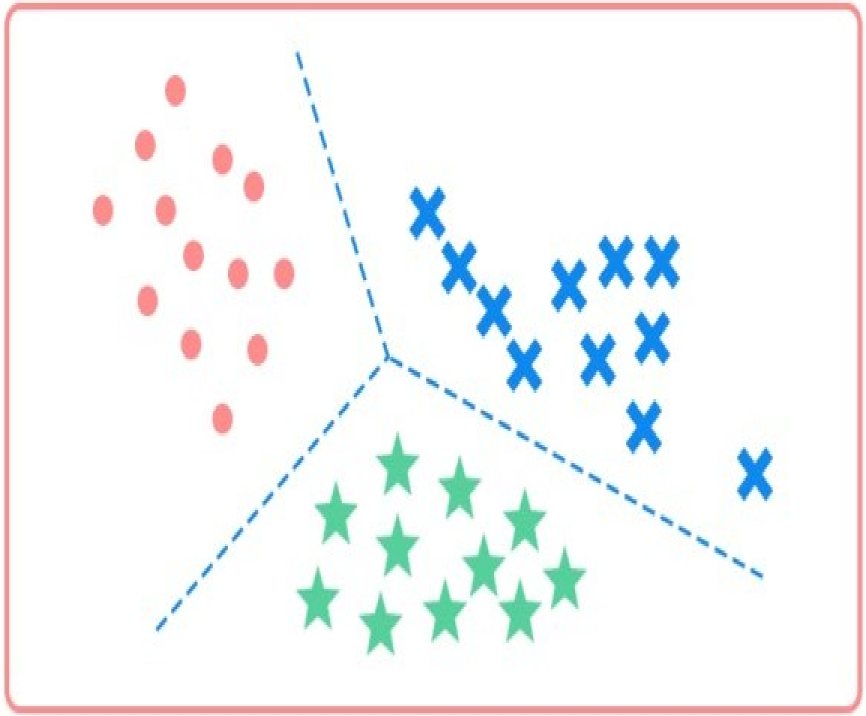
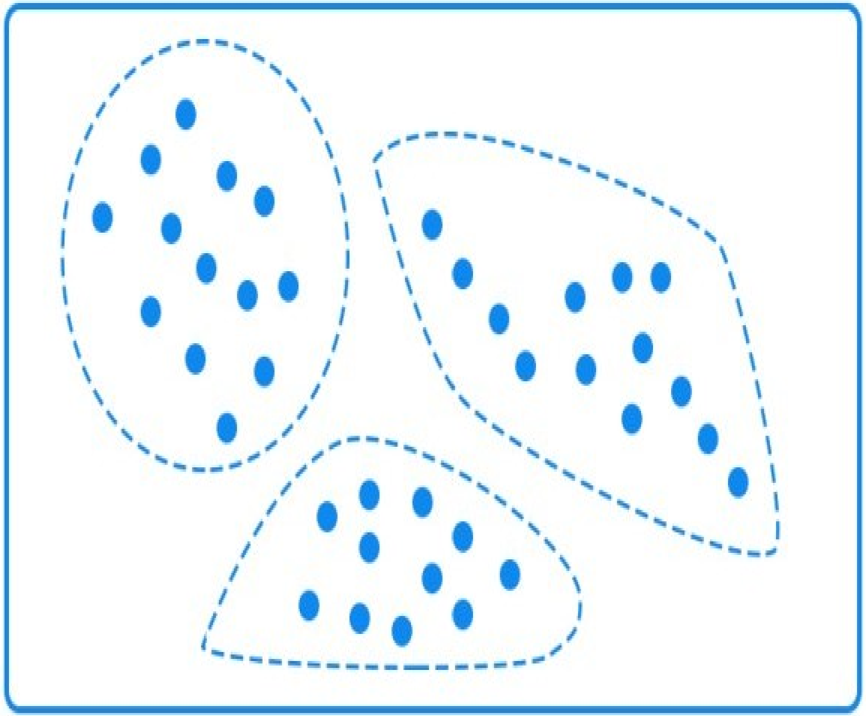
Supervised vs. Unsupervised Learning

Classification



Supervised learning

Clustering



Unsupervised learning

Supervised learning

In *supervised learning*, the training set you feed to the algorithm includes the desired solutions, called *labels* (Figure 1-5).

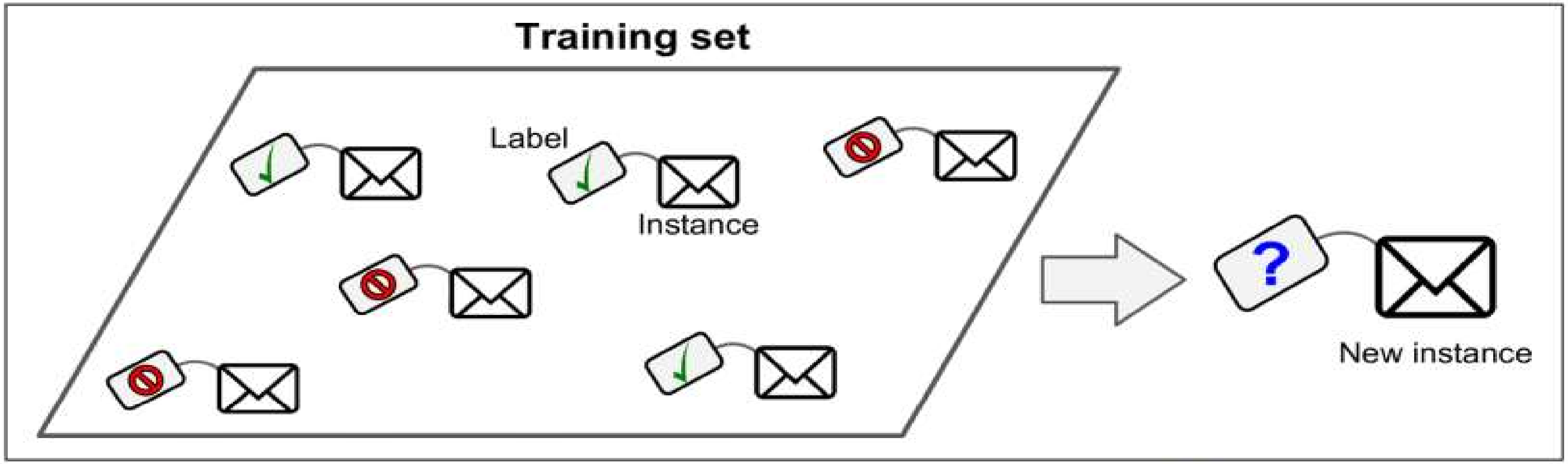
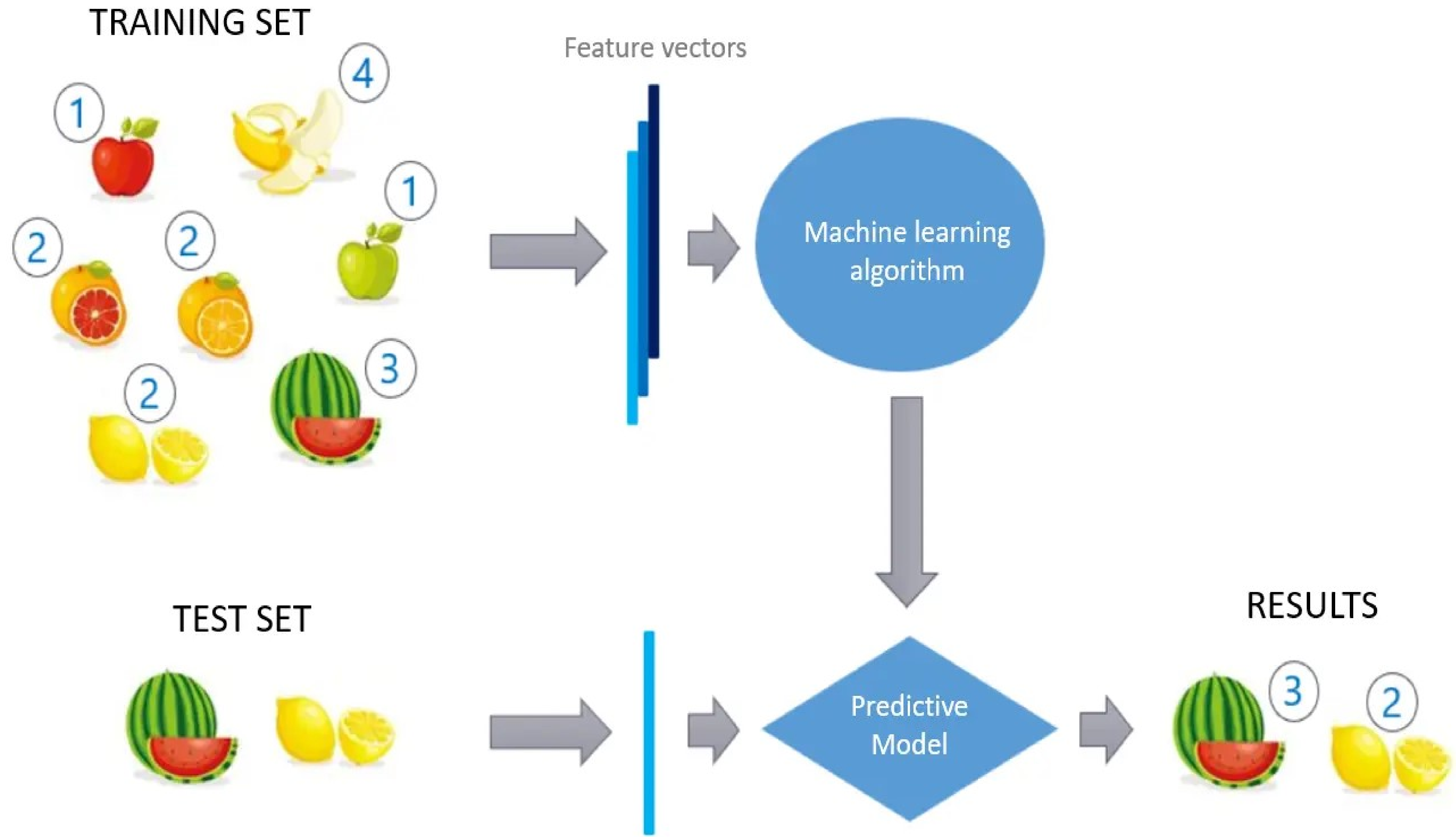


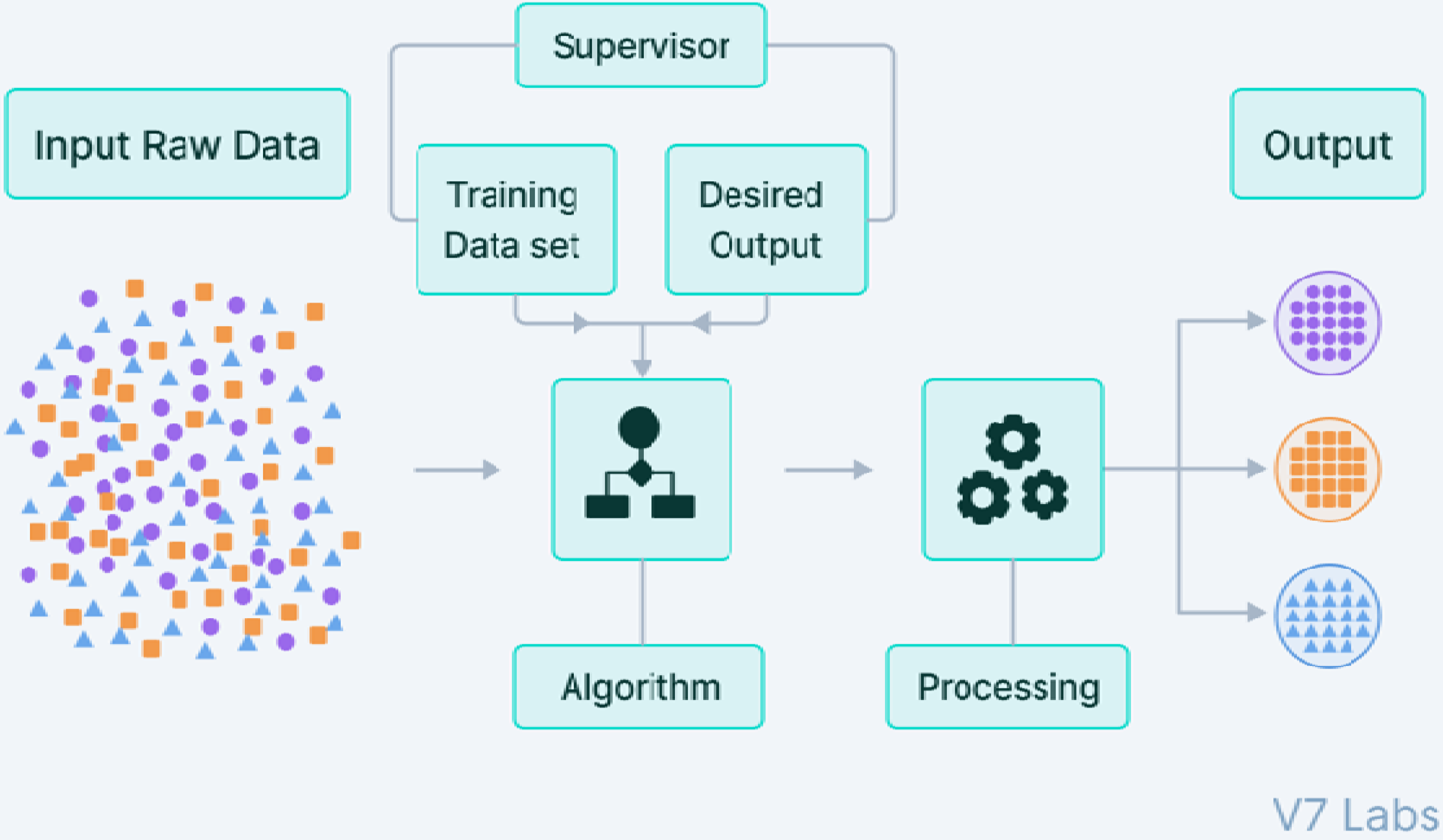
Figure 1-5. A labeled training set for spam classification (an example of supervised learning)

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks[2]

Supervised Learning



Supervised Learning



Supervised Learning

Supervised learning algorithms or methods are the most commonly used ML algorithms. This method or learning algorithm take the data sample i.e. the training data and its associated output i.e. labels or responses with each data samples during the training process.

