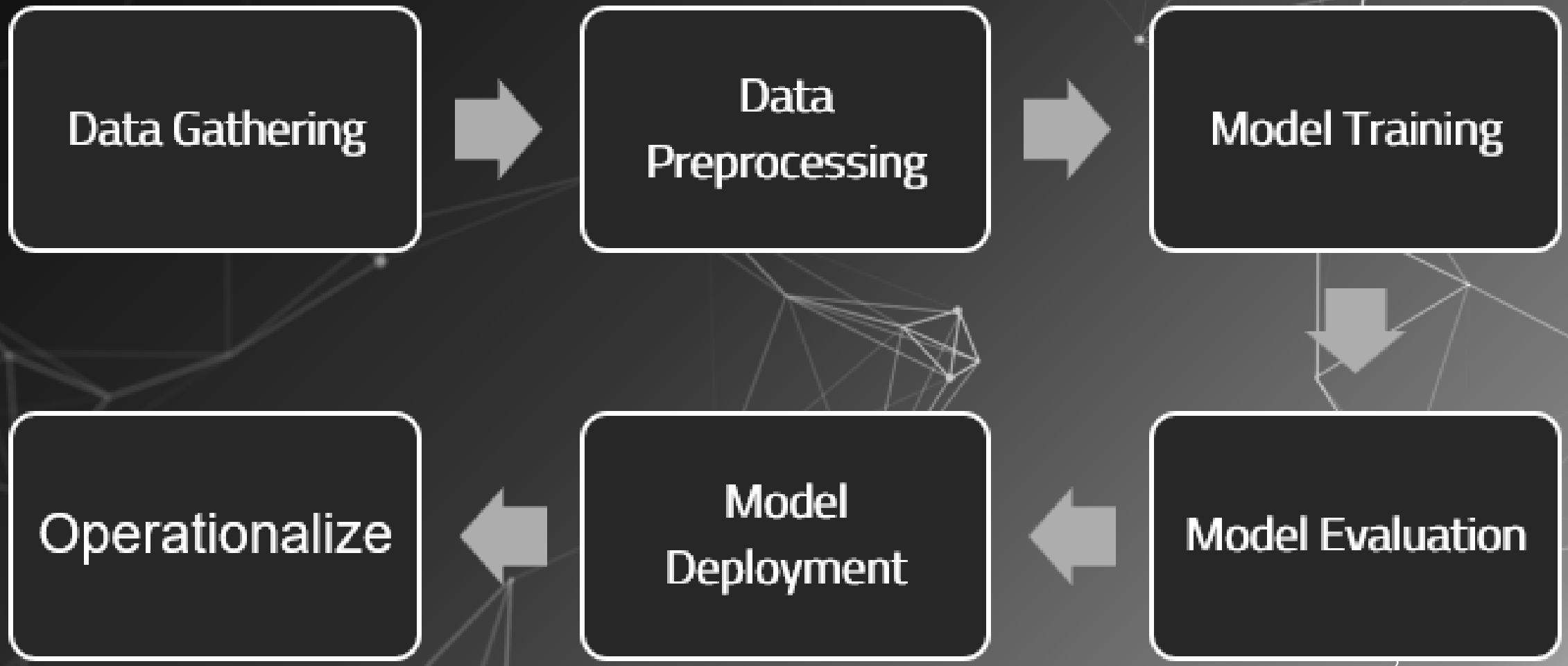
SUPERVISED MACHINE LEARNING

ENG- MOHAMED KHALED IDRIS

ENG- MAYAR SWILAM

Machine Learning Process



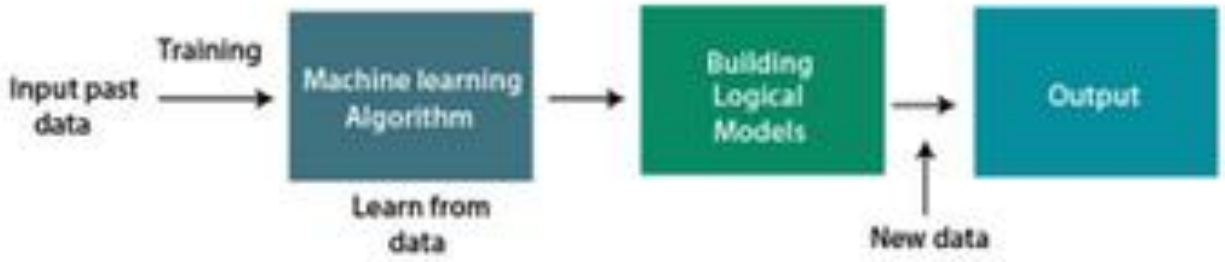
WHAT IS MACHINE LEARNING ?

- **Machine Learning is a branch of artificial intelligence that is concerned with the development of algorithms which allow a computer to learn automatically from the past data and past experiences.**
- **Machine learning builds mathematical or statistical models to make predictions using historical data or information.**

Historical data: Known as *Training data*

ALWAYS IN MACHINE LEARNING

- The accuracy of predictions depends on the **amount** of data: **More** data → **better** predictions.



HOW MACHINE LEARNS ?

Data storage utilizes observation, memory, and recall to provide a factual basis for further reasoning.

Abstraction involves transforming raw data into higher-level representations or features that capture essential information. This process often includes data preprocessing, feature engineering, and data transformation to make it suitable for modeling. Abstraction is essential for reducing noise and highlighting relevant patterns.

Generalization uses abstracted data to create knowledge and inferences that drive action in new contexts.

Evaluation provides a feedback mechanism to measure the utility of learned knowledge and inform potential improvements.

DIFFERENT TYPES OF MODELS

Mathematical equations

Statistical Models

Relational diagrams such as trees and graphs

Logical if/else rules

Groupings of data known as clusters

ABSTRACTION

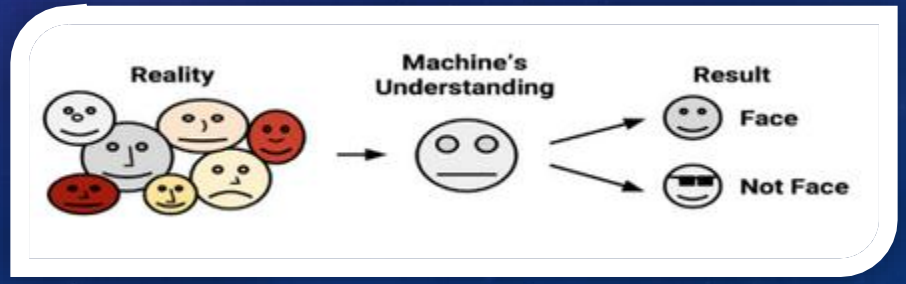
- The process of fitting a model to a dataset is known as **training**. When the model has been trained, the data is transformed into an abstract form that summarizes the original information.

Note ! You might wonder why this step is called training rather than learning.

- **First**, note that the process of learning does not end with data abstraction; the learner must still generalize and evaluate its training.
- **Second**, the word training better connotes the fact that the human teacher trains the machine student to understand the data in a specific way.

GENERALIZATION

- The term **generalization** describes the process of turning abstracted knowledge into a form that can be utilized for future action, on tasks that are similar, but not identical, to those it has seen before.
- In **generalization**, the learner is tasked with limiting the patterns it discovers to only those that will be most relevant to its future tasks.
- The algorithm is said to have a bias if the conclusions are systematically erroneous, or wrong in a predictable manner.



EVALUATION

- Therefore, the final step in the generalization process is to **evaluate or measure** the learner's success in spite of its biases and use this information to inform additional training if needed.

- Generally, evaluation occurs after a model has been trained on an initial training dataset. Then, the model is **evaluated on a new test dataset** in order to judge how well its characterization of the training data generalizes to new, **unseen data**.

EVALUATION (CONT.)

