بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

# Machine Learning

Course Teacher: **Dr. Tania Islam** ( also 5th and 4th Batch)

"Sharing your notes with your batchmates can greatly benefit average and backseat students. Your 1 minute of effort can reduce the efforts of 100 minutes of others"

-Zahid Hasan, CSE 6th Batch, University of Barishal.

১৪ টা স্লাইডে যে টপিক আছে আমি ওই টপিক গুলা আলাদা আলদা ভাগ করে সাজাইছি। টপিক গুলা এমন ভাবে সাজানোর চেস্টা করছি যাতে আলাদা ভাবে আর প্রশ্ন সল্ভ না করা লাগে, এই টপিক গুলা শেষ করলে স্লাইড+ ৪র্থ+৫ম ব্যাচের ফাইনাল+মিডের সব প্রশ্ন টপিকের সাথে সল্ভ হয়ে যায়। যেগুলা সল্ভ করতে পারিনি আশা করি ওগুলা শেয়ার করবি। লাগলে এডিট এক্সেস দিব নইলে আমারে ইনবক্সে দিলেও হবে।

শেষ রিকুয়েস্ট, আশা করি অনার্স লাইফের লাস্ট এক্সামে এসেও কেও ক্লাস ২ এর বাচ্চাদের মত আচারন করবি না। আমি সল্ভ না দিলেও অনেক টপাররা খাতায়/ডকে যে প্রশ্ন সল্ভ করো তা সবাই জানে। এই কথা গুলা বার বার বলি, কারন আমার যে পরিমান কস্ট হয় ( লাস্ট ৪ টা সেমিস্টার ধরে) এই ডক করতে তা টপাররাও' করো নায় কখনো। কারন আমি সব করি এক্সামের এই ৩ দিনে। মিডে খালি খাতা জমা দিছি এই সাবজেক্টের। ন্যূনতম সহানুভূতি কি দেখনো যেতে পারে না..?

Doc Link: Machine Learning by Zahid

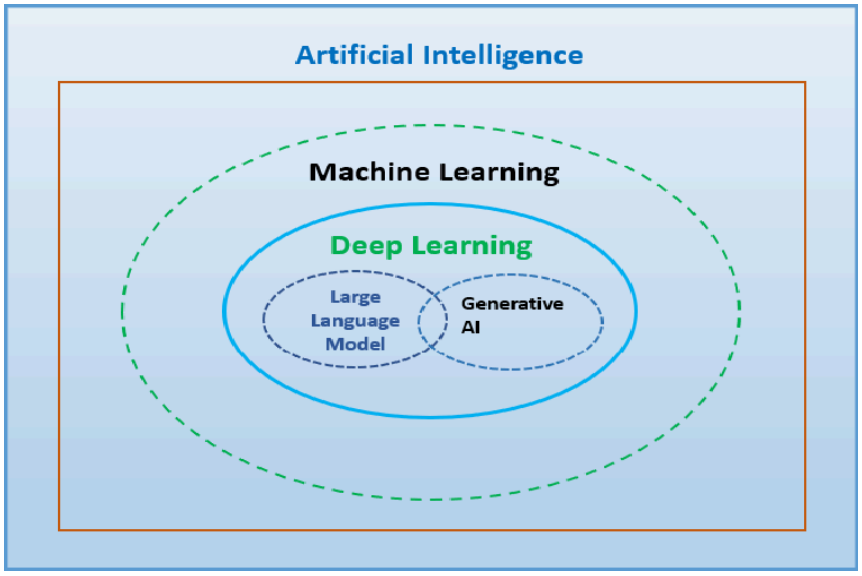
# Introduction of ML

## Def:

Machine learning is a subset of artificial intelligence (AI) that allows machines to learn and improve from data without being explicitly programmed.

## Relation between artificial intelligence (AI) and Machine Learning (ML):

Machine learning (ML) is a specific branch of artificial intelligence (AI). ML has a limited scope and focus compared to AI. AI includes several strategies and technologies that are outside the scope of machine learning.



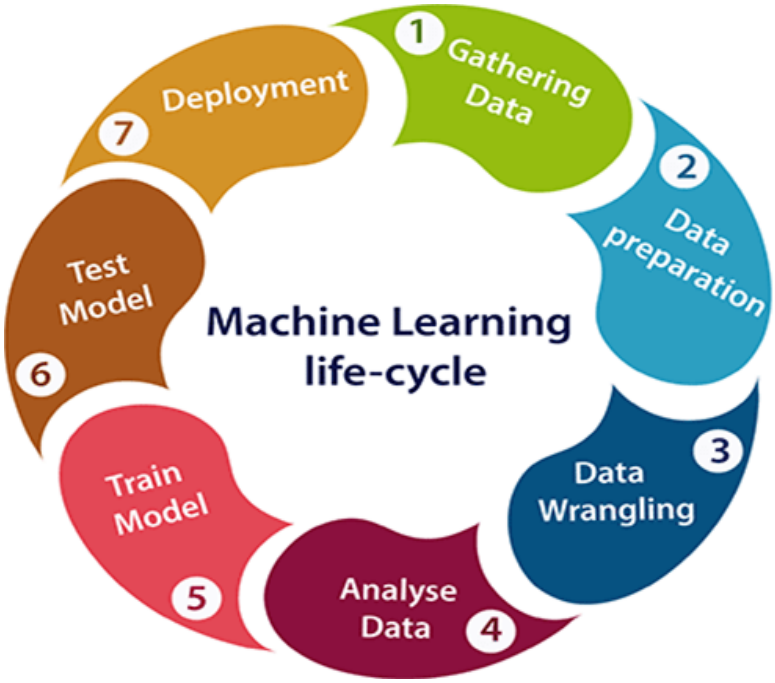

## Machine learning life cycle

The machine learning lifecycle is a process that guides the development and deployment of machine learning models in a structured way. It consists of various steps.

Each step plays a crucial role in ensuring the success and effectiveness of the machine-learning solution. By following the machine-learning lifecycle, organizations can solve complex problems

systematically, leverage data-driven insights, and create scalable and sustainable machine-learning solutions that deliver tangible value.

## Steps involved in ML life cycle:




**Gathering Data**: Collecting data from different sources.

**Data Preparation**: Cleaning and organizing the data for use.

**Data Wrangling**: Fixing errors and adjusting data to be ready for analysis.

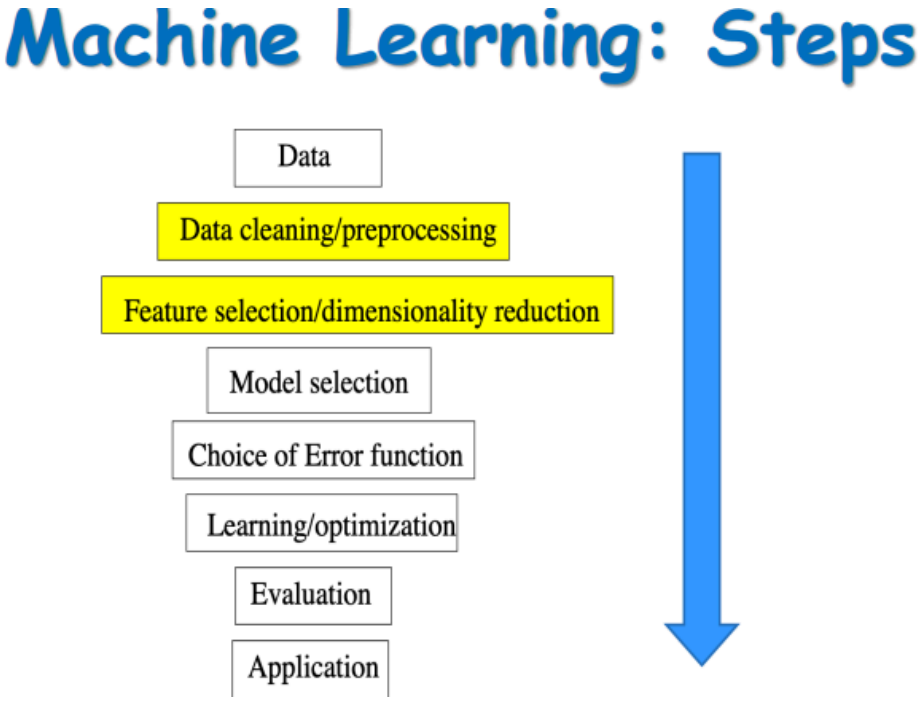
**Analyze Data**: Looking at the data to find patterns and insights.

**Train the Model**: Teaching the model to recognize patterns using the data.

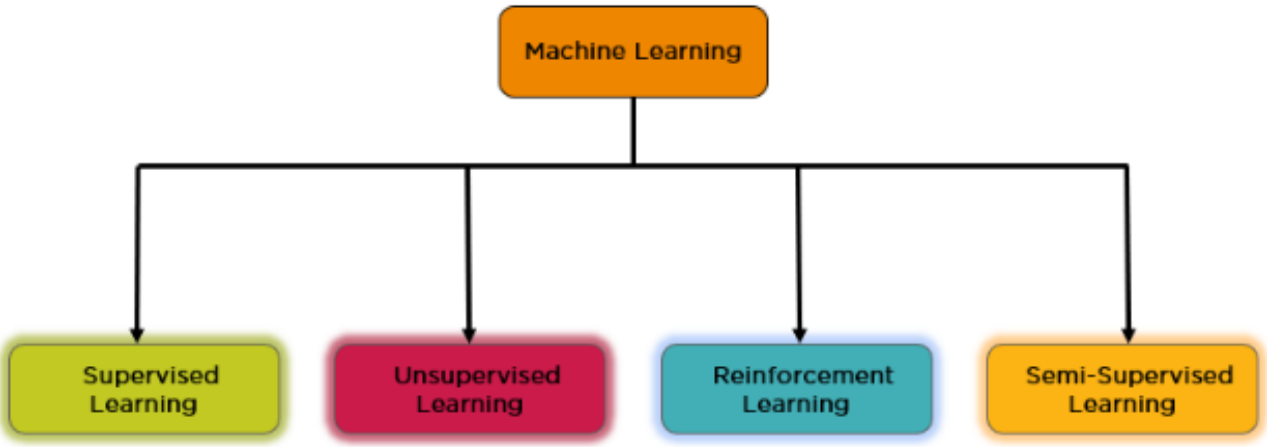
**Test the Model**: Checking how well the model works with new data.

**Deployment**: Putting the model to use in a real-world setting

Or, from Slide

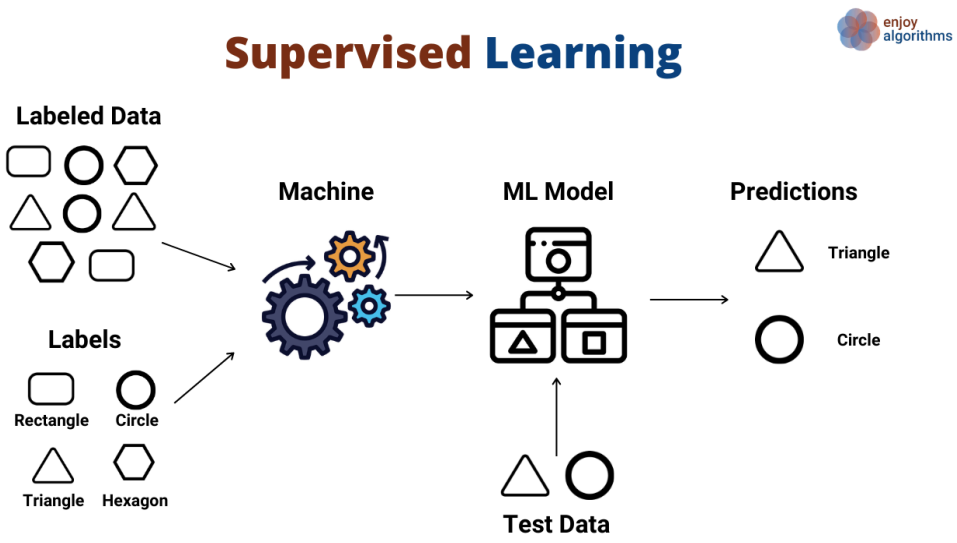



## Types Of Machine Learning




## Supervised learning

Supervised learning is a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns



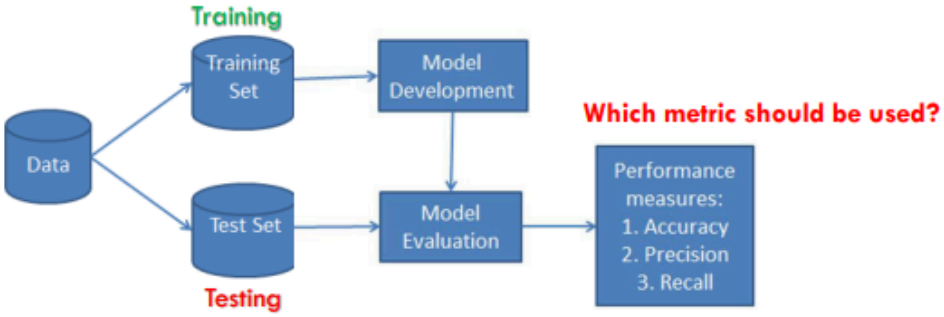

Examples:

- Random Forest
- Decision Trees
- Logistic Regression
- Support Vector Machines
- KNN

**From Slide**

# Supervised Learning: Classification

- The classification has two phases, a training (learning) phase, and the testing (evaluation) phase.
- In the training phase, classifier trains its model on a given dataset.
- The model is **developed** during the training phase.
  - **Development of the model** means that several **parameters are adjusted** to predict a value (weights in ANN).
- In the evaluation phase, it tests the classifier performance.
- **Performance is evaluated** on the basis of various parameters such as accuracy, error, precision, and recall.

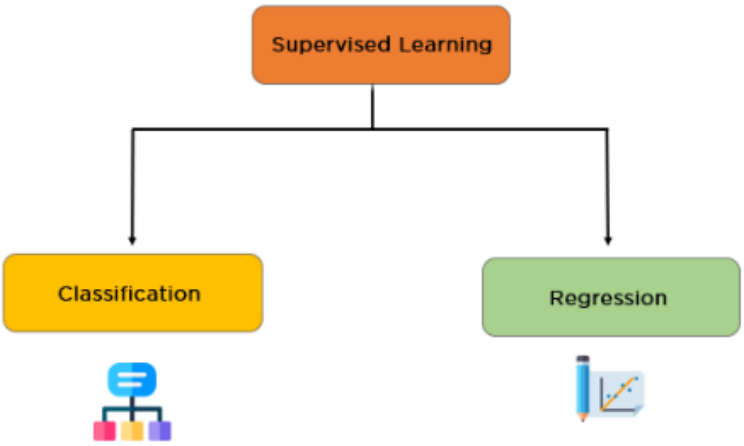



Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.

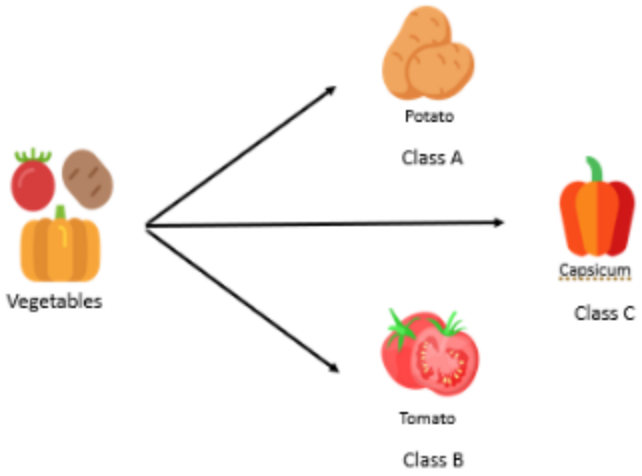
- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, it means our model is accurate.

### Method of Supervised Machine Learning




### Classification

Supervised classification is a machine learning technique that uses labeled data sets to train algorithms to classify data into categories



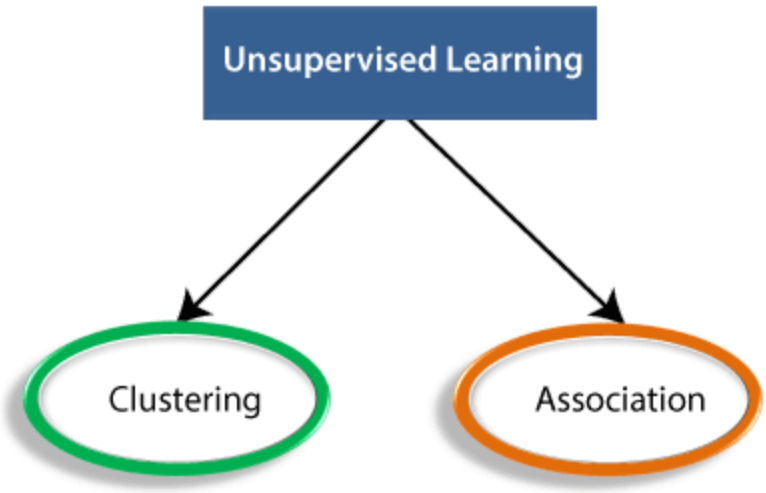

Regression er jonno alada section korechi niche

## Unsupervised learning

Unsupervised learning is a type of machine learning that uses algorithms to analyze unlabeled data without human guidance

### Method of Unsupervised Learning :

The unsupervised learning algorithm can be further categorized into two types of problems:




Clustering: Clustering is a method of grouping objects into clusters such that objects with the most similarities remain in a group and have less or no similarities with the objects of another group.

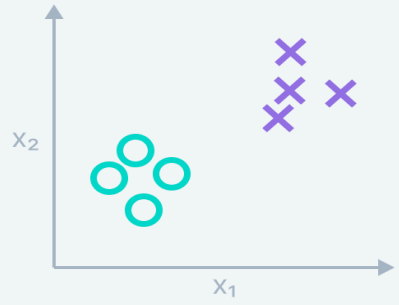
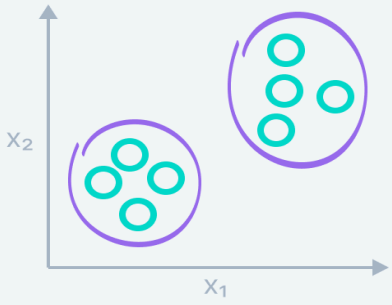
Association: An association rule is an unsupervised learning method that is used for finding the relationships between variables in a large database.