It is called supervised because the whole process of learning can be thought as it is being supervised by a teacher or supervisor. Examples of supervised machine learning algorithms includes **Decision tree, Random Forest, KNN, Logistic Regression** etc.

Classification

The key objective of classification-based tasks is to predict categorial output labels or responses for the given input data. The output will be based on what the model has learned in training phase. As we know that the categorial output responses means unordered and discrete values, hence each output response will belong to a specific class or category. We will discuss

Regression

The key objective of regression-based tasks is to predict output labels or responses which are continues numeric values, for the given input data. The output will be based on what the model has learned in its training phase. Basically, regression models use the input data features (independent variables) and their corresponding continuous numeric output values (dependent or outcome variables) to learn specific association between inputs and corresponding outputs.

Unsupervised learning

In *unsupervised learning*, as you might guess, the training data is unlabeled (Figure 1-7). The system tries to learn without a teacher.

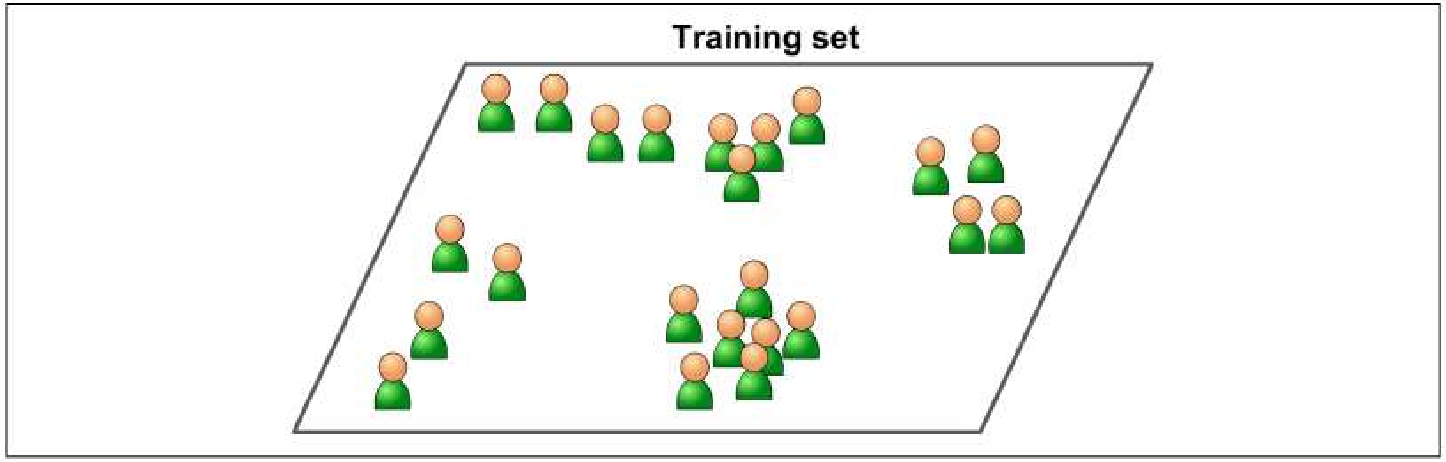


Figure 1-7. An unlabeled training set for unsupervised learning

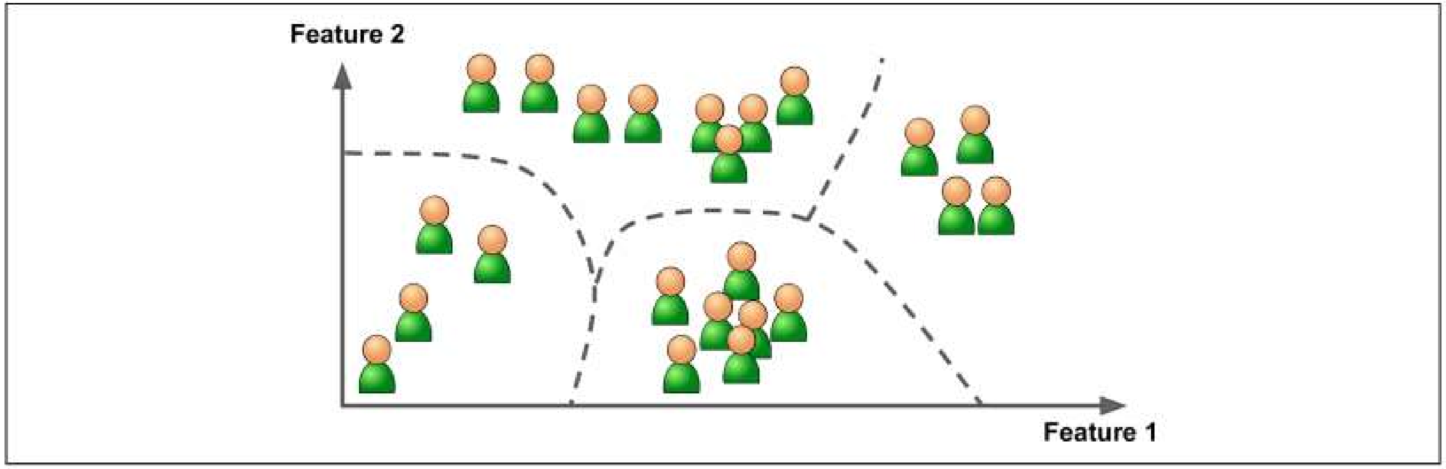
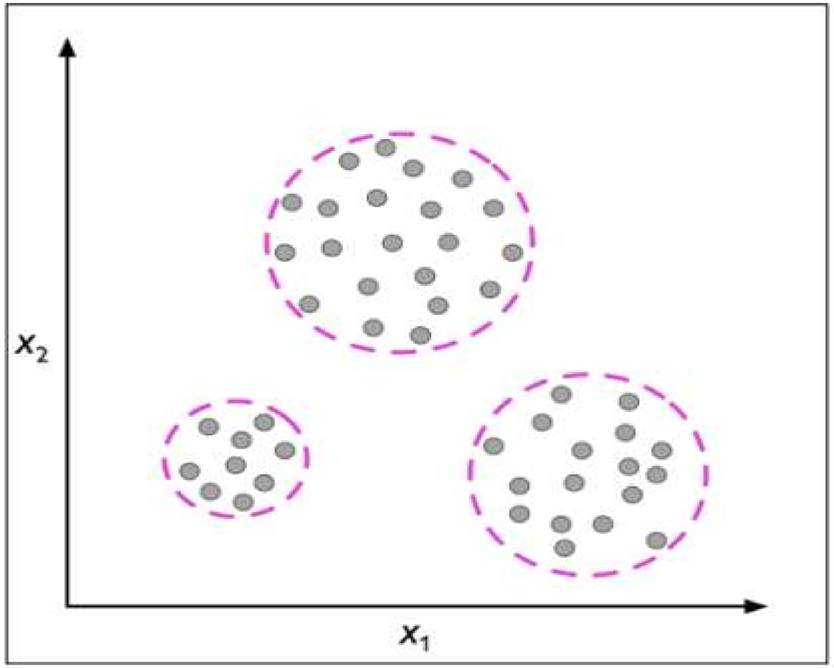


Figure 1-8. Clustering

Discovering hidden structures with unsupervised learning

In supervised learning, we know the right answer beforehand when we train our model, and in reinforcement learning, we define a measure of reward for particular actions by the agent. In unsupervised learning, however, we are dealing with unlabeled data or data of unknown structure. Using unsupervised learning techniques, we are able to explore the structure of our data to extract meaningful information without the guidance of a known outcome variable or reward function.

Finding subgroups with clustering



Unsupervised Learning

As the name suggests, it is opposite to supervised ML methods or algorithms which means in unsupervised machine learning algorithms we do not have any supervisor to provide any sort of guidance. Unsupervised learning algorithms are handy in the scenario in which we do not have the liberty, like in supervised learning algorithms, of having pre-labeled training data and we want to extract useful pattern from input data.

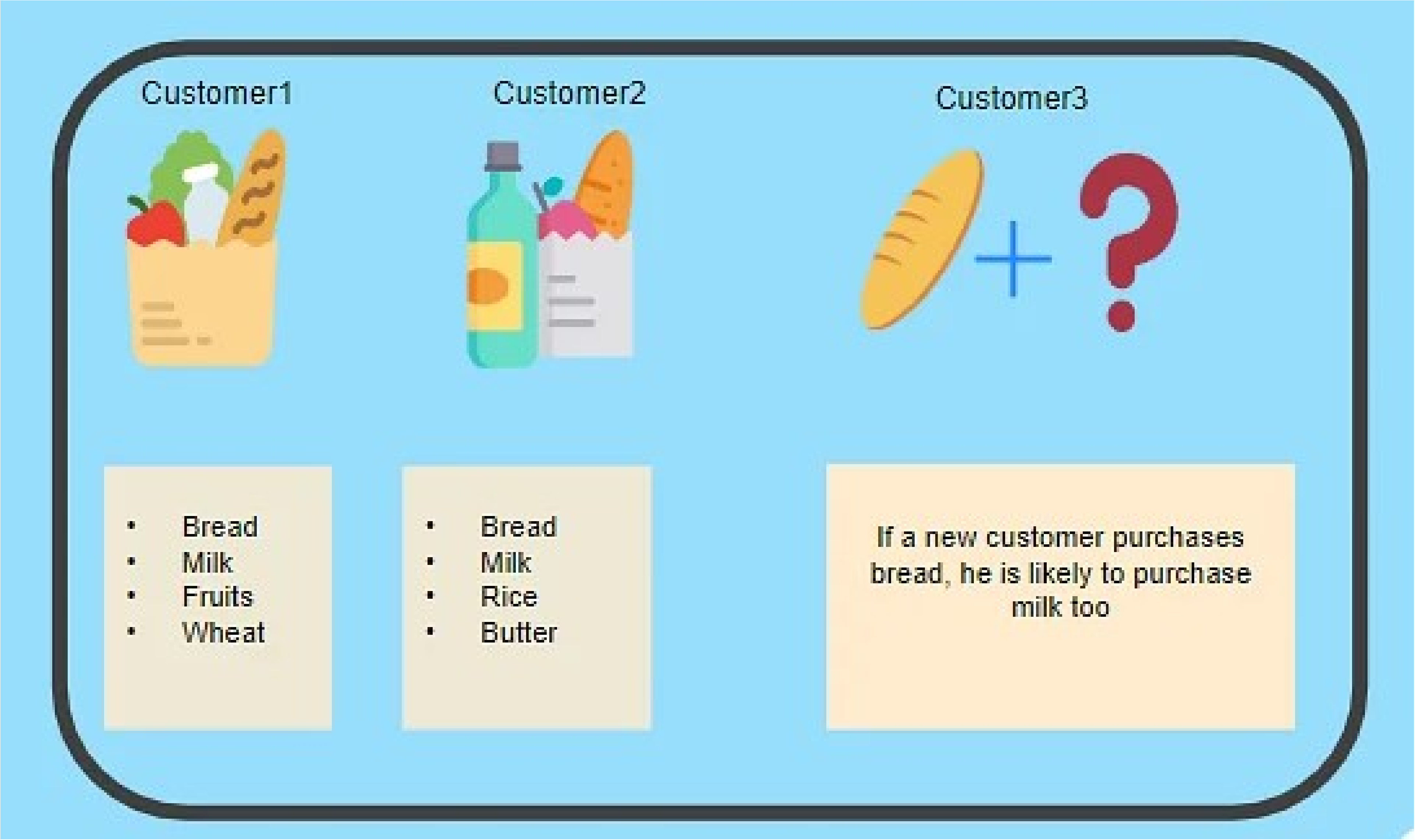
Clustering

Clustering methods are one of the most useful unsupervised ML methods. These algorithms used to find similarity as well as relationship patterns among data samples and then cluster those samples into groups having similarity based on features. The real-world example of clustering is to group the customers by their purchasing behavior.

