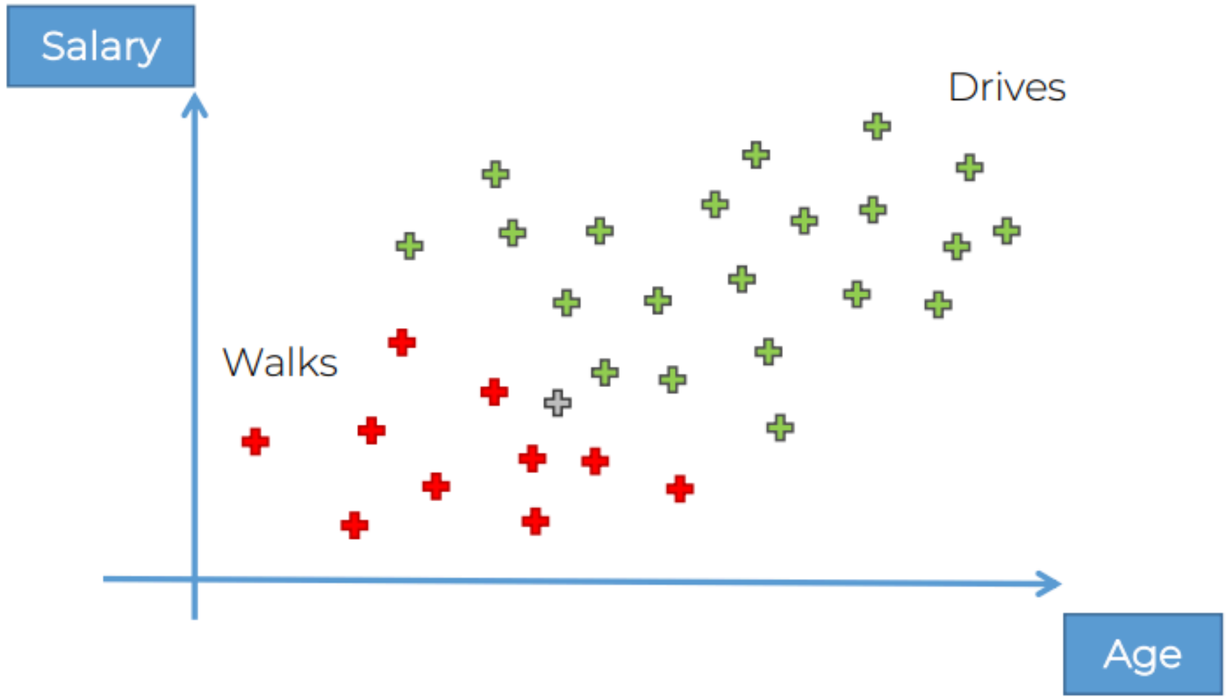
Category 1: 3 neighbors

Category 2: 2 neighbors

Naive Bayes



Naive Bayes - Step 1



#1. P(Walks)