## Examples

K-means clustering
CNN
Neural Networks

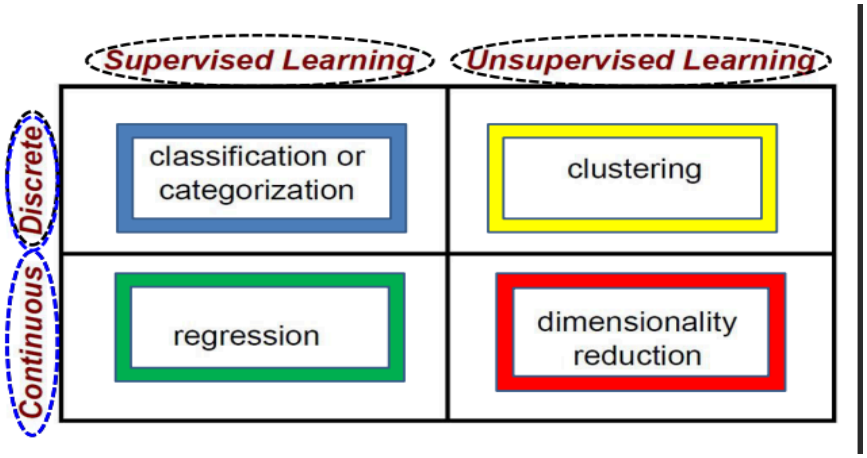
**Difference between Supervised and Unsupervised**

**Supervised, Unsupervised and Reinforcement Learning in Artificial Intelligence in Hindi**

| Supervised learning | Unsupervised learning |
| --- | --- |
| Input data is labeled | Input data is unlabeled |
| Has a feedback mechanism | Has no feedback mechanism |
| Data is classified based on the training dataset | Assigns properties of given data to classify it |
| Divided into Regression & Classification | Divided into Clustering & Association |
| Used for prediction | Used for analysis |
| Algorithms include: decision trees, logistic regressions, support vector machine | Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm |
| A known number of classes | A unknown number of classes |
|  |  |

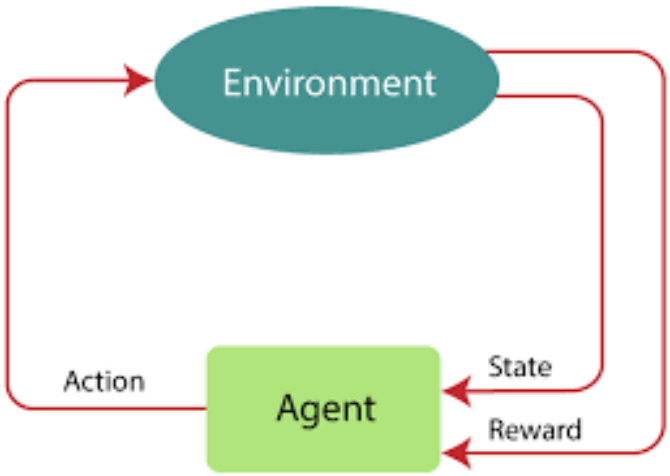


V7 Labs




## Reinforcement Machine Learning

Reinforcement learning (RL) is a machine learning (ML) technique that teaches machines how to make decisions and learn from their experiences.
It's based on the idea that machines can learn to achieve the best results by mimicking the human process of trial and error



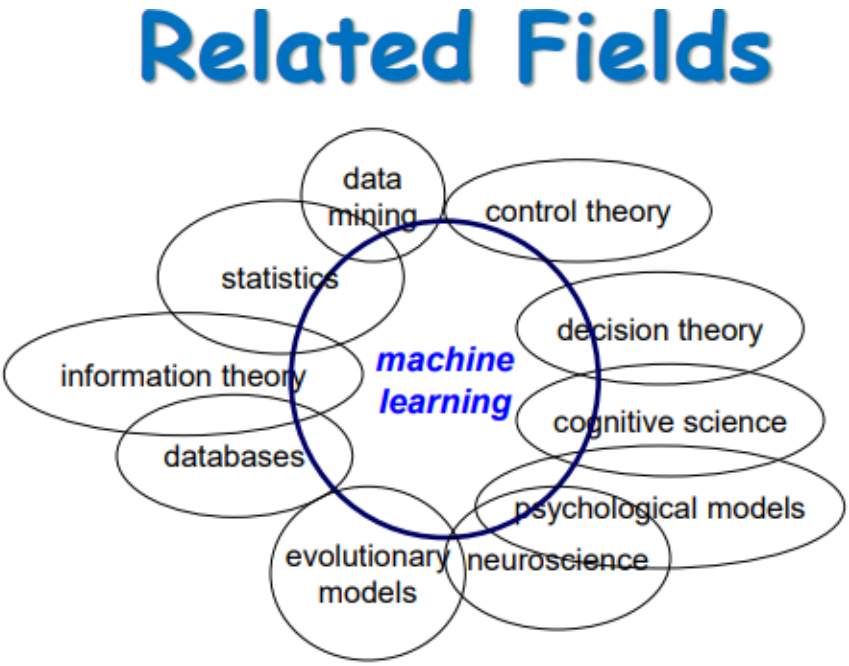

## Advantages of Machine Learning

I. **Improved Accuracy and Precision**: ML models can analyze data and make more accurate predictions than humans.
II. **Automation of Repetitive Tasks:** ML automates routine tasks, increasing efficiency and saving time.
III. **Enhanced Decision-Making**: ML helps make smarter decisions by analyzing large datasets.
IV. **Personalization and Customer Experience**: ML tailors products and services to individual preferences.
V. **Predictive Analytics:** ML predicts future outcomes based on past data.
VI. **Scalability:** ML handles large data and grows with increasing data easily.
VII. I**mproved Security**: ML detects threats and fraud in real-time.
VIII. **Cost Reduction**: ML reduces operational costs by automating processes.
**IX.** **Innovation and Competitive Advantage:** ML drives new ideas and gives businesses an edge.
X. **Enhanced Human Capabilities**: ML helps humans perform tasks more effectively with better insights.

## Disadvantages of Machine Learning

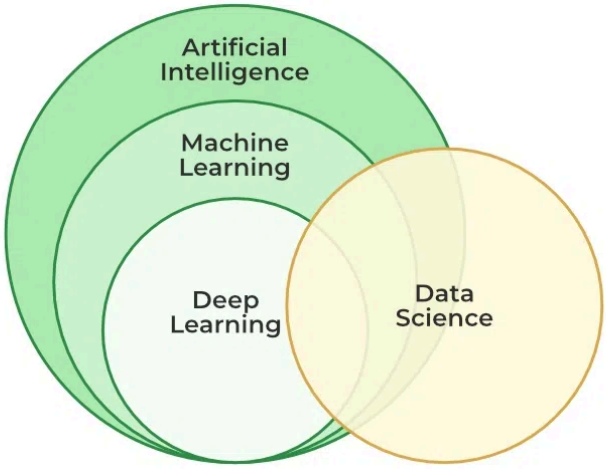
I. **Data Dependency**: ML needs a lot of high-quality data to work well.
II. **High Computational Costs**: Training ML models can be expensive and energy-consuming.
III. **Complexity and Interpretability**: Some ML models are hard to understand or explain.
IV. **Overfitting and Underfitting**: ML models can perform poorly if they learn too much or too little from the data.
V. **Ethical Concerns**: ML can raise issues related to privacy and bias.
VI. **Lack of Generalization**: ML models may struggle to work well in different situations.
VII. **Dependency on Expertise**: ML development requires skilled professionals.
VIII. **Security Vulnerabilities**: ML can be tricked by malicious data inputs.
IX. **Maintenance and Updates**: ML models need regular updates to stay accurate.
X. **Legal and Regulatory Challenges**: ML faces legal challenges, especially with data privacy.

## Applications of ML




### Deep learning relations with ML

*Deep learning*, is a subset of machine learning that uses neural networks with multiple layers to analyze complex patterns and relationships in data.




## Key differences

Deep learning relies heavily on deep artificial neural networks with multiple hidden layers, while traditional machine learning algorithms often use simpler models with fewer layers or different structures altogether.

**Feature Extraction:**
Deep learning automatically learns features from raw data through its network layers, whereas other machine learning methods usually require manual feature engineering by humans to identify relevant data attributes.

**Data Requirements:**
Deep learning typically demands large amounts of data to train effectively, while some traditional machine learning algorithms can work with smaller datasets.

**Complexity of Tasks:**
Deep learning is better suited for complex tasks like image classification, natural language translation, and speech recognition, where traditional machine learning might struggle

**Now practice yourself based on the last 2 years' previous ques!**

a) What is machine learning? Write down the relation between artificial intelligence (AI) and Machine Learning (ML). [2.5]

b) What is the difference between supervised and unsupervised machine learning? Give example.[3]

c) How can you define deep learning, and how does it contrast with other machine learning algorithms?

d) What do mean by machine learning life cycle?

e) Describe the steps involved in ML life cycle.

f) How does supervised learning work?

g) Briefly describe the steps involved in Supervised Learning.

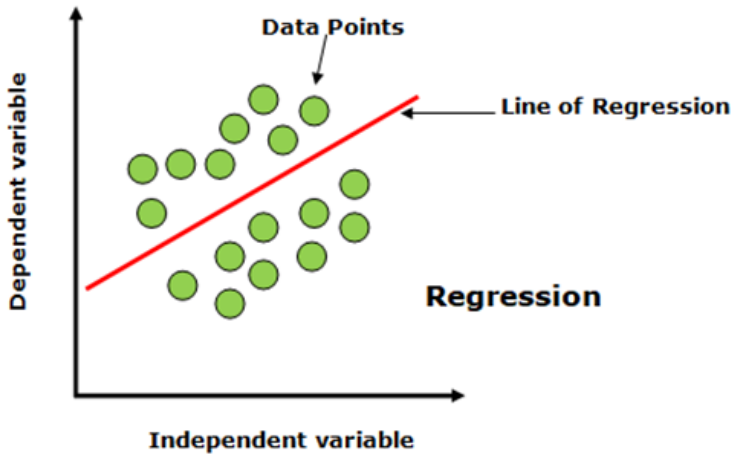
h) Briefly describe the types of unsupervised learning algorithms.

# Reggeression

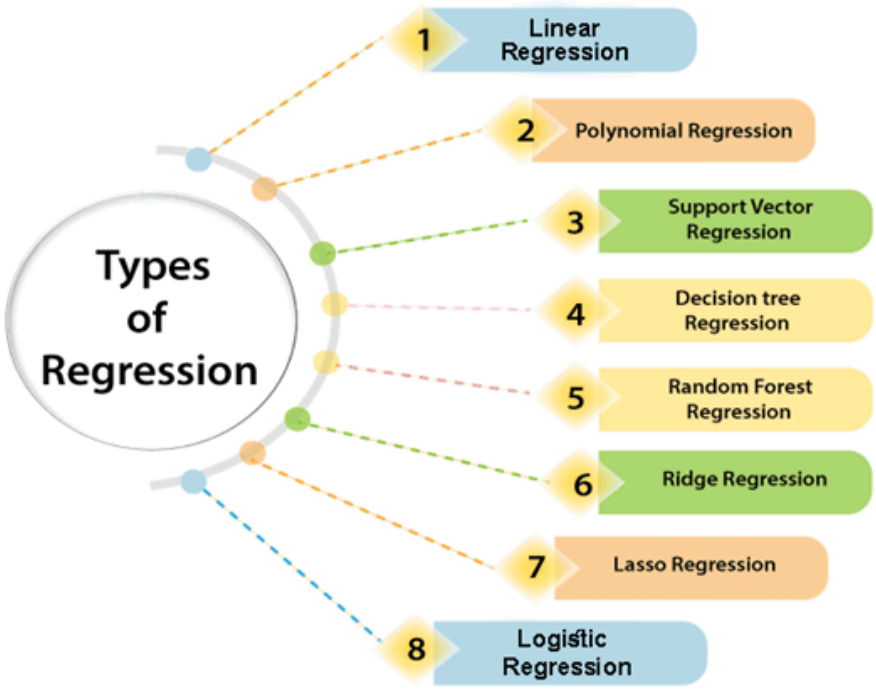
Reggeression—> Lec-3: Introduction to Regression with Real Life Examples

## Def:

Regression is a machine learning technique that uses algorithms to find the relationship between independent and dependent variables to predict outcomes. It's a common method in supervised machine learning, which uses labeled training data for both input and output

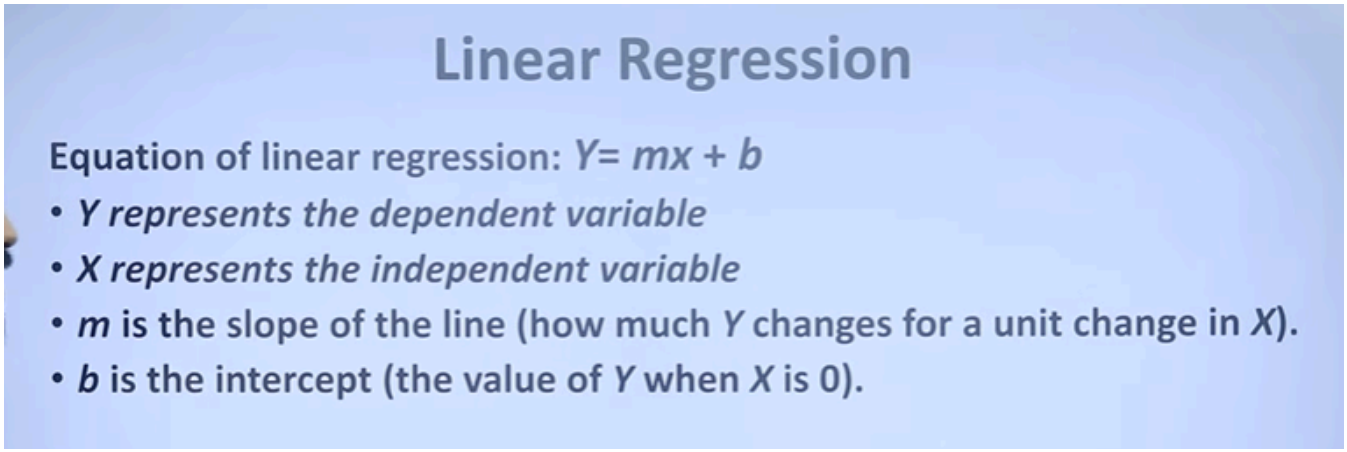


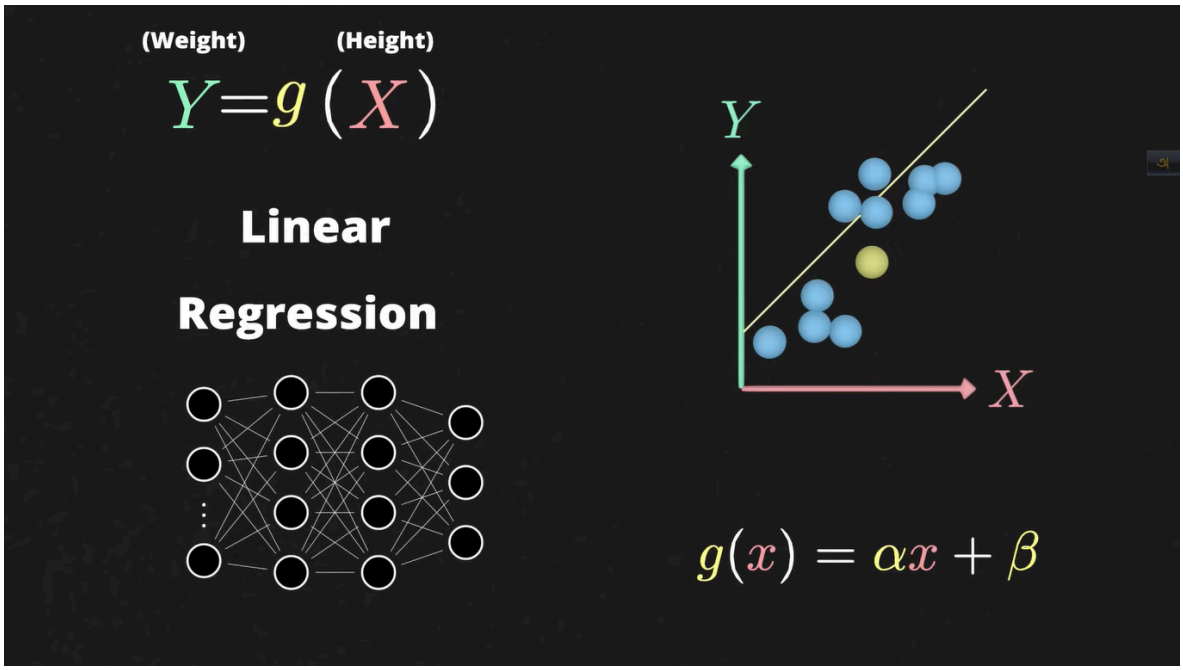

## Types of Regression



# Linear Regression

**Def & How it performs:**
Linear regression is an analytics procedure that can generate predictions by using an easily interpreted mathematical formula.

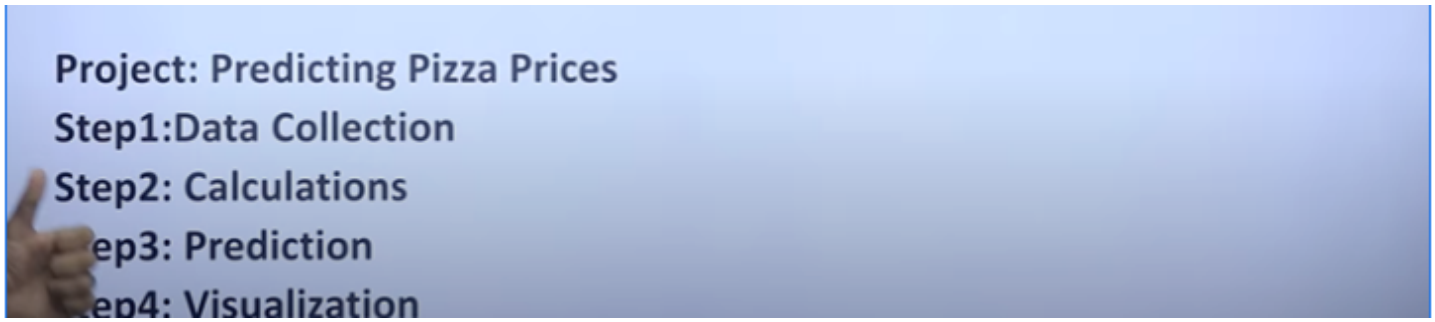


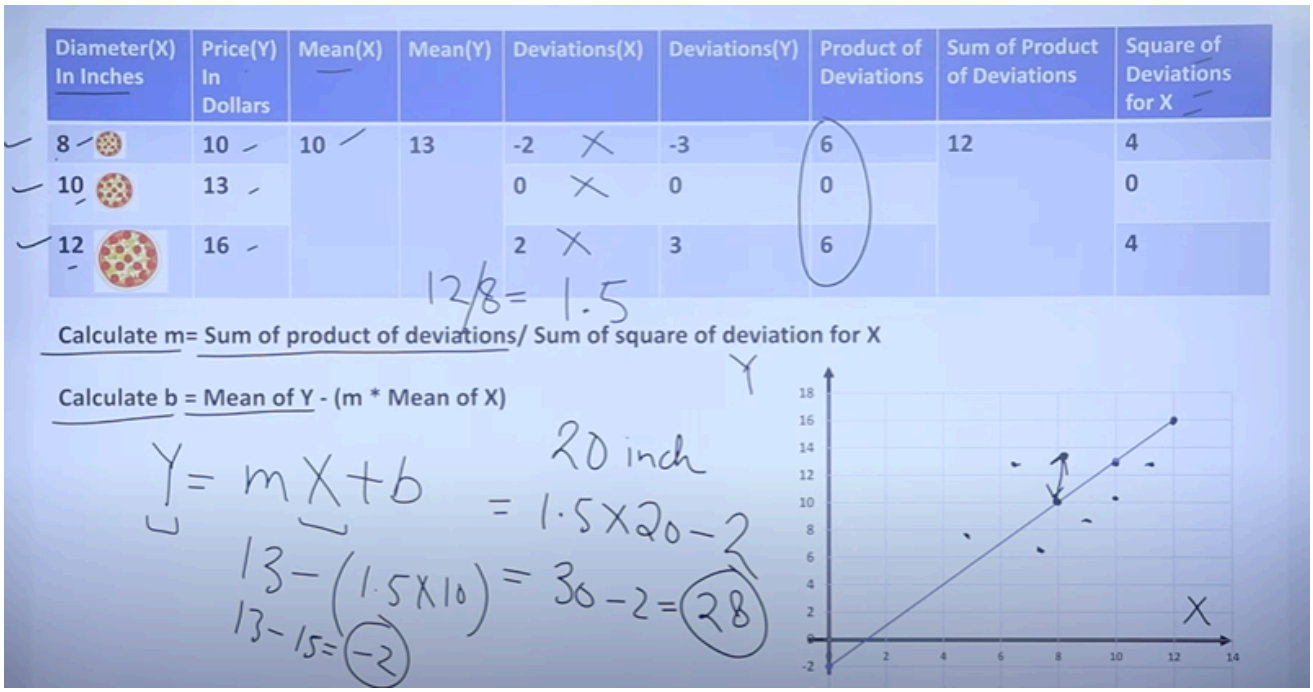



## Mathematical Example

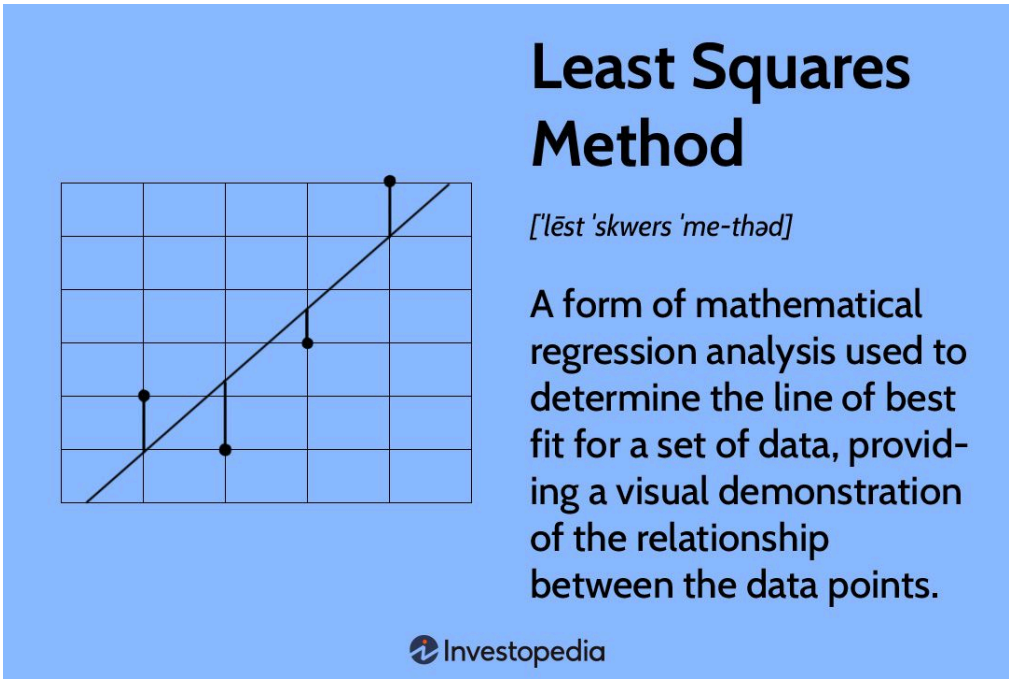
To solve See that →

Lec-4: Linear Regression📈 with Real life examples & Calculations | Easiest Explanation



