Supervised Learning: Classification

Classification in machine learning is a type of supervised learning where the goal is to categorize or classify data points into predefined classes or labels.

Given input data (features), a classification algorithm learns **a mapping function** that assigns a label (or class) to each data point based on the training data. It is used when the output variable is categorical, meaning it falls into one of several distinct categories.

Examples of Classification Algorithms:

Logistic Regression

K-Nearest Neighbors (KNN)

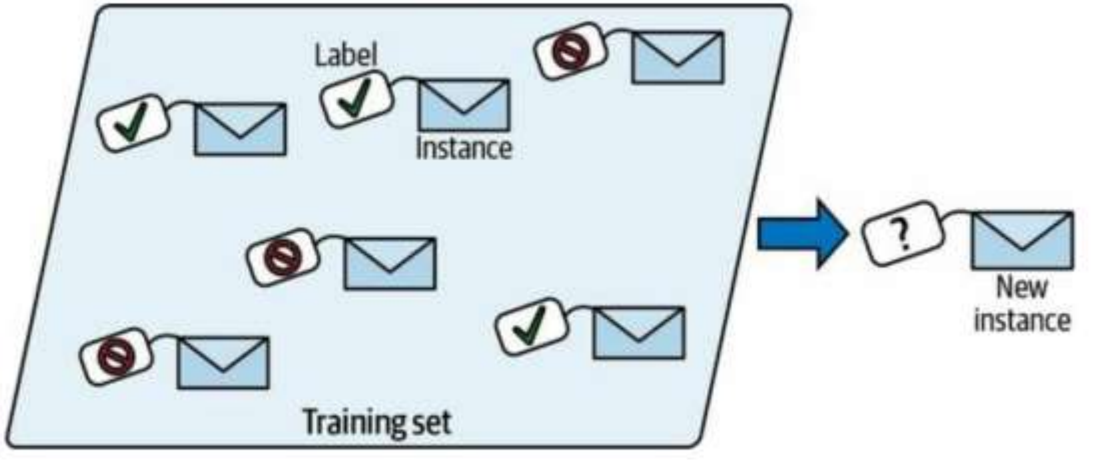
Decision Trees

Naive Bayes

Support Vector Machine (SVM

Random Forest

Neural Networks



Supervised Learning: Classification

K-Nearest Neighbors (KNN)

It is a simple, **non-parametric**, **instance-based** machine learning algorithm. It can be also used as regression tasks.

Non-parametric: k-NN doesn’t assume any specific form for the underlying data distribution, which makes it flexible for various data types. It can model complex data patterns without needing to fit data to a predefined parameterized function.

Instance-based: Instead of learning a general rule from the training data, k-NN memorizes the entire training set. For prediction, it looks up the k closest instances in the training data (based on a distance metric like Euclidean or Manhattan) and uses those instances to classify new points or make regression predictions.

Supervised Learning: Classification

K-Nearest Neighbors (KNN)

KNN works by finding the "k" closest data points (neighbors) to a new, unknown data point and making predictions based on their classes.

KNN **does not build a model** during the training phase. Instead, it stores the training data and makes predictions by referencing it directly during testing. This is why it is also called a **lazy learner**.

Advantages of KNN:

KNN is easy to understand and implement. Since KNN doesn’t build a model, the training phase is **fast**. It can be used for both classification and regression problems. KNN doesn’t make assumptions about the underlying data distribution (Non-parametric).

Supervised Learning: Classification

How KNN Works:

Select k Neighbors: The parameter "k" refers to the number of nearest neighbors to consider when making a prediction.

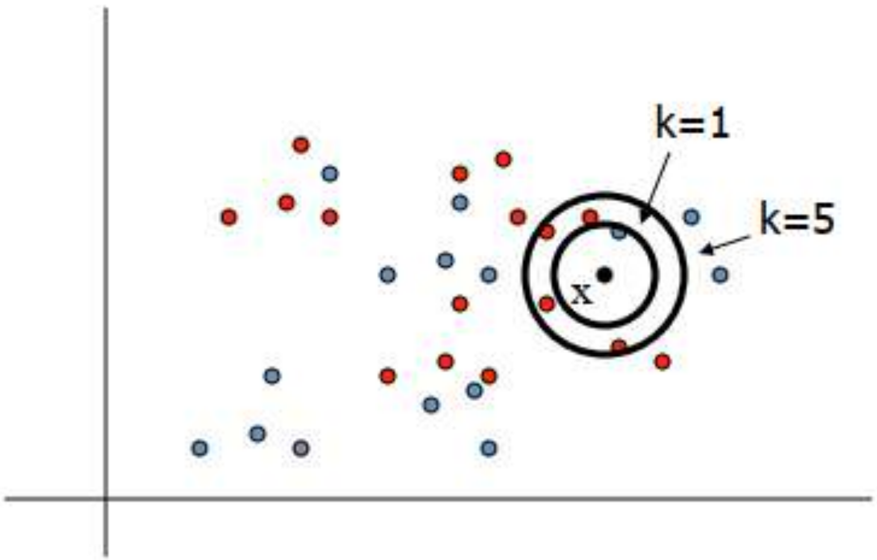
Calculate Distance: The algorithm calculates the distance between the new data point and all other points in the training dataset. Common distance metrics include: Euclidean Distance, Manhattan Distance.