$$P(Walks) = \frac{Number\ of\ Walkers}{Total\ Observations}$$

$$P(Walks) = \frac{10}{30}$$

Naive Bayes - Step 1

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Naive Bayes - Step 1

#4 Posterior Probability

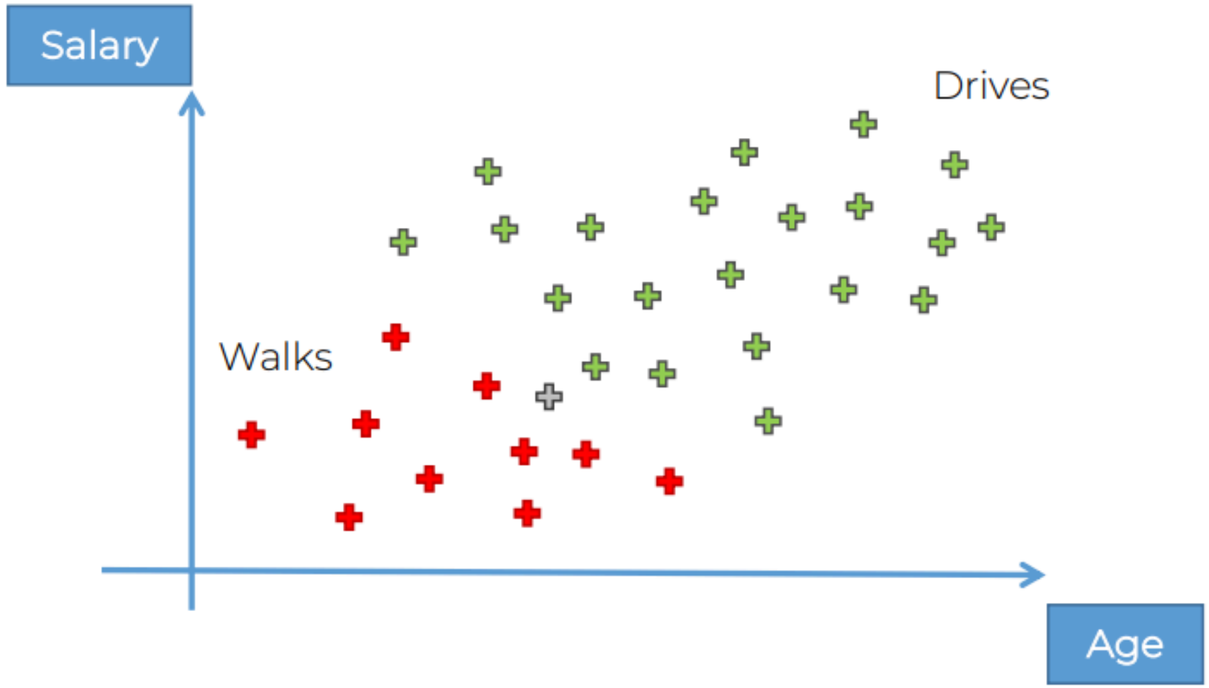
#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

#2 Marginal Likelihood

Naive Bayes - Step 1

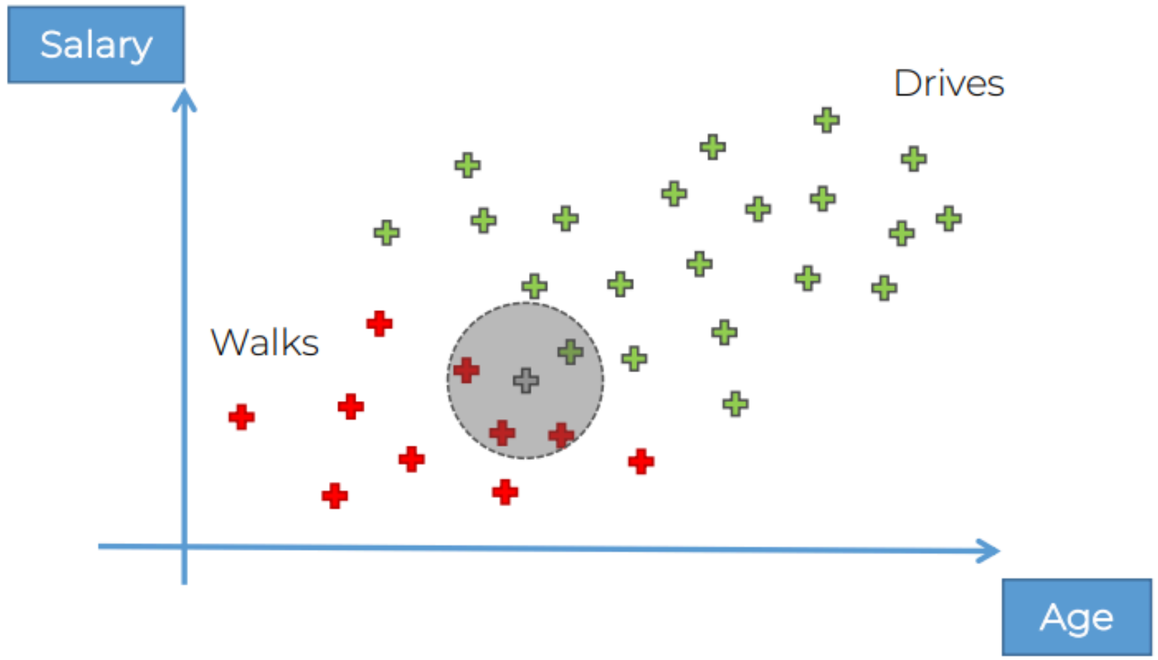


#1. P(Walks)