## Least Squares Method




### Mathematical Example for Least Squares Method

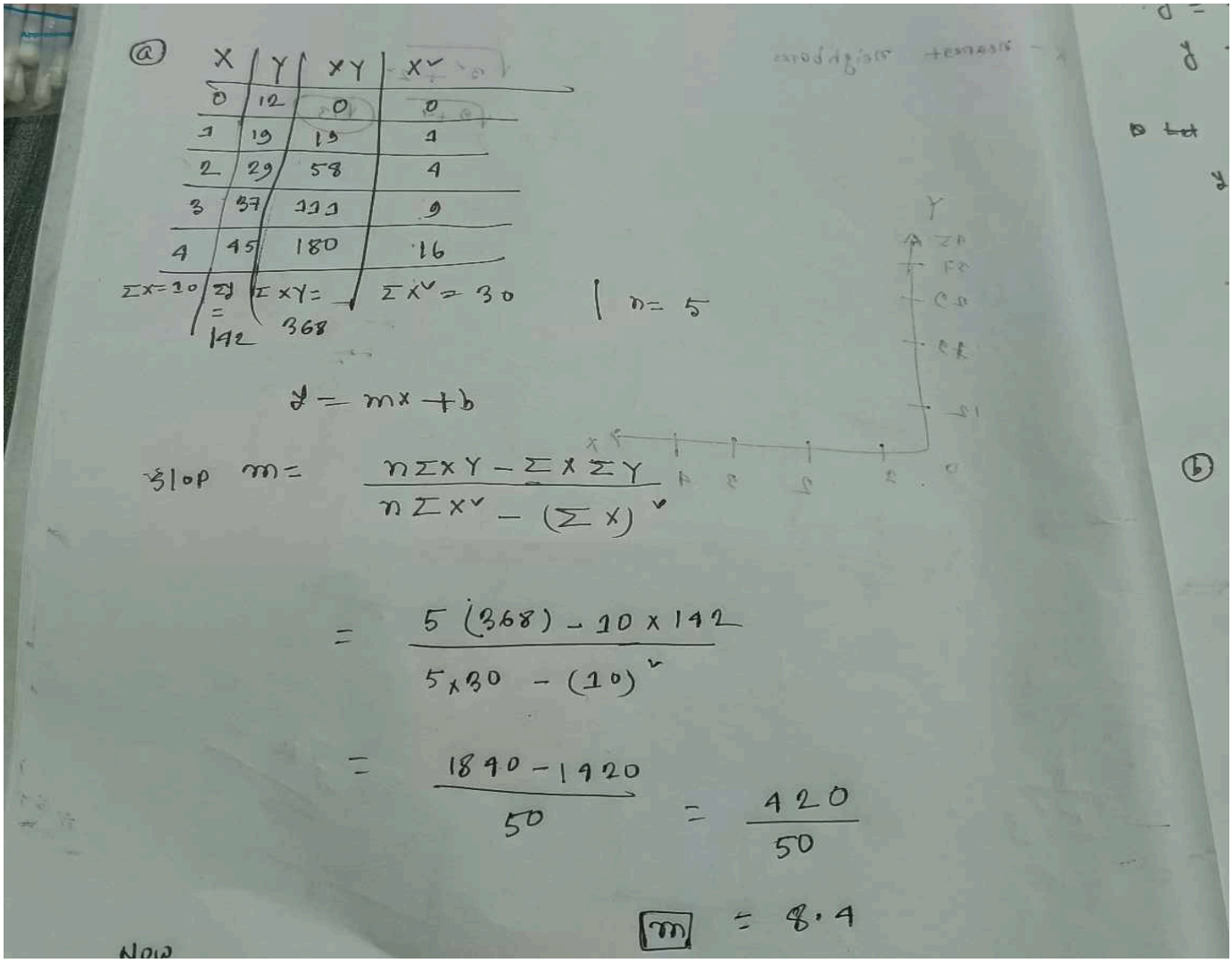
The sales of a company (in million dollars) for each year are shown in the table below.

| x (year) | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| y (sales) | 12 | 19 | 29 | 37 | 45 |

i) Find the least square regression line y = ax + b.
ii) Use the least squares regression line as a model to estimate the sales of the company in year 7.

Sol: Linear Regression Using Least Squares Method - Line of Best Fit Equation

l:



| X | Y | XY | $X^2$ |
|---|---|---|---|
| 0 | 12 | 0 | 0 |
| 1 | 19 | 19 | 1 |
| 2 | 29 | 58 | 4 |
| 3 | 37 | 111 | 9 |
| 4 | 45 | 180 | 16 |
| $\Sigma X = 10$ | $\Sigma Y = 142$ | $\Sigma XY = 368$ | $\Sigma X^2 = 30$ |

$n = 5$

$y = mx + b$

Slop $m = \dfrac{n\Sigma XY - \Sigma X \Sigma Y}{n\Sigma X^2 - (\Sigma X)^2}$

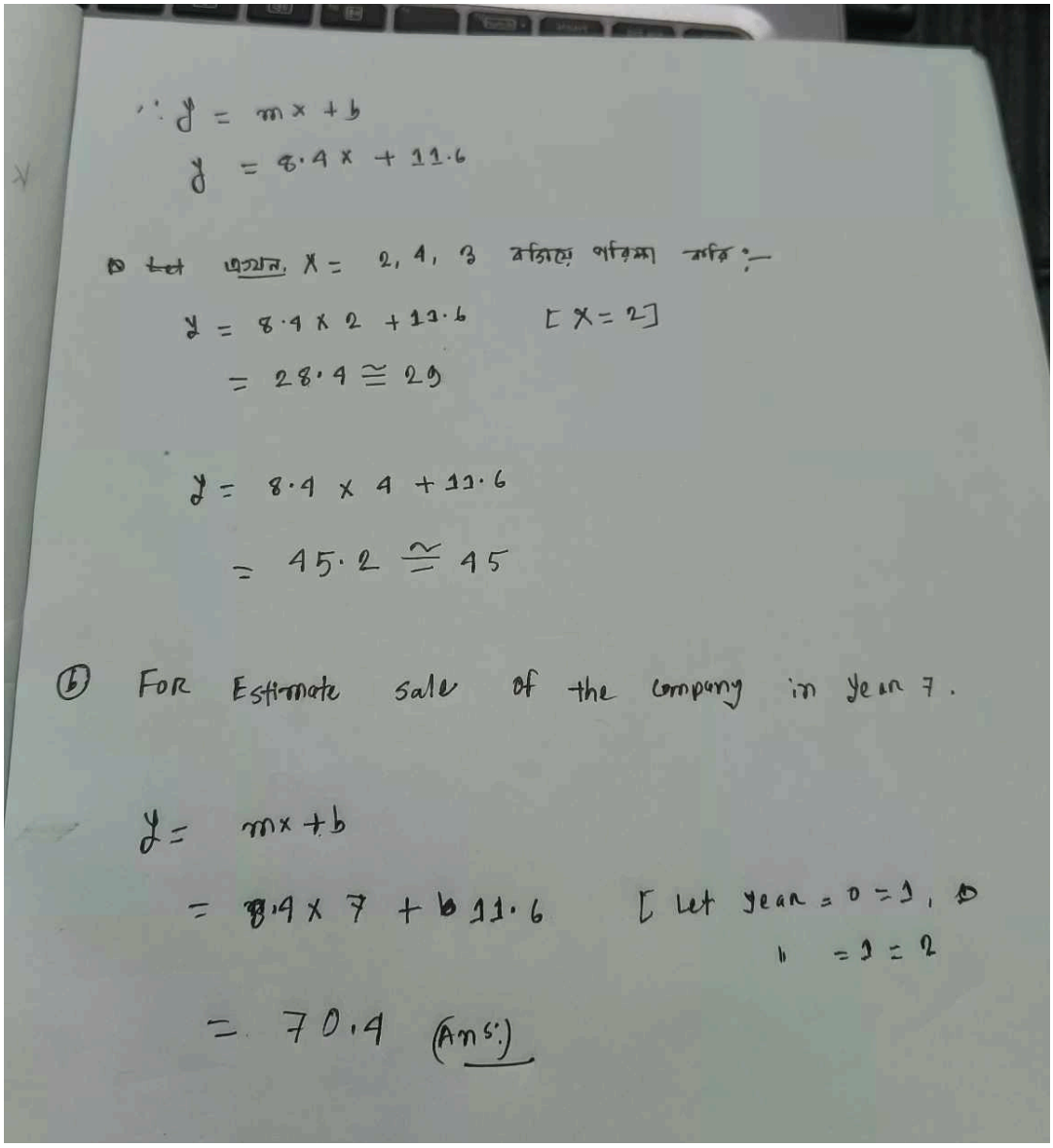
$= \dfrac{5(368) - 10 \times 142}{5 \times 30 - (10)^2}$

$= \dfrac{1840 - 1420}{50} = \dfrac{420}{50}$

$\boxed{m} = 8.4$

Now

$b = \dfrac{\Sigma y - m\Sigma X}{n} = \dfrac{142 - 8.4 \times 10}{5} = \dfrac{58}{5} = 11.6$




## When should multiple regression analysis be used :

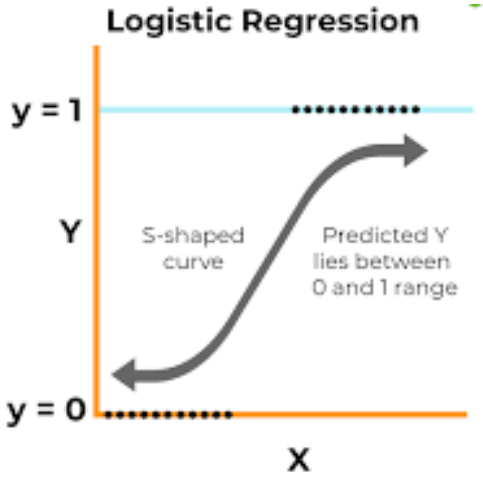
Multiple regression is an extension of simple linear regression. It is used when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable ( the outcome or target variable).

## Logistic Regression

**Def:**

Videos→ Lec-5: Logistic Regression with Simplest & Easiest Example | Machine Learning

Logistic regression is a supervised machine learning algorithm used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not




## Linear vs Logistics Regression

| Linear Regression | Logistic Regression |
|---|---|
| Linear regression is used to predict the continuous dependent variable using a given set of independent variables. | Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables. |

| | |
|---|---|
| Linear Regression is used for solving Regression problems. | Logistic regression is used for solving Classification problems. |
| In Linear regression, we predict the value of continuous variables. | In logistic Regression, we predict the values of categorical variables. |
| In linear regression, we find the best-fit line, by which we can easily predict the output. | In Logistic Regression, we find the S-curve by which we can classify the samples. |
| In Linear regression, it is required that the relationship between a dependent variable and the independent variable must be linear. | In Logistic regression, it is not required to have a linear relationship between the dependent and independent variables. |
| In linear regression, there may be collinearity between the independent variables. | In logistic regression, there should not be collinearity between the independent variables. |

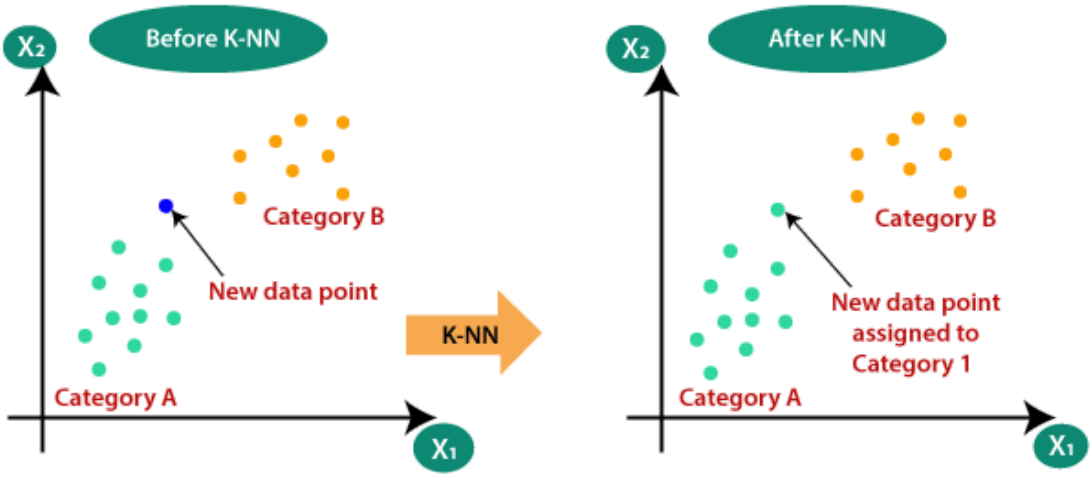
# KNN Algorithm

The k-nearest neighbors (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

Basic videos: Lec-7: kNN Classification with Real Life Example | Movie Imdb Example | Supervised Lea…

## 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:

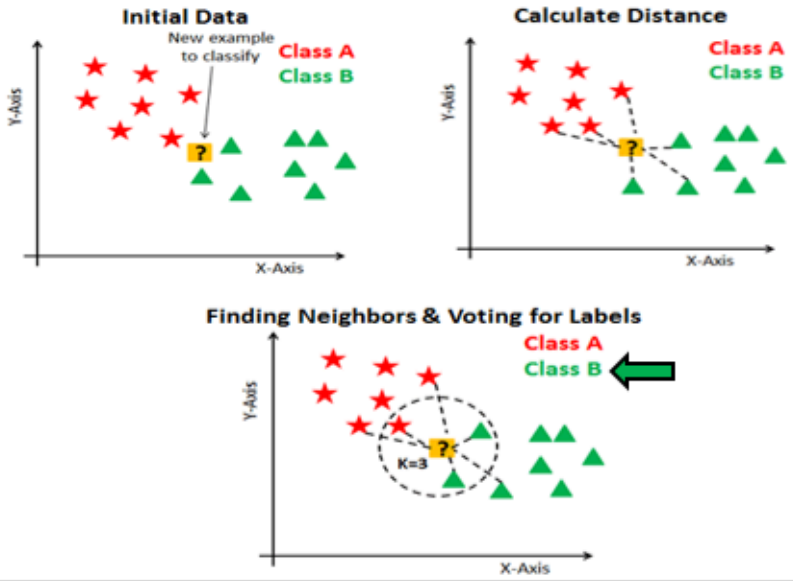



## How does K-NN work?

# KNN Algorithm: General Information

KNN has the following *basic steps*:

1. Prepare data (split to training/testing)
2. Get first test data point
3. Calculate distance
4. Find closest neighbors
5. Vote for labels (majority rule)




7

## Steps

**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

## Simulation

### Step-1

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

○ There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.

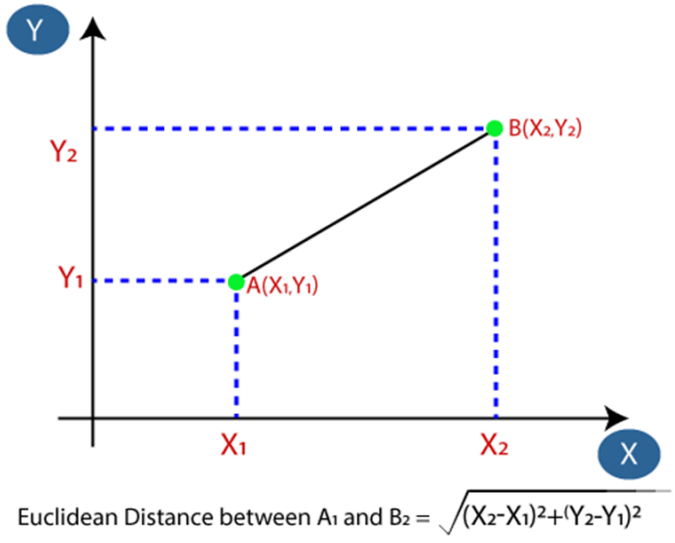
○ A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.

○ Large values for K are good, but it may find some difficulties.

○ Try to take odd values.

