K nearest neighbours

BCSE0105: MACHINE LEARNING

Different names of KNN

- K-Nearest Neighbors
- Memory-Based Reasoning
- Example-Based Reasoning
- Instance-Based Learning
- Lazy Learning

Introduction

- Classification belongs to the category of supervised learning where the output is also provided with the input data.

- There are many applications in classification such as credit approval, medical diagnosis etc.

Type of Learners

- There are two types of learners in classification:

- Eager learners
- Lazy learners

Eager Learners

- Construct a classification model based on the given training data before receiving data for classification.
- It must be able to commit to a single hypothesis that covers the entire instance space.
- Due to the model construction, eager learners take a long time for training and less time to predict.
- Ex. Decision Tree, Naive Bayes, Artificial Neural Networks

Lazy Learners

- Simply store the training data and wait until a testing data appear.
- Classification is conducted based on the most related data in the stored training data.
- Compared to eager learners, lazy learners have less training time but more time in predicting.
- Ex. k-nearest neighbor, Case-based reasoning

k-Nearest Neighbors

- “kNN which stand for K Nearest Neighbors is a Supervised Machine Learning algorithm that classifies a new data point into the target class, depending on the features of its neighboring data points.”

- The k-nearest neighbors (kNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

“K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure.”

Features of kNN Algorithm

- kNN is a Supervised Learning algorithm that uses labeled input data set to predict the output of the data points.
- It is one of the simplest Machine learning algorithms and it can be easily implemented for a varied set of problems.
- It is mainly based on feature similarity.
- kNN checks how similar a data point is to its neighbor and classifies the data point into the class it is most similar to.

Features of kNN Algorithm

- Unlike most algorithms, kNN is a non-parametric model which means that it does not make any assumptions about the data set.
- This makes the algorithm more effective since it can handle realistic data.
- kNN is a lazy algorithm, this means that it memorizes the training data set instead of learning a discriminative function from the training data.
- kNN can be used for solving both classification and regression problems.

The kNN Algorithm

Assumption: similar things exist in close proximity.

Step 1 − For implementing any algorithm, we need dataset. So during the first step of kNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of k i.e. the nearest data points. K can be any integer(preferably odd)

Step 3 − For each point in the test data do the following −

3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean or Manhattan distance. The most commonly used method to calculate distance is Euclidean.

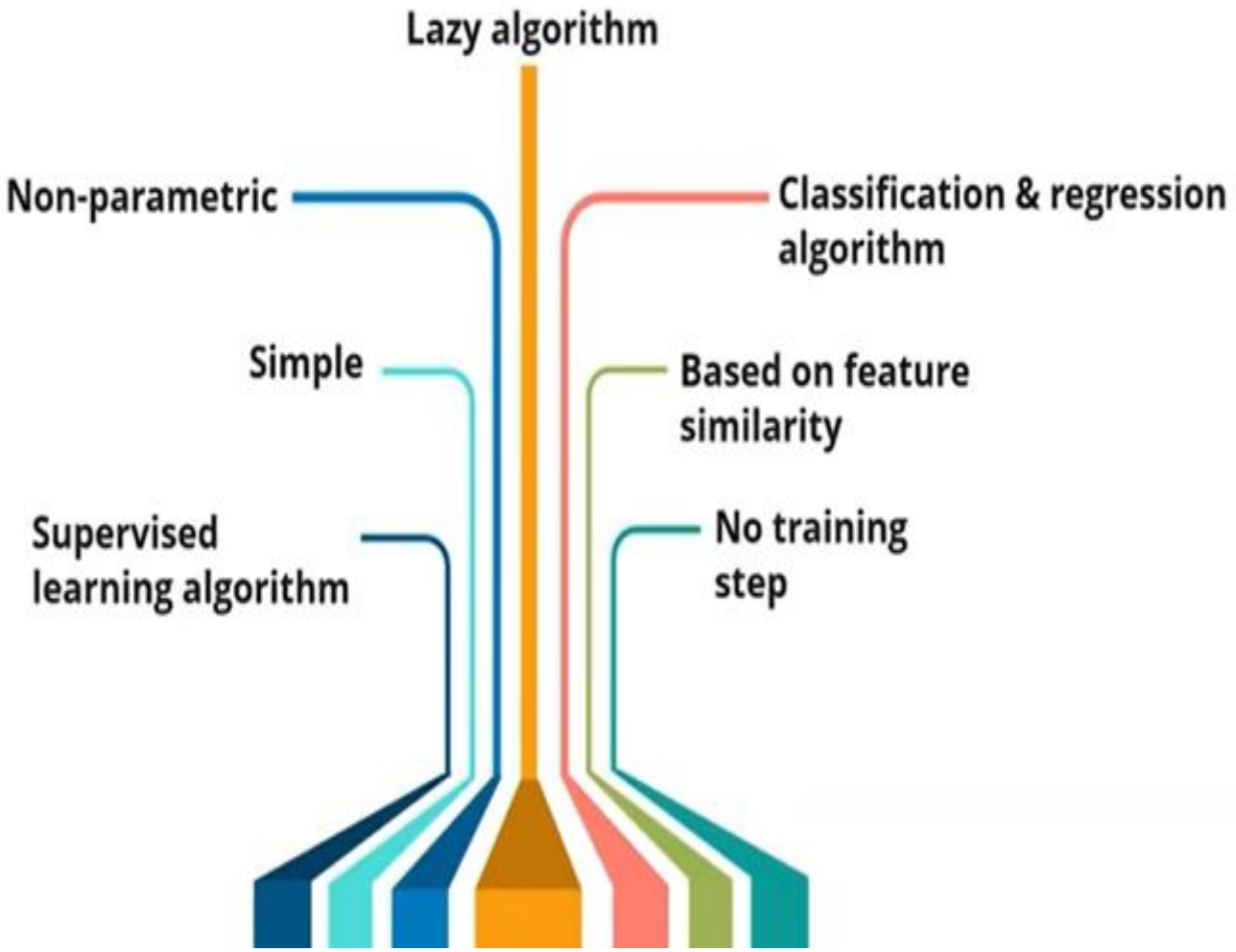
3.2 − Now, based on the distance value, sort them in ascending order.