- **In parts, models fail to perfectly generalize due to the problem of noise, Noisy data is caused by seemingly random events, such as:**

- Measurement error due to imprecise sensors that sometimes add or subtract a bit from the readings.
- Issues with human subjects, such as survey respondents reporting random answers to survey questions, in order to finish more quickly.
- Data quality problems, including missing, null, truncated, incorrectly coded, or corrupted values.
- Phenomena that are so complex or so little understood that they impact the data in ways that appear to be unsystematic.

Trying to model noise is the basis of a problem called OVERFITTING !

UNDERFITTING VS. OVERFITTING VS. GOOD FITTING

TRAINING DATASET

- The **training set** is the portion of the dataset that is used to train the machine learning model. It contains labeled examples (input data and corresponding target labels) that the model uses to learn the patterns and relationships in the data.

- The goal is for the model to generalize from the training data and capture the underlying patterns so that it can make accurate predictions on new, unseen data.

VALIDATION DATASET

- The **validation set** is used during the training process to fine-tune the model's hyperparameters and monitor its performance. It is separate from the training set and contains examples that the model has not seen during training.

- The validation set allows you to assess the model's performance on unseen data and make decisions about parameter adjustments, such as adjusting the learning rate or choosing the number of hidden units in a neural network

- By evaluating the model's performance on the validation set, you can make informed decisions to improve the model's generalization capabilities.

TEST DATASET

- The **test set** is used to evaluate the final performance of the trained machine learning model. It is a separate, independent dataset that the model has never seen before during training or validation.

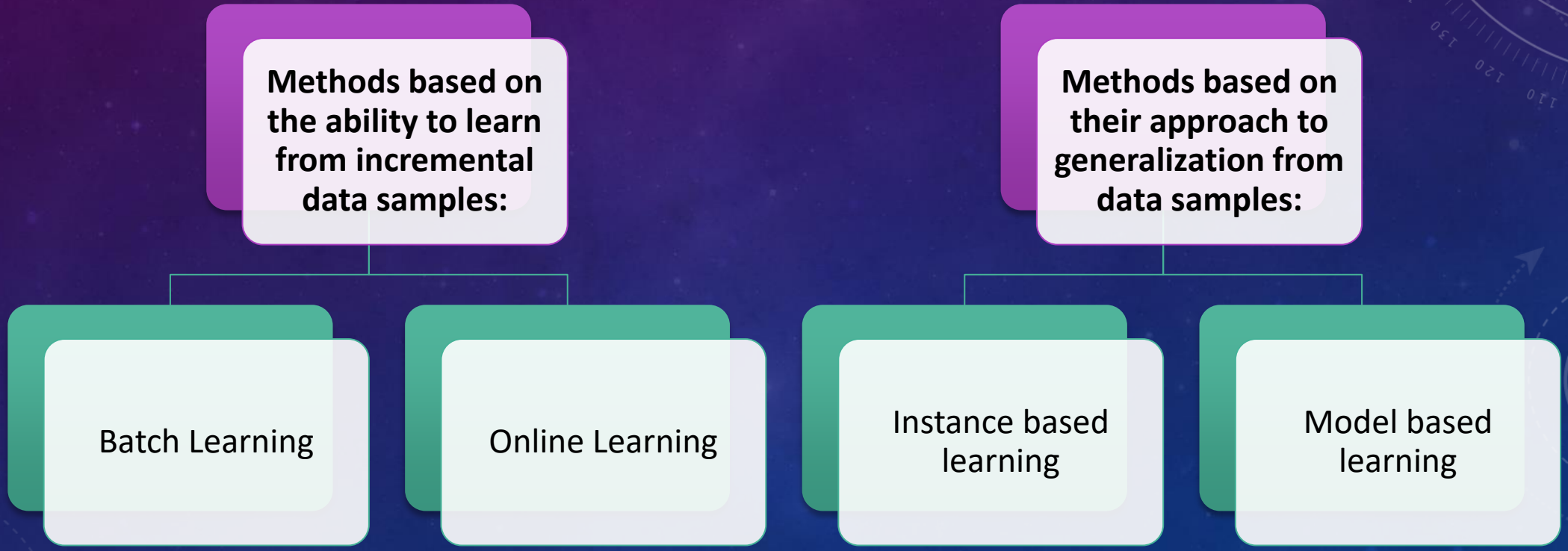
- The purpose of the test set is to provide an unbiased assessment of the model's performance on completely new data.

- By evaluating the model on a test set, you can estimate how well the model is likely to perform in real-world scenarios.

TRAINING VS. VALIDATION VS. TEST

- Training Set: Used for training the model by adjusting its parameters based on labeled examples.
- Validation Set: Used to fine-tune hyperparameters and monitor performance during training.
- Test Set: Used to assess the final performance and generalization capabilities of the trained model.

ADDITIONAL MACHINE LEARNING METHODS



SUPERVISED LEARNING (CLASSIFICATION)

WHAT IS CLASSIFICATION ?

- **Classification** is a fundamental task in machine learning and data analysis. It involves categorizing data instances into predefined classes or categories based on their features or attributes.

- The goal of classification is to build a predictive model that can assign the correct class label to new, unseen data based on patterns and relationships learned from a labeled dataset.

In a classification problem, you have a dataset containing examples with their corresponding class labels.

TYPES OF CLASSIFICATION

Binary Classification

Multi Classification

NEAREST NEIGHBOR

NEAREST NEIGHBOR

- **Neighbors-based classification is a type of *instance-based learning* or *non-generalizing learning*: it does not attempt to construct a general internal model, but simply stores instances of the training data.**

- **Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.neighbours**

WHAT IS K-NEAREST NEIGHBORS (KNN) ?

- **K-Nearest Neighbors (KNN)** is a simple yet powerful machine learning algorithm used for both classification and regression tasks.

- It is a non-parametric and instance-based learning algorithm, meaning it makes predictions based on the similarity between input data points and the data points in its training set.

Sure! Imagine you have a pet dog, and you want to figure out if your dog is more like a "friendly" dog or a "shy" dog. You have a bunch of pictures of other dogs that are labeled as either "friendly" or "shy." You want to use these pictures to decide if your dog is more like the "friendly" dogs or the "shy" dogs.

STORYTELLING OF GPT

K-Nearest Neighbors, or KNN for short, is a bit like asking your friends who have dogs for their opinions. Here's how it works:

1. You show your friends the picture of your dog, and they compare it to pictures of their own dogs. They look at the dogs that are most similar to yours.
2. Let's say you ask three friends (k = 3), and two of them say their dogs are "friendly" while one says their dog is "shy."
3. Since more of your friends have "friendly" dogs, you might guess that your dog is more likely to be "friendly" as well.

STORYTELLING OF GPT (CONT.)

In KNN, the "k" represents the number of friends you ask. The algorithm looks at the features of the dogs (like size, color, and fur type) and compares them to decide if your dog is more like the "friendly" dogs or the "shy" dogs. The decision is based on what the majority of your friends' dogs are like.

So, KNN helps you make a guess about whether your dog is more like "friendly" dogs or "shy" dogs by asking its closest neighbors (other dogs) for their opinions. It's like asking your friends for advice about your dog's behavior based on what their dogs are like.

STORYTELLING OF GPT (CONT.)

HOW IT WORKS ?

Training Phase:

- The algorithm is provided with a labeled training dataset, where each data point consists of features (attributes) and a corresponding class label (for classification) or a target value (for regression).

- KNN does not explicitly "train" a model like many other algorithms. Instead, it memorizes the entire training dataset.

HOW IT WORKS ? (CONT.)

Prediction Phase:

- When a new, unlabeled data point needs to be classified or predicted, KNN identifies the "k" closest data points (neighbors) from the training dataset based on a **distance metric**, often using Euclidean distance.

- The value of "k" is a user-defined parameter that determines how many neighbors influence the prediction. For example, if k = 3, the algorithm considers the labels or values of the three closest neighbors.

It has no loss function !

HOW IT WORKS ? (CONT.)

Classification:

- For classification tasks, the algorithm assigns the class label that is most common among the k nearest neighbors. This is often done using majority voting. For instance, if out of the k neighbors, 2 belong to class A and 1 belongs to class B, the algorithm would predict class A for the new data point.