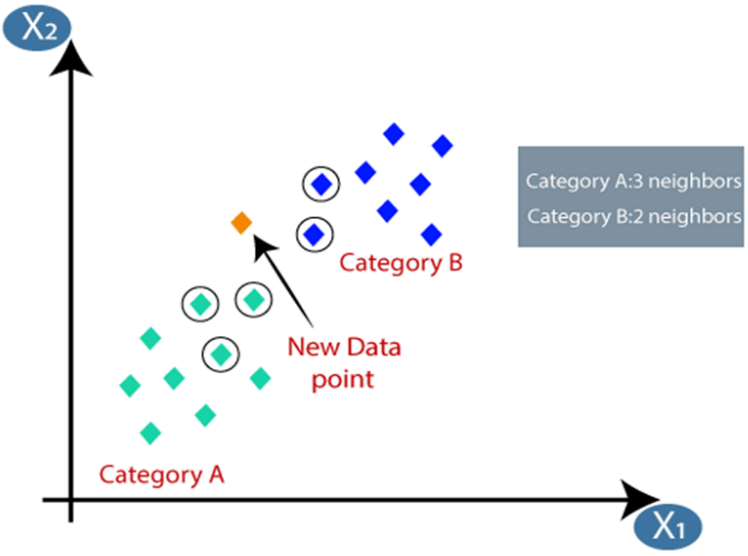
### Step-2

We will calculate the **Euclidean distance** between test data and each row of training data. The Euclidean distance is the distance between two points. It can be calculated as:

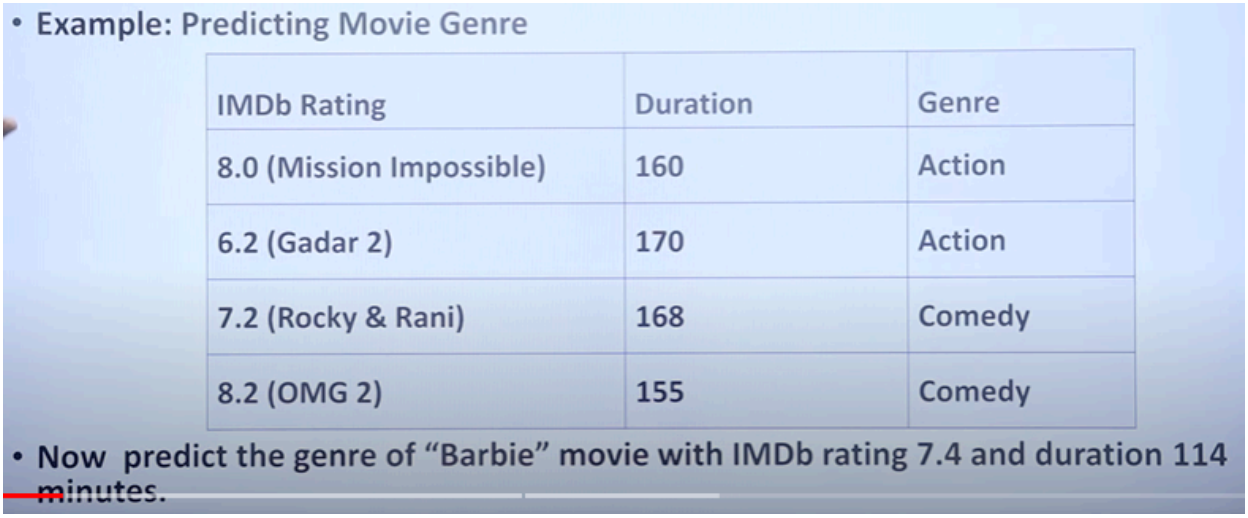



### Step-3

By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B.

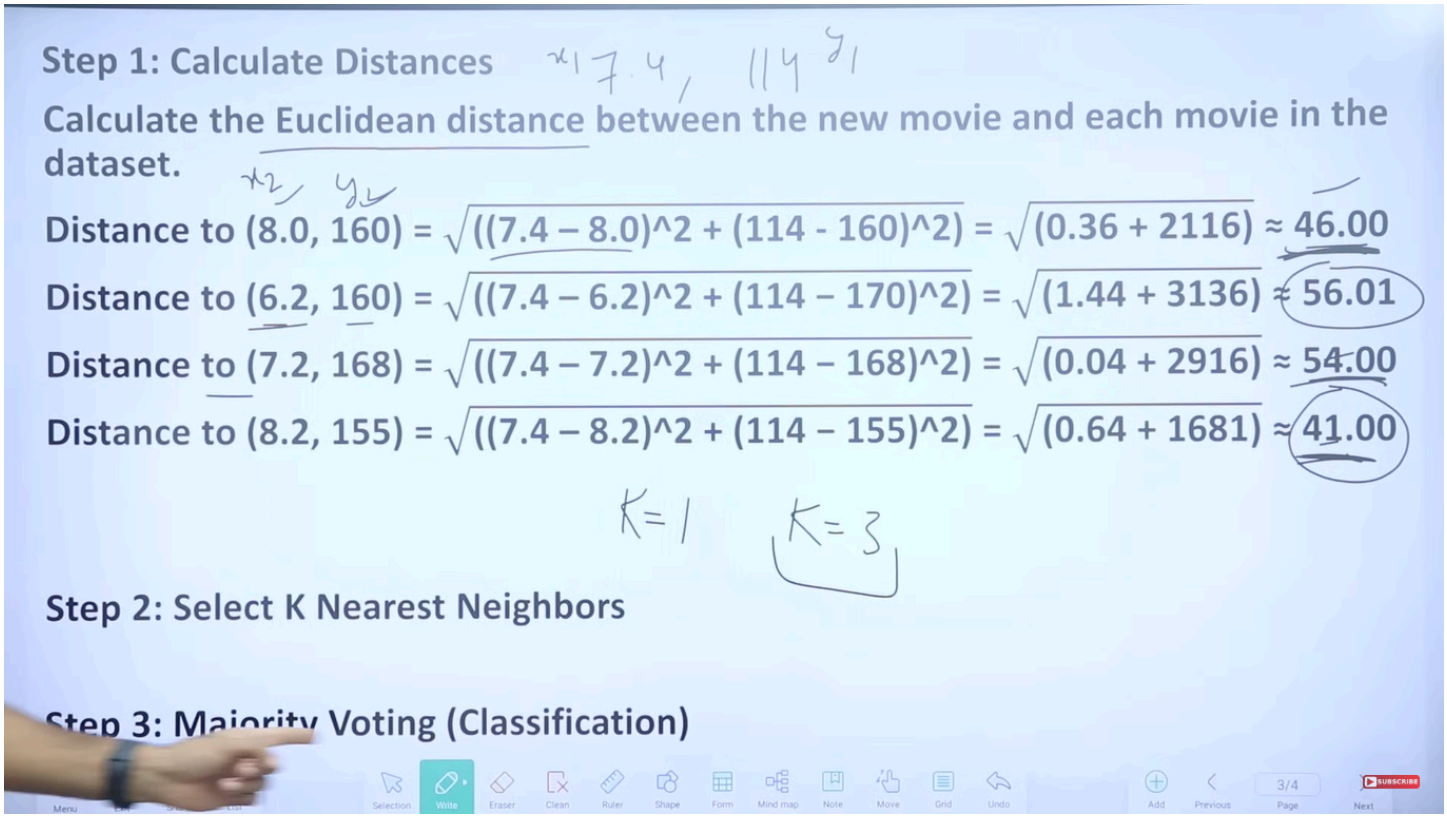



## Mathematical Example




Sol:

Video: Lec-7: kNN Classification with Real Life Example | Movie Imdb Example | Supervised Learning




### Importance of the KNN algorithm?

We need the K-NN algorithm because it is simple, easy to use, and works well for both classification and regression tasks. It doesn't make assumptions about the data and can handle both numbers and categories. K-NN finds the closest data points to a new one and makes predictions based on their similarity, adapting to the local data patterns. It's also less affected by outliers compared to other methods.
Also When the dataset is small, labeled and noise-free

## Advantages of KNN

- It is simple to implement.
- It is robust to the noisy training data.
- It can be more effective if the training data is large.
- It is very useful for nonlinear data because there is no assumption about data in this algorithm.
- It is a versatile algorithm as we can use it for classification as well as regression.

## Disadvantage of KNN

- It is computationally a bit expensive algorithm because it stores all the training data.
- High memory storage required as compared to other supervised learning algorithms.
- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

## How to choose the value of K

# Selecting the Number of Neighbors

K is a hyperparameter that is important to the KNN algorithm performance

Increase k:

- Makes KNN less sensitive to noise

Decrease k:

- Allows capturing finer structure of space

➔Pick k not too large, but not too small (depends on data)

- When we increase K, the training error will increase (increase bias), but the test error may decrease at the same time (*decrease variance*).
- The bias will be 0 (*but variance will be high*) when K is small (K=1)

### Parametic and Non-parametic model

Parametric models, like logistic regression (Log. R), linear regression (LR), and artificial neural networks (ANN), have a fixed number of parameters. No matter how much data you add, the model doesn't change how many parameters it uses, and its complexity stays the same

Non-parametric models, like KNN, decision trees (DT), and gradient boosting (GB), adapt to the data as it grows, meaning their complexity increases with more data. The model changes based on the amount and type of data given.

## Distance Measures Types

- **Minkowski Distance:** $\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$
  - *P = 1* → Manhattan Distance $d = \sum_{i}^{n} |x_i - y_i|$
  - *P = 2* → Euclidean Distance $d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- **Cosine Distance:** $\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$

  determines whether two vectors are pointing in the same direction
- **Jaccard Distance:** $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$

  Ratio of overlapping items to Total Items

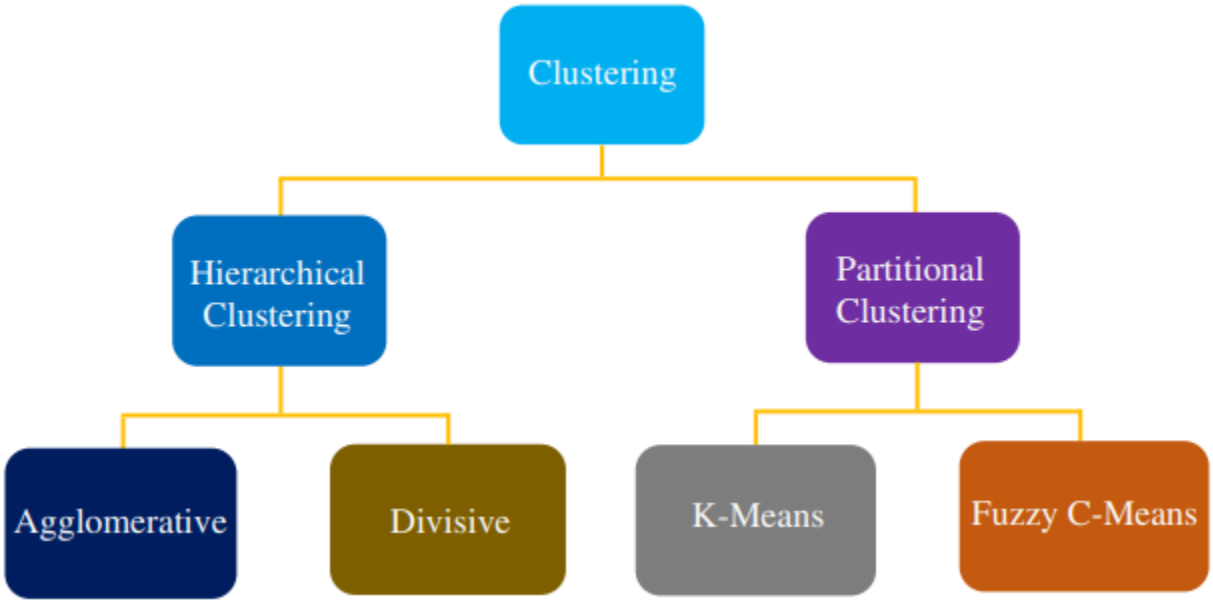
# K-Means Clustering

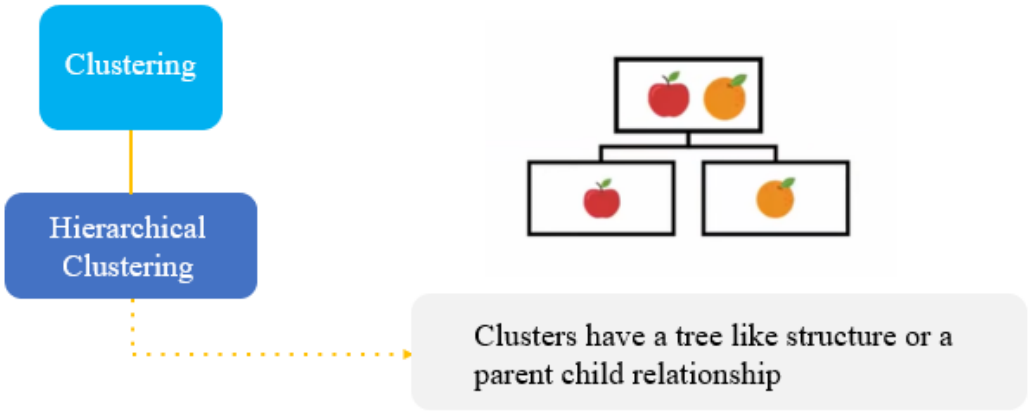
K-mean Clustering with Numerical Example | Unsupervised Learning | Machine🖥️ Learning 👨‍💻🧑‍🦰

**Def:**

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.

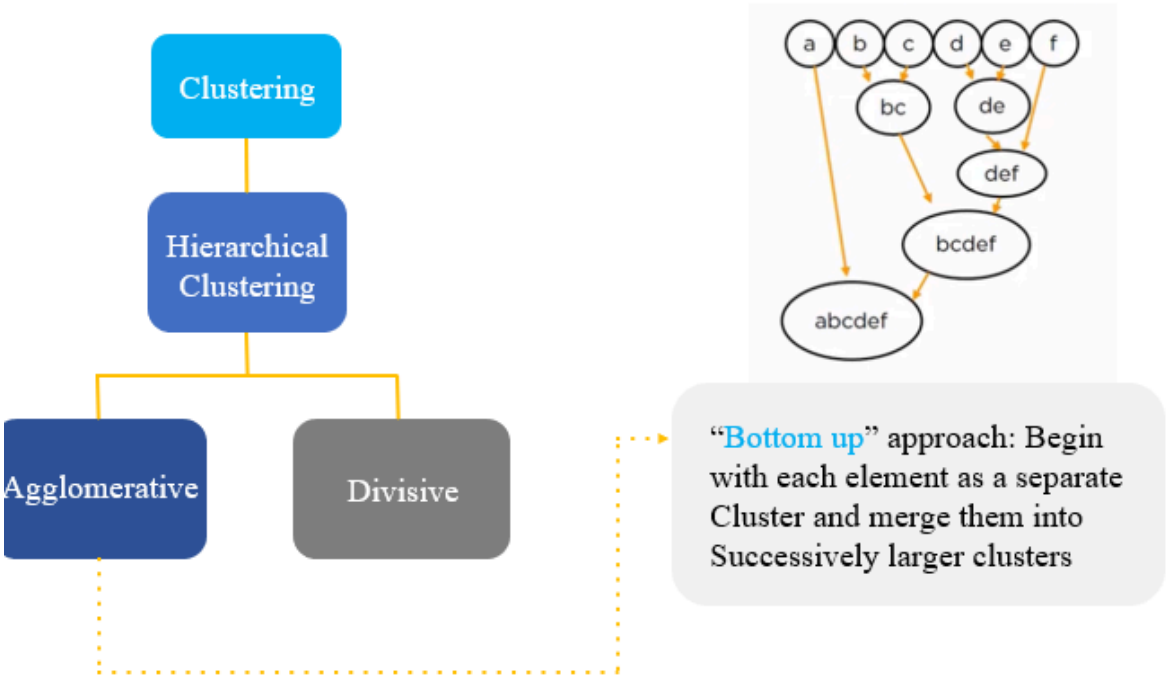
Types of K Means Clustering:



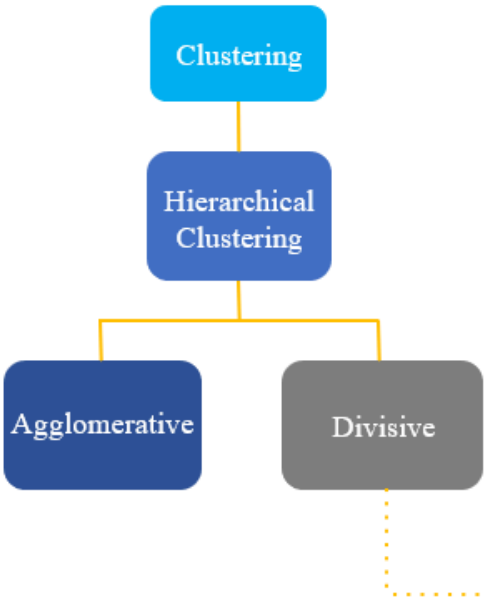



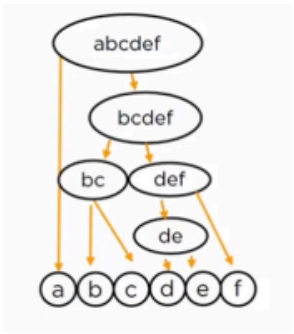
## Agglomerative clustering

There is a bottom-up approach. We begin with each element as a separate cluster and merge them into successively more massive clusters, as shown below:



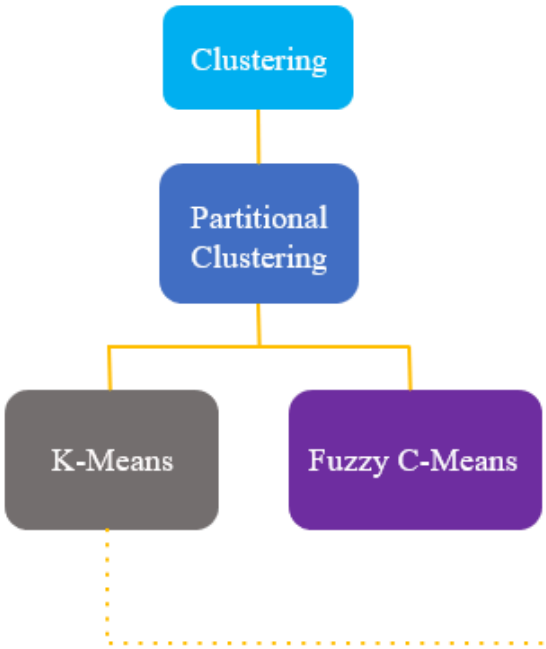

## Divisive clustering

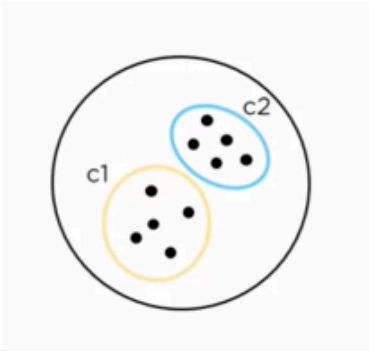



"Top down" approach begin with the whole set and proceed to divide it into successively smaller clusters

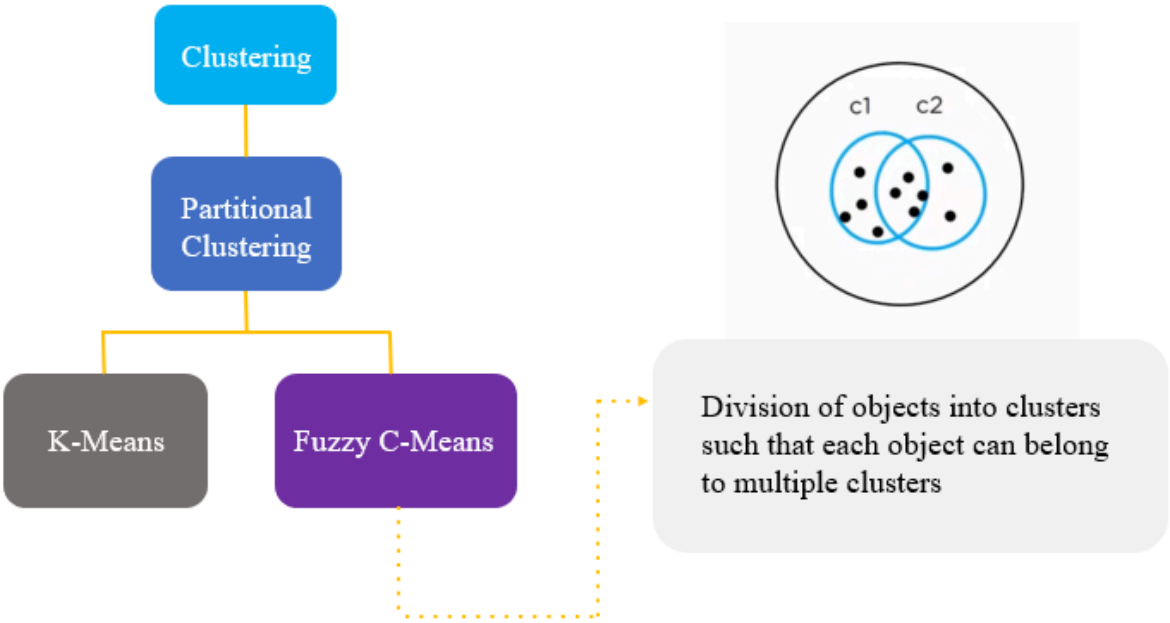
**Partitioning Clustering**




Division of objects into clusters such that each object is in exactly one cluster, not several

**Fuzzy c-means**




## Advantages of k-means

1. **Simple and easy to implement**: The k-means algorithm is easy to understand and implement, making it a popular choice for clustering tasks.
2. **Fast and efficient**: K-means is computationally efficient and can handle large datasets with high dimensionality.
3. **Scalability**: K-means can handle large datasets with a large number of data points and can be easily scaled to handle even larger datasets.
4. **Flexibility:** K-means can be easily adapted to different applications and can be used with different distance metrics and initialization methods.