$$P(Walks) = \frac{Number\ of\ Walkers}{Total\ Observations}$$

$$P(Walks) = \frac{10}{30}$$

Naive Bayes - Step 1

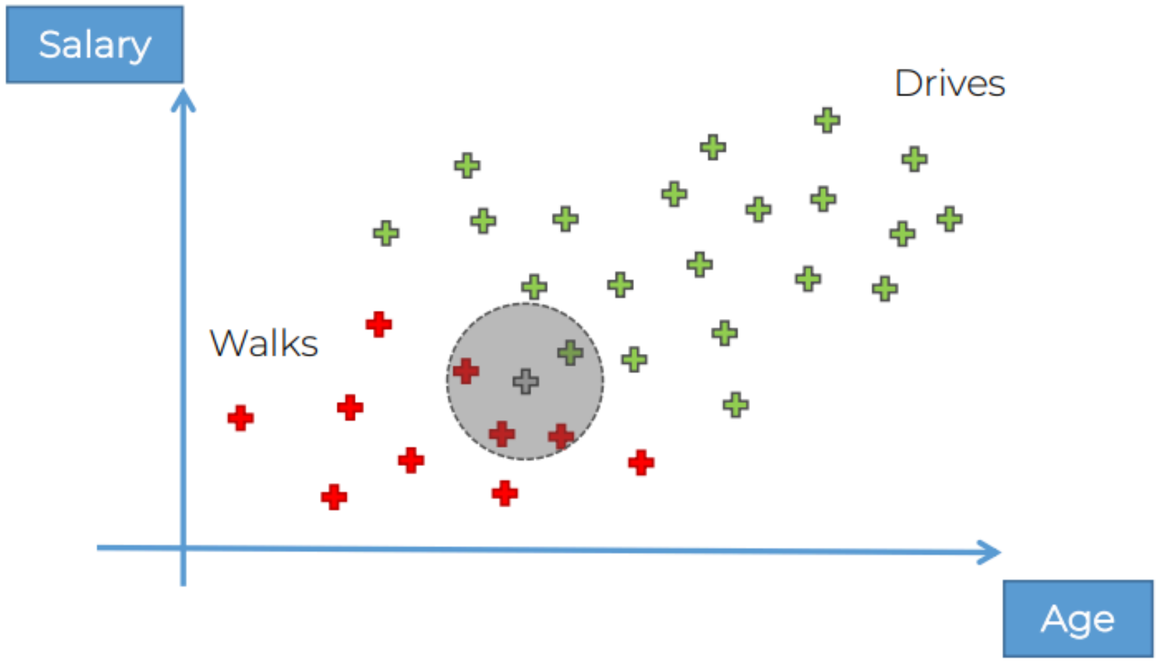


#2. P(X)

$$P(X) = \frac{Number\ of\ Similar\ Observations}{Total\ Observations}$$

$$P(X) = \frac{4}{30}$$

Naive Bayes - Step 1



#3. P(X|Walks)

$$P(X|Walks) = \frac{Number\ of\ Similar\ Observations\ Among\ those\ who\ Walk}{Total\ number\ of\ Walkers}$$

$$P(X|Walks) = \frac{3}{10}$$

Naive Bayes - Step 1

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{\frac{3}{10} * \frac{10}{30}}{\frac{4}{30}} = 0.75$$