Association

Another useful unsupervised ML method is **Association** which is used to analyze large dataset to find patterns which further represents the interesting relationships between various items. It is also termed as **Association Rule Mining** or **Market basket analysis** which is mainly used to analyze customer shopping patterns.

Association Rule Mining



Customer1

- Bread
- Milk
- Fruits
- Wheat

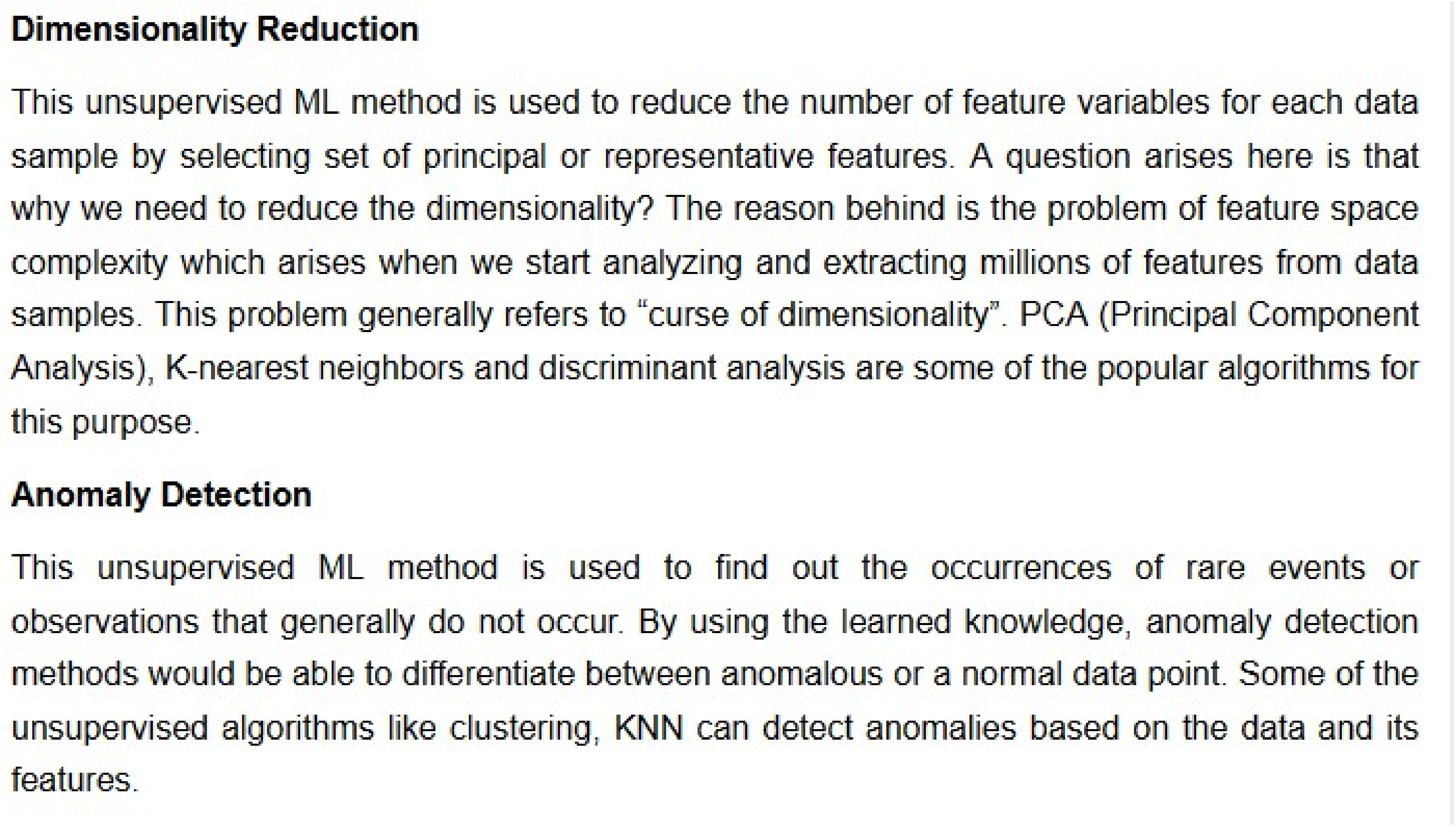
Customer2

- Bread
- Milk
- Rice
- Butter

Customer3

If a new customer purchases bread, he is likely to purchase milk too

Unsupervised Learning



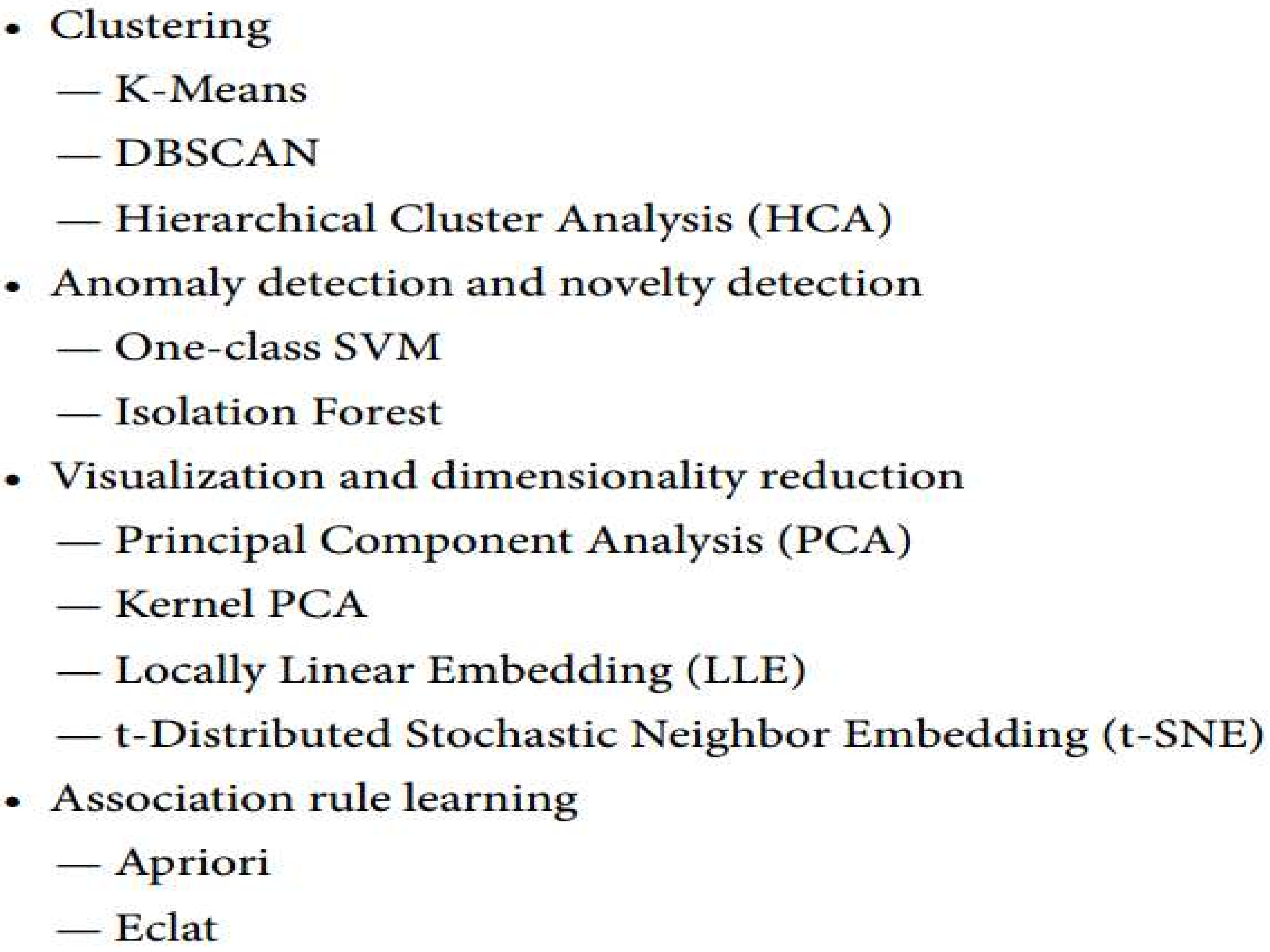
Dimensionality Reduction

This unsupervised ML method is used to reduce the number of feature variables for each data sample by selecting set of principal or representative features. A question arises here is that why we need to reduce the dimensionality? The reason behind is the problem of feature space complexity which arises when we start analyzing and extracting millions of features from data samples. This problem generally refers to “curse of dimensionality”. PCA (Principal Component Analysis), K-nearest neighbors and discriminant analysis are some of the popular algorithms for this purpose.

Anomaly Detection

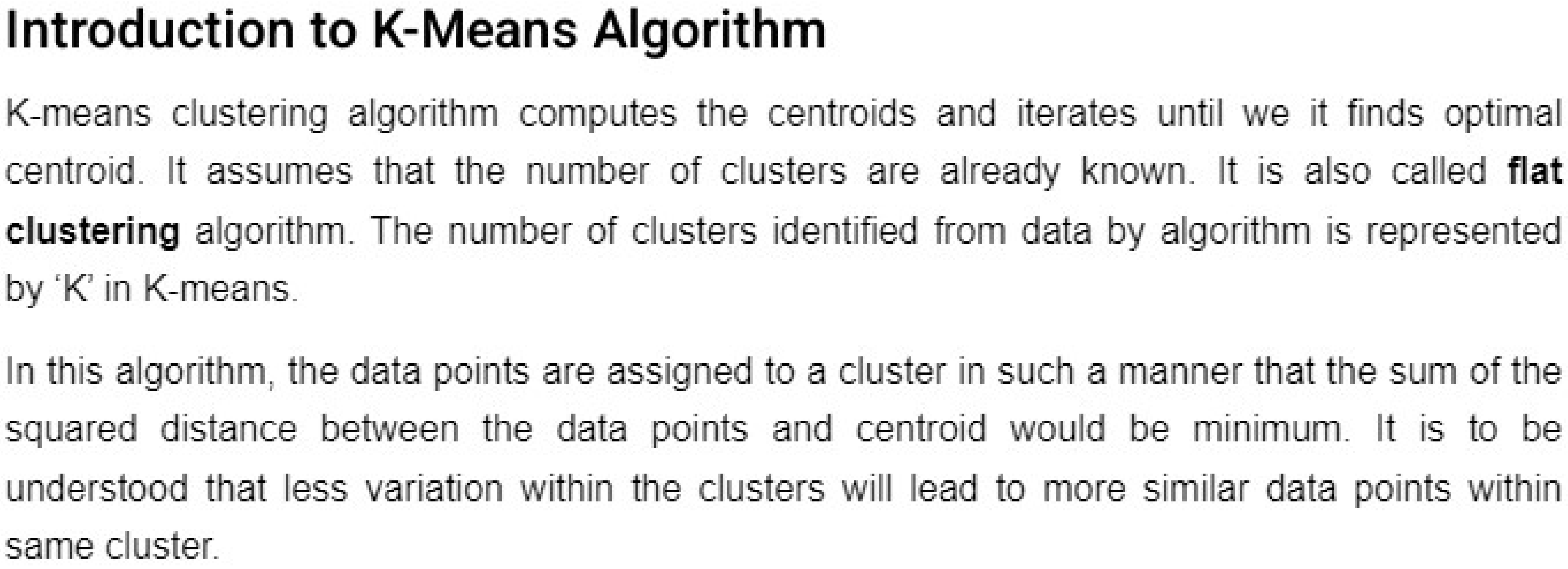
This unsupervised ML method is used to find out the occurrences of rare events or observations that generally do not occur. By using the learned knowledge, anomaly detection methods would be able to differentiate between anomalous or a normal data point. Some of the unsupervised algorithms like clustering, KNN can detect anomalies based on the data and its features.