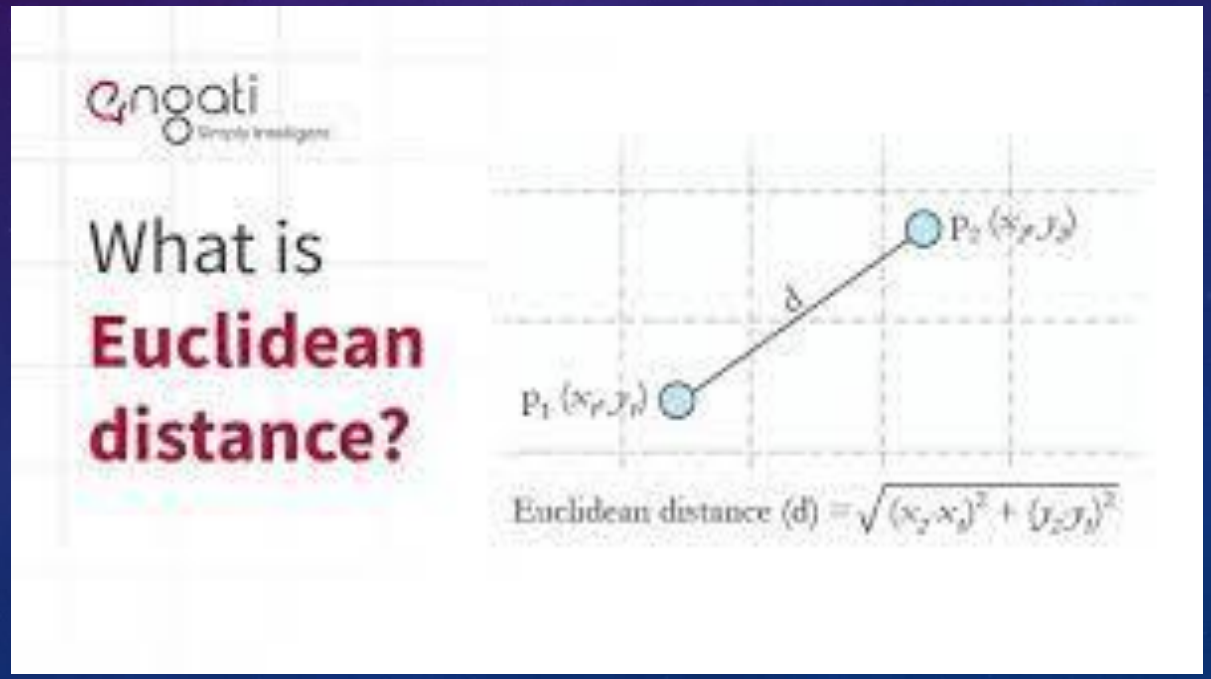
In other words, it is the using of MODE

SIMILARITY METRICS FOR KNN

Euclidean Distance	**Manhattan Distance**	**Chebyshev Distance**	**Minkowski Distance**
Hamming Distance	**L1 Norm**	**L2 Norm**	**Jaccard Similarity**

EUCLIDEAN DISTANCE (L2 NORM)

- This is the most common distance metric and is used for continuous numerical data.
- It calculates the straight-line distance between two points in the feature space. If you have two points with coordinates (x1, y1) and (x2, y2), the Euclidean distance is:



MANHATTAN DISTANCE (L1 NORM)

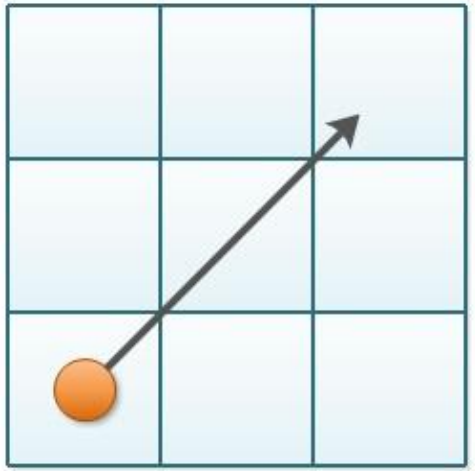
- Also known as the "city block" or "taxicab" distance, this metric calculates the distance by summing the absolute differences between corresponding coordinates.

- It's often used when movement can only occur along grid lines (like streets in a city).

CHEBYSHEV DISTANCE

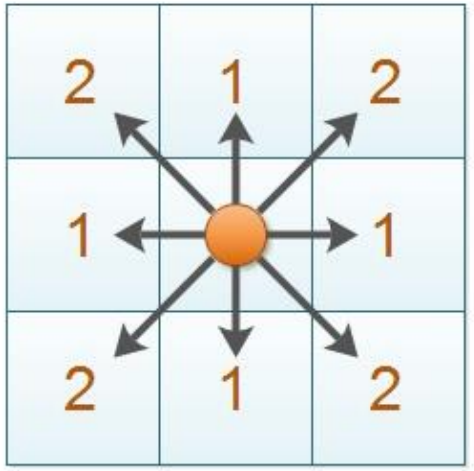
- This metric calculates the maximum absolute difference between coordinates.
- It's useful when you want to emphasize the largest difference between dimensions.

Euclidean Distance



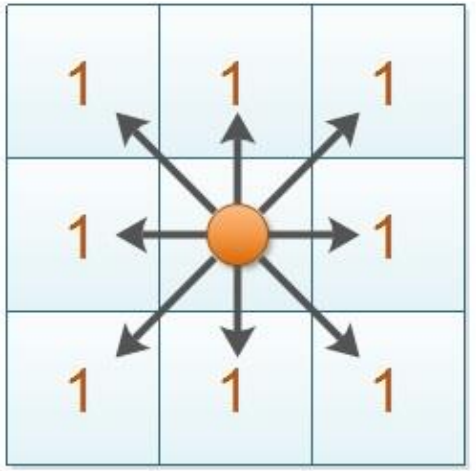
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan Distance



$$|x_1 - x_2| + |y_1 - y_2|$$

Chebyshev Distance



$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

MINKOWSKY DISTANCE

- This is a generalization of both **Euclidean and Manhattan distances**. It includes a parameter "p" that allows you to control the distance calculation.

- When p = 1, it's equivalent to the Manhattan distance, and when p = 2, it's the Euclidean distance.

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

PROS & CONS.

- KNN is simple to understand and implement.
- It is a lazy learner, meaning it doesn't perform explicit training and retains the entire training dataset.
- It can handle multi-class classification and can be adapted for multi-label classification as well.
- KNN's performance can be sensitive to the choice of distance metric, data scaling, and the value of "k."
- It works well when the decision boundaries are irregular and when the dataset is relatively small.
- KNN can be computationally expensive, especially for large datasets, as it requires calculating distances for all data points.