3.3 − Next, choose the top k rows from the sorted array.

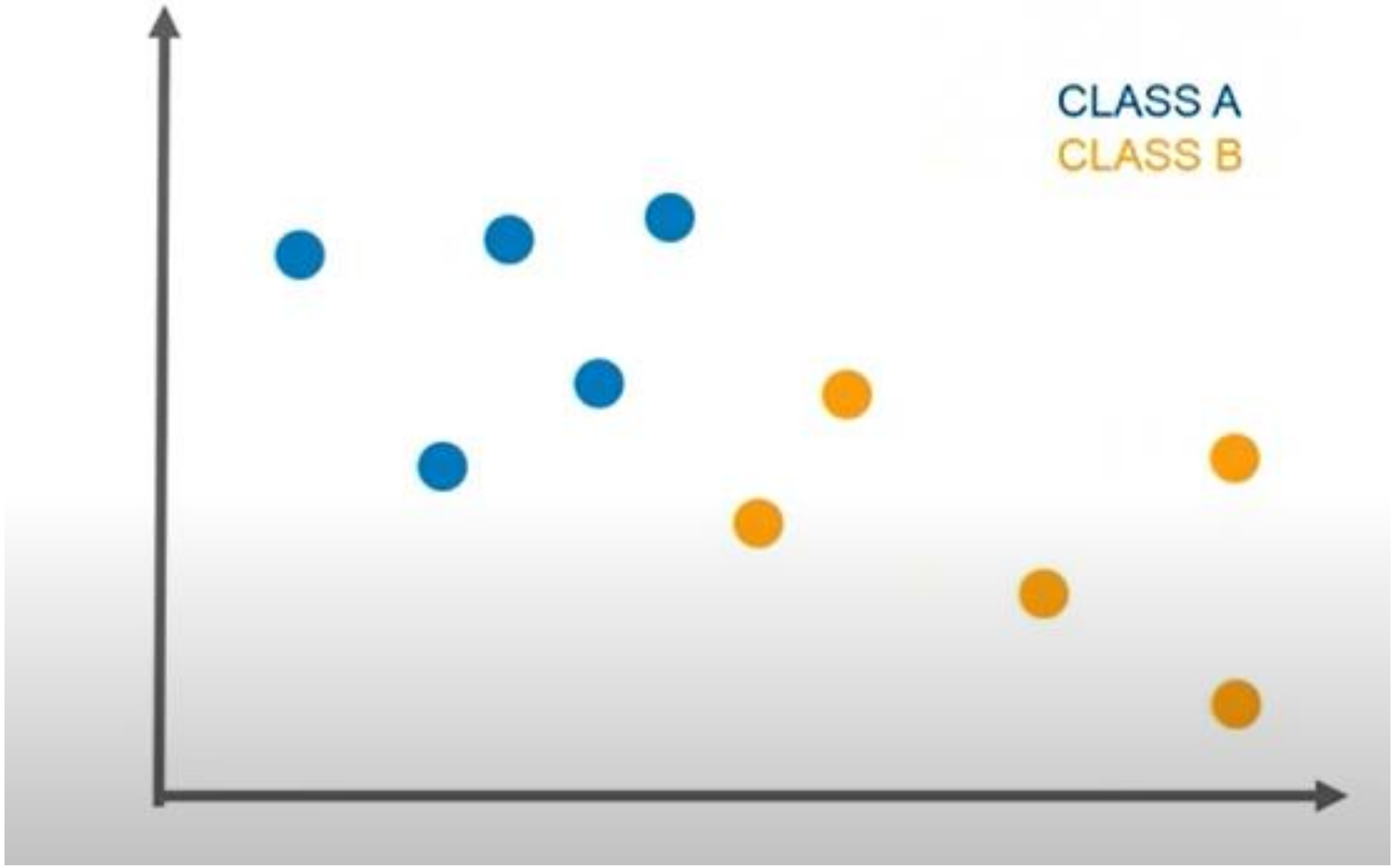
3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 − End