Unsupervised Learning



- Clustering
  - — K-Means
  - — DBSCAN
  - — Hierarchical Cluster Analysis (HCA)
- Anomaly detection and novelty detection
  - — One-class SVM
  - — Isolation Forest
- Visualization and dimensionality reduction
  - — Principal Component Analysis (PCA)
  - — Kernel PCA
  - — Locally Linear Embedding (LLE)
  - — t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning
  - — Apriori
  - — Eclat

K-means Clustering Algorithm
(unsupervised machine learning)



Introduction to K-Means Algorithm

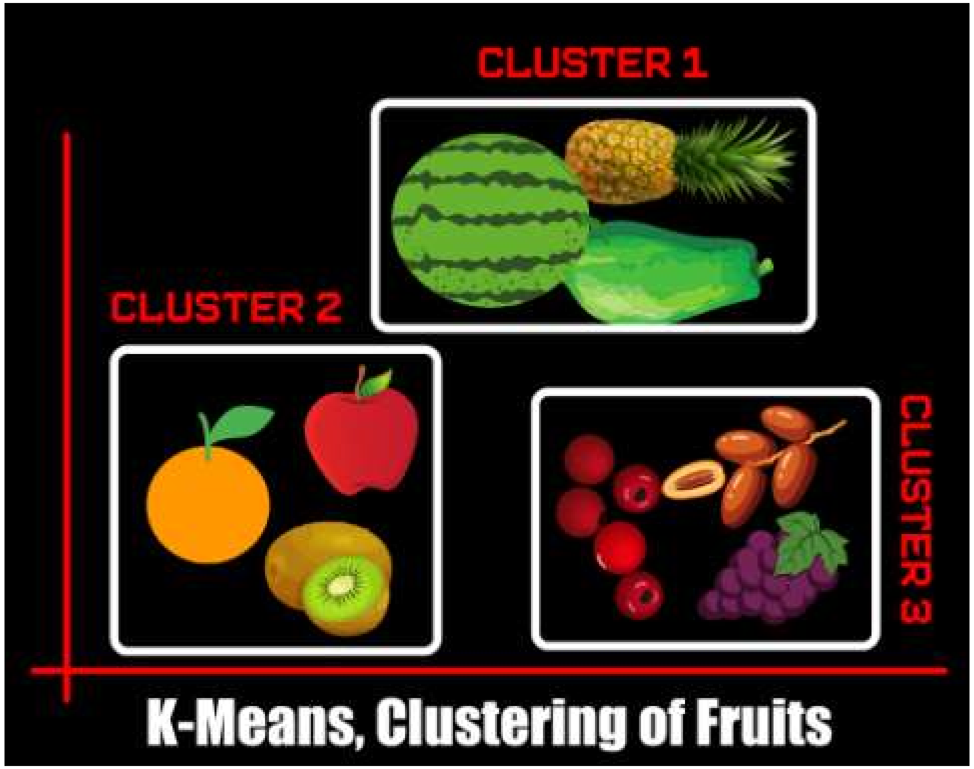
K-means clustering algorithm computes the centroids and iterates until we it finds optimal centroid. It assumes that the number of clusters are already known. It is also called **flat clustering** algorithm. The number of clusters identified from data by algorithm is represented by 'K' in K-means.

In this algorithm, the data points are assigned to a cluster in such a manner that the sum of the squared distance between the data points and centroid would be minimum. It is to be understood that less variation within the clusters will lead to more similar data points within same cluster.

What Is K-Means Clustering In Machine Learning?

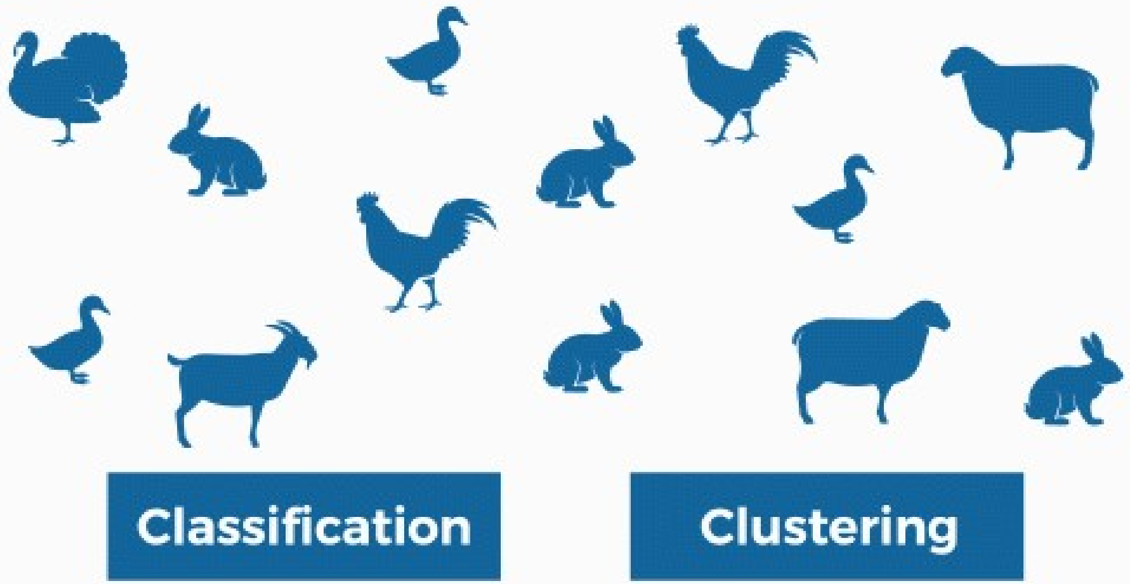
K-Means is a clustering algorithm, which is a part of unsupervised machine learning(data with no labels). It is used to create clusters out of lots of unlabeled data points where data holding a similar property or pattern are stored in the same cluster. "K" in K means represents the number of clusters it forms, which is decided by the practitioner.

For example, if you have data of lots of fruit without labels(means you just have their property like shape, size, color, taste, etc and you don't know which fruit it is) then based on their property it will create different clusters and put the fruits with same property in one cluster and others in other clusters.



Clusterização

A clusterização ou agrupamento é uma técnica de aprendizado de máquina **não-supervisionada**. Na criação do modelo de clusterização não usamos a target, por isso é chamado de não-supervisionado. Também não fazemos a separação do dataset entre treino e teste. Usamos todos os dados, sem fazer divisões das linhas dos dados. No aprendizado não-supervisionado, chamamos as colunas do dataset de **dimensões** e não mais de atributos ou variáveis. Por exemplo, no dataset de Flores (iris dataset) temos 4 dimensões: altura da pétala, largura da pétala, altura da sépala e largura da sépala.



How Does KMeans Work?

It is very easy to understand how K means work. To understand it better, let’s see it’s process step by step.

Step 1 → After collecting and feeding the data to the model we first decide the value of “K” which is nothing but the number of clusters. The value of “K” is generally decided on the basis of understanding of domain and data. There is also one other method to decide the value of “K” which we will discuss later in this article.

Step 2 → Then we randomly choose the K number of Centroid (center of data points, initially we choose it randomly so it is actually not in a center) which represents the center of a particular cluster. For example, if we choose K’s value to be 2 then we will have two clusters and each cluster will have one Centroid.

Step 3 → Then we calculate the distance of each data point from each centroid we created. And then we assign the data points to a certain cluster based on the closest distance it has from a centroid. For example If the distance of a data point ‘P’ from the centroid “A” is 5 and from the centroid “B” is 7 then the data point ‘P’ will be assigned to centroid “A”.

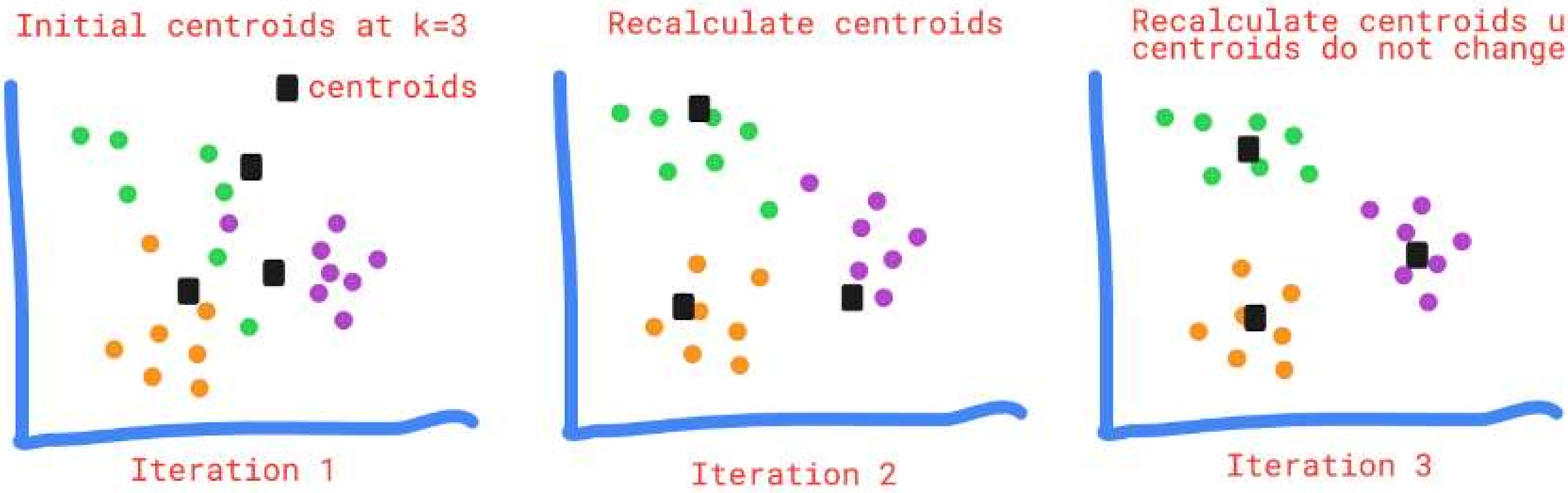
Step 4 → After we assign each data point to its closest cluster, we then calculate the actual centroid of each cluster.

Step 5 → And after that we repeat the step 3 and 4 until it’s fully optimized.

k-means clustering

Steps involved in k-means clustering algorithm,

1. *Choose the k number of clusters and determine their centroids*
2. *Assign each data point to its nearest centroid using distance measures*
3. *Recalculate the new centroids and again assign each data point to its nearest centroid*
4. *Repeat step 3 and 4 until centroids do not change or no change in criterion function (J)*