DECISION TREE

A SEQUENCE OF DECISIONS

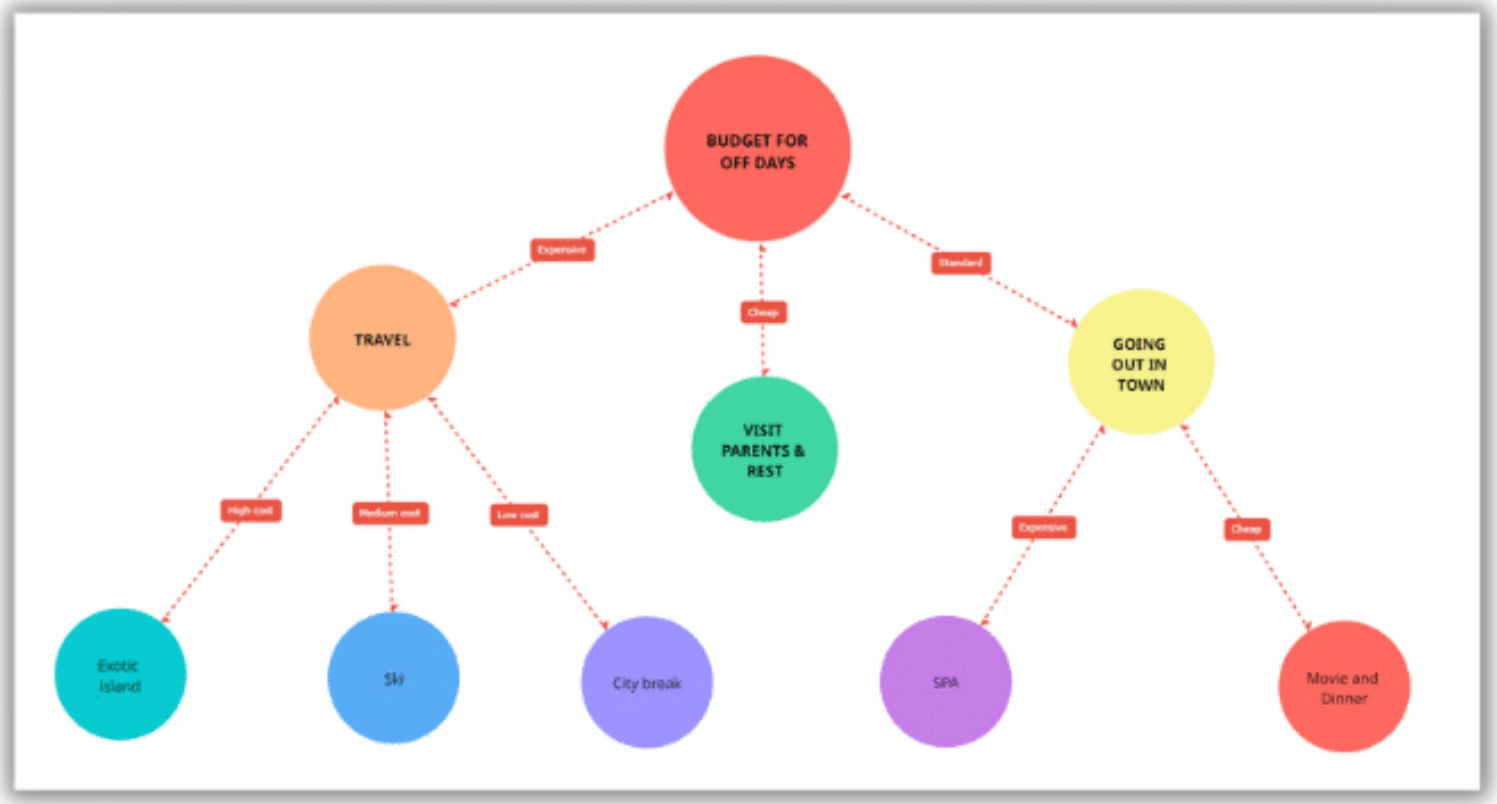
In the summer days, do you have any criteria to get up from your bed and get out into the hot weather for important mission ?

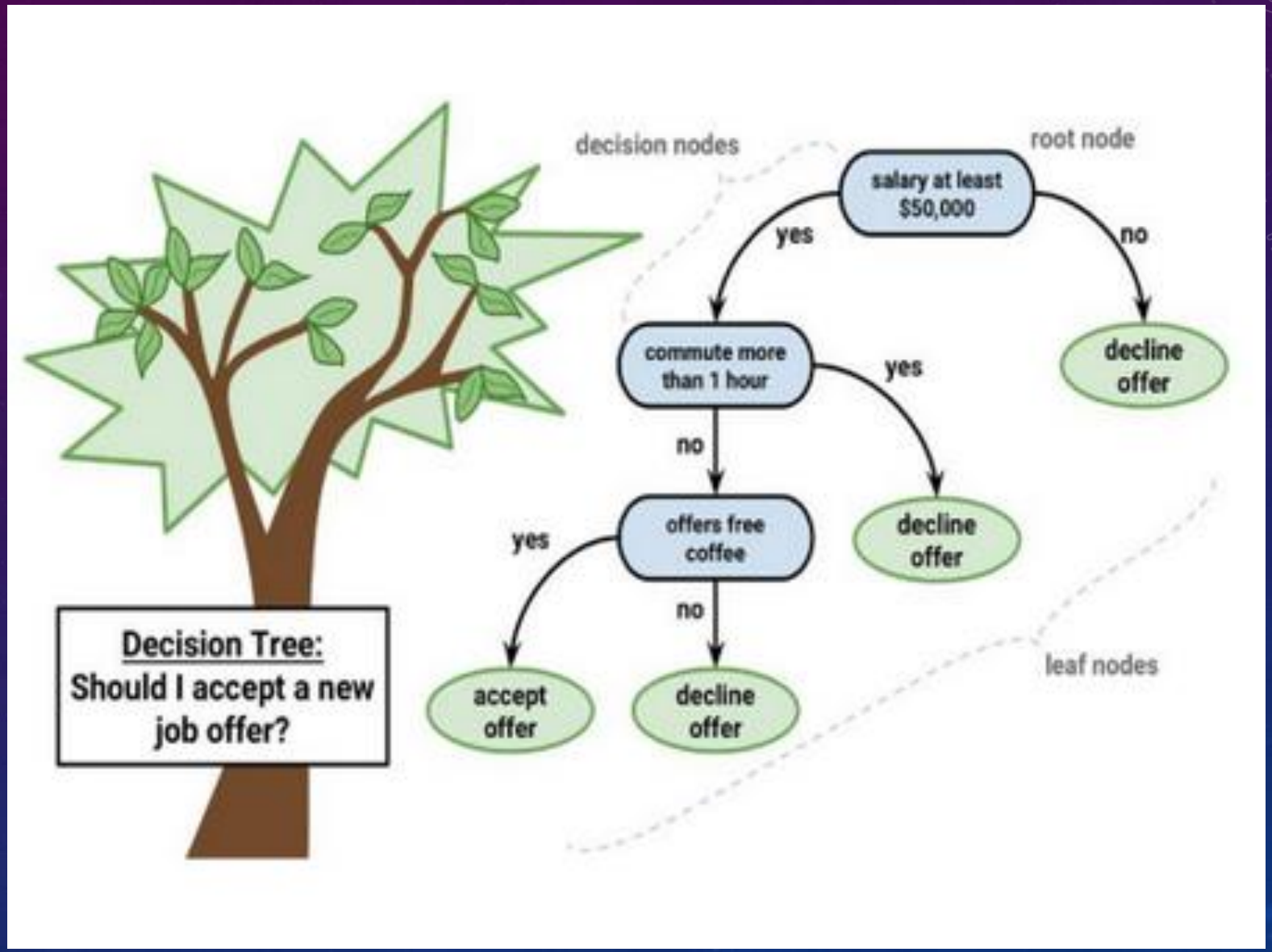
Is there a truly something important to make you go out in that hot weather ?

You should re-consider you Decisions based on your priorities. You can use such a thing that makes logic rules that help you make a decision such as:

Decision Tree

Decision tree examples




Decision Tree:
Should I accept a new job offer?
decision nodes
root node
salary at least $50,000
yes
no
commute more than 1 hour
yes
decline offer
no
offers free coffee
yes
decline offer
yes
no
accept offer
decline offer
leaf nodes

```
public static string DetermineGender(int input) {
  string gender = string.Empty;

  if (input == 0) {
    gender = "male";
  } else if (input == 1) {
    gender = "woman";
  } else {
    gender = "unknown";
  }

  return gender;
}
```

SOMETHING YOU MAY BE FAMILIAR WITH (GROUP OF IF-ELSEIF STATEMENTS)

WHAT IS “DECISION TREE” ?

- **Decision tree** is one of the most powerful tools of **supervised learning** algorithms used for both classification and regression tasks in machine learning.
- It builds a **flowchart-like tree structure** where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

- It is constructed by recursively splitting the training data into subsets based on the values of the attributes until a stopping criterion is met, such as the maximum depth of the tree or the minimum number of samples required to split a node.