## Disadvantages of K-Means:

1. **Sensitivity to initial centroids**: K-means is sensitive to the initial selection of centroids and can converge to a suboptimal solution.
2. **Requires specifying the number of clusters**: The number of clusters k needs to be specified before running the algorithm, which can be challenging in some applications.
3. **Sensitive to outliers**: K-means is sensitive to outliers, which can have a significant impact on the resulting clusters.

## Applications of K-Means Clustering


### Difference between KNN and K Means Clustering

| K-Means | KNN |
|---|---|
| It is an **Unsupervised** learning technique | It is a **Supervised** learning technique |
| It is used for **Clustering** | It is used mostly for **Classification**, and sometimes even for **Regression** |
| 'K' in K-Means is the number of clusters the algorithm is trying to identify/learn from the data. The clusters are often unknown since this is used with Unsupervised learning. | 'K' in KNN is the number of nearest neighbours used to classify or (predict in case of continuous variable/regression) a test sample |
| It is typically used for scenarios like understanding the population demomgraphics, market segmentation, social media trends, anomaly detection, etc. where the clusters are unknown to begin with. | It is used for classification and regression of known data where usually the target attribute/variable is known before hand. |
| In training phase of K-Means, K observations are arbitrarily selected (known as centroids) and the clusters are formed around (similar to) them. Once the clusters are formed, for each cluster the centroid is updated to the mean of all cluster members. And the cluster formation restarts with new centroids. This repeats until best centroids of the clusters are identified. The prediction of a test observation is done based on nearest centroid. | K-NN doesn't have a training phase as such. But the prediction of a test observation is done based on the K-Nearest (often euclidean distance) Neighbours (observations) based on weighted averages/votes. |

## Distance Measure

Euclidean distance measure

Manhattan distance measure

Distance measure will determine the similarity between two elements and it will influence the shape of the clusters

Squared Euclidean Distance measure

Cosine distance measure

01 Euclidean distance measure

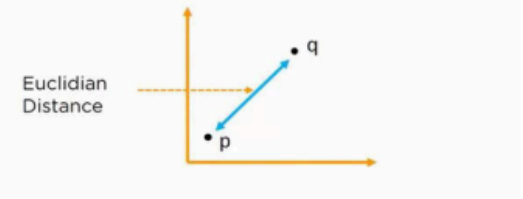
02 Squared Euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

- The Euclidean distance is the "ordinary" straight line
- It is the distance between two points in Euclidean space

$$d=\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$$



## Squared Euclidean Distance Measure

01 Euclidean distance measure

02 Squared Euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The Euclidean squared distance matric uses the same equation as the Euclidean distance metric, but does not take the square root.

$$d = \sum_{i=1}^{n} (q_i - p_i)^2$$

## Manhattan Distance Measure

01 Euclidean distance measure

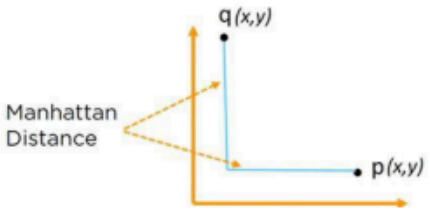
02 Squared Euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The Manhattan distance is the simple sum of the horizontal and vertical Components or the distance between two points measured along axes at right angles

$$d = \sum_{i=1}^{n} |q_x - p_x| + |q_y - p_y|$$



## Cosine Distance Measure

01 Euclidean distance measure

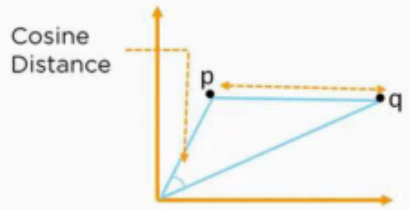
02 Squared Euclidean distance measure

03 Manhattan distance measure

04 Cosine distance measure

The cosine distance similarity measures the angle between the two vectors

$$d = \frac{\sum_{i=0}^{n-1} q_i - p_x}{\sum_{i=0}^{n-1}(q_i)^2 \times \sum_{i=0}^{n-1}(p_i)^2}$$




# Distance measure Mathematical Example

For videos: How to find Euclidean Manhattan Minkowski distance Supremum distance Cosine Similarity …

## 1. Euclidian Distance

## Euclidian Distance – Solved Example – Data Mining

| | $A_1$ | $A_2$ | Euclidian Distance |
|---|---|---|---|
| 1 | 1.5 | 1.7 | $\sqrt{(1.4-1.5)^2+(1.6-1.7)^2} = 0.1414$ |
| 2 | 2 | 1.9 | $\sqrt{(1.4-2.0)^2+(1.6-1.9)^2} = 0.6708$ |
| 3 | 1.6 | 1.8 | $\sqrt{(1.4-1.6)^2+(1.6-1.6)^2} = 0.2828$ |
| 4 | 1.2 | 1.5 | $\sqrt{(1.4-1.2)^2+(1.6-1.5)^2} = 0.2236$ |
| 5 | 1.5 | 1.0 | $\sqrt{(1.4-1.5)^2+(1.6-1.0)^2} = 0.6083$ |

$x = (1.4, 1.6)$

- *Let* $P_1(x_1, y_1)$ *and* $P_2(x_2, y_2)$
- *Euclidian Distance* $= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

## 2. Manhattan Distance

## Manhattan Distance – Solved Example – Data Mining

| | $A_1$ | $A_2$ | Manhattan Distance |
|---|---|---|---|
| 1 | 1.5 | 1.7 | $\|1.4-1.5\| + \|1.6-1.7\| = 0.2$ |
| 2 | 2 | 1.9 | $\|1.4-2.0\| + \|1.6-1.9\| = 0.9$ |
| 3 | 1.6 | 1.8 | $\|1.4-1.6\| + \|1.6-1.8\| = 0.4$ |
| 4 | 1.2 | 1.5 | $\|1.4-1.2\| + \|1.6-1.5\| = 0.3$ |
| 5 | 1.5 | 1.0 | $\|1.4-1.5\| + \|1.6-1.0\| = 0.7$ |

$x = (1.4, 1.6)$

- *Let* $P_1(x_1, y_1)$ *and* $P_2(x_2, y_2)$
- *Manhattan Distance* $= |x_2 - x_1| + |y_2 - y_1|$

## 3. Minkowski Distance

- $Minkowski\ Distance = \sqrt[h]{(x_2 - x_1)^h + (y_2 - y_1)^h}$

**For h=2**

### Minkowski Distance – Solved Example – Data Mining

| | $A_1$ | $A_2$ | Minkowski Distance |
|---|---|---|---|
| 1 | 1.5 | 1.7 | $\sqrt{(1.4-1.5)^2 + (1.6-1.7)^2} = 0.1414$ |
| 2 | 2 | 1.9 | $\sqrt{(1.4-2.0)^2 + (1.6-1.9)^2} = 0.6708$ |
| 3 | 1.6 | 1.8 | $\sqrt{(1.4-1.6)^2 + (1.6-1.6)^2} = 0.2828$ |
| 4 | 1.2 | 1.5 | $\sqrt{(1.4-1.2)^2 + (1.6-1.5)^2} = 0.2236$ |
| 5 | 1.5 | 1.0 | $\sqrt{(1.4-1.5)^2 + (1.6-1.0)^2} = 0.6083$ |

$x = (1.4, 1.6)$
$h = 2$

- $Let\ P_1(x_1, y_1)\ and\ P_2(x_2, y_2)$
- $Minkowski\ Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

## 4. Cosine Similarity

## Cosine Similarity – Solved Example – Data Mining

| | $A_1$ | $A_2$ | Cosine Similarity |
|---|---|---|---|
| 1 | 1.5 | 1.7 | $\frac{(1.4*1.5)+(1.6*1.7)}{\sqrt{(1.5^2+1.7^2)}*\sqrt{(1.4^2+1.6^2)}} = 0.9999$ |
| 2 | 2 | 1.9 | |
| 3 | 1.6 | 1.8 | |
| 4 | 1.2 | 1.5 | |
| 5 | 1.5 | 1.0 | |

$x$ = (1.4, 1.6)

- $Cosine\ Similarity = \frac{A.B}{||A||||B||} = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2}\sqrt{\sum B_i^2}}$

## 5. Supremum Distance

## Supremum Distance – Solved Example – Data Mining

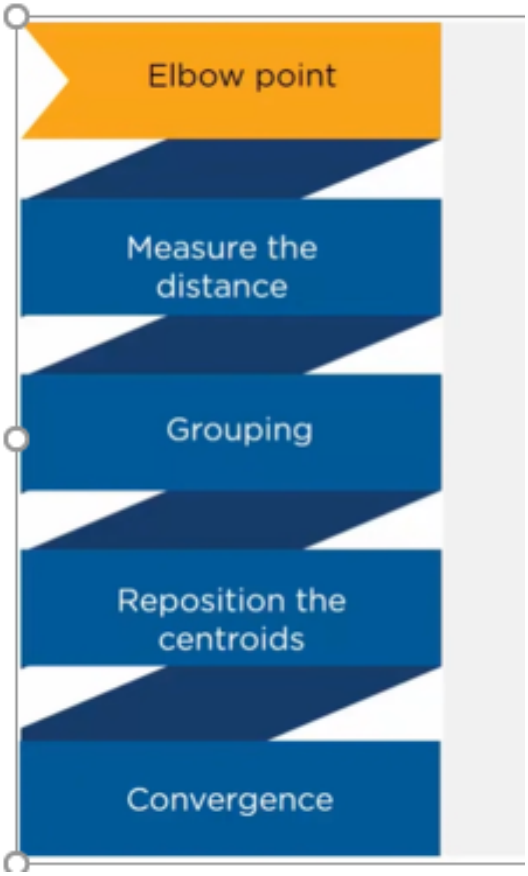
| | $A_1$ | $A_2$ | Supremum Distance |
|---|---|---|---|
| 1 | 1.5 | 1.7 | $max(\|(1.4-1.5)\|, \|(1.6,-1.7)\|) = 0.1$ |
| 2 | 2 | 1.9 | $max(\|(1.4-2.0)\|, \|(1.6,-1.9)\|) = 0.6$ |
| 3 | 1.6 | 1.8 | $max(\|(1.4-1.6)\|, \|(1.6,-1.8)\|) = 0.2$ |
| 4 | 1.2 | 1.5 | $max(\|(1.4-1.2)\|, \|(1.6,-1.5)\|) = 0.2$ |
| 5 | 1.5 | 1.0 | $max(\|(1.4-1.5)\|, \|(1.6,-1.0)\|) = 0.6$ |

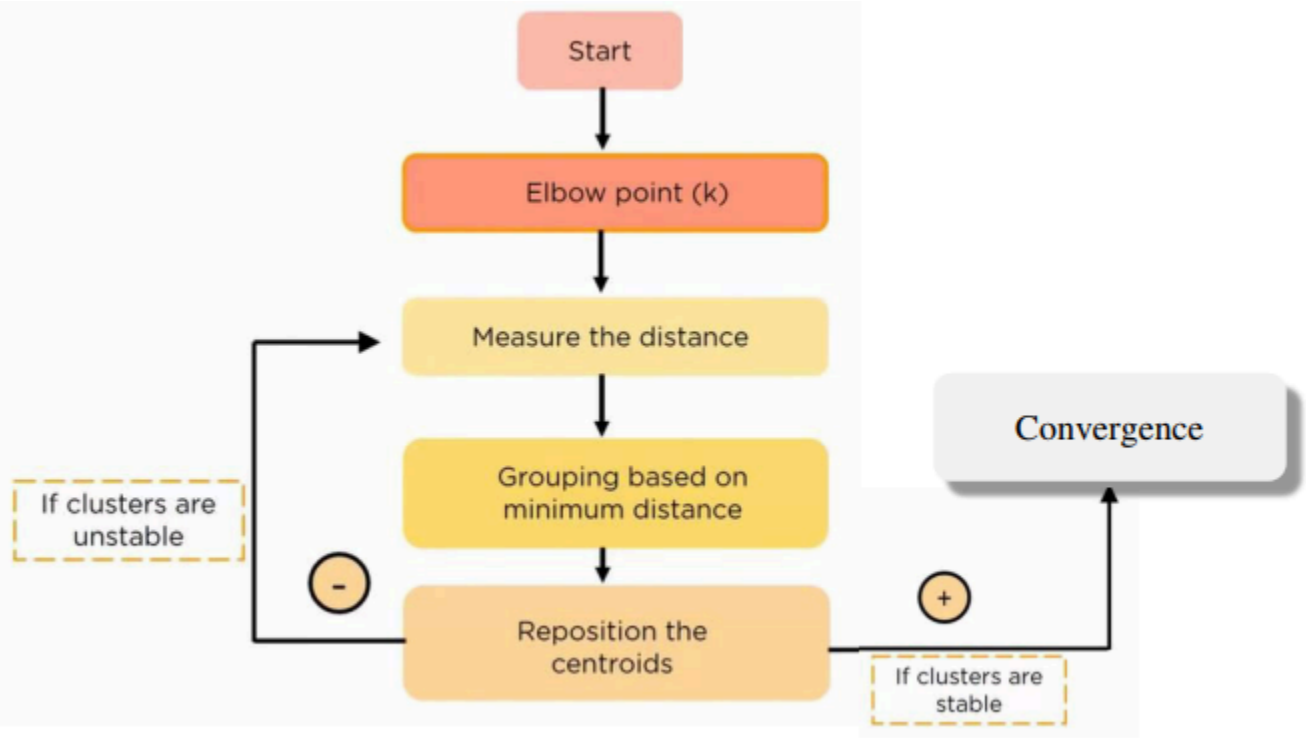
$x$ = (1.4, 1.6)

- $Let\ P_1(x_1, y_1)\ and\ P_2(x_2, y_2)$
- $Supremum\ Distance\ =\ max(|(x_2 - x_1)|, |(y_2 - y_1)|)$

## K-Means clustering working process

For Videos: StatQuest: K-means clustering






## Advantages of K-Means Clustering Algorithm

- ❑ Relatively simple to implement.
- ❑ Scales to large data sets.
- ❑ Guarantees convergence.
- ❑ Can warm-start the positions of centroids.
- ❑ Easily adapts to new examples.
- ❑ Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

## Disadvantages of K-Means Clustering Algorithm

- ❑ Choosing K manually
- ❑ Being dependent on initial values
- ❑ Clustering data of varying sizes and density
- ❑ Clustering outliers
- ❑ Scaling with number of dimensions

## Mathematical Example of M-Mans Clustering

**Random Data**

| X | Y |
|---|---|
| 2 | 4 |
| 2 | 6 |
| 5 | 6 |
| 4 | 7 |
| 8 | 3 |
| 6 | 6 |
| 5 | 2 |
| 5 | 7 |
| 6 | 3 |
| 4 | 4 |

**Sol:**

See the video: K Means Clustering Algorithm | K Means Solved Numerical Example Euclidean Distance …

Or solve from slide

**Iteration - 1**

C1 - Seed Point1 – (1, 5)
C2 - Seed Point2 – (4, 1)
C3 - Seed Point3 – ( 8, 4)

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

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (4.5, 3)
C3 – Centroid – ( 6, 5)

| | | Distance to | | | Cluster |
|---|---|---|---|---|---|
| X | Y | (1, 5) | (4, 1) | (8, 4) | Number |
| 2 | 4 | 1.41 | 3.61 | 6.00 | C1 |
| 2 | 6 | 1.41 | 5.39 | 6.32 | C1 |
| 5 | 6 | 4.12 | 5.10 | 3.61 | C3 |
| 4 | 7 | 3.61 | 6.00 | 5.00 | C1 |
| 8 | 3 | 7.28 | 4.47 | 1.00 | C3 |
| 6 | 6 | 5.10 | 5.39 | 2.83 | C3 |
| 5 | 2 | 5.00 | 1.41 | 3.61 | C2 |
| 5 | 7 | 4.47 | 6.08 | 4.24 | C3 |
| 6 | 3 | 5.39 | 2.83 | 2.24 | C3 |
| 4 | 4 | 3.16 | 3.00 | 4.00 | C2 |

**Iteration - 2**

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (4.5, 3)
C3 – Centroid – ( 6, 5)

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5, 3)
C3 – Centroid – ( 6, 5.5)

| | | Distance to | | | Cluster |
|---|---|---|---|---|---|
| X | Y | (2.66, 5.66) | (4.5, 3) | (6, 5) | Number |
| 2 | 4 | 1.79 | 2.69 | 4.12 | C1 |
| 2 | 6 | 0.74 | 3.91 | 4.12 | C1 |
| 5 | 6 | 2.36 | 3.04 | 1.41 | C3 |
| 4 | 7 | 1.90 | 4.03 | 2.83 | C1 |
| 8 | 3 | 5.97 | 3.5 | 2.83 | C3 |
| 6 | 6 | 3.36 | 3.35 | 1 | C3 |
| 5 | 2 | 4.34 | 1.12 | 3.16 | C2 |
| 5 | 7 | 2.70 | 4.03 | 2.24 | C3 |
| 6 | 3 | 4.27 | 1.5 | 2 | C2 |
| 4 | 4 | 2.13 | 1.12 | 2.24 | C2 |

**Iteration - 3**

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5, 3)
C3 – Centroid – ( 6, 5.5)

| X | Y | Distance to (2.66, 5.66) | Distance to (5, 3) | Distance to (6, 5.5) | Cluster Number |
|---|---|---|---|---|---|
| 2 | 4 | 1.79 | 3.16 | 4.27 | C1 |
| 2 | 6 | 0.74 | 4.24 | 4.03 | C1 |
| 5 | 6 | 2.36 | 3.00 | 1.12 | C3 |
| 4 | 7 | 1.90 | 4.12 | 2.50 | C1 |
| 8 | 3 | 5.97 | 3.00 | 3.20 | C2 |
| 6 | 6 | 3.36 | 3.16 | 0.50 | C3 |
| 5 | 2 | 4.34 | 1.00 | 3.64 | C2 |
| 5 | 7 | 2.70 | 4.00 | 1.80 | C3 |
| 6 | 3 | 4.27 | 1.00 | 2.50 | C2 |
| 4 | 4 | 2.13 | 1.41 | 2.50 | C2 |

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5.75, 3)
C3 – Centroid – ( 5.33, 6.33)

**Iteration - 4**

C1 – Centroid – (2.66, 5.66)
C2 – Centroid – (5.75, 3)
C3 – Centroid – ( 5.33, 6.33)

C1 – Centroid – (2, 5)
C2 – Centroid – (5.75, 3)
C3 – Centroid – ( 5, 6.5)