Identify Nearest Neighbors: After calculating the distances, the algorithm selects the k nearest points (neighbors) from the training set.

Make a Prediction: Classification: The new point is assigned to the class that is most common among its k nearest neighbors (majority voting).

Supervised Learning: Classification

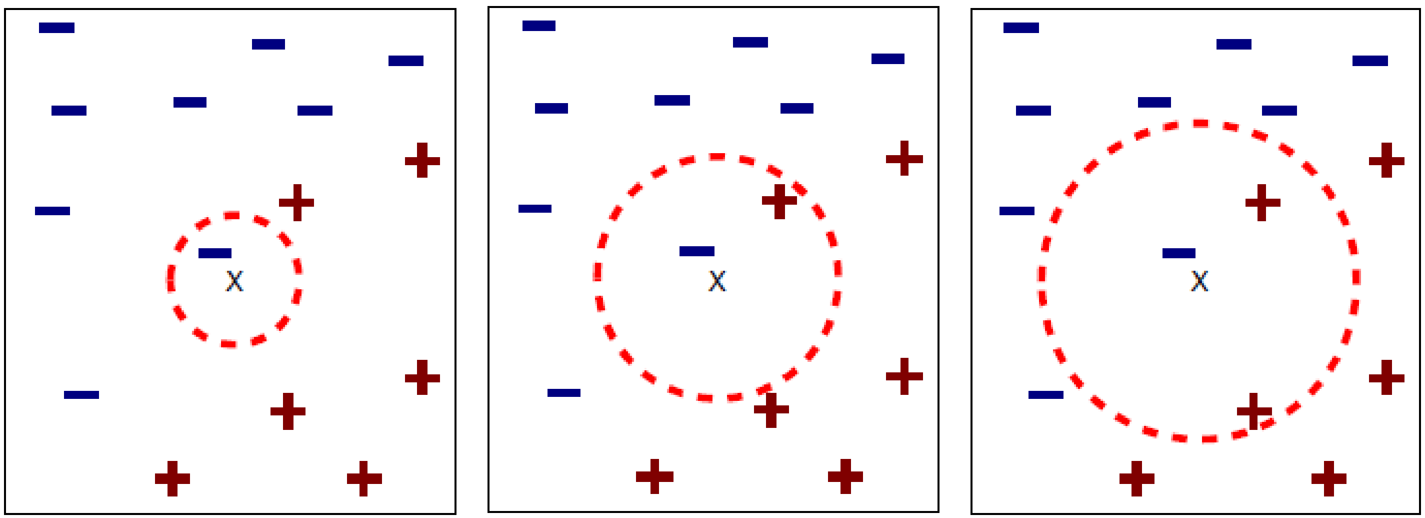
To classify a new input vector x, examine the k-closest traning data point to x and assign the object to the most frequently occurring class.



common values for k: 3, 5

Supervised Learning: Classification

How KNN Works:



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Supervised Learning: Classification

Example of KNN:

Height (cm)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?

Supervised Learning: Classification

Example of KNN:

$$\boldsymbol{d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

$$\boldsymbol{d_1 = \sqrt{(170 - 167)^2 + (57 - 51)^2}}$$
$$\boldsymbol{d_2 = \sqrt{(170 - 182)^2 + (57 - 62)^2}}$$

.

.

.

$$\boldsymbol{d_9 = \sqrt{(170 - 170)^2 + (57 - 55)^2}}$$

Height (cm)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?

Supervised Learning: Classification

Example of KNN:

Height (cm)	Weight (KG)	Class	d
167	51	Underweight	6.7
182	62	Normal	13
176	69	Normal	13.4
173	64	Normal	7.6
172	65	Normal	8.2
174	56	Underweight	4.1
169	58	Normal	1.4
173	57	Normal	3
170	55	Normal	2
170	57	?	

Supervised Learning: Classification

Example of KNN:

Height (cm)	Weight (KG)	Class	d	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

Supervised Learning: Classification

Example of KNN:

Height (cm)	Weight (KG)	Class	d	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

K=3

Supervised Learning: Classification

Example of KNN:

Height (cm)	Weight (KG)	Class	d	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

K=4

Supervised Learning: Classification

Example of KNN:

Height (cm)	Weight (KG)	Class	d	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

K=5

Supervised Learning: Classification

Disadvantages of KNN:

Computationally Expensive: As the size of the dataset increases, the algorithm becomes slow because it has to calculate the distance for each test point with every training point.

Sensitive to Noisy Data: Outliers or irrelevant features can significantly affect predictions.

Feature Scaling Required: Since KNN relies on distance calculations, features must be scaled properly to prevent features with larger ranges from dominating the distance calculation.

Memory-Intensive: The algorithm needs to store the entire training dataset, making it less memory-efficient for large datasets.

Supervised Learning: Classification

KNN Code:

https://colab.research.google.com/drive/1WvzjLHrb2Yf0ocPfUPUAIL7lYTIAvGwl?usp=sharing

Lets go for understanding the coding of KNN using Python