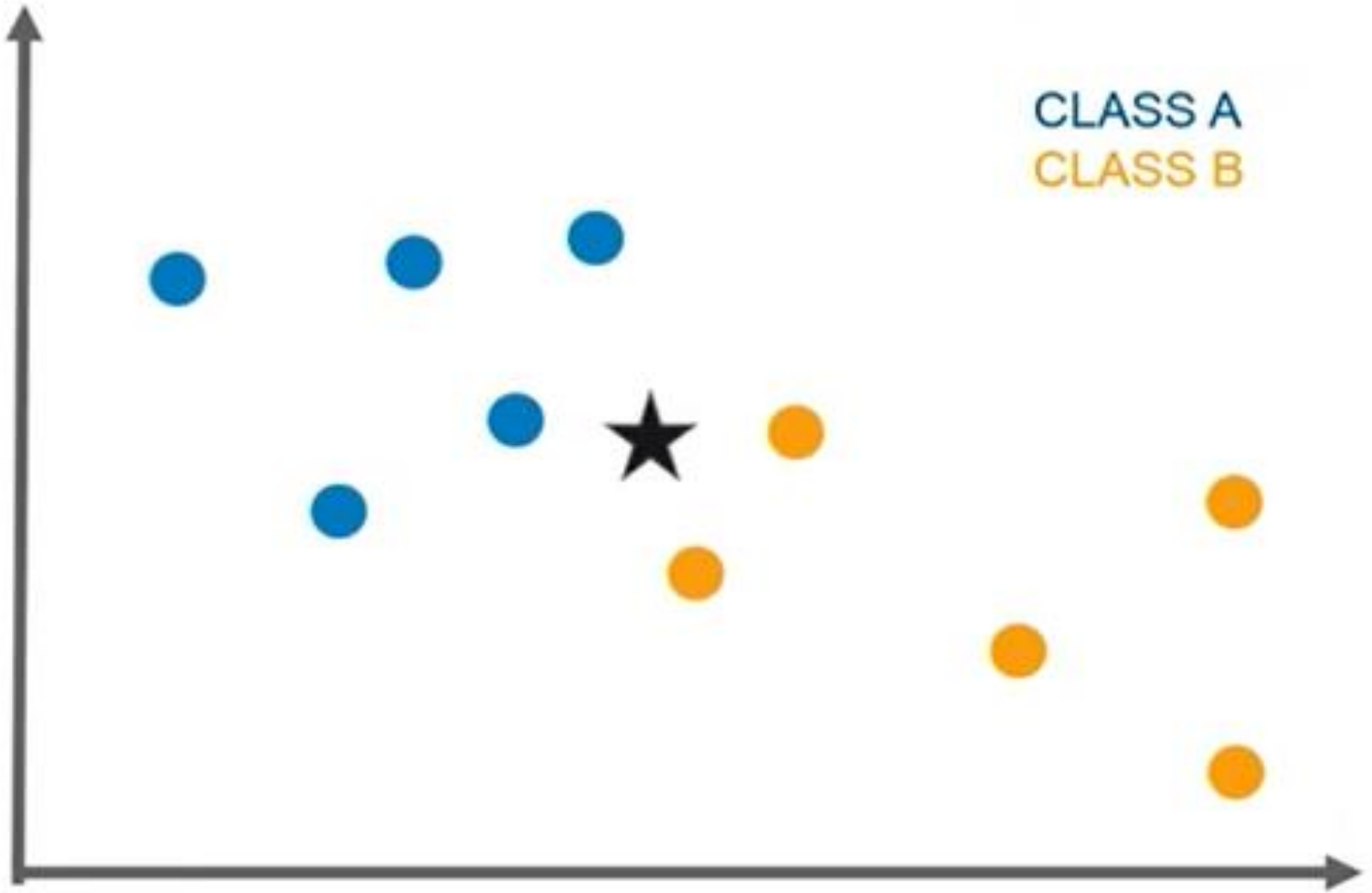
Features of KNN



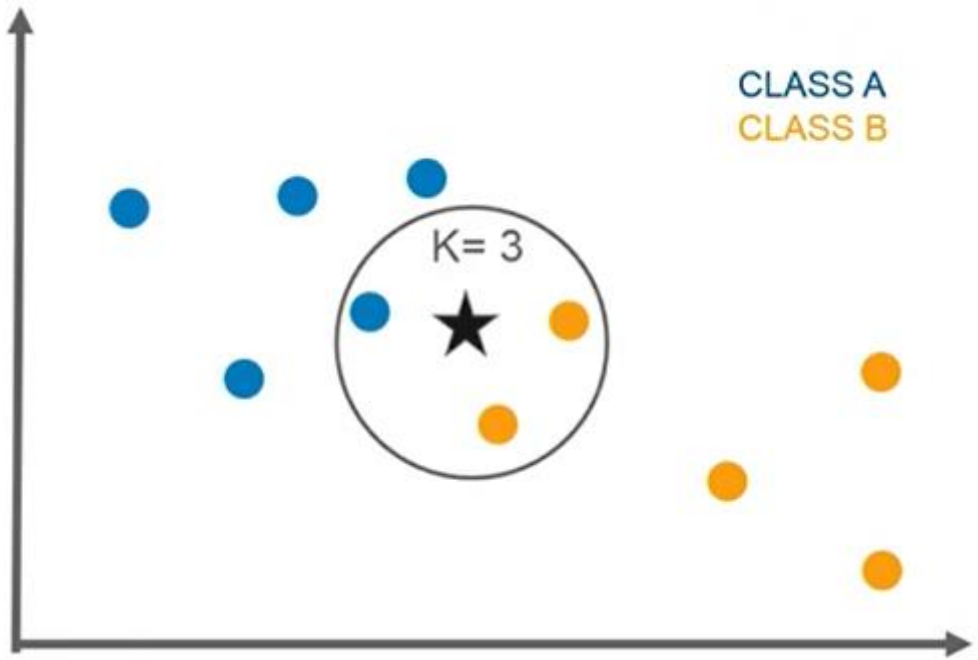
How KNN algorithm works?