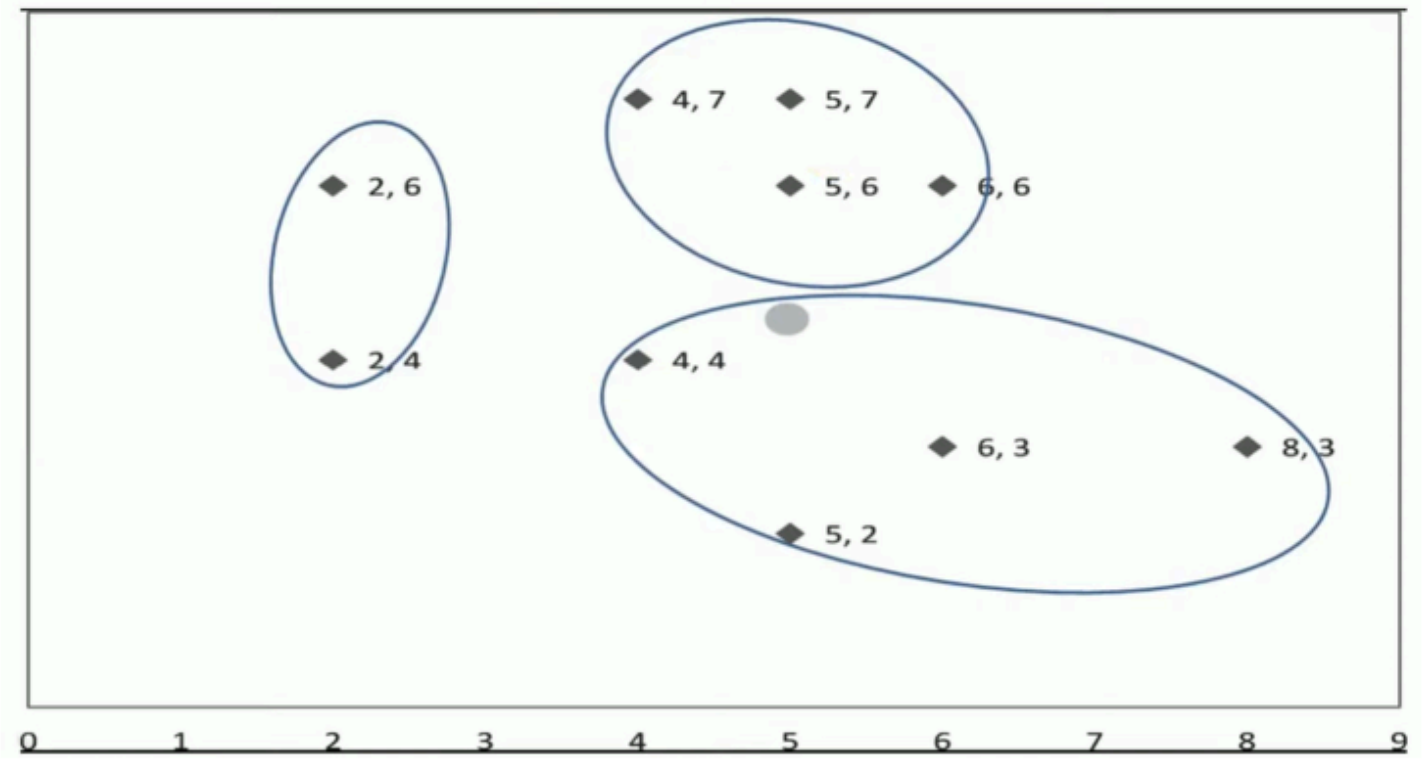
| | | Distance to | | | Cluster |
|---|---|---|---|---|---|
| X | Y | (2.66, 5.66) | (5.75, 3) | (5.33, 6.33) | Number |
| 2 | 4 | 1.79 | 3.88 | 4.06 | C1 |
| 2 | 6 | 0.74 | 4.80 | 3.35 | C1 |
| 5 | 6 | 2.36 | 3.09 | 0.47 | C3 |
| 4 | 7 | 1.90 | 4.37 | 1.49 | C3 |
| 8 | 3 | 5.97 | 2.25 | 4.27 | C2 |
| 6 | 6 | 3.36 | 3.01 | 0.75 | C3 |
| 5 | 2 | 4.34 | 1.25 | 4.34 | C2 |
| 5 | 7 | 2.70 | 4.07 | 0.75 | C3 |
| 6 | 3 | 4.27 | 0.25 | 3.40 | C2 |
| 4 | 4 | 2.13 | 2.02 | 2.68 | C2 |

**Iteration - 5**

C1 – Centroid – (2, 5)
C2 – Centroid – (5.75, 3)
C3 – Centroid – ( 5, 6.5)

No movement of data Points
Hence these are the final positions

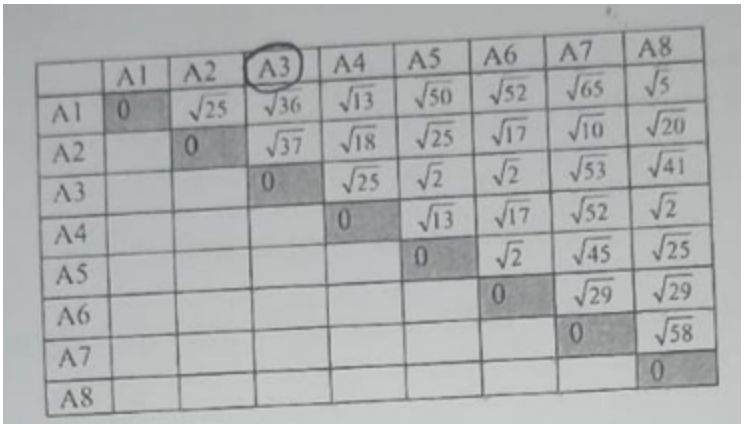
| | | Distance to | | | Cluster |
|---|---|---|---|---|---|
| X | Y | (2, 5) | (5.75, 3) | (5, 6.5) | Number |
| 2 | 4 | 1.00 | 3.88 | 3.91 | C1 |
| 2 | 6 | 1.00 | 4.80 | 3.04 | C1 |
| 5 | 6 | 3.16 | 3.09 | 0.50 | C3 |
| 4 | 7 | 2.83 | 4.37 | 1.12 | C3 |
| 8 | 3 | 6.32 | 2.25 | 4.61 | C2 |
| 6 | 6 | 4.12 | 3.01 | 1.12 | C3 |
| 5 | 2 | 4.24 | 1.25 | 4.50 | C2 |
| 5 | 7 | 3.61 | 4.07 | 0.50 | C3 |
| 6 | 3 | 4.47 | 0.25 | 3.64 | C2 |
| 4 | 4 | 2.24 | 2.02 | 2.69 | C2 |




## Another Example

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters; A1-(2,10), A2-(2,5), A3-(8,4). A4-(5,8), A5-(7,5), A6 (6,4), A7-(1,2), A8-(4,9)

The distance matrix based on the Euclidean distance is given below:



| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 | $\sqrt{25}$ | $\sqrt{36}$ | $\sqrt{13}$ | $\sqrt{50}$ | $\sqrt{52}$ | $\sqrt{65}$ | $\sqrt{5}$ |
| A2 | | 0 | $\sqrt{37}$ | $\sqrt{18}$ | $\sqrt{25}$ | $\sqrt{17}$ | $\sqrt{10}$ | $\sqrt{20}$ |
| A3 | | | 0 | $\sqrt{25}$ | $\sqrt{2}$ | $\sqrt{2}$ | $\sqrt{53}$ | $\sqrt{41}$ |
| A4 | | | | 0 | $\sqrt{13}$ | $\sqrt{17}$ | $\sqrt{52}$ | $\sqrt{2}$ |
| A5 | | | | | 0 | $\sqrt{2}$ | $\sqrt{45}$ | $\sqrt{25}$ |
| A6 | | | | | | 0 | $\sqrt{29}$ | $\sqrt{29}$ |
| A7 | | | | | | | 0 | $\sqrt{58}$ |
| A8 | | | | | | | | 0 |

Suppose that the initial seeds (centers of each cluster) are A1, A4, and A7. Run the k-means algorithm for
1 epoch only. At the end of this epoch show:

a. The new clusters (i.e. the examples belonging to each cluster)
b. The centers of the new clusters
c. Draw a 10 by 10 space with all the 8 points and show the clusters after the first epoch and the new centroids.
d. How many more iterations are needed to converge? Draw the result for each epoch.

Sol:

K Means Clustering Algorithm | K Means Solved Numerical Example Euclidea…

a)
d(a,b) denotes the Eucledian distance between a and b. It is obtained directly from the distance matrix or calculated as follows: $d(a,b)=sqrt((x_b-x_a)^2+(y_b-y_a)^2))$
seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

epoch1 – start:

A1:
d(A1, seed1)=0 as A1 is seed1
d(A1, seed2)= $\sqrt{13}$ >0
d(A1, seed3)= $\sqrt{65}$ >0
➔A1 ∈ cluster1

A2:
d(A2,seed1)= $\sqrt{25}$ = 5
d(A2, seed2)= $\sqrt{18}$ = 4.24
d(A2, seed3)= $\sqrt{10}$ = 3.16 ← smaller
➔ A2 ∈ cluster3

A3:
d(A3, seed1)= $\sqrt{36}$ = 6
d(A3, seed2)= $\sqrt{25}$ = 5 ← smaller
d(A3, seed3)= $\sqrt{53}$ = 7.28
➔ A3 ∈ cluster2

A4:
d(A4, seed1)= $\sqrt{13}$
d(A4, seed2)=0 as A4 is seed2
d(A4, seed3)= $\sqrt{52}$ >0
➔ A4 ∈ cluster2

A5:
d(A5, seed1)= $\sqrt{50}$ = 7.07
d(A5, seed2)= $\sqrt{13}$ = 3.60 ← smaller
d(A5, seed3)= $\sqrt{45}$ = 6.70
➔ A5 ∈ cluster2

A6:
d(A6, seed1)= $\sqrt{52}$ = 7.21
d(A6, seed2)= $\sqrt{17}$ = 4.12 ← smaller
d(A6, seed3)= $\sqrt{29}$ = 5.38
➔ A6 ∈ cluster2

A7:
d(A7, seed1)= $\sqrt{65}$ >0
d(A7, seed2)= $\sqrt{52}$ >0
d(A7, seed3)=0 as A7 is seed3
➔ A7 ∈ cluster3

A8:
d(A8, seed1)= $\sqrt{5}$
d(A8, seed2)= $\sqrt{2}$ ← smaller
d(A8, seed3)= $\sqrt{58}$
➔ A8 ∈ cluster2

end of epoch1

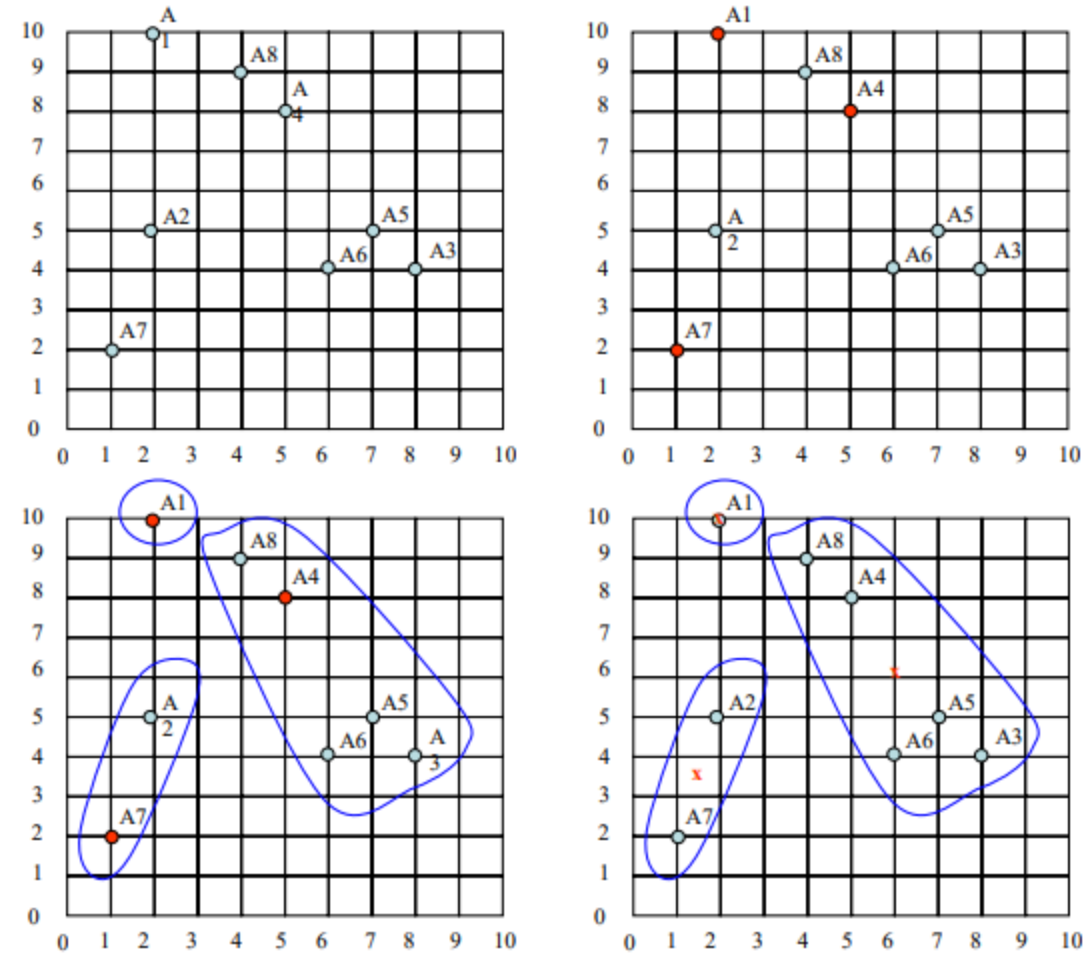
new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}

B.

b) centers of the new clusters:
C1= (2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)

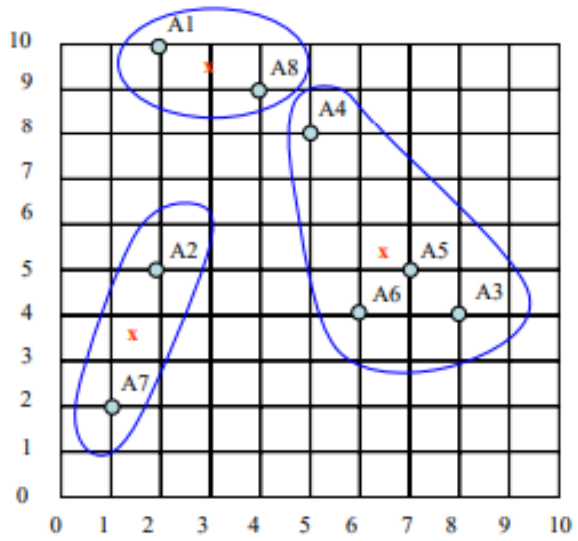
C.

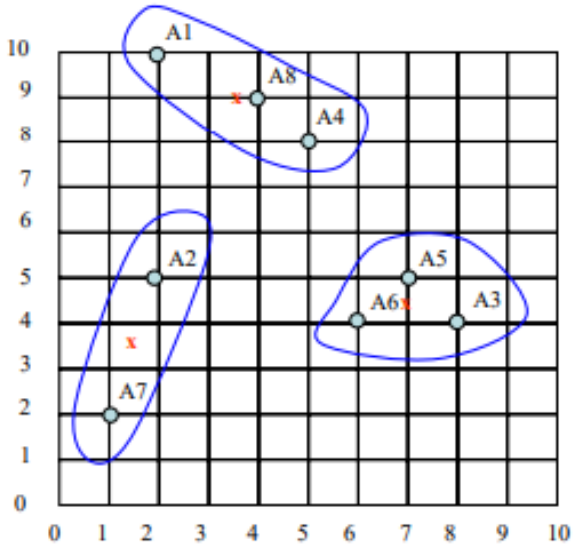
c)




D.

d)
We would need two more epochs. After the 2nd epoch the results would be:
1: {A1, A8}, 2: {A3, A4, A5, A6}, 3: {A2, A7}
with centers C1=(3, 9.5), C2=(6.5, 5.25) and C3=(1.5, 3.5).
After the 3rd epoch, the results would be:
1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7}
with centers C1=(3.66, 9), C2=(7, 4.33) and C3=(1.5, 3.5).




# SVM

For Basic: How Support Vector Machine (SVM) Works Types of SVM Linear SVM Non-Linear SVM ML D…

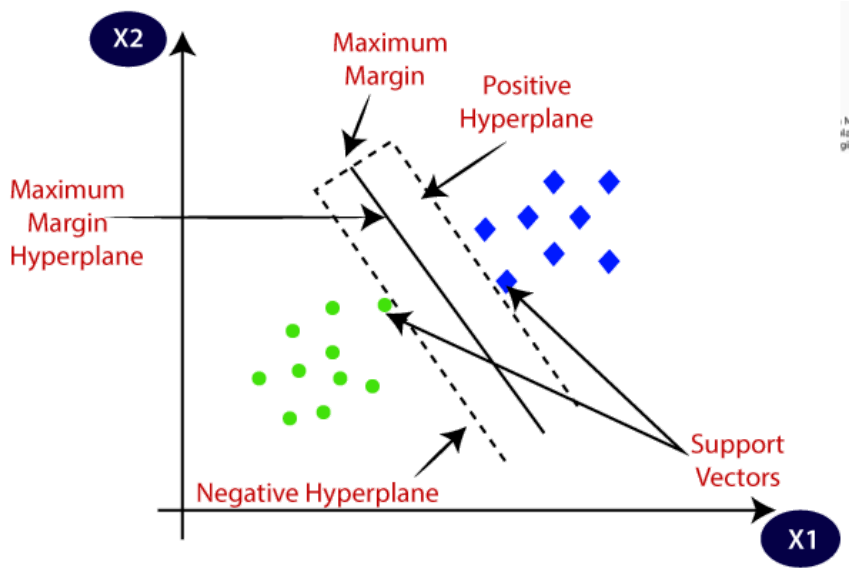
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.

A support vector machine (SVM) is a machine learning algorithm that classifies data by finding a hyperplane that separates data into two classes. SVMs are known for being simple, flexible, and computationally efficient

## Why Used?

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVMs are particularly good at solving binary classification problems, which require classifying the elements of a data set into two groups.

Consider the below diagram in which two different categories are classified using a decision boundary or hyperplane:



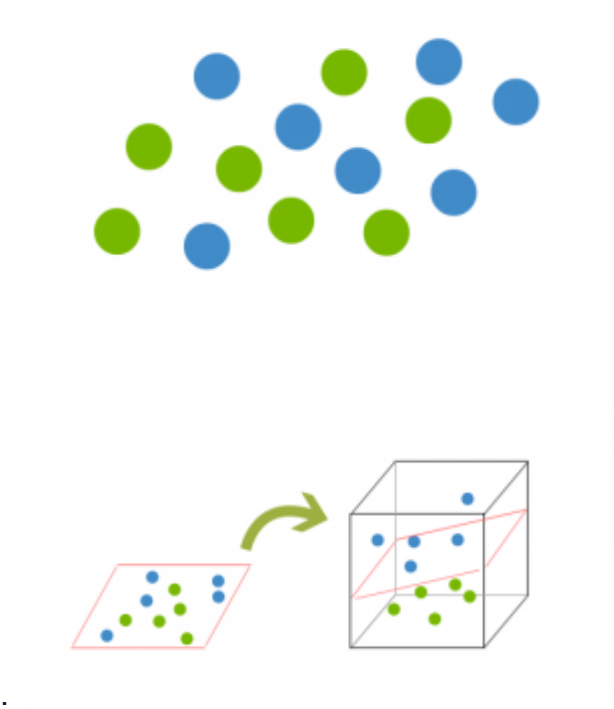

## What is a hyperplane?

A hyperplane is a decision boundary that differentiates the two classes in SVM. A data point falling on either side of the hyperplane can be attributed to different classes. The dimension of the hyperplane depends on the number of input features in the dataset.