Initial centroids at k=3

Recalculate centroids

Recalculate centroids until centroids do not change

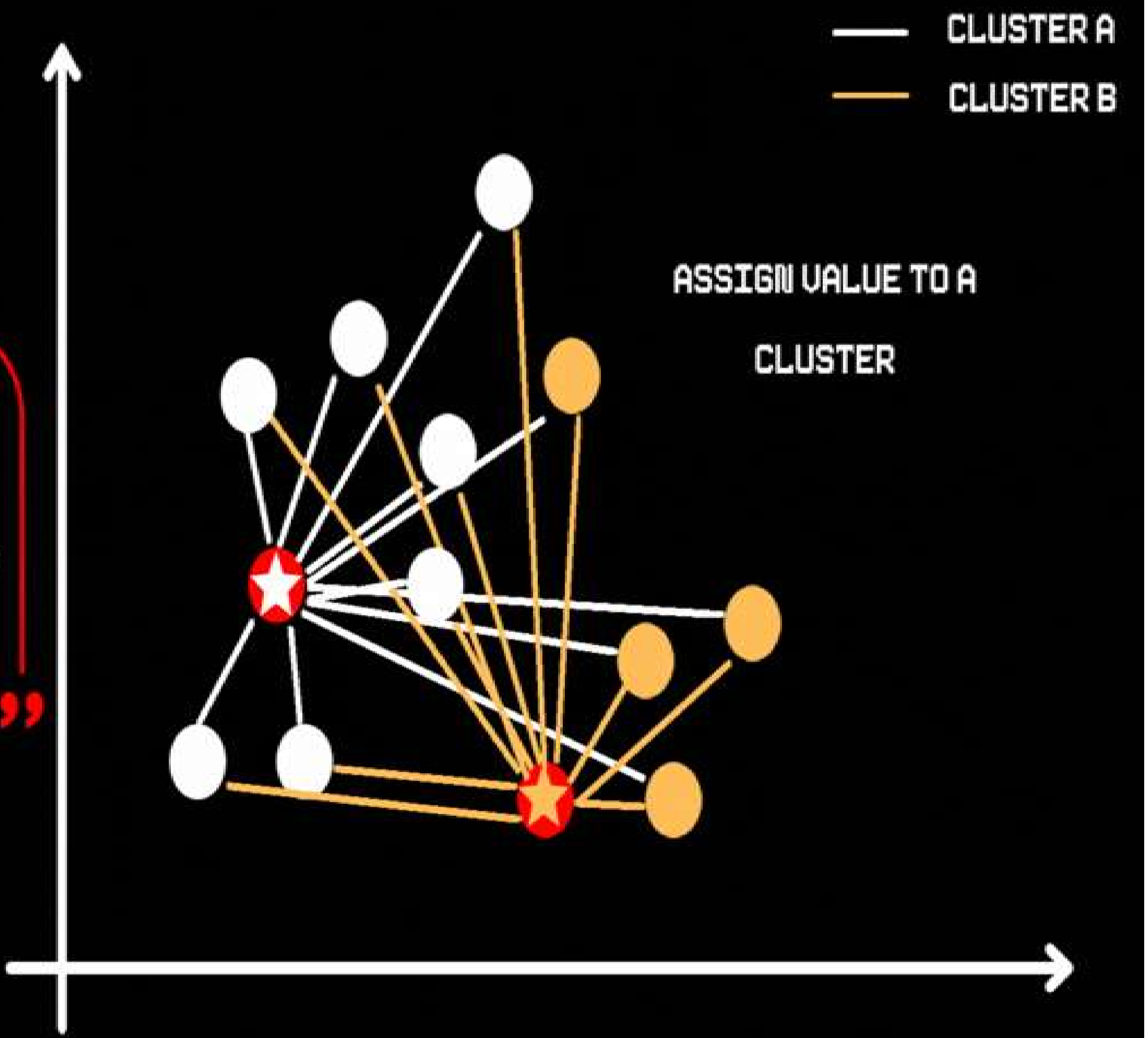
Iteration 1

Iteration 2

Iteration 3

Step 3

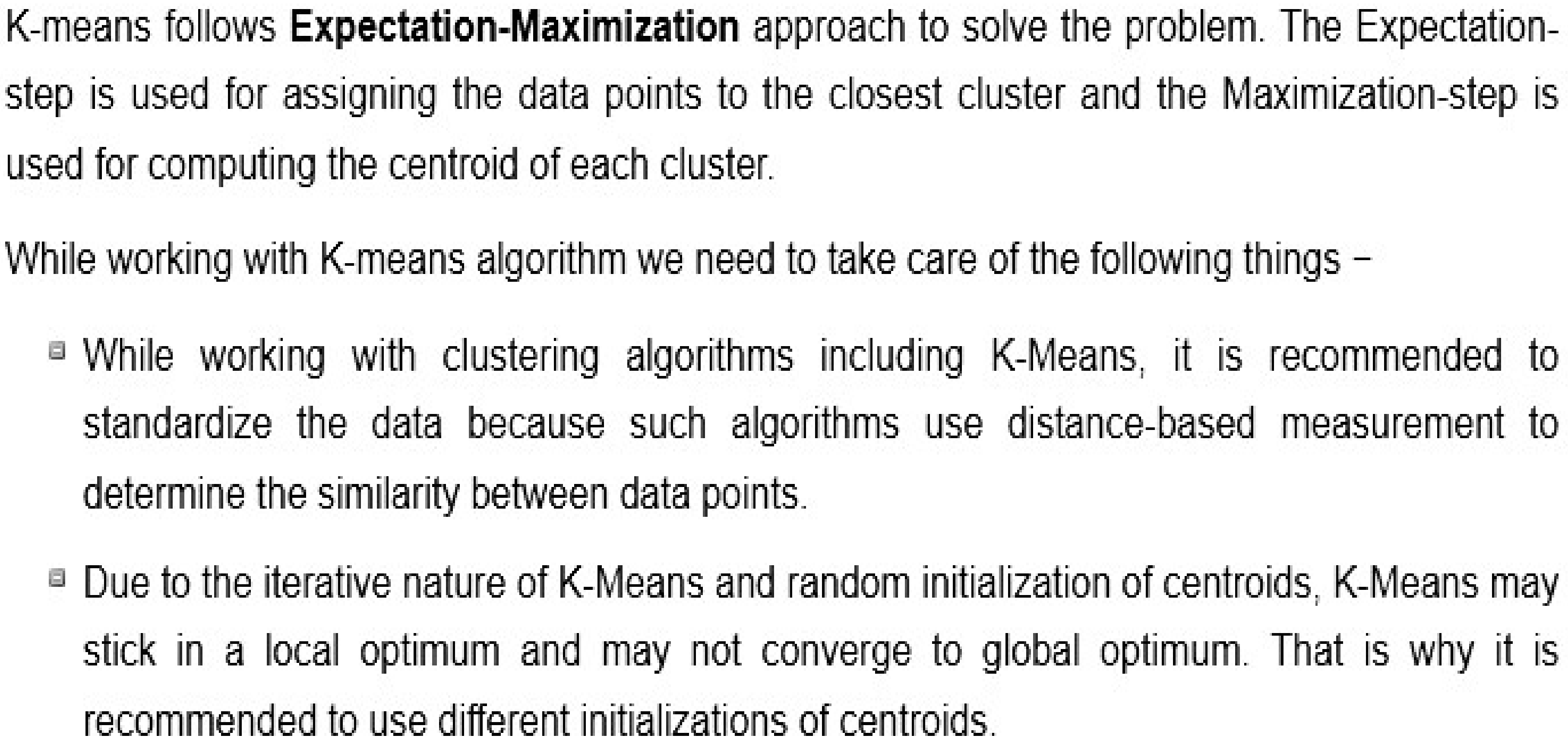
> “WE WILL ASSIGN THE POINTS TO A CLUSTER BASED ON IT'S NEAREST CENTROID”



ASSIGN VALUE TO A CLUSTER

Buggyprogrammer.com

K-means Care



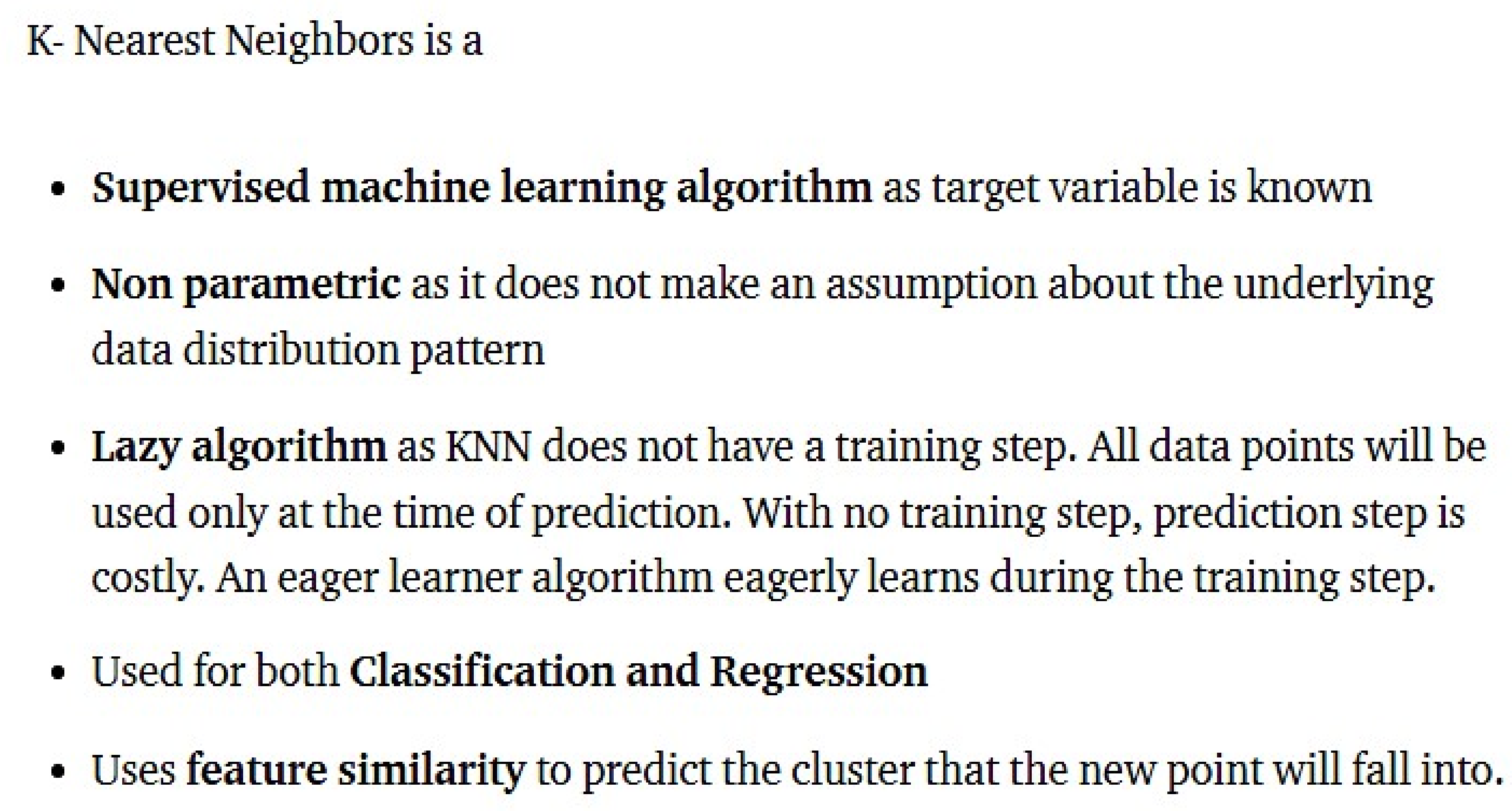
K-means follows **Expectation-Maximization** approach to solve the problem. The Expectation-step is used for assigning the data points to the closest cluster and the Maximization-step is used for computing the centroid of each cluster.

While working with K-means algorithm we need to take care of the following things −

- While working with clustering algorithms including K-Means, it is recommended to standardize the data because such algorithms use distance-based measurement to determine the similarity between data points.
- Due to the iterative nature of K-Means and random initialization of centroids, K-Means may stick in a local optimum and may not converge to global optimum. That is why it is recommended to use different initializations of centroids.

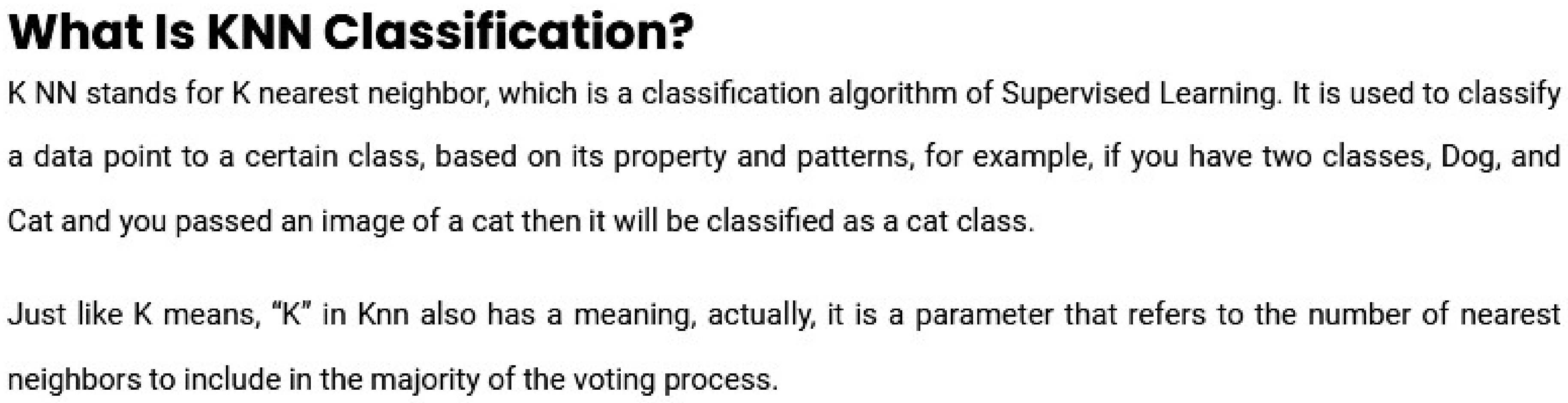
K- Nearest Neighbors (KNN)

(supervised machine learning)



K- Nearest Neighbors is a

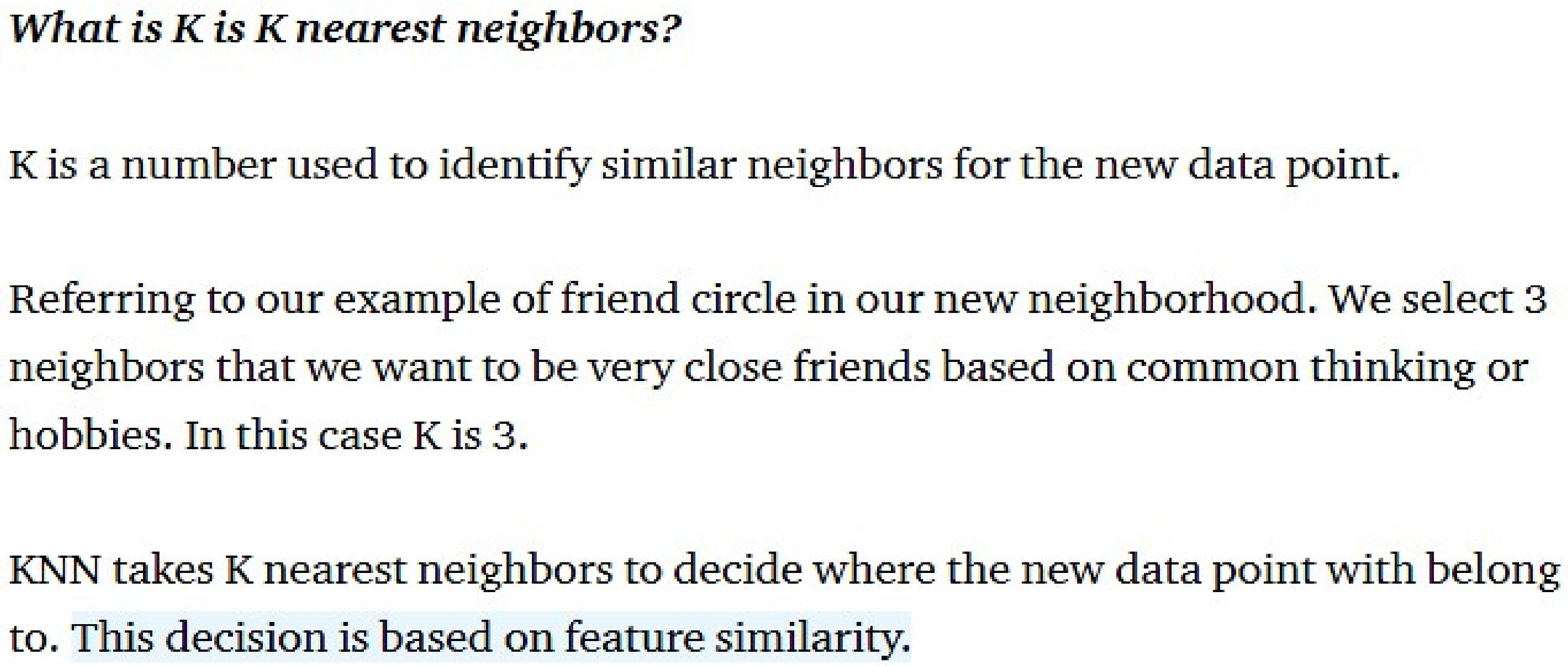
- **Supervised machine learning algorithm** as target variable is known
- **Non parametric** as it does not make an assumption about the underlying data distribution pattern
- **Lazy algorithm** as KNN does not have a training step. All data points will be used only at the time of prediction. With no training step, prediction step is costly. An eager learner algorithm eagerly learns during the training step.
- Used for both **Classification and Regression**
- Uses **feature similarity** to predict the cluster that the new point will fall into.