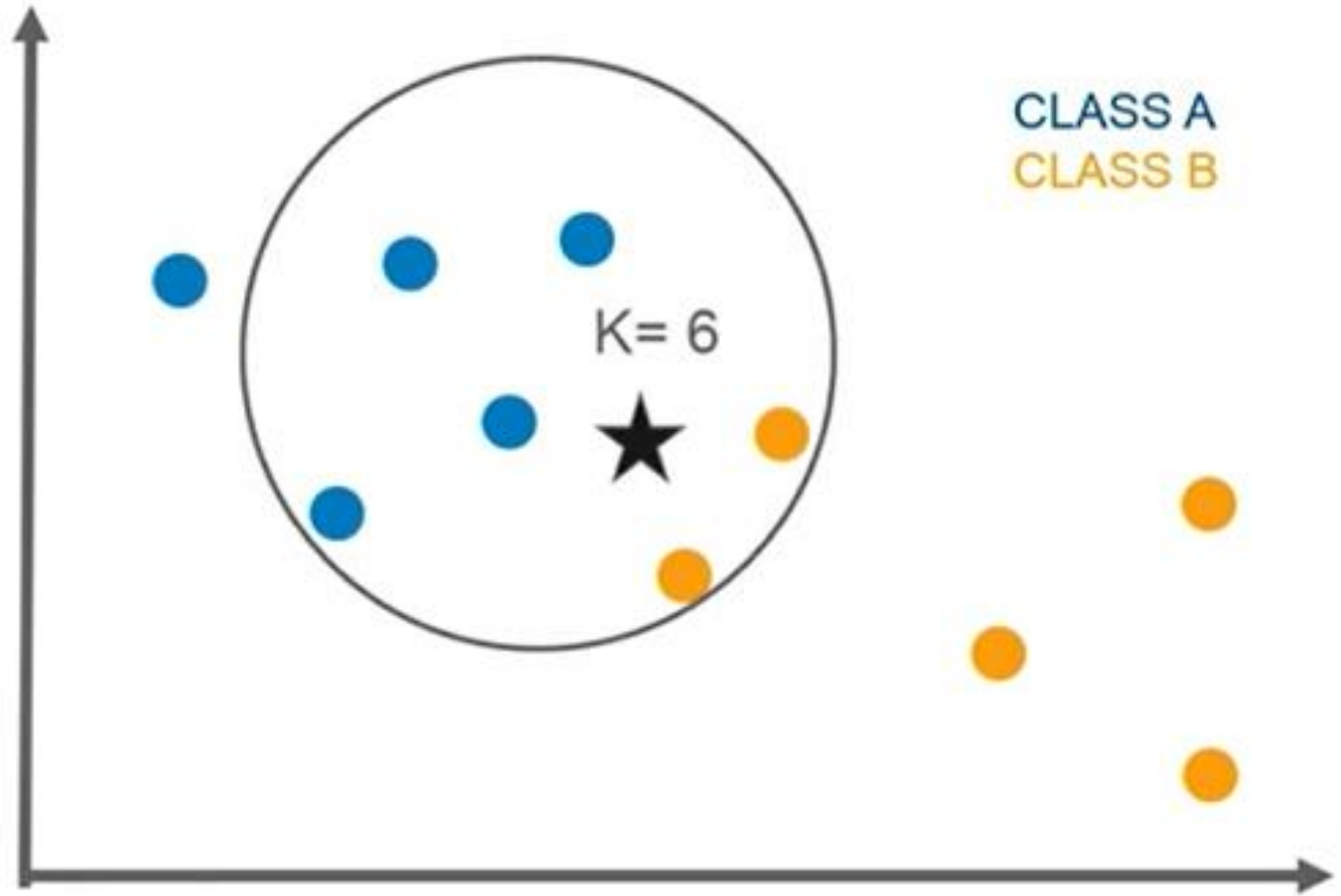
Test data



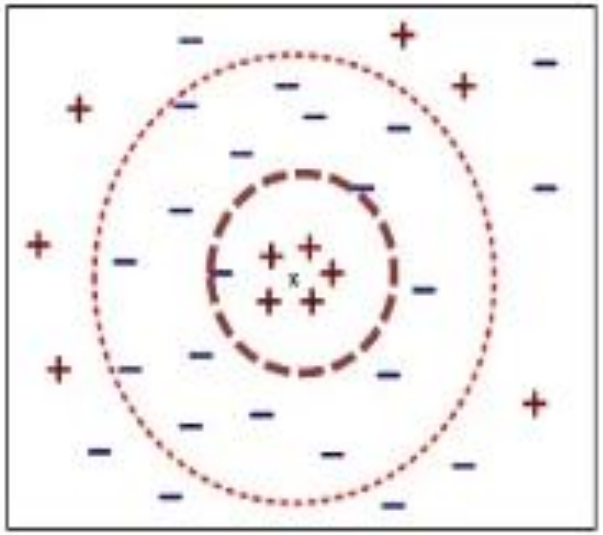
For K=3 (number of nearest neighbours that we want to select)



For K=6



We need to try with several values of k in order to determine which works best for your data

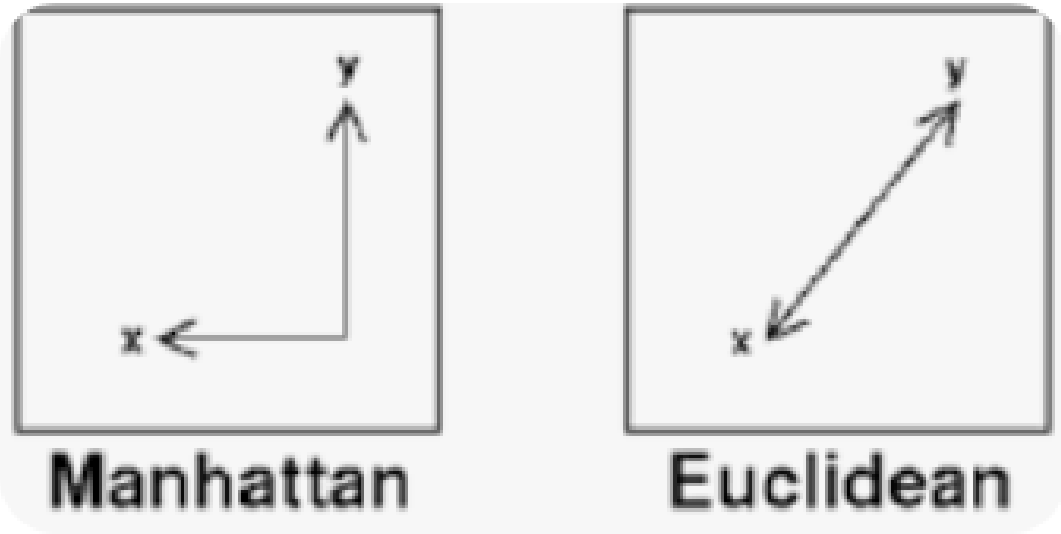


- Rule of thumb is K < sqrt(n), n is number of examples.