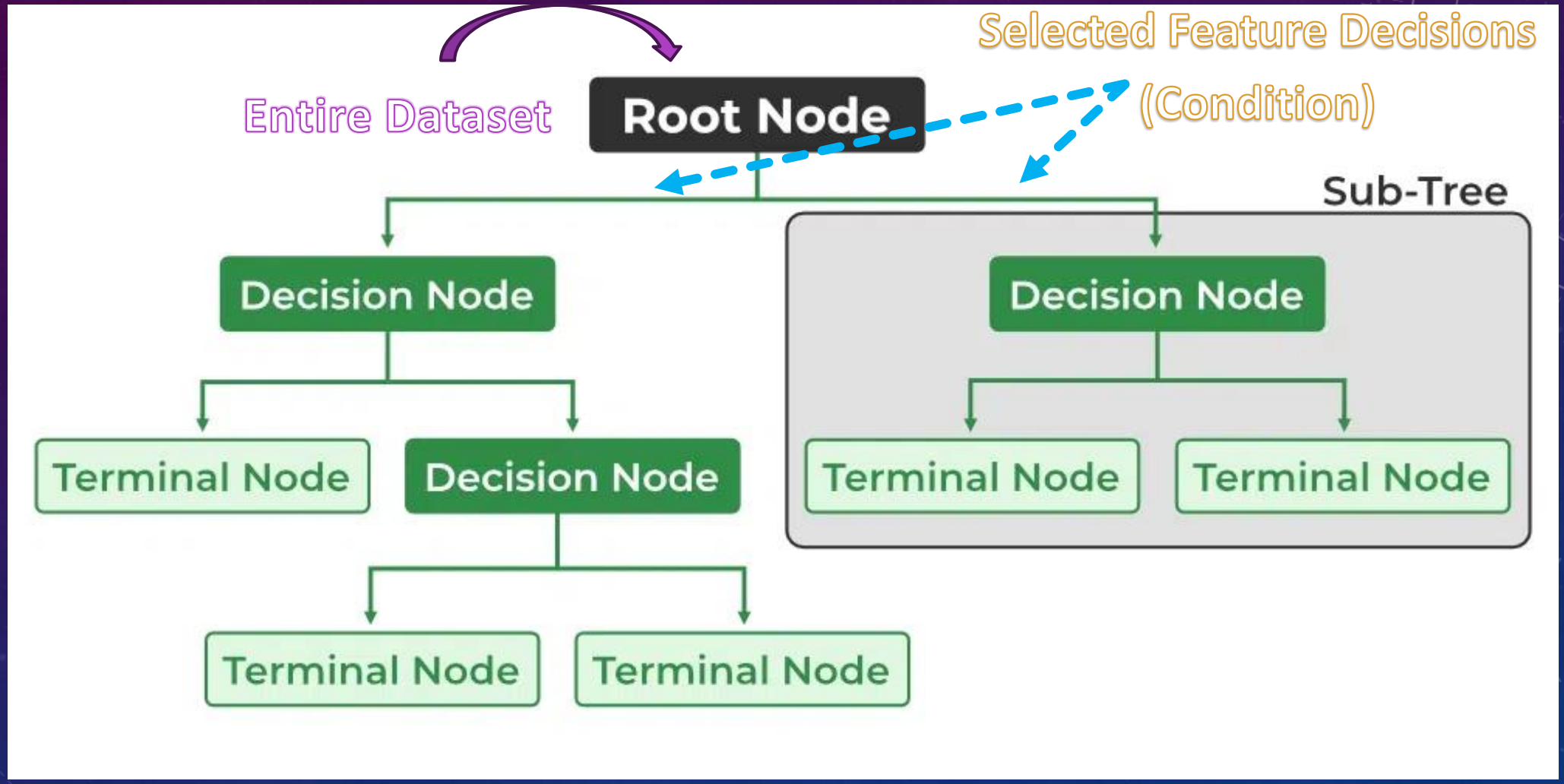
WHY DECISION TREE ?

- A decision tree simply asks a question and based on the answer (Yes/No), it further split the tree into sub-trees.

Before going to Decision Tree You should know some TERMINOLOGIES

DECISION TREE TERMINOLOGIES

Terminology	Description
Root Node	Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
Decision/Internal Node	A node that symbolizes a choice regarding an input feature. Branching off of internal nodes connects them to leaf nodes or other internal nodes.
Leaf/Terminal Node	A node without any child nodes that indicates a class label or a numerical value.
Splitting	The process of splitting a node into two or more sub-nodes **using a split criterion** and a selected feature.
Branch/Sub-Tree	A subsection of the decision tree starts at an internal node and ends at the leaf nodes
Pruning	The process of removing branches from the tree that do not provide any additional information or lead to overfitting **(Optimization)**


Entire Dataset
Root Node
Selected Feature Decisions
(Condition)
Sub-Tree
Decision Node
Decision Node
Terminal Node
Decision Node
Terminal Node
Terminal Node
Terminal Node
Terminal Node

SOME NOTES YOU SHOULD REMEMBER !

Decision tree is not unique, as different ordering of internal nodes can give different decision tree.

DECISION TREE MAY BE N-ARY, N ≥ 2.

ALL NODES DRAWN WITH **CIRCLE (ELLIPSE)** ARE CALLED INTERNAL NODES.

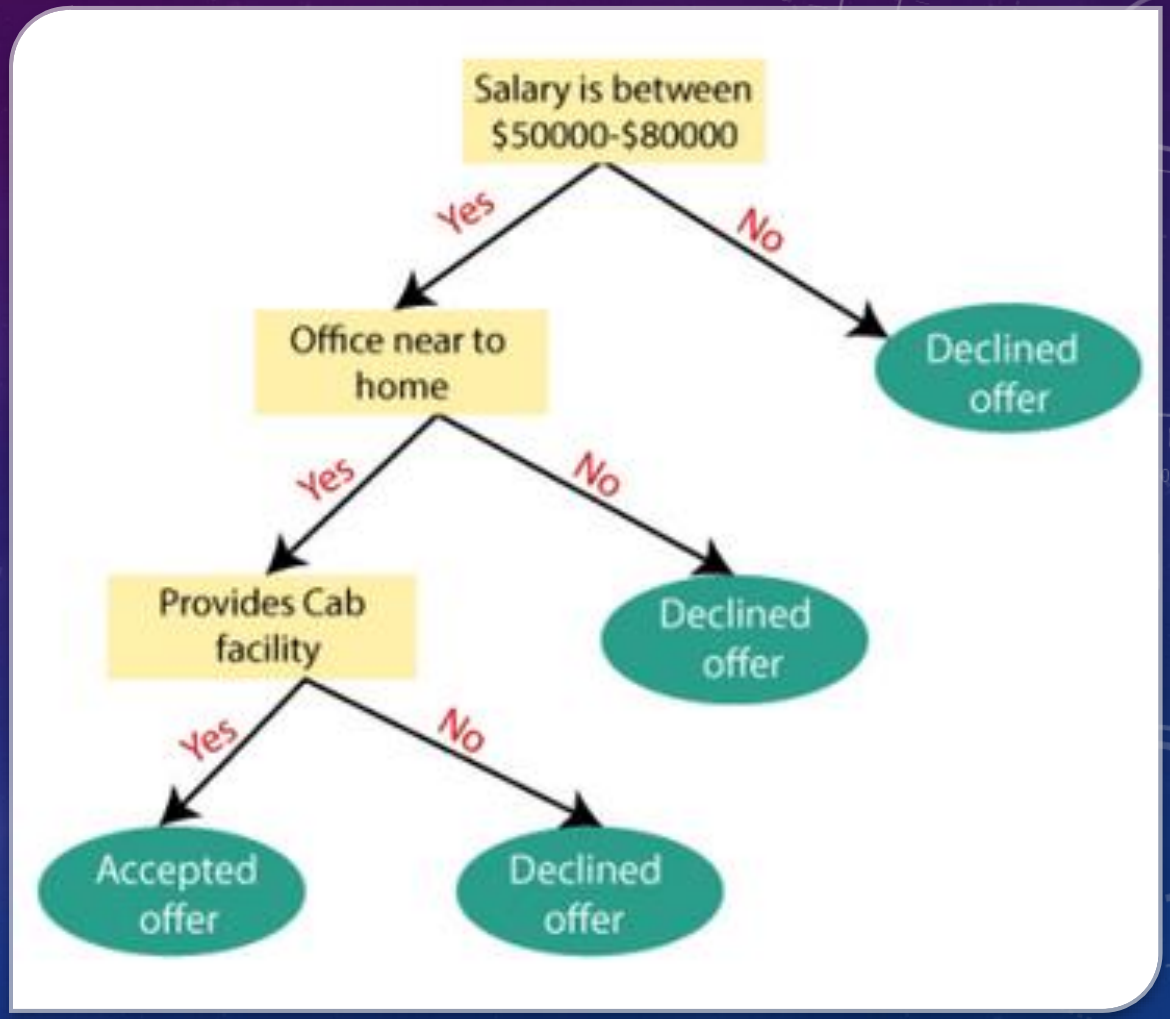
ALL NODES DRAWN WITH **SQUARE BOXES** ARE CALLED TERMINAL NODES OR LEAF NODES.