What Is KNN Classification?

K NN stands for K nearest neighbor, which is a classification algorithm of Supervised Learning. It is used to classify a data point to a certain class, based on its property and patterns, for example, if you have two classes, Dog, and Cat and you passed an image of a cat then it will be classified as a cat class.

Just like K means, “K” in Knn also has a meaning, actually, it is a parameter that refers to the number of nearest neighbors to include in the majority of the voting process.

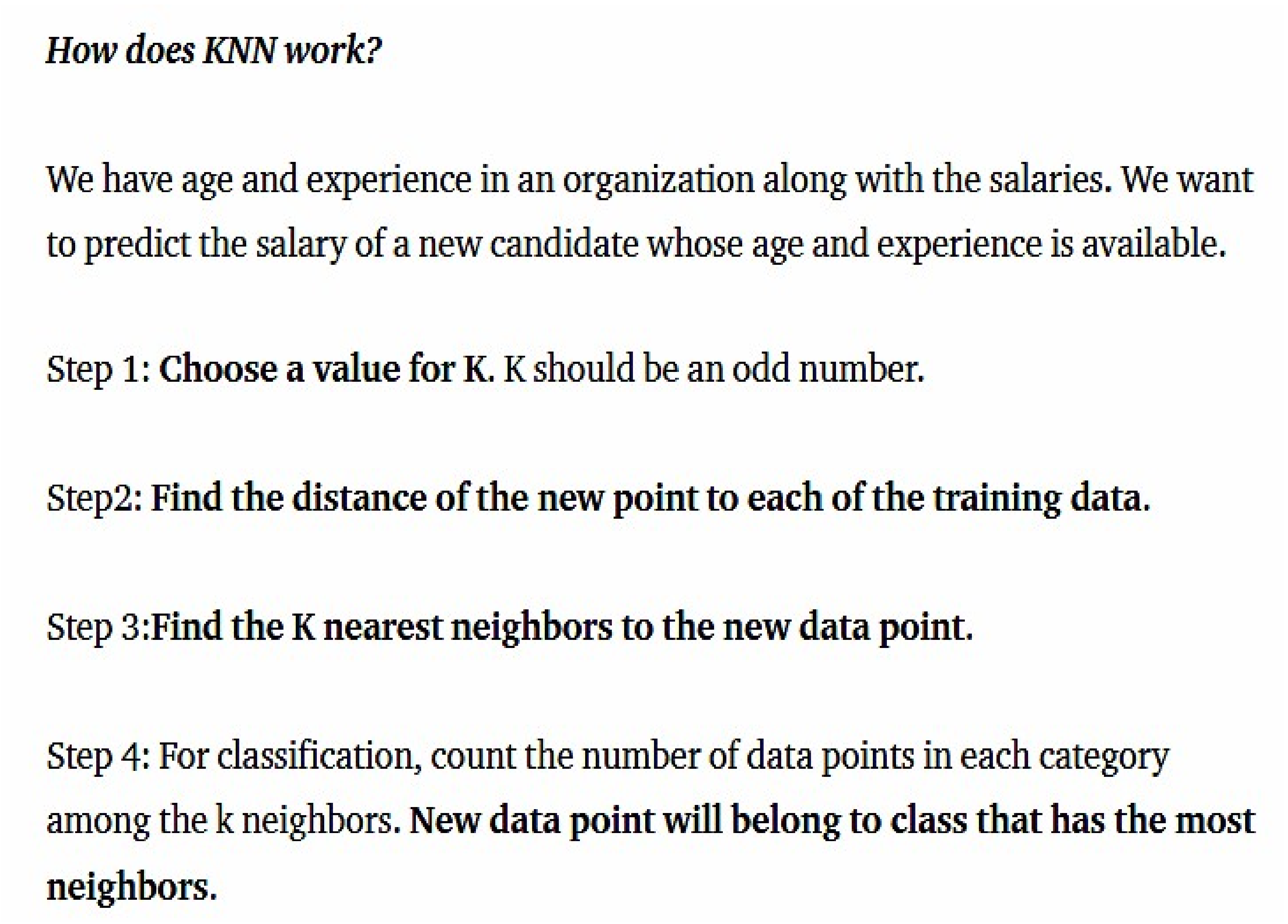


What is K is K nearest neighbors?

K is a number used to identify similar neighbors for the new data point.

Referring to our example of friend circle in our new neighborhood. We select 3 neighbors that we want to be very close friends based on common thinking or hobbies. In this case K is 3.

KNN takes K nearest neighbors to decide where the new data point with belong to. This decision is based on feature similarity.



How does KNN work?

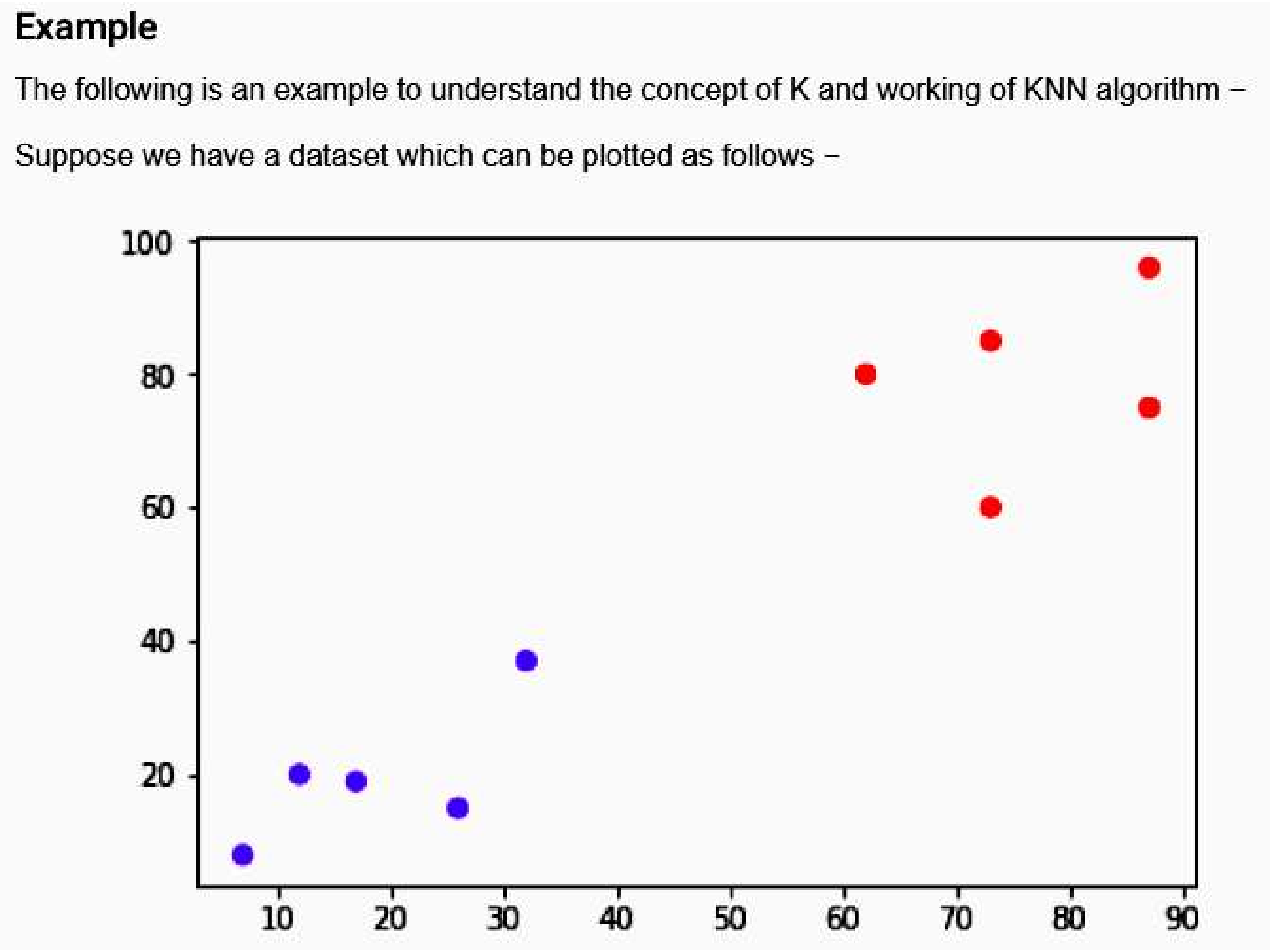
We have age and experience in an organization along with the salaries. We want to predict the salary of a new candidate whose age and experience is available.

Step 1: **Choose a value for K.** K should be an odd number.

Step2: **Find the distance of the new point to each of the training data.**

Step 3:**Find the K nearest neighbors to the new data point.**

Step 4: For classification, count the number of data points in each category among the k neighbors. **New data point will belong to class that has the most neighbors.**

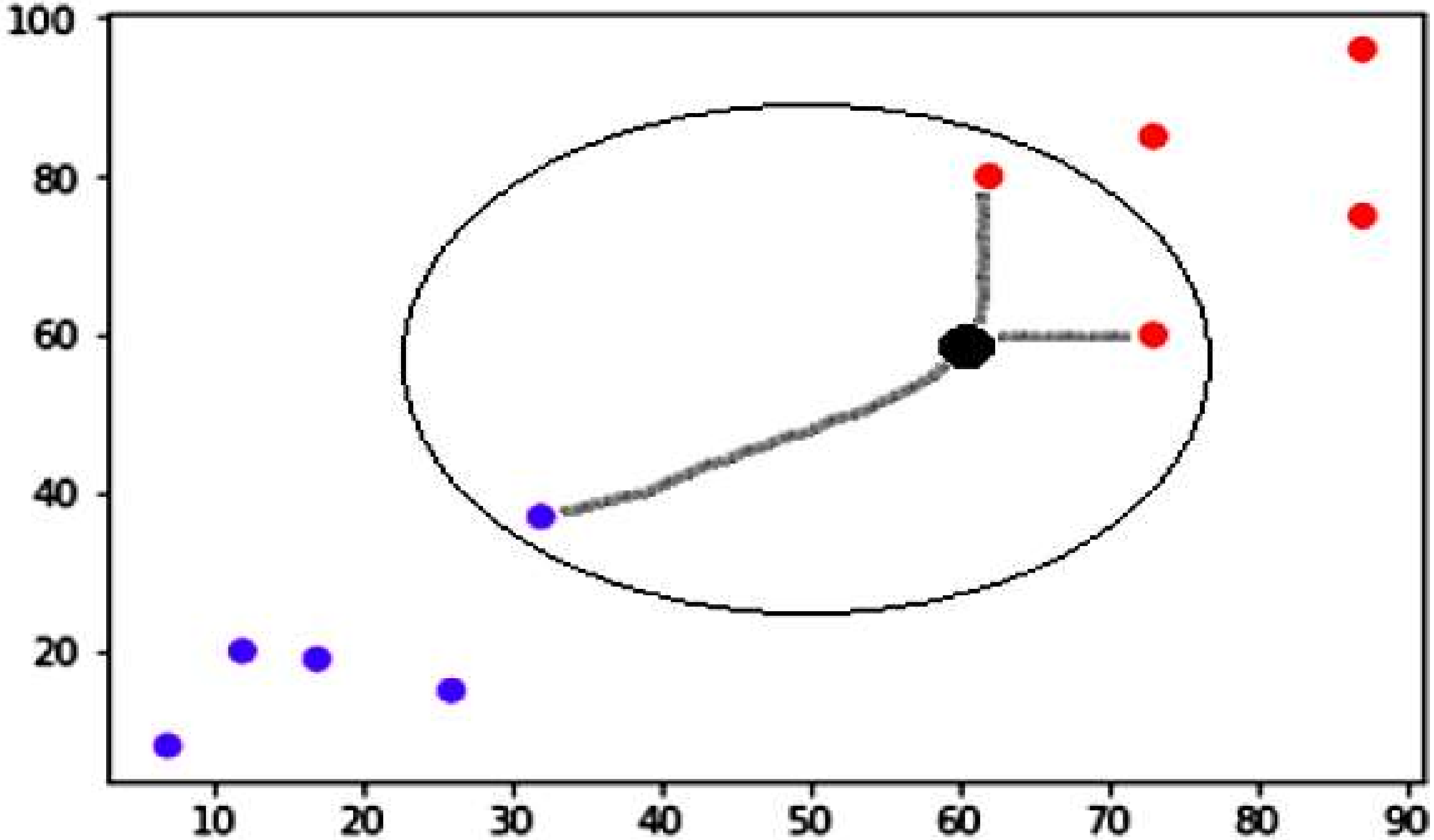


Example

The following is an example to understand the concept of K and working of KNN algorithm −