To find the nearest neighbours

- Following distance measures can be used in KNN algorithm:

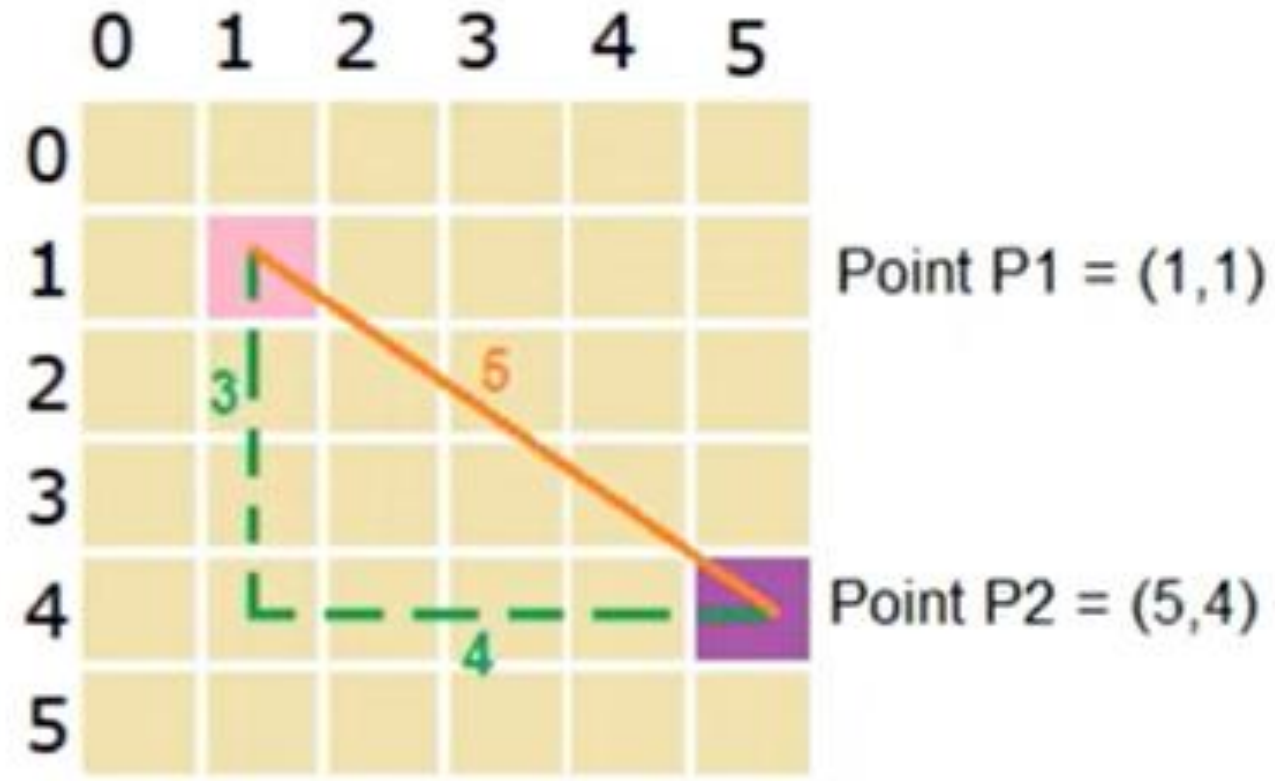
- **Euclidean distance**
- **Manhattan distance**



Euclidean distance

The Euclidean distance formula says:

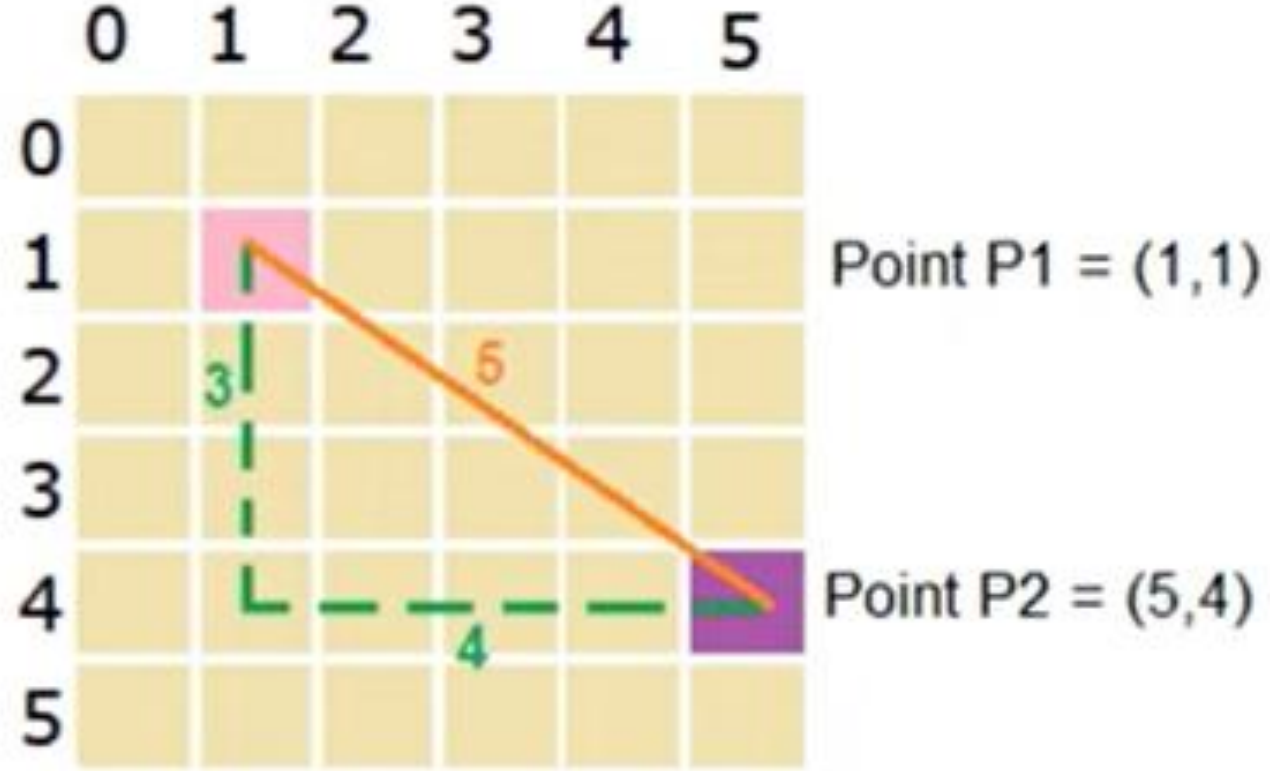
$d = \sqrt{[\,(x_2 - x_1)^2 + (y_2 - y_1)^2]}$



$$\text{Euclidean distance} = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

Manhattan distance

$$|x1 - x2| + |y1 - y2|$$



Manhattan distance = $|5-1| + |4-1| = 7$

Euclidean distance is the least possible distance between point A and B whereas Manhattan distance is measured along the axes at right angles

Example (k=3)

Predict the class for X (P1=3 and P2=7)

P_1	P_2	Class
7	7	False
7	4	False
3	4	True
1	4	True

Find the Euclidean distance of X (P1=3 and P2=7) from every other (P1 and P2)

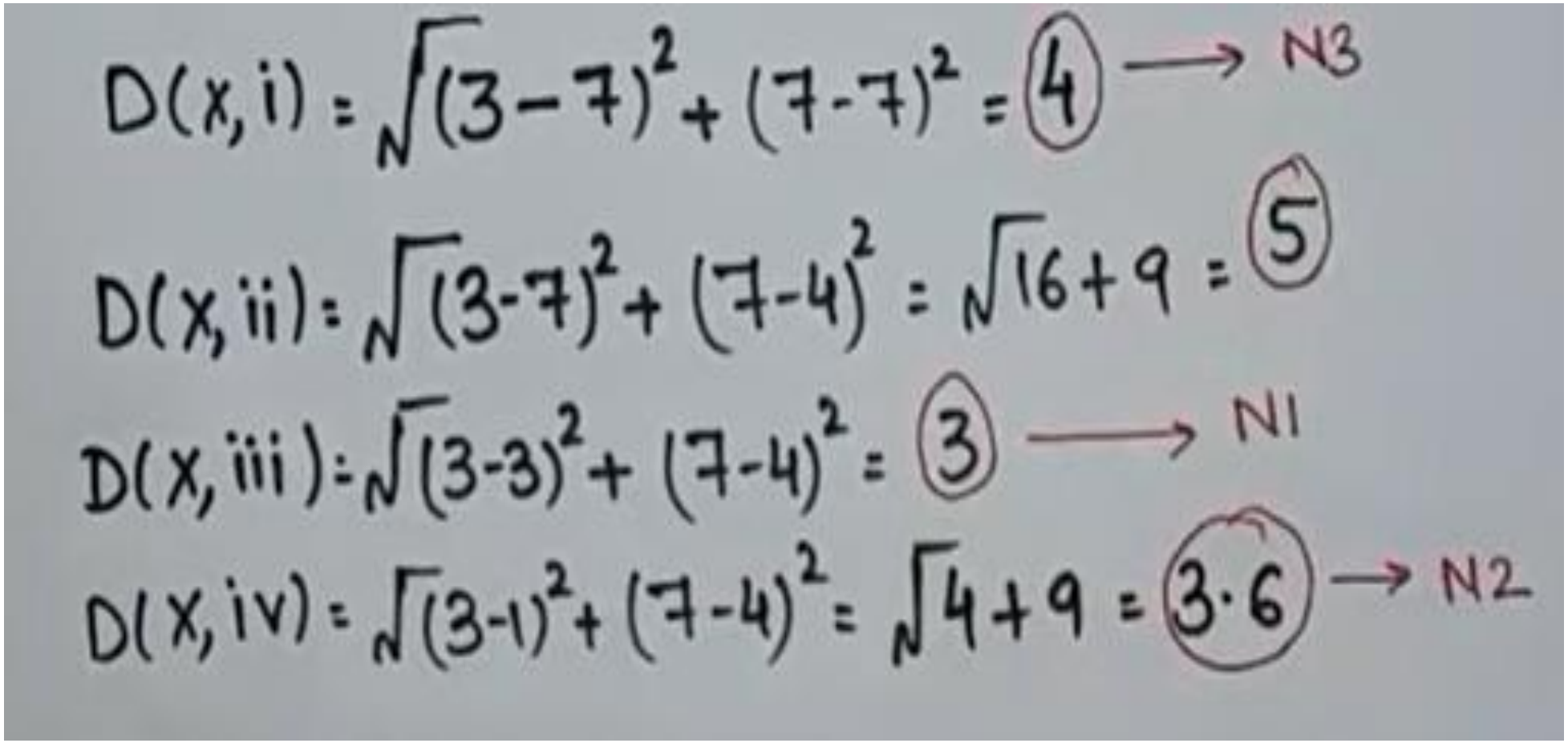
$$D(x, i) = \sqrt{(3-7)^2 + (7-7)^2} = 4$$