EDGES OF A NODE REPRESENT **THE OUTCOME FOR A VALUE OF THE NODE.**

IN A PATH, A NODE WITH SAME LABEL IS **NEVER REPEATED**.

EXAMPLE

- Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by **Attribute Selection Measure –ASM-**).



Advantages of the Decision Tree:

1. It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
2. It can be very useful for solving decision-related problems.
3. It helps to think about all the possible outcomes for a problem.
4. There is less requirement of data cleaning compared to other algorithms.

DECISION TREE ADVANTAGES

Disadvantages of the Decision Tree:

1. The decision tree contains lots of layers, which makes it complex.
2. It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
3. For more class labels, the computational complexity of the decision tree may increase.

DECISION TREE DISADVANTAGES

SUPPORT-VECTOR MACHINE

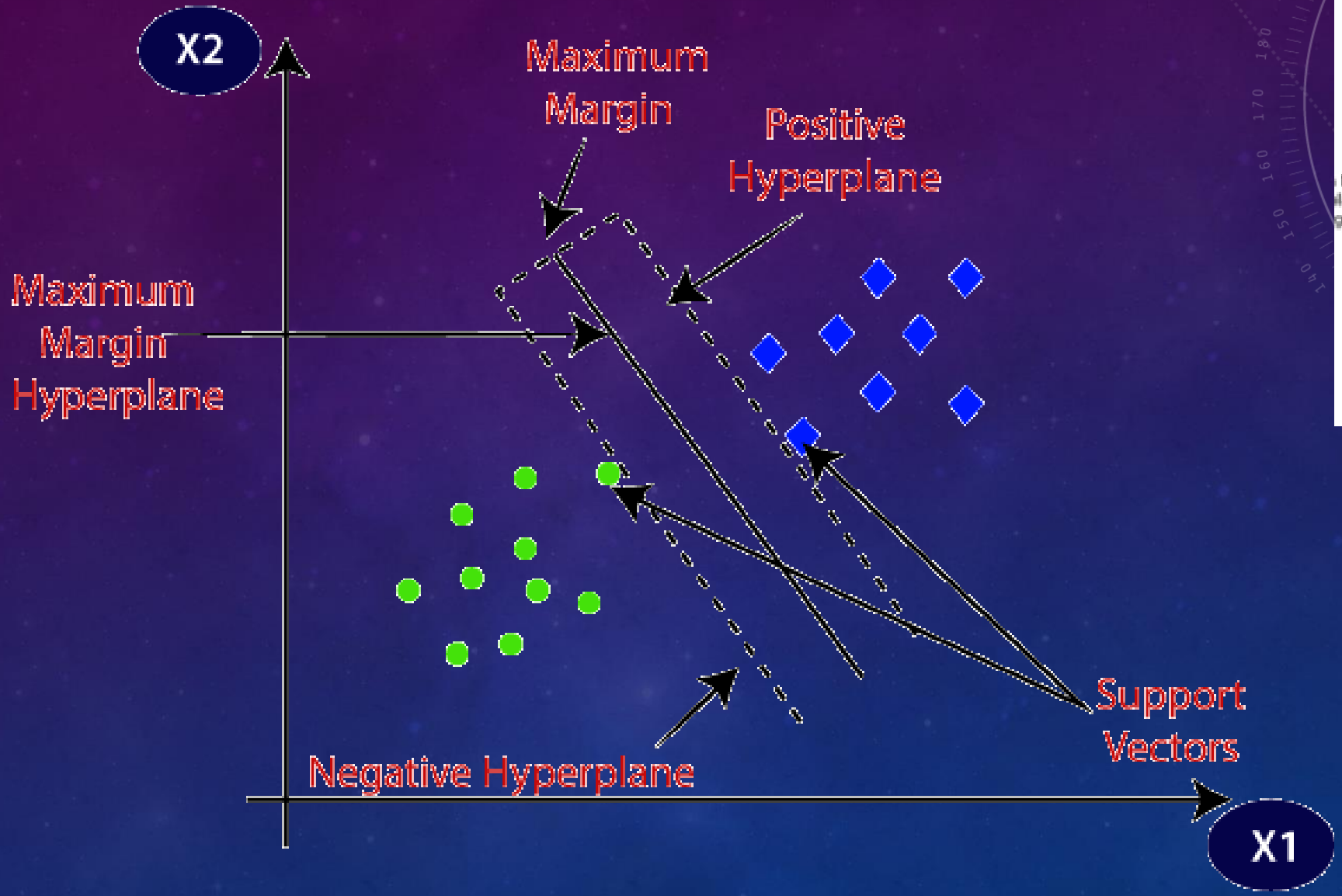
WHAT IS SUPPORT-VECTOR MACHINE ?

- A **Support Vector Machine (SVM)** is a powerful and versatile supervised machine learning algorithm used for both classification and regression tasks.

- SVMs are particularly effective in scenarios where there is a clear separation between classes or when the data is not linearly separable.

What is the main goal ?

- The main goal of an SVM is to find a hyperplane (or decision boundary) that best separates the data into different classes while maximizing the margin between the classes.