Suppose we have a dataset which can be plotted as follows −

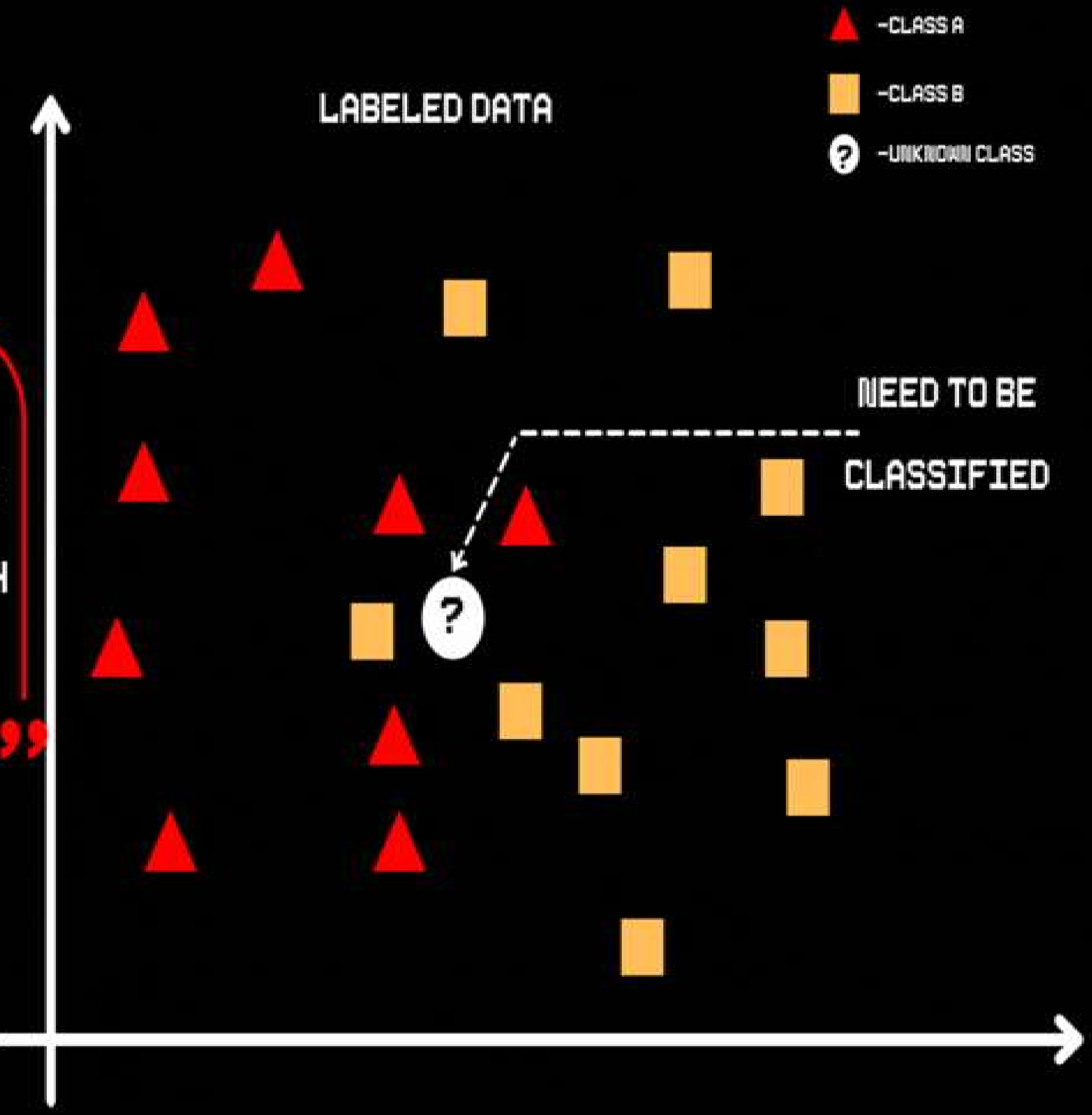
Now, we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming K = 3 i.e. it would find three nearest data points. It is shown in the next diagram –



We can see in the above diagram the three nearest neighbors of the data point with black dot. Among those three, two of them lies in Red class hence the black dot will also be assigned in red class.