### What happens when there is no clear hyperplane?

When there is no clear hyperplane in an SVM (Support Vector Machine), it means the data is not linearly separable, and to classify it, the model needs to utilize a technique called "kernel trick" to

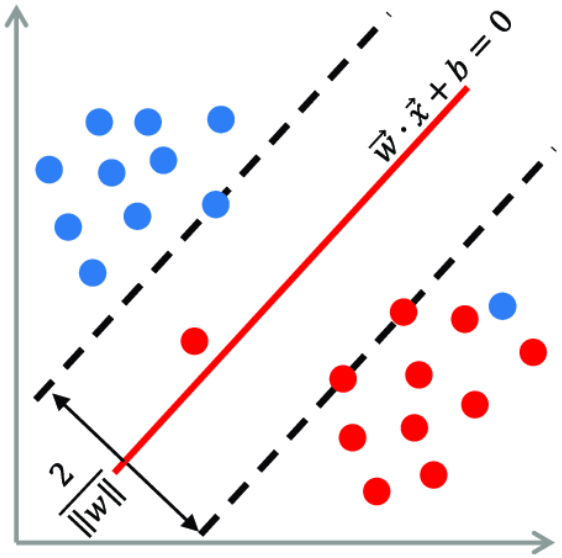
transform the data into a higher dimensional space where a clear separating hyperplane can be found; essentially, this "lifts" the data into a space where separation becomes possible, allowing the SVM to classify the data effectively even when it appears non-linear in the original space.
To classify a dataset, it's necessary to move away from a 2D view of the data to a 3D view.



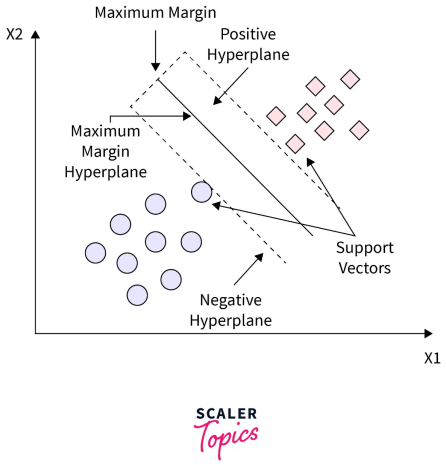
.

## Types of Support Vector Machine (SVM)

**Linear SVM:** Linear SVM is used for linearly separable data.

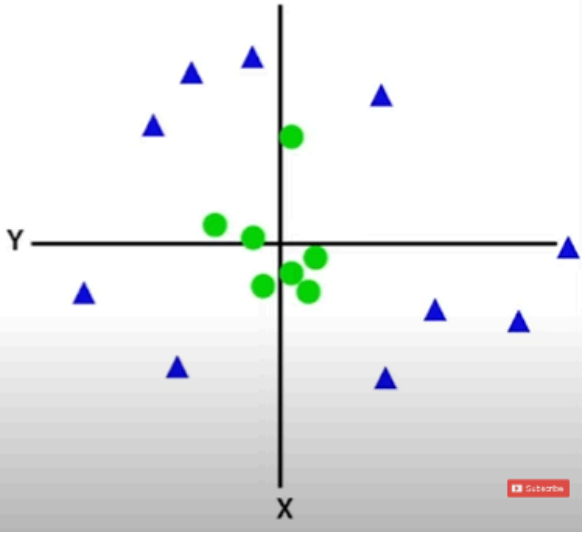


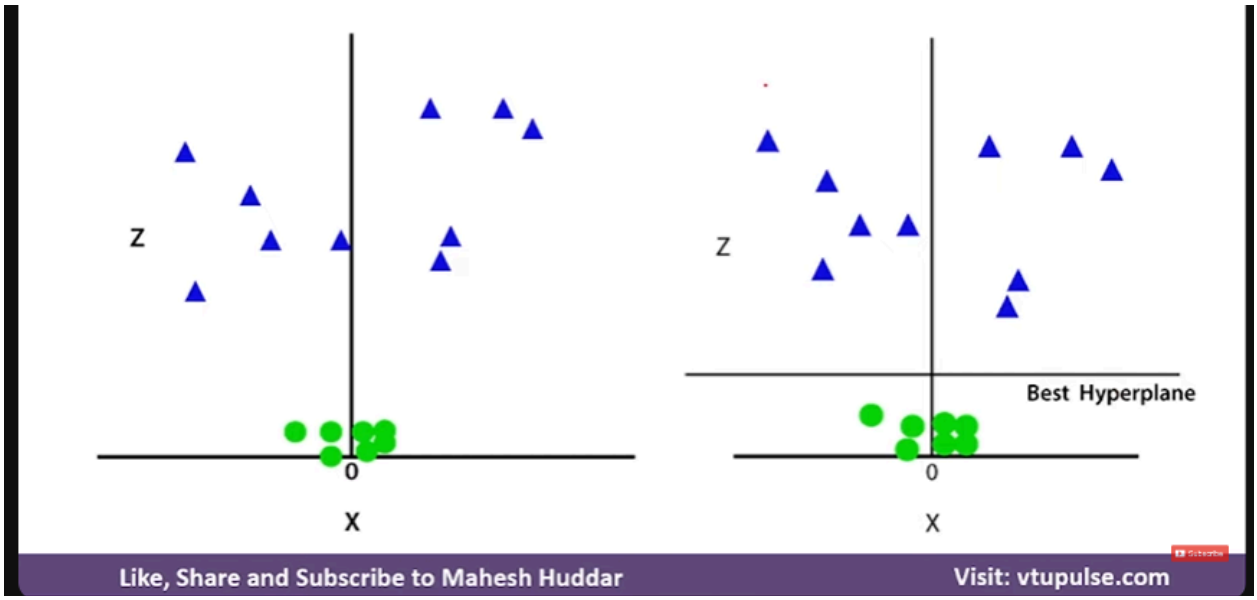

i.e. for a dataset that can be categorized into two categories by utilizing a single straight line. Such data points are termed as linearly separable data, and the classifier is described as a Linear SVM classifier.




**Non-linear SVM:** Non-linear SVM is used for data that are non-linearly separable data

- **Non-Linear SVM:**
- If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.
- So to separate these data points, we need to add one more dimension.
- For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z.
- It can be calculated as: $z=x^2+y^2$



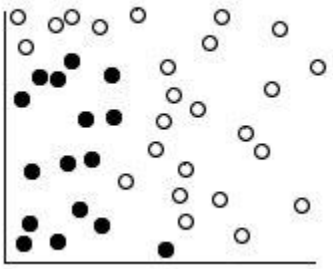

i.e. a straight line cannot be used to classify the dataset. For this, we use something known as a kernel trick that sets data points in a higher dimension where they can be separated using planes or other mathematical functions. Such data points are termed as non-linear data, and the classifier used is termed as a Non-linear SVM classifier.

## How SVM Works

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

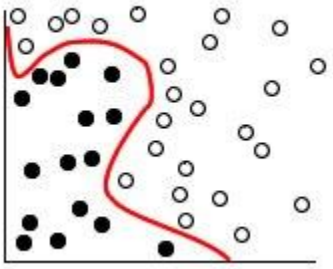
For example, consider the following figure, in which the data points fall into two different categories.

*Figure 1. Original dataset*



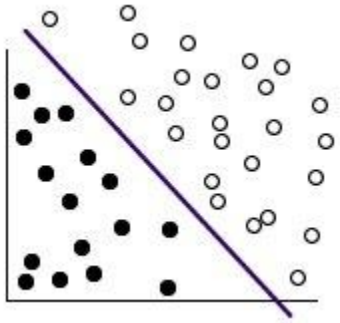
The two categories can be separated with a curve, as shown in the following figure

*Figure 2. Data with separator added*



After the transformation, the boundary between the two categories can be defined by a hyperplane, as shown in the following figure.

*Figure 3. Transformed data*

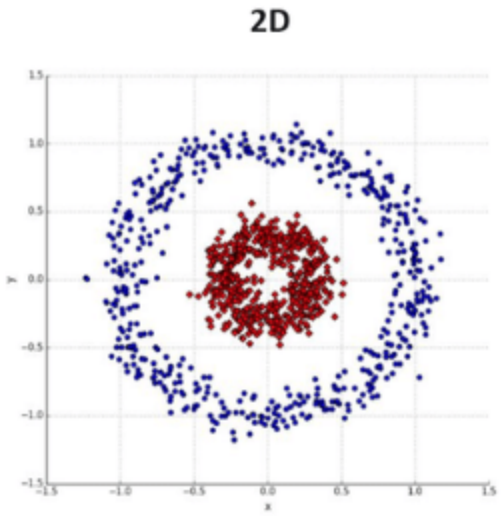


## Why do we use the Kernel Trick in the Support Vector Machine? For example.

The Kernel Trick in Support Vector Machines (SVMs) allows for the classification of non-linearly separable data by implicitly mapping it to a higher-dimensional space, enabling linear separation without explicitly calculating the new coordinates. This approach uses a kernel function to compute similarities between data points, enabling SVMs to solve non-linear problems with a linear classifier

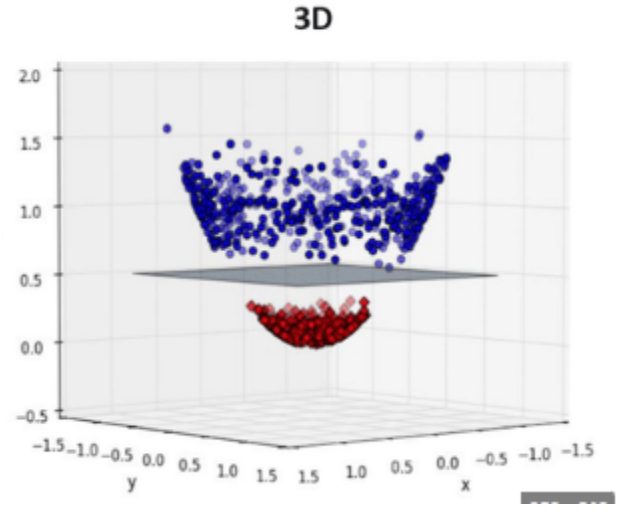
Let's take an example to understand the kernel trick in more detail.

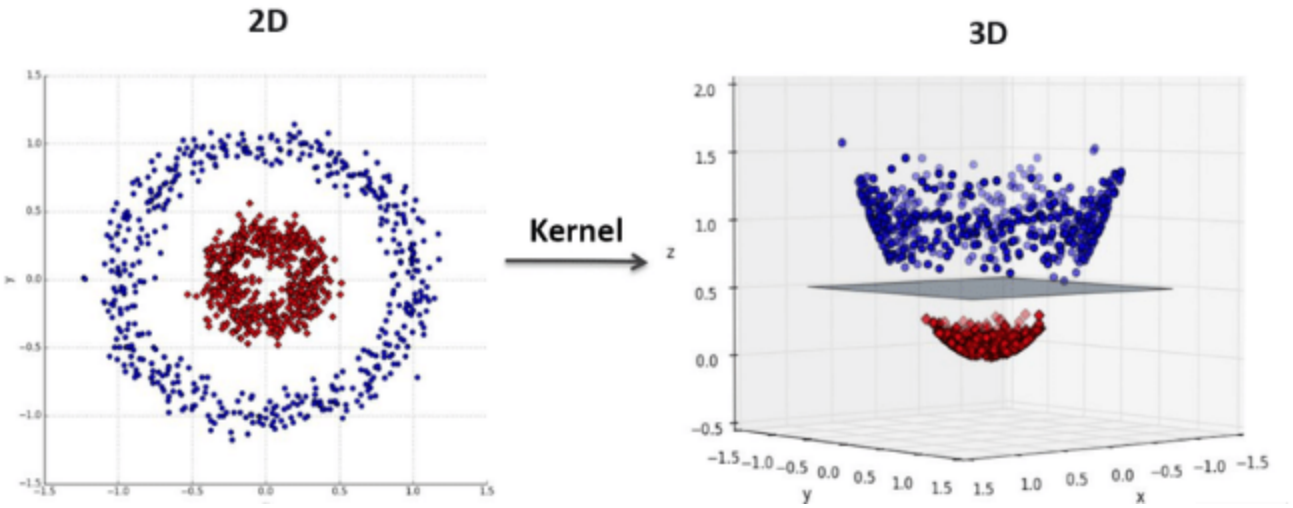
Consider a binary classification problem where we have two classes of data points: red and blue. The data is not linearly separable in the 2D space. We can see this in the plot below:




To make this data linearly separable, we can use the kernel trick.

By applying the kernel trick to the data, we transform it into a higher-dimensional feature space where the data becomes linearly separable. We can see this in the plot below, where the red and blue data points have been separated by a hyperplane in the 3D space:






As we can see, the kernel trick has helped us find a solution for a non-linearly separable dataset.

## Hard and Soft Margin

Hard-margin and soft-margin are two types of Support Vector Machine (SVM) algorithms that differ in how they handle data separability and misclassifications:

**Hard-margin**

Used when data is linearly separable and aims to find a hyperplane that separates the classes without any misclassifications. However, hard-margin SVMs can fail to find a hyperplane when data is not linearly separable, resulting in high training errors.

**Soft-margin**

Used when a linear boundary is not feasible or when some misclassifications are acceptable. Soft-margin SVMs are more flexible and can better generalize to unseen data than hard-margin SVMs. They are also more robust to outliers, which are data points that deviate significantly from the majority of the data

## Mathematical Example

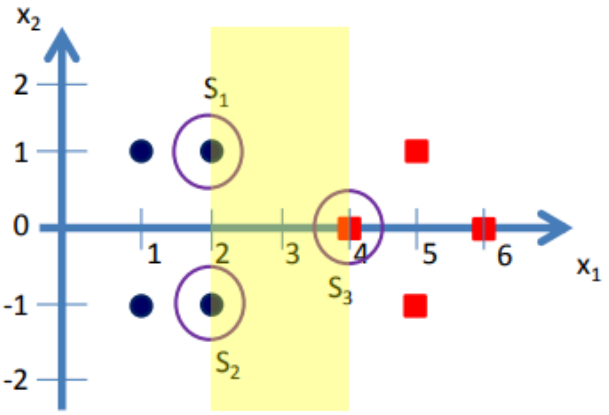
How to draw a hyper plane in Support Vector Machine | Linear SVM – Solved Example by Mahesh Huddar

## Example 1 for Linear SVM

Scholastic
home.video.tutor
@gmail.com

# Support Vector Machines

- Here we select 3 Support Vectors to start with.
- They are $S_1$, $S_2$ and $S_3$.




$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

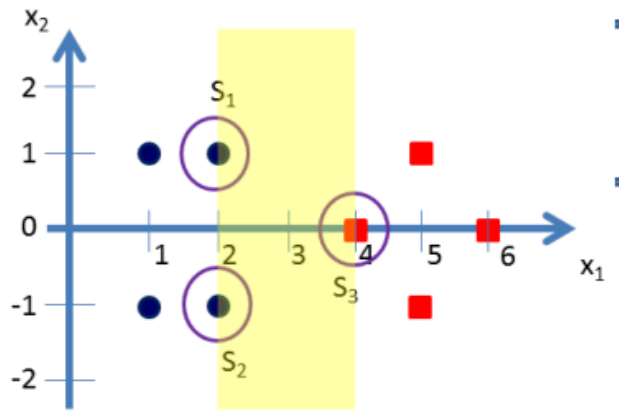
$$S_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

- Here we will use vectors augmented with a 1 as a bias input, and for clarity we will differentiate these with an over-tilde. That is:

$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad \widetilde{S_1} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \qquad \widetilde{S_2} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \qquad \widetilde{S_3} = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$




- Now we need to find 3 parameters $\alpha_1, \alpha_2$, and $\alpha_3$ based on the following 3 linear equations:

$$\alpha_1\widetilde{S_1}.\widetilde{S_1} + \alpha_2\widetilde{S_2}.\widetilde{S_1} + \alpha_3\widetilde{S_3}.\widetilde{S_1} = -1 \ (-ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_2} + \alpha_2\widetilde{S_2}.\widetilde{S_2} + \alpha_3\widetilde{S_3}.\widetilde{S_2} = -1\ (-ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_3} + \alpha_2\widetilde{S_2}.\widetilde{S_3} + \alpha_3\widetilde{S_3}.\widetilde{S_3} = +1 \ (+ve\ class)$$

- Let's substitute the values for $\widetilde{S_1}$, $\widetilde{S_2}$ and $\widetilde{S_3}$ in the above equations.

$$\widetilde{S_1} = \begin{pmatrix}2\\1\\1\end{pmatrix} \qquad \widetilde{S_2} = \begin{pmatrix}2\\-1\\1\end{pmatrix} \qquad \widetilde{S_3} = \begin{pmatrix}4\\0\\1\end{pmatrix}$$

$$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}.\begin{pmatrix}2\\1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}.\begin{pmatrix}2\\1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}.\begin{pmatrix}2\\1\\1\end{pmatrix} = -1$$

$$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}.\begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}.\begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}.\begin{pmatrix}2\\-1\\1\end{pmatrix} = -1$$

$$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}.\begin{pmatrix}4\\0\\1\end{pmatrix} + \alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}.\begin{pmatrix}4\\0\\1\end{pmatrix} + \alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}.\begin{pmatrix}4\\0\\1\end{pmatrix} = +1$$

$$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}.\begin{pmatrix}2\\1\\1\end{pmatrix}+\alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}.\begin{pmatrix}2\\1\\1\end{pmatrix}+\alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}.\begin{pmatrix}2\\1\\1\end{pmatrix}=-1$$

$$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}.\begin{pmatrix}2\\-1\\1\end{pmatrix}+\alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}.\begin{pmatrix}2\\-1\\1\end{pmatrix}+\alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}.\begin{pmatrix}2\\-1\\1\end{pmatrix}=-1$$

$$\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}.\begin{pmatrix}4\\0\\1\end{pmatrix}+\alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}.\begin{pmatrix}4\\0\\1\end{pmatrix}+\alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}.\begin{pmatrix}4\\0\\1\end{pmatrix}=+1$$

- After simplification we get:

$$6\alpha_1+4\alpha_2+9\alpha_3=-1$$

$$4\alpha_1+6\alpha_2+9\alpha_3=-1$$

$$9\alpha_1+9\alpha_2+17\alpha_3=+1$$

- Simplifying the above 3 simultaneous equations we get: $\alpha_1=\alpha_2=-3.25$ and $\alpha_3=3.5$.