#2 Marginal Likelihood

Naive Bayes - Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$$

#2 Marginal Likelihood

Naive Bayes - Step 2

#4 Posterior Probability

#3 Likelihood

#1 Prior Probability

$$P(Drives|X) = \frac{\frac{1}{20} * \frac{20}{30}}{\frac{4}{30}} = 0.25$$

#2 Marginal Likelihood

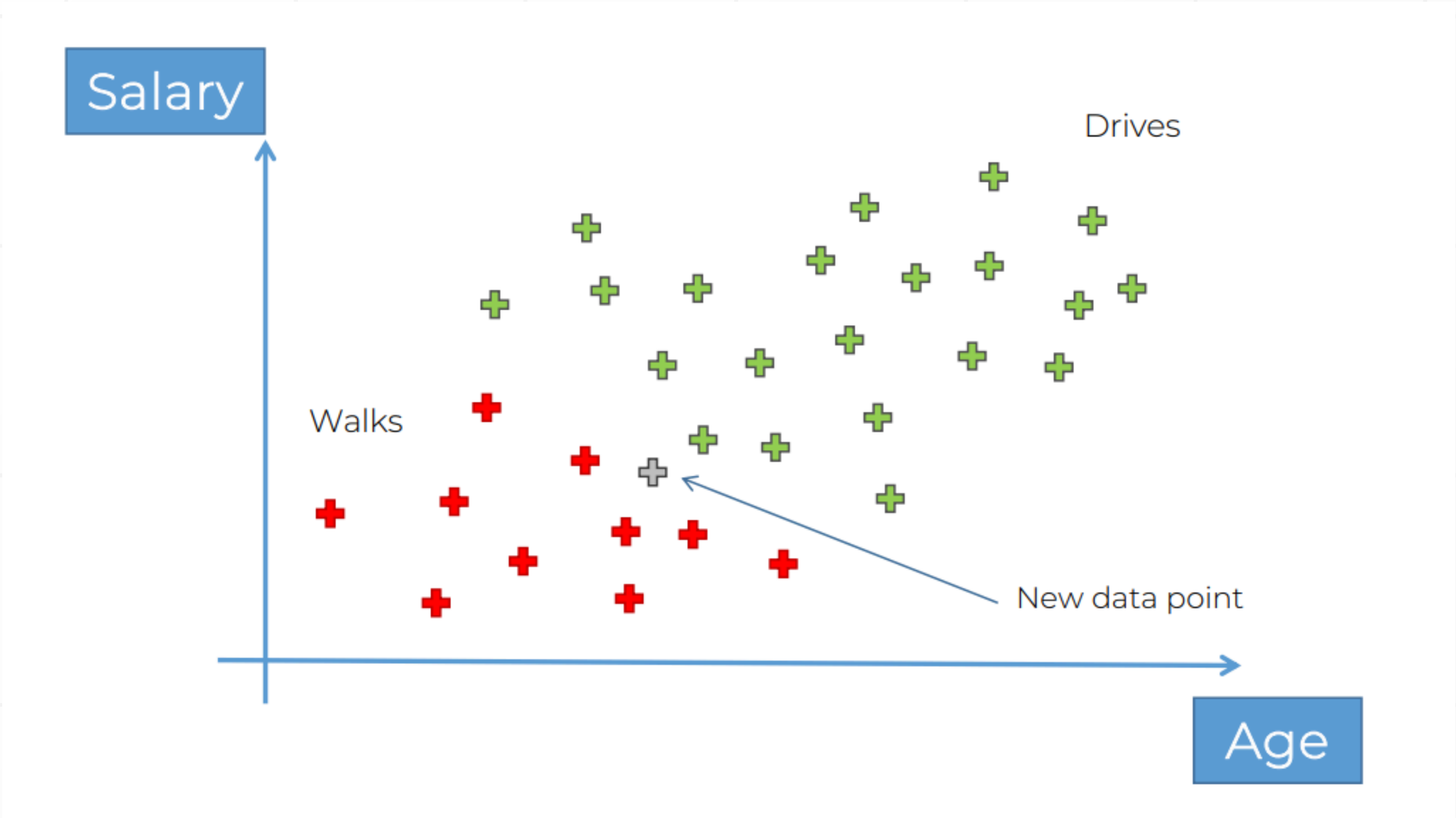
Naive Bayes - Step 3

$$P(Walks|X) \; v.s. \; P(Drives|X)$$

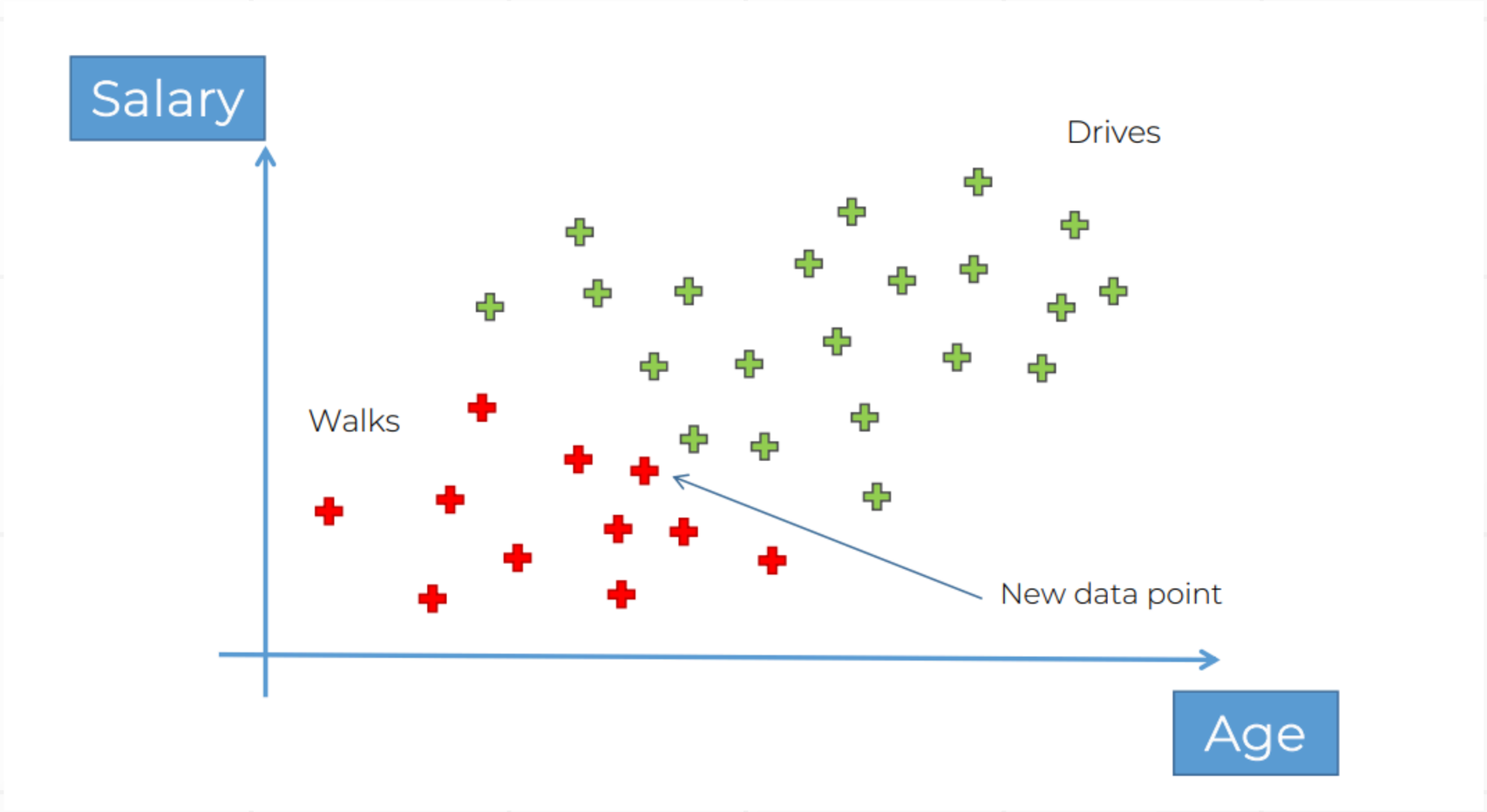
$$0.75 > 0.25$$

$$P(Walks|X) > P(Drives|X)$$

Naive Bayes - Step 3



Naive Bayes - Step 3



Performance Measure

- Once you have developed your classification model, you can apply it with a set of testing data
- Then, you can compare the results of the predicted values of the testing data with the actual values to assess the performance of the decision tree developed
- Confusion matrix can be used for this purpose

Confusion Matrix

- Displays the number of correct and incorrect predictions made by the model compared to the actual classifications in the test data

		Predicted status: Positive	Predicted status: Negative
Actual Status	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Confusion Matrix

- Let’s say you want to test your classifier with 1000 rows of testing data, which indicate as whether your existing customer will buy a new product from your company or not.
- Out of this 1000 rows, the model correctly predicted 565 of the existing customer will buy the new product, and another 375 will not buy.
- In addition, the model predicted that 50 customer will buy the product, but the actual data is not. And the remaining 11 customers were predicted as will not be buying their product, but they actually did.

Confusion Matrix

- Now, transform this information into the confusion matrix.
- Correctly predicted will buy – 565 (true positive)
- Correctly predicted will not buy – 375 (true negative)
- Incorrectly predicted will buy – 50 (false positive)
- Incorrectly predicted will not buy – 10 (false negative)