X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

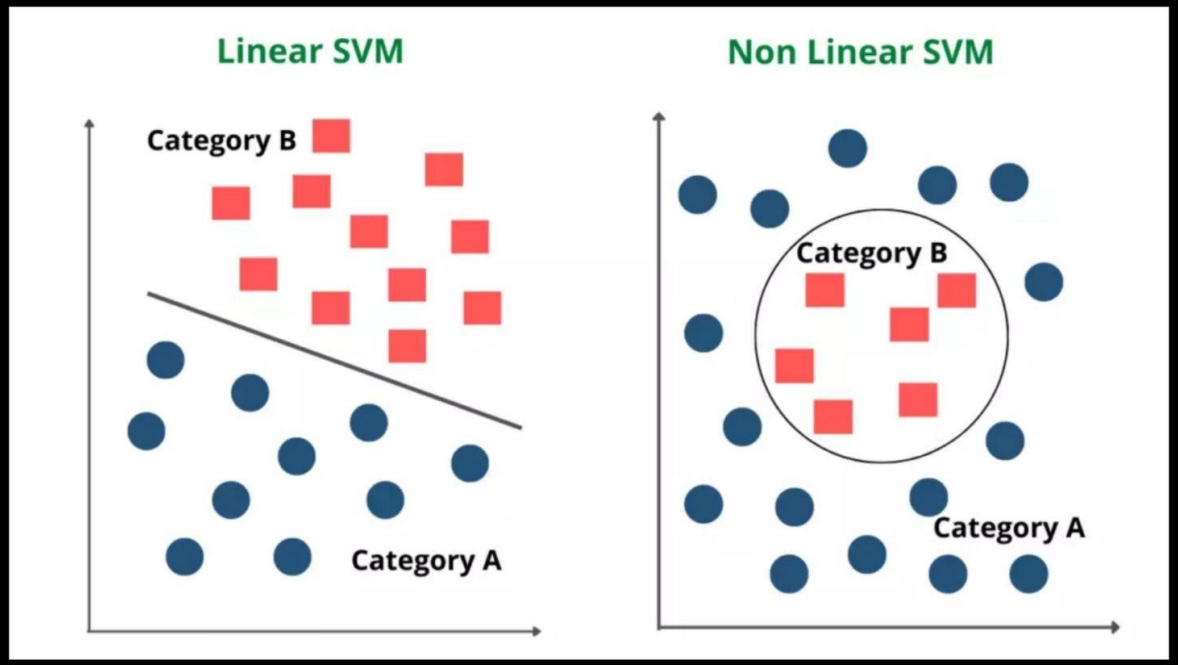
IMPORTANT TERMS

- **Hyperplane:** A hyperplane is a decision boundary that separates data points of different classes. In a binary classification problem, it's a line in two dimensions or a hyperplane in higher dimensions.
- **Support Vectors:** These are the data points that are closest to the hyperplane and have the most influence in determining its position. Support vectors define the margin and play a crucial role in SVM.
- **Margin:** The margin is the distance between the hyperplane and the nearest data points (support vectors) of each class. SVM aims to maximize this margin.
- **Kernel:** A kernel function is used to transform the data into a higher-dimensional space, allowing SVM to find a hyperplane in a transformed feature space.

TYPES OF SVM

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

SVM in ML


Linear SVM
Category B
Category A
Non Linear SVM
Category B
Category A

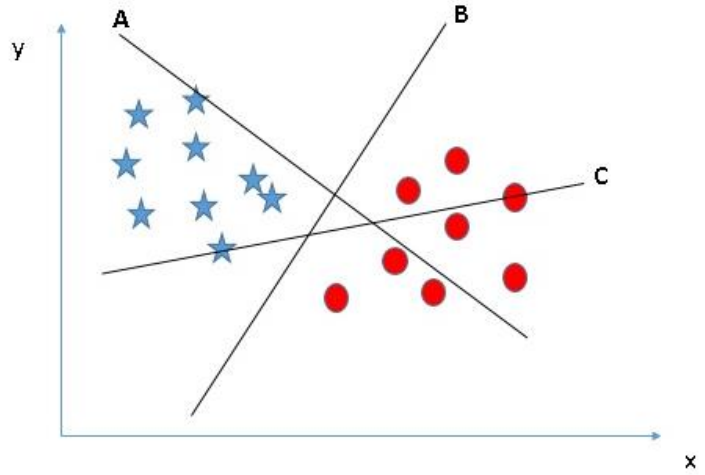
EXISTENCE OF THE KERNELS (HYPERPLANE)

The mathematical function used for the transformation is known as the ***kernel function.*** Following are the popular functions.

- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid

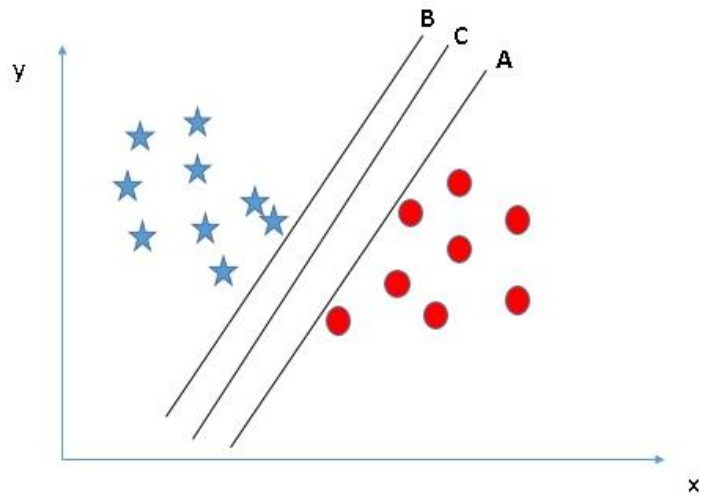
WHY "MARGIN" ?

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.



WHY "MARGIN" ? (CONT.)

- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B, and C), and all segregate the classes well. Now, How can we identify the right hyper-plane?

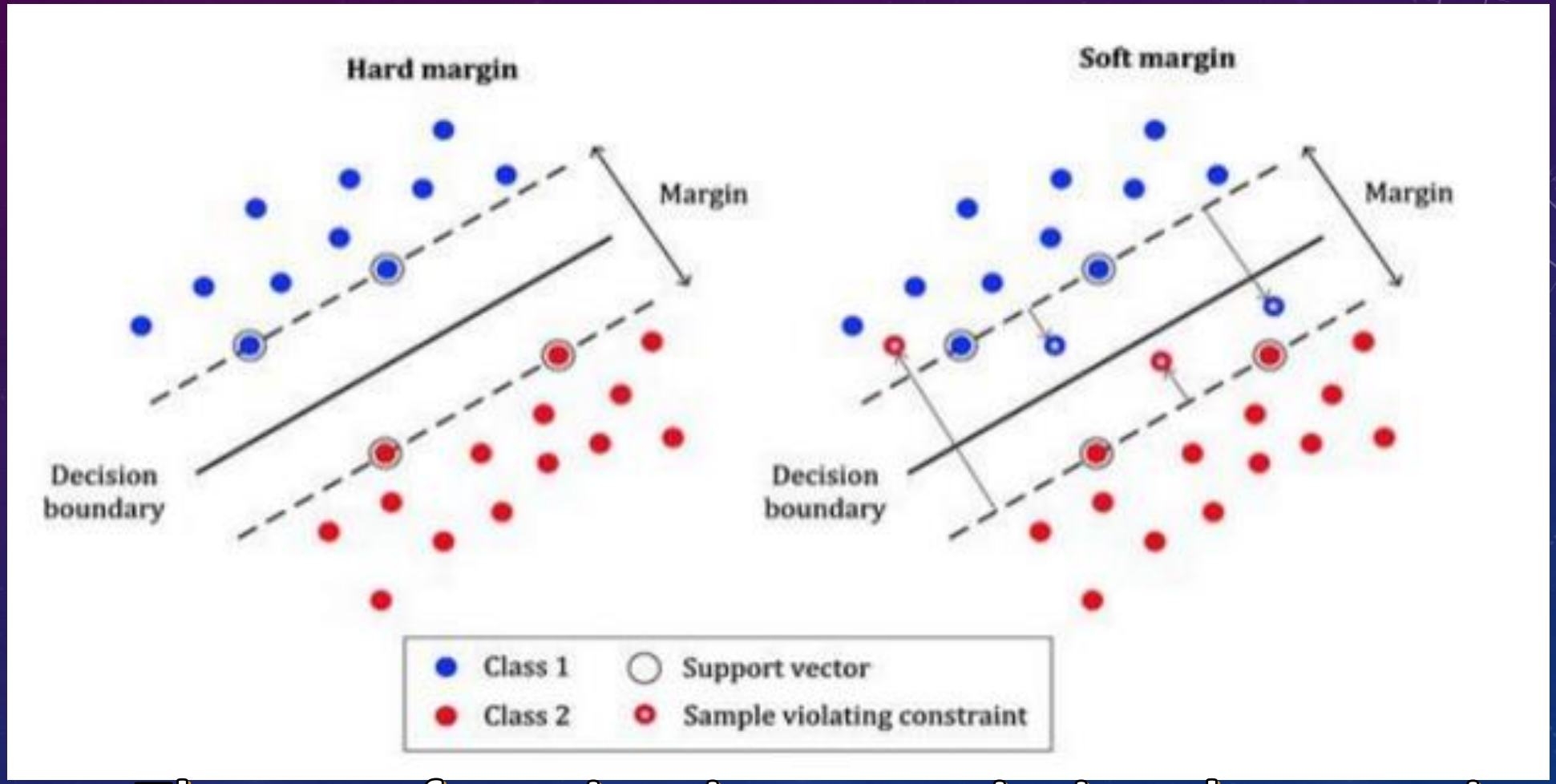


THE LEARNING PROCESS

- The distance of the vectors from the hyperplane is called the margin which is a separation of a line to the closest class points.