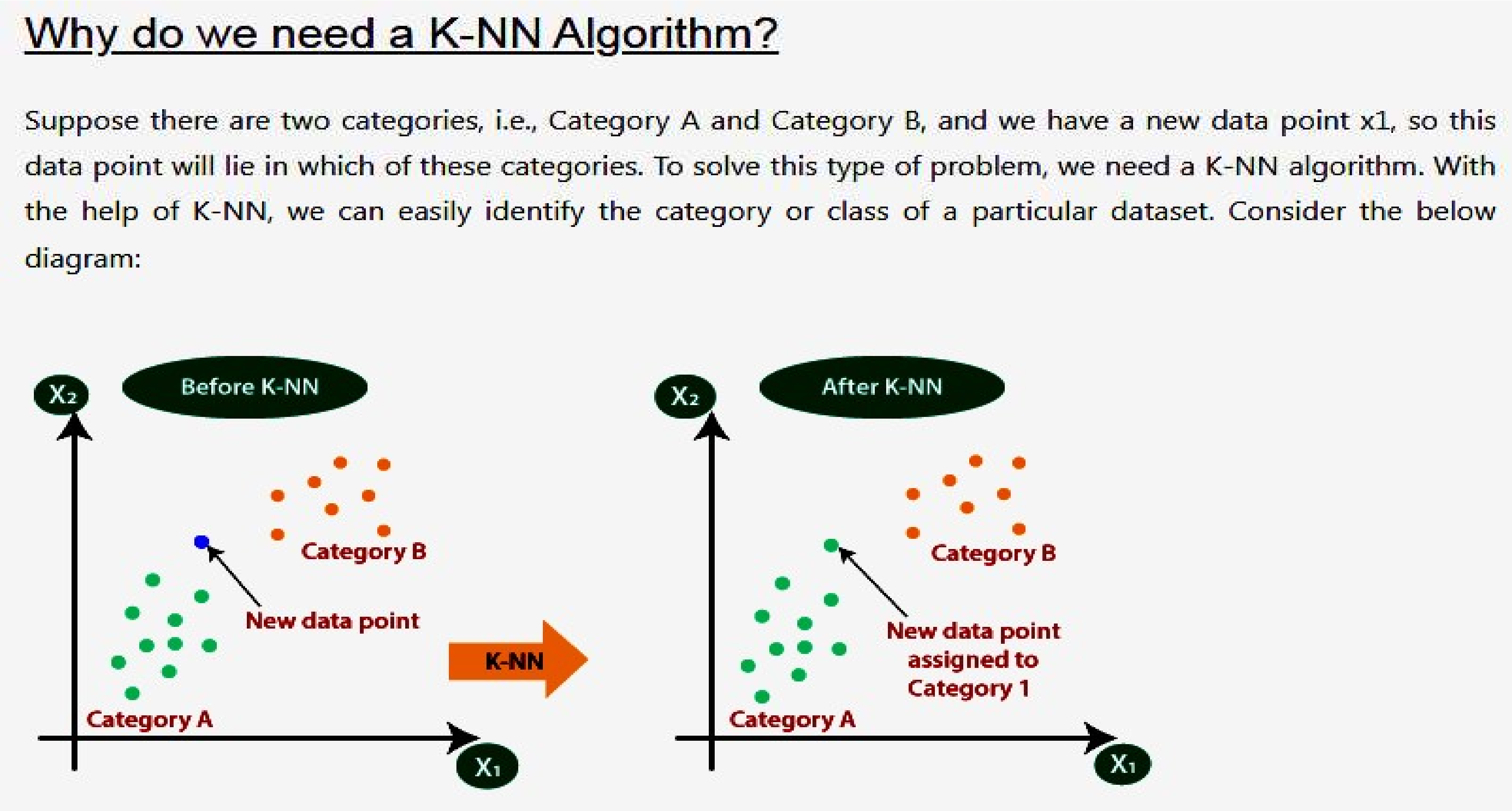
K NN demonstration

> AND WE GOT AN UNKNOWN POINT ? AND WE WANT TO PREDICT IT'S CLASS WITH KNN



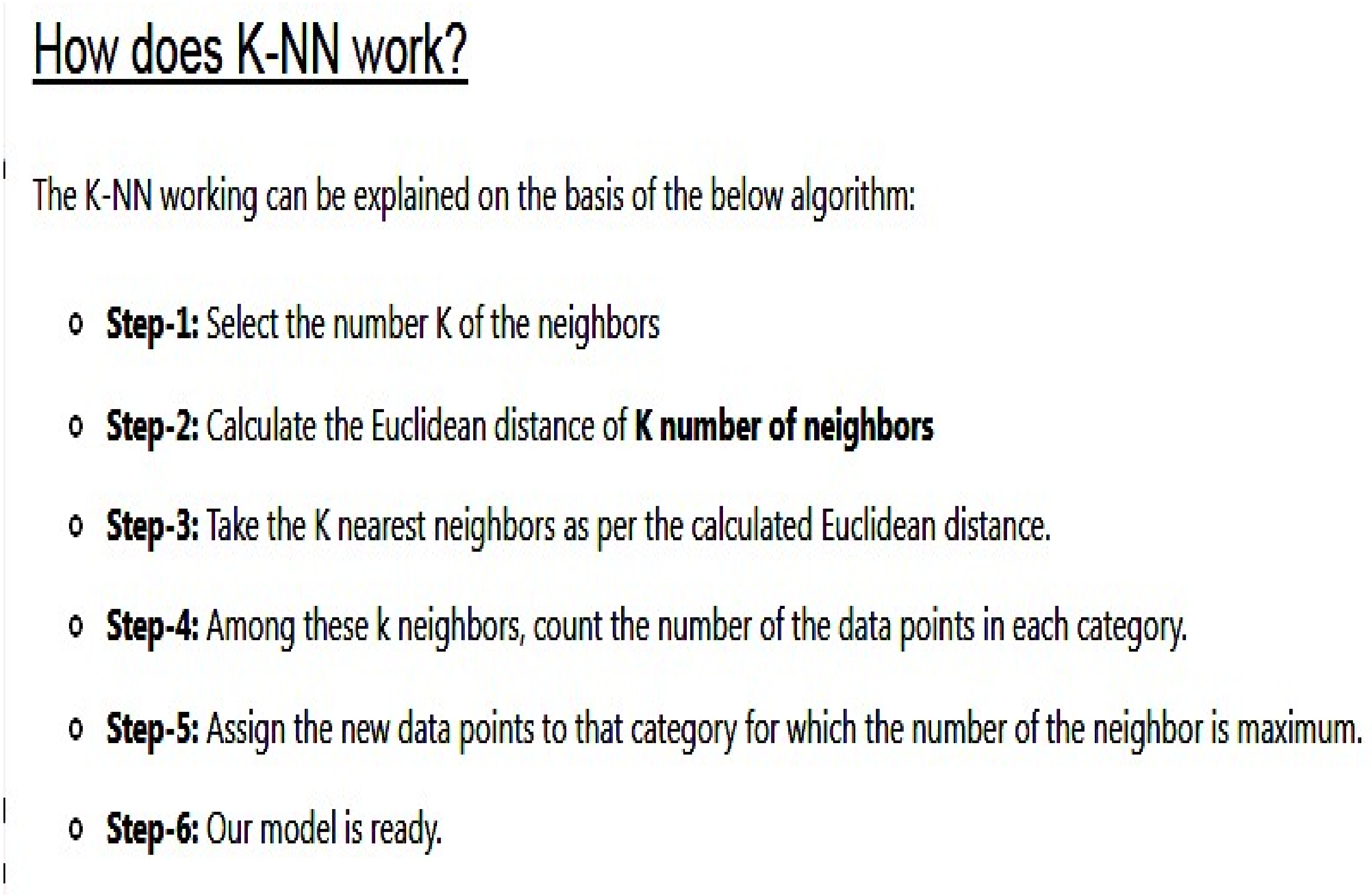
KNN Classifier

Input value
Predicted Output



Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

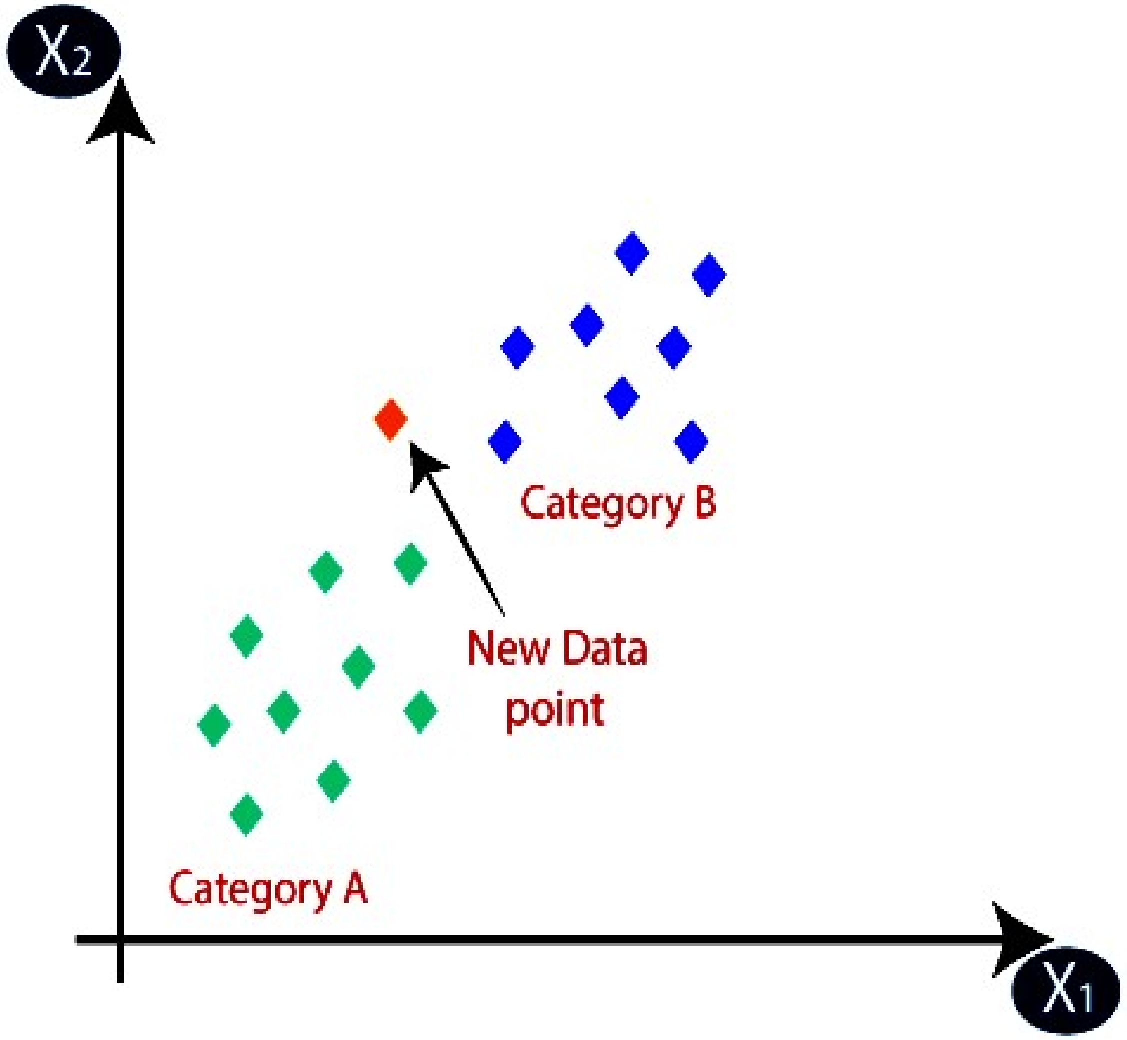


How does K-NN work?

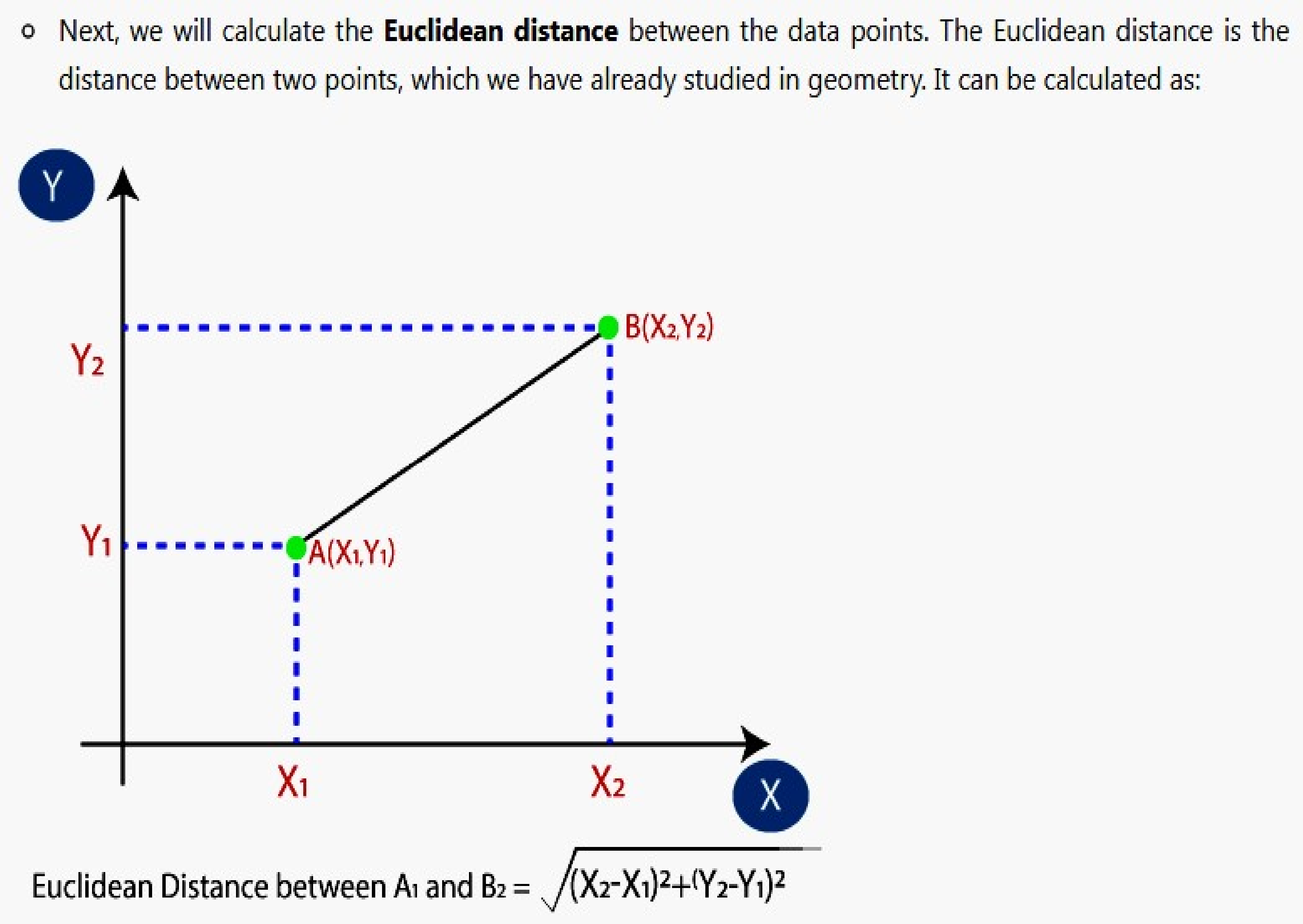
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



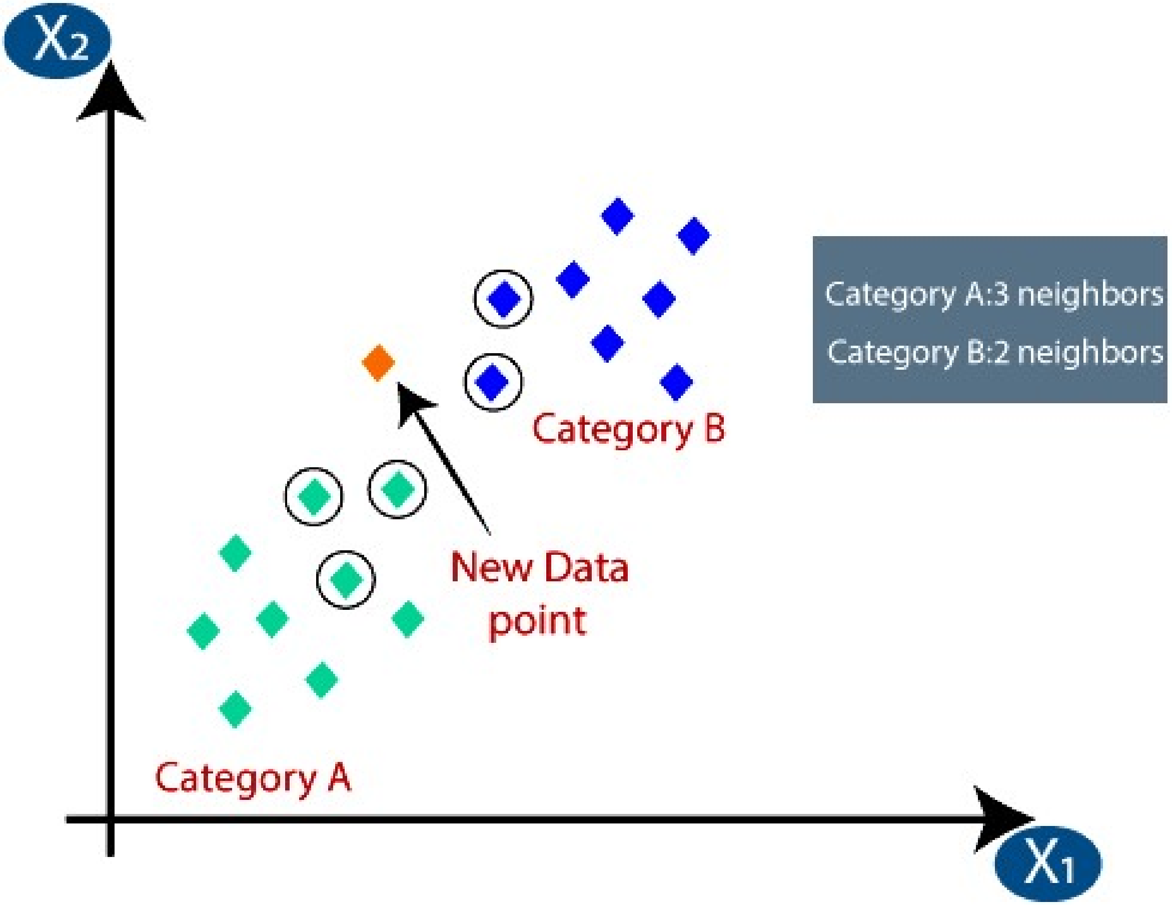
- Firstly, we will choose the number of neighbors, so we will choose the k=5.



- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

Referências - K-means

- https://realpython.com/k-means-clustering-python/
- https://www.datacamp.com/tutorial/k-means-clustering-python
- https://uc-r.github.io/kmeans_clustering
- https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/

Referências - Knn

- https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
- https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm
- https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn
- https://realpython.com/knn-python/

Notebooks Machine Learning (github)

- https://github.com/PacktPublishing/Python-Machine-Learning-Cookbook
- https://github.com/rasbt/python-machine-learning-book-2nd-edition
- https://github.com/dipanjanS/practical-machine-learning-with-python/tree/master/notebooks
- https://github.com/the-deep-learners/deep-learning-illustrated/tree/master/notebooks
- https://github.com/TrainingByPackt/Applied-Deep-Learning-with-Keras

Tutoriais Machine Learning with Python

- https://www.w3schools.com/python/default.asp
- https://www.w3schools.com/python/python_ml_getting_started.asp
- https://www.tutorialspoint.com/machine_learning_with_python/index.htm
- https://playground.tensorflow.org/
- https://matheusfacure.github.io/mltutorial/
- https://becominghuman.ai/step-by-step-neural-network-tutorial-for-beginner-cc71a04eedeb
- https://medium.com/turing-talks/turing-talks-19-modelos-de-predi%C3%A7%C3%A3o-redes-neurais-1f165583a927
- https://www.deeplearningbook.com.br/o-que-sao-redes-neurais-artificiais-profundas/
- https://lamfo-unb.github.io/2017/06/18/itro-ao-deep-learning/
- http://neuralnetworksanddeeplearning.com/chap1.html