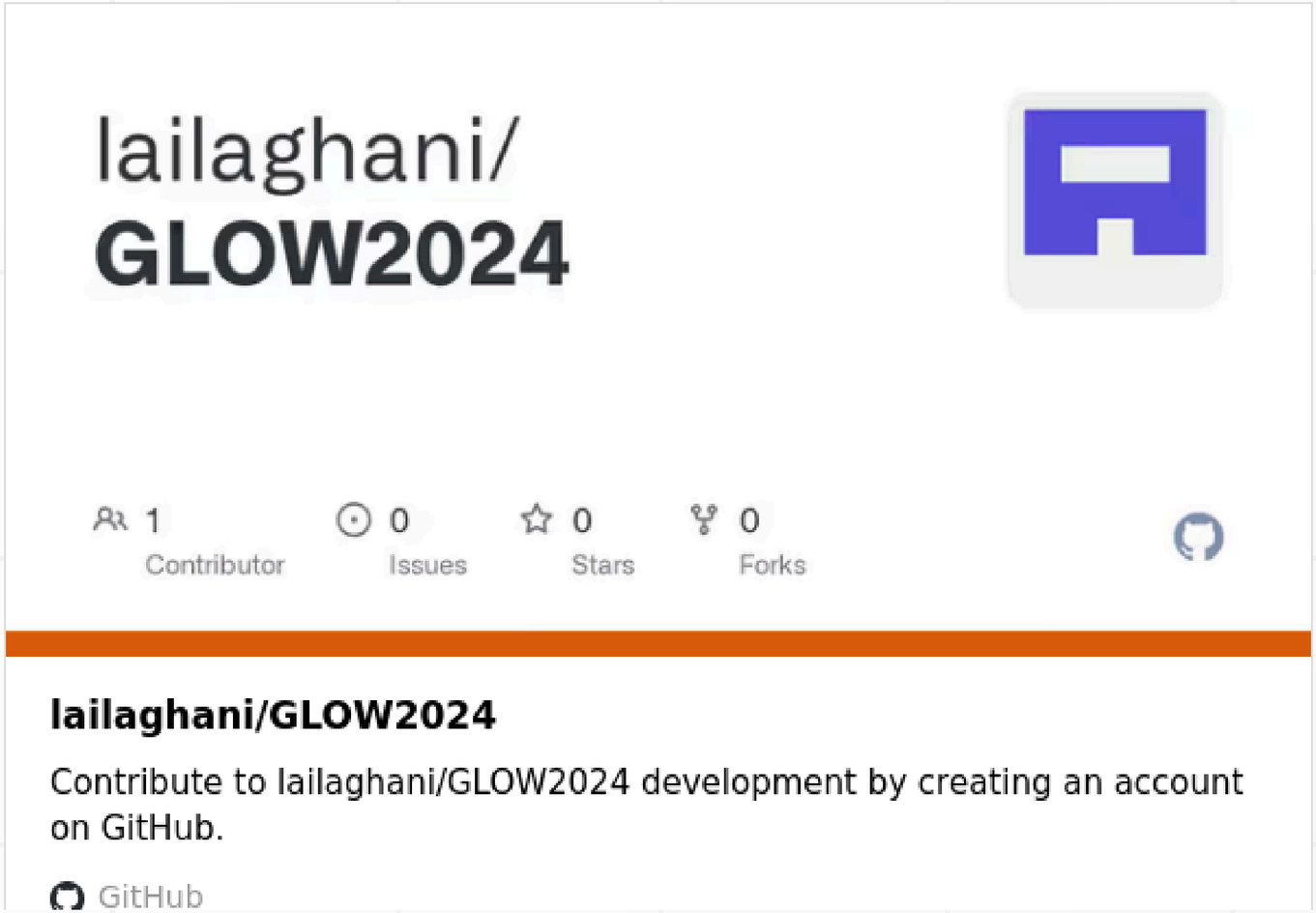
		Predicted status: Positive	Predicted status: Negative
Actual status	Positive	**565**	**10**
	Negative	**50**	**375**

Confusion Matrix

- Now, we know that the classifier model has:
  - Correctly predicted will buy – 565 (true positive)
  - Correctly predicted will not buy – 375 (true negative)
  - Incorrectly predicted will buy – 50 (false positive)
  - Incorrectly predicted will not buy – 10 (false negative)
- So, based on this:
  - The model made 940 correct predictions (565 + 375).
  - The model made 60 incorrect predictions (50 + 10).
  - There are total 1000 scored cases (565 + 10 + 50 + 375).
  - The error rate is 60/1000 = 0.06

The overall accuracy rate is 940/1000 = 0.94

GLOW 2024


lailaghani/
GLOW2024
1
Contributor
0
Issues
0
Stars
0
Forks
lailaghani/GLOW2024
Contribute to lailaghani/GLOW2024 development by creating an account on GitHub.
GitHub