- We would like to choose a hyperplane that **maximizes the margin** between classes.

  - Soft Margin
  - Hard Margin



The cost function is to maximize the margin

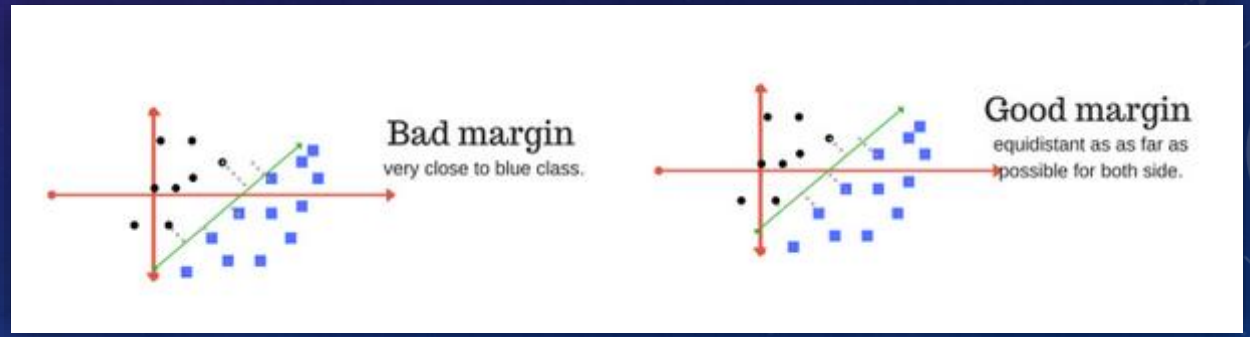
1- Soft Margin – As most of the real-world data are not fully linearly separable, we will allow some margin violation to occur which is called soft margin classification. It is better to have a large margin, even though some constraints are violated. Margin violation means choosing a hyperplane, which can allow some data points to stay on either the incorrect side of the hyperplane and between the margin and correct side of the hyperplane.

2- Hard Margin – If the training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible.

SOFT VS. HARD MARGIN

GOOD FITTING

- A good margin is one where this separation is larger for both the classes. Images below gives to visual example of good and bad margin.



ADVANTAGES

- Effective in high-dimensional spaces and with small to medium-sized datasets.
- Works well in cases where classes are not linearly separable through the use of kernel functions.
- Tends to generalize well and avoid overfitting, especially when the regularization parameter C is properly tuned.

LIMITATIONS

- SVMs can be computationally intensive, especially for large datasets.
- Choosing the appropriate kernel and hyperparameters requires careful tuning and experimentation.
- Interpreting the model's decisions can be challenging, especially in high-dimensional spaces.
- SVMs may not perform well when dealing with noisy data or data with a high degree of overlap between classes.

EVALUATION METRICS

EVALUATION METRICS

Each Machine Learning Problem has it is own Evaluation Metrics

CLASSIFICATION EVALUATION METRICS

- **Evaluation metrics** for classification are used to assess the performance of a classification model by comparing its predictions to the actual class labels in a dataset.

- These metrics help quantify how well the model is performing and provide insights into its strengths and weaknesses.

Accuracy	Precision	Recall	F1-Score	ROC	AUC

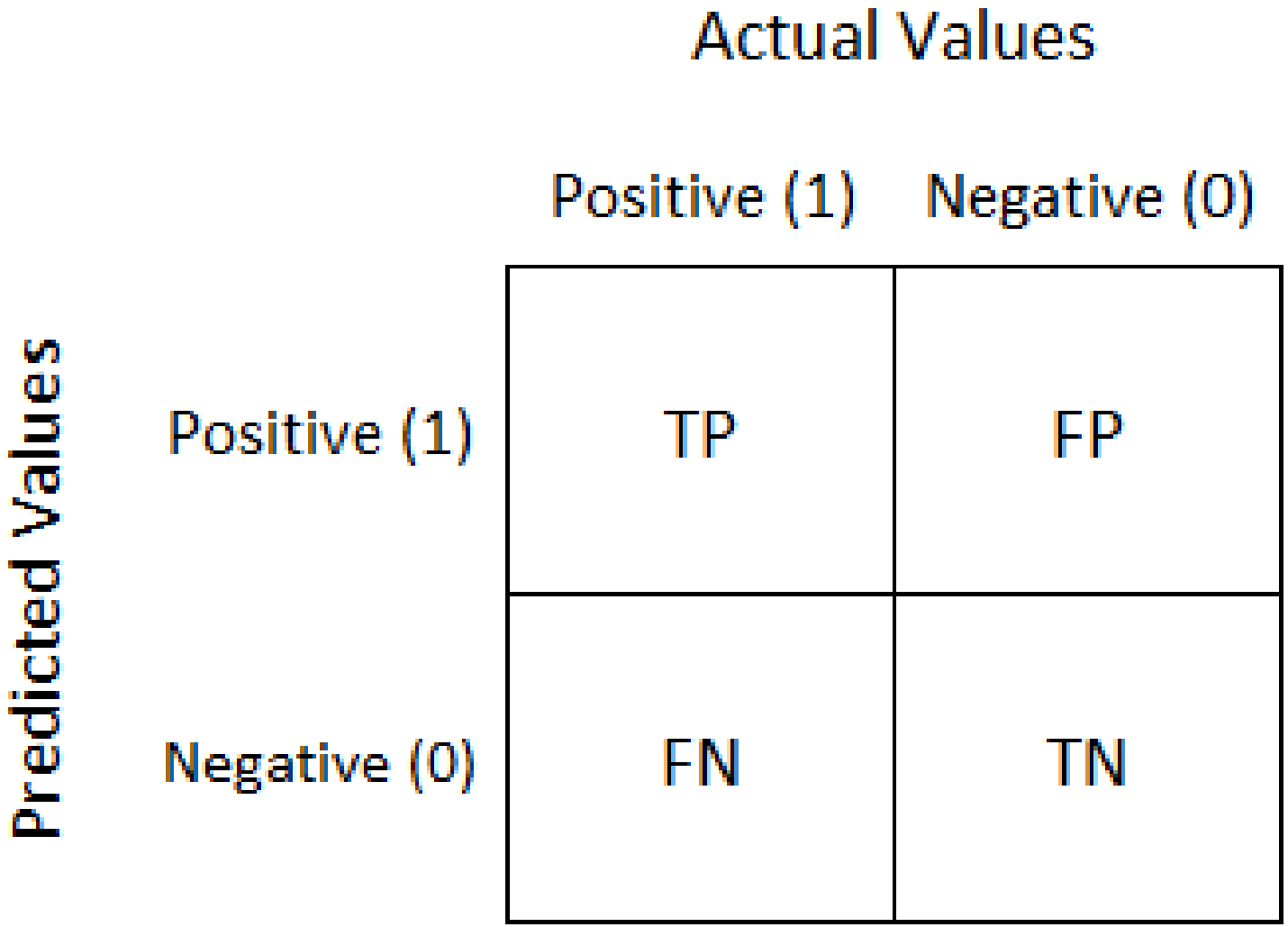
ACCURACY

Accuracy: Accuracy measures the proportion of correctly predicted instances out of the total instances in the dataset. It's a simple and intuitive metric but may not be suitable for imbalanced datasets.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Confusion Matrix: A confusion matrix provides a detailed breakdown of true positive, true negative, false positive, and false negative predictions. It's a useful tool for understanding the distribution of prediction errors.

CONFUSION MATRIX


Actual Values
Positive (1)
Negative (0)
Predicted Values
Positive (1)
Negative (0)
TP
FP
FN
TN

Precision: Precision measures the proportion of true positive predictions (correctly predicted positives) out of all predicted positives. It focuses on the correctness of positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Used for Medical Diagnosis

PRECISION

Recall (Sensitivity or True Positive Rate): Recall measures the proportion of true positive predictions out of all actual positives. It focuses on the completeness of positive predictions.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Used for information Retreiving

RECALL

QUESTIONS

THANK YOU