$$D(x, ii) = \sqrt{(3-7)^2 + (7-4)^2} = \sqrt{16+9} = 5$$

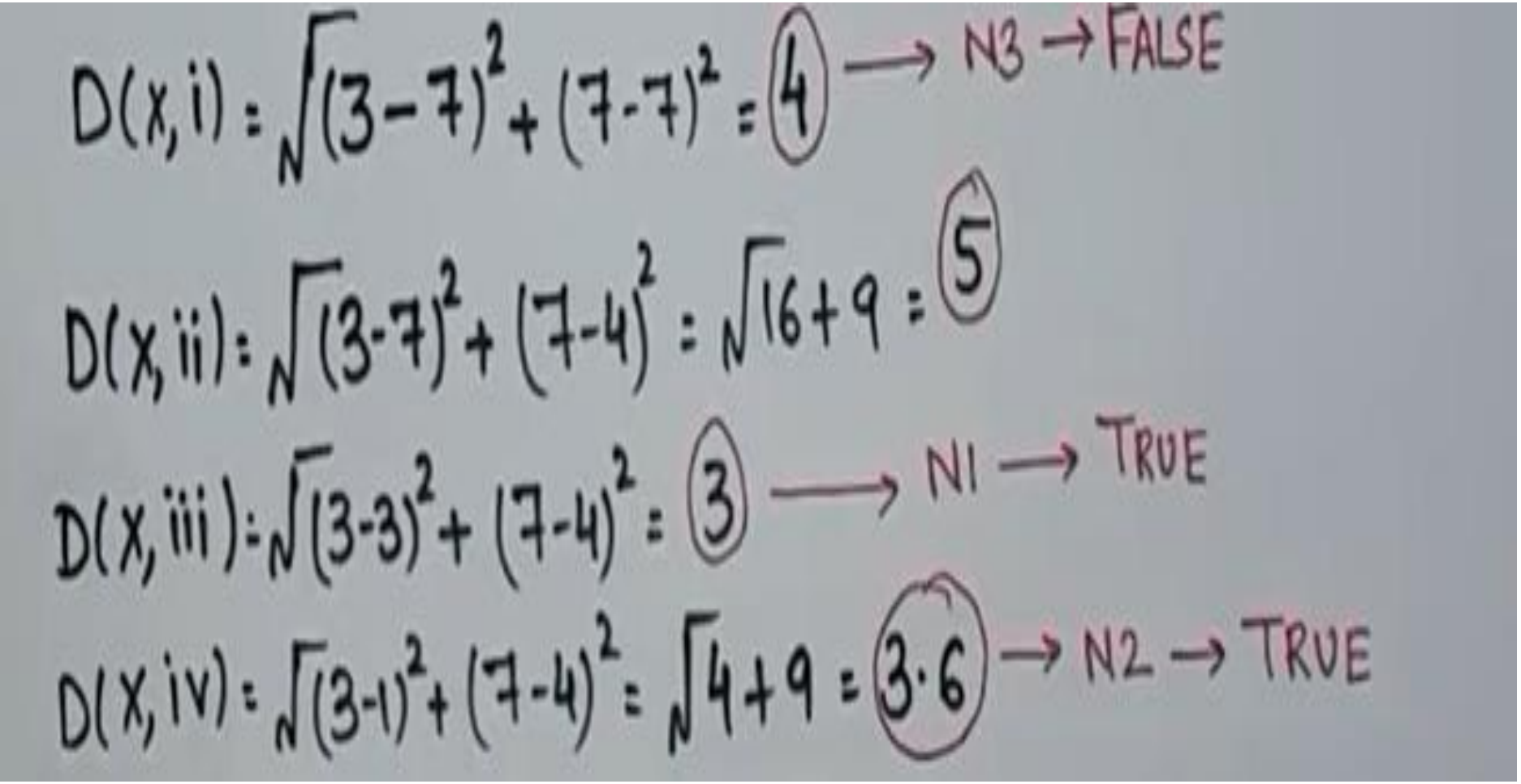
$$D(x, iii) = \sqrt{(3-3)^2 + (7-4)^2} = 3$$

$$D(x, iv) = \sqrt{(3-1)^2 + (7-4)^2} = \sqrt{4+9} = 3.6$$

3 nearest neighbours (N1,N2 and N3)

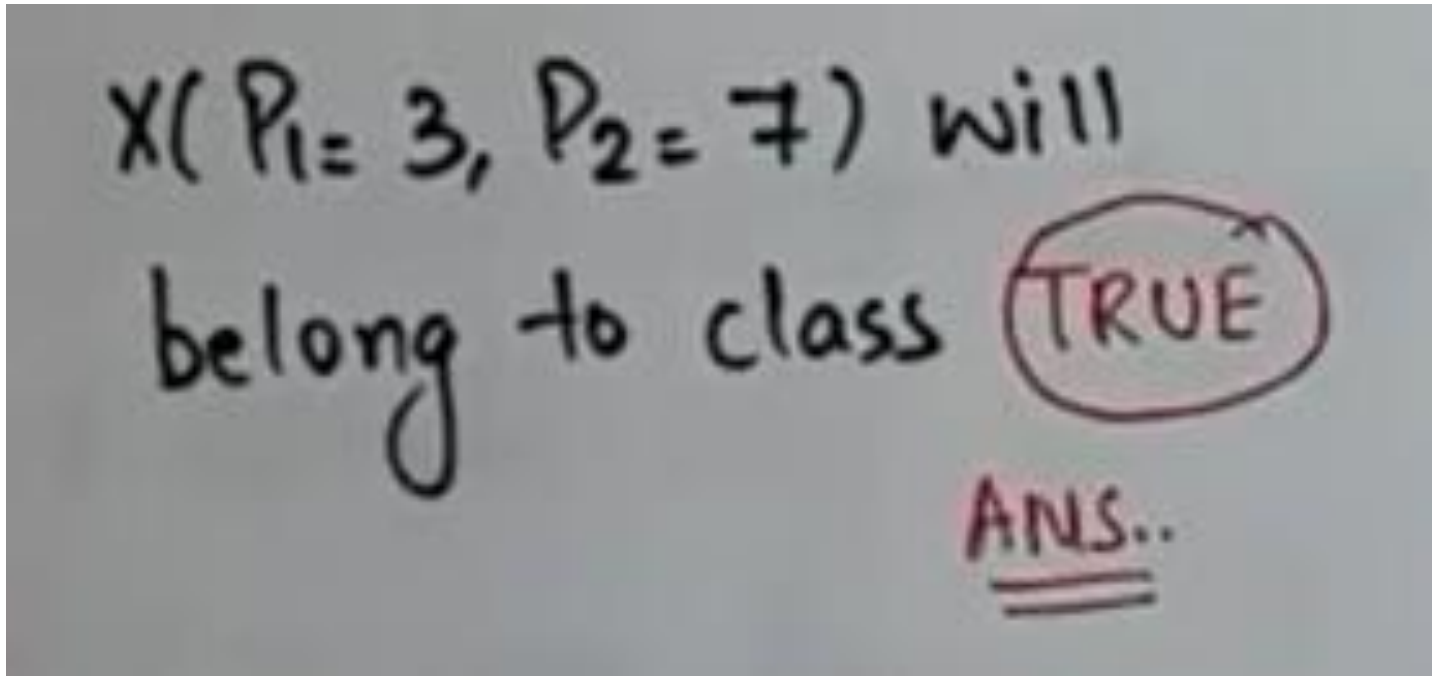


Check for the classes of nearest neighbours



2 TRUE > 1 FALSE

Classify the test data



KNN Example

Customer	Age	Loan	Default
John	25	40000	N
Smith	35	60000	N
Alex	45	80000	N
Jade	20	20000	N
Kate	35	120000	N
Mark	52	18000	N
Anil	23	95000	Y
Pat	40	62000	Y
George	60	100000	Y
Jim	48	220000	Y
Jack	33	150000	Y
Andrew	48	142000	?

We need to predict Andrew default status by using Euclidean distance

Calculate Euclidean distance for all the data points.

Customer	Age	Loan	Default	Euclidean distance
John	25	40000	N	1,02,000.00
Smith	35	60000	N	82,000.00
Alex	45	80000	N	62,000.00
Jade	20	20000	N	1,22,000.00
Kate	35	120000	N	22,000.00
Mark	52	18000	N	1,24,000.00
Anil	23	95000	Y	47,000.01
Pat	40	62000	Y	80,000.00
George	60	100000	Y	42,000.00
Jim	48	220000	Y	78,000.00
Jack	33	150000	Y	8,000.01
Andrew	**48**	**142000**	**?**	

The Euclidean distance formula says:

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

$$= \text{Sq.rt}(48-25)^2 + (142000-40000)^2$$

$$\text{dist}(d_1) = 1,02,000.$$

We need to calcuate the distance for all the datapoints

Customer	Age	Loan	Default	Euclidean distance	Minimum Euclidean Distance
John	25	40000	N	1,02,000.00	
Smith	35	60000	N	82,000.00	
Alex	**45**	**80000**	**N**	**62,000.00**	**5**
Jade	20	20000	N	1,22,000.00	
Kate	**35**	**120000**	**N**	**22,000.00**	**2**
Mark	52	18000	N	1,24,000.00	
Anil	**23**	**95000**	**Y**	**47,000.01**	**4**
Pat	40	62000	Y	80,000.00	
George	**60**	**100000**	**Y**	**42,000.00**	**3**
Jim	48	220000	Y	78,000.00	
Jack	**33**	**150000**	**Y**	**8,000.01**	**1**
Andrew	48	142000	?		

Let assume K = 5

Find minimum euclidean distance and rank in order (ascending)

In this case, 5 minimum euclidean distance.
With k=5, there are two Default = N and three Default = Y out of five closest neighbors.

We can say Andrew default stauts is 'Y' (Yes)

Strengths of KNN

- Very simple and intuitive.
- Can be applied to the data from any distribution.
- Good classification if the number of samples is large enough.

Weaknesses of KNN

- Takes more time to classify a new example.
  - need to calculate and compare distance from new example to all other examples.
- Choosing k may be tricky.
- Need large number of samples for accuracy.

Conclusion

KNN is an effective **machine learning algorithm** that can be used in credit scoring, prediction of cancer cells, image recognition, and many other applications. The main importance of using **KNN** is that it's easy to implement and works well with small datasets.

- **KNN can be used for regression problem statements**.

- In other words, the KNN algorithm can be applied when the dependent variable is continuous.

- For regression problem statements, the predicted value is given by the average of the values of its k nearest neighbours.