$\alpha_1=\alpha_2=$ -3.25 and $\alpha_3=$ 3.5

$$\widetilde{S_1}=\begin{pmatrix}2\\1\\1\end{pmatrix}\quad \widetilde{S_2}=\begin{pmatrix}2\\-1\\1\end{pmatrix}\quad \widetilde{S_3}=\begin{pmatrix}4\\0\\1\end{pmatrix}$$

- The hyper plane that discriminates the positive class from the negative class is give by:

$$\widetilde{w}=\sum_i \alpha_i \widetilde{S_i}$$

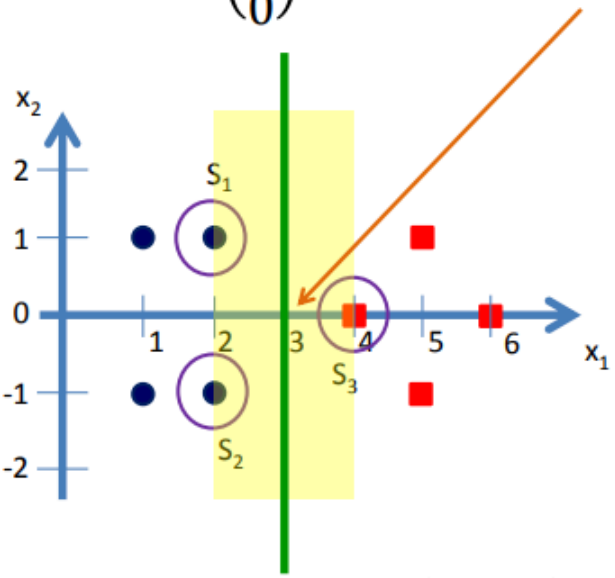
- Substituting the values we get:

$$\widetilde{w}=\alpha_1\begin{pmatrix}2\\1\\1\end{pmatrix}+\alpha_2\begin{pmatrix}2\\-1\\1\end{pmatrix}+\alpha_3\begin{pmatrix}4\\0\\1\end{pmatrix}$$

$$\widetilde{w}=(-3.25).\begin{pmatrix}2\\1\\1\end{pmatrix}+(-3.25).\begin{pmatrix}2\\-1\\1\end{pmatrix}+(3.5).\begin{pmatrix}4\\0\\1\end{pmatrix}=\begin{pmatrix}1\\0\\-3\end{pmatrix}$$

- Therefore the separating hyper plane equation $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and offset $b = -3$.


- $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and offset $b = -3$.

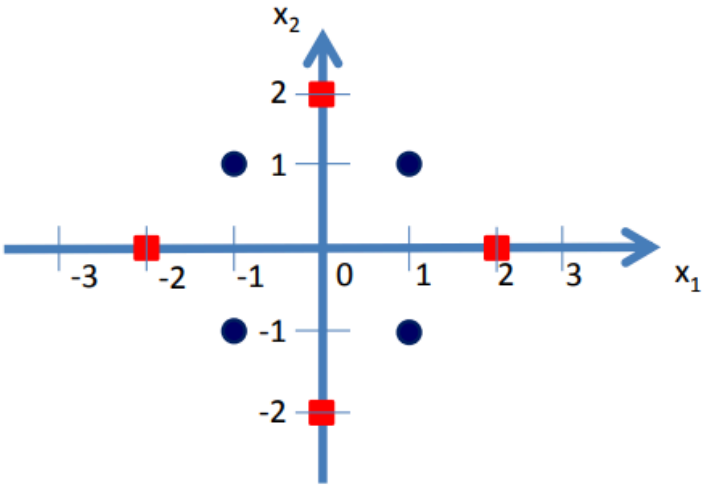




- This is the expected decision surface of the LSVM.

How to solve linear equations using a calculator:

How to solve linear equations with 3 variables using scientific calculator CASIO fx-991ES PLUS

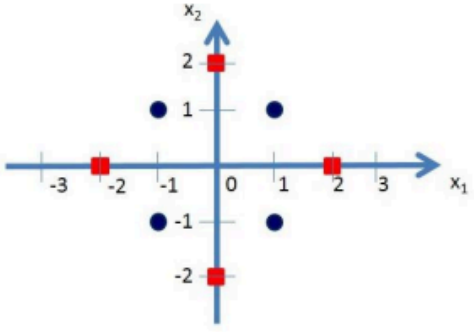
## Example-2 for Non-Linear SVM




- Blue class vectors are: $\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}-1\\1\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1\\-1\end{pmatrix}$
- Red class vectors are: $\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}-2\\0\end{pmatrix}, \begin{pmatrix}0\\-2\end{pmatrix}$

- Here we need to find a non-linear mapping function $\Phi$ which can transform these data in to a new feature space where a seperating hyperplane can be found.
- Let us consider the following mapping function.

- $$\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases}\begin{pmatrix}6 - x_1 + (x_1 - x_2)^2\\6 - x_2 + (x_1 - x_2)^2\end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} \geq 2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$$



- Now let us transform the blue and red calss vectors using the non-linear mapping function $\Phi$ .

- $$\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases}\begin{pmatrix}6 - x_1 + (x_1 - x_2)^2\\6 - x_2 + (x_1 - x_2)^2\end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} \geq 2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$$

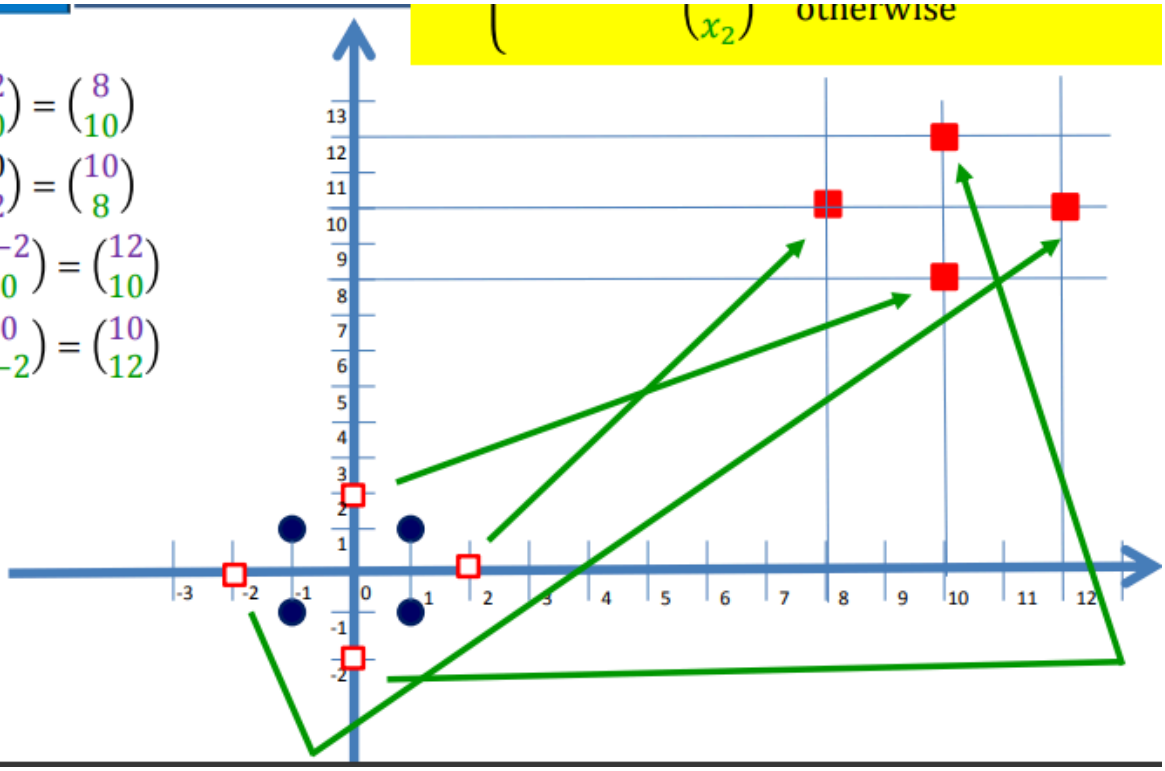
- Blue class vectors are: $\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}-1\\1\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1\\-1\end{pmatrix}$ no change since $\sqrt{x_1^2 + x_2^2} < 2$ for all the vectors

- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases}\begin{pmatrix}6-x_1+(x_1-x_2)^2\\6-x_2+(x_1-x_2)^2\end{pmatrix} & \text{if } \sqrt{x_1^2+x_2^2} \geq 2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$
- Let us take Red class vectors : $\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}-2\\0\end{pmatrix}, \begin{pmatrix}0\\-2\end{pmatrix}$

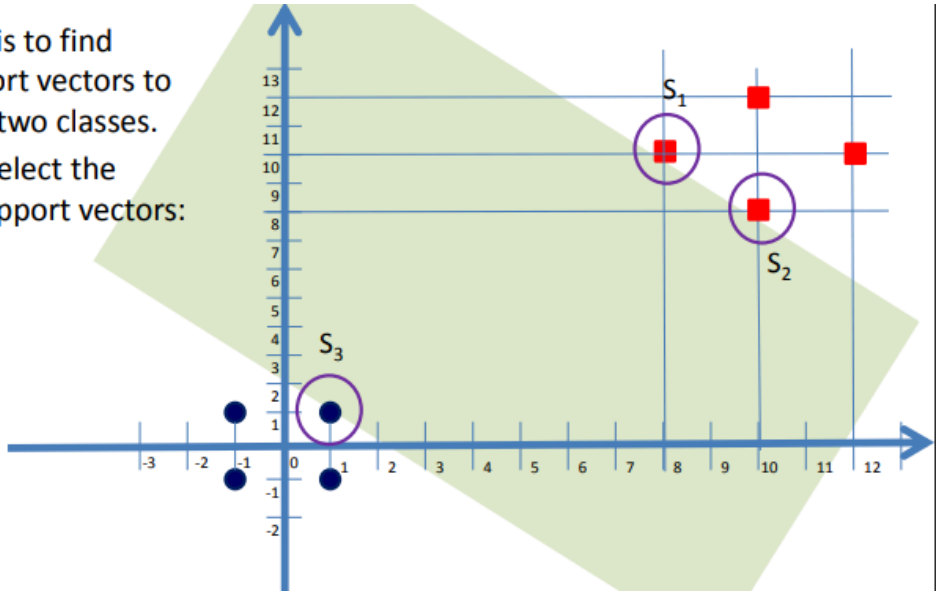
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}6-2+(2-0)^2\\6-0+(2-0)^2\end{pmatrix} = \begin{pmatrix}8\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}6-0+(0-2)^2\\6-2+(0-2)^2\end{pmatrix} = \begin{pmatrix}10\\8\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}6+2+(-2-0)^2\\6-0+(-2-0)^2\end{pmatrix} = \begin{pmatrix}12\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\-2\end{pmatrix} = \begin{pmatrix}6-0+(0+2)^2\\6+2+(0+2)^2\end{pmatrix} = \begin{pmatrix}10\\12\end{pmatrix}$

$\begin{pmatrix}x_2\end{pmatrix}$ otherwise

- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}8\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}10\\8\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}12\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\-2\end{pmatrix} = \begin{pmatrix}10\\12\end{pmatrix}$



- Now our task is to find suitable support vectors to classify these two classes.
- Here we will select the following 3 support vectors:
- $S_1 = \begin{pmatrix} 8 \\ 10 \end{pmatrix}$,
- $S_2 = \begin{pmatrix} 10 \\ 8 \end{pmatrix}$,
- and $S_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$




Here we will use vectors augmented with a 1 as a bias input, and for clarity we will differentiate these with an over-tilde. That is:

$$S_1 = \begin{pmatrix} 8 \\ 10 \end{pmatrix} \qquad \widetilde{S_1} = \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 10 \\ 8 \end{pmatrix} \qquad \widetilde{S_2} = \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \widetilde{S_3} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

- Now we need to find 3 parameters $\alpha_1, \alpha_2$, and $\alpha_3$ based on the following 3 linear equations:

$$\alpha_1\widetilde{S_1}.\widetilde{S_1} + \alpha_2\widetilde{S_2}.\widetilde{S_1} + \alpha_3\widetilde{S_3}.\widetilde{S_1} = +1 \ (+ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_2} + \alpha_2\widetilde{S_2}.\widetilde{S_2} + \alpha_3\widetilde{S_3}.\widetilde{S_2} = +1\ (+ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_3} + \alpha_2\widetilde{S_2}.\widetilde{S_3} + \alpha_3\widetilde{S_3}.\widetilde{S_3} = -1 \ (-ve\ class)$$

- Let's substitute the values for $\widetilde{S_1}$, $\widetilde{S_2}$ and $\widetilde{S_3}$ in the above equations.

$$\widetilde{S_1} = \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} \quad \widetilde{S_2} = \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} \quad \widetilde{S_3} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}.\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}.\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} = +1$$

$$\alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}.\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}.\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} = +1$$

$$\alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1$$

- After multiplication we get:

$$165\,\alpha_1 + 161\,\alpha_2 + 19\,\alpha_3 = +1$$

$$161\,\alpha_1 + 165\,\alpha_2 + 19\,\alpha_3 = +1$$

$$19\,\alpha_1 + 19\,\alpha_2 + 3\,\alpha_3 = -1$$

- Simplifying the above 3 simultaneous equations we get: $\alpha_1 = \alpha_2 = 0.859$ and $\alpha_3 = -1.4219$.

- The hyper plane that discriminates the positive class from the negative class is given by:

$$\tilde{w} = \sum_i \alpha_i \tilde{S}_i$$

- Substituting the values we get:

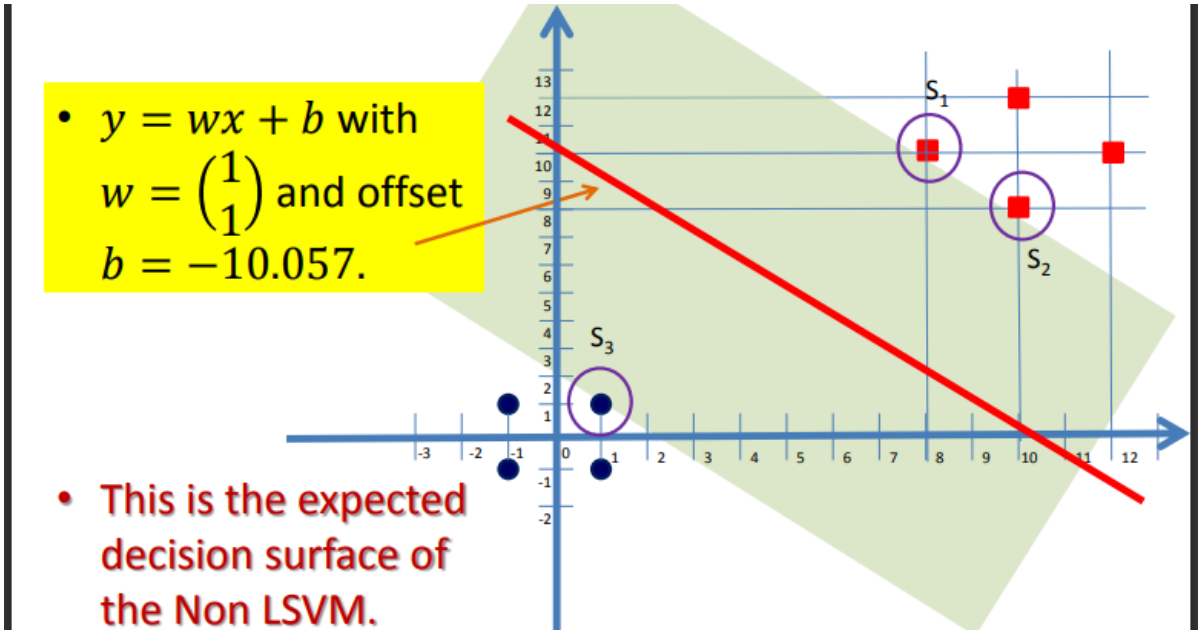
$$\tilde{w} = \alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\tilde{w} = (0.0859). \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + (0.0859). \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + (-1.4219). \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.1243 \\ 0.1243 \\ -1.2501 \end{pmatrix}$$

- Therefore the separating hyper plane equation

$y = wx + b$ with $w = \begin{pmatrix} 0.1243/0.1243 \\ 0.1243/0.1243 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

and an offset $b = -\frac{1.2501}{0.1243} = -10.057$.




# Ensemble learning (Bagging and Boosting)

Introduction to Ensemble Learning with Real Life Examples | Machine⚙️ Learning

Def:

Ensemble learning is a machine learning technique that combines the predictions from multiple models to improve the overall performance of a model.

# Ensembles

- Ensemble machine learning methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms.
- Many of the popular modern machine learning algorithms are actually ensembles. For example, Random Forest (RF) and Gradient Boosting Machine (GBM) are both ensemble learners.
- Both bagging (e.g. Random Forest) and boosting (e.g. GBM) are methods for ensembling that take a collection of weak learners (e.g. decision tree) and form a single, strong learner.

# Different types of Ensembles

**How can we design/create ML Ensembles?**

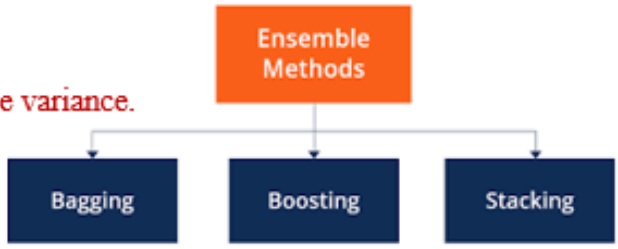
Different learning algorithms (Collaborate)

Similar algorithms with different choice for parameters

Data set with different features (e.g. random subspace)

Data set = different subsets (e.g. bagging, boosting, stacking)

Bagging helps decrease variance.

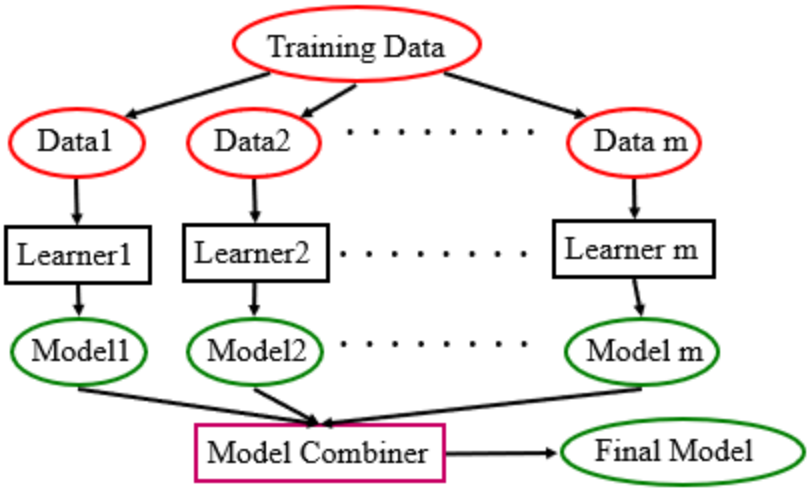



Boosting helps to decrease the model's bias.

# Ensemble Learning

Learn multiple alternative definitions of a concept using different training data or different learning algorithms.

Combine decisions of multiple definitions, e.g. using weighted voting.




14

# Key Ensemble Questions

## Which components to combine?

- Different learning algorithms
- Same learning algorithm trained in different ways
- Same learning algorithm trained the same way

## How to combine classifications?

- Majority vote
- Weighted (confidence of classifier) vote
- Weighted (confidence in classifier) vote
- Learned combiner

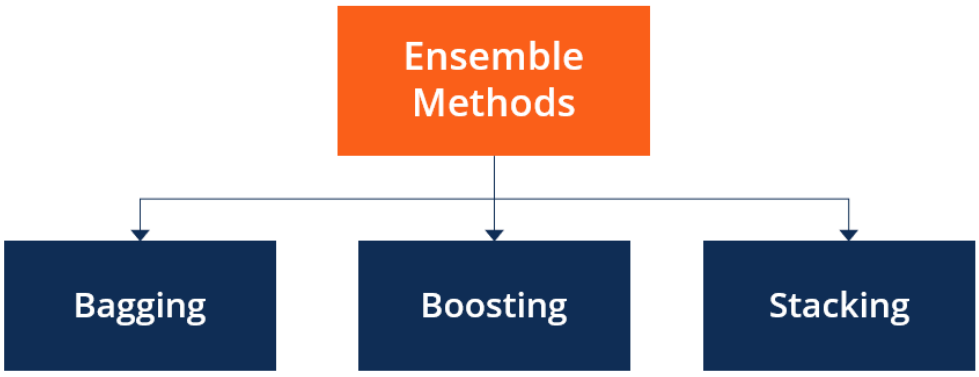
## What makes a good (accurate) ensemble?

- Produce better results than single traditional ML
- Utilize similar CPU time as regular ML algorithms

# Diversity of Ensemble

- Objective: create many classifiers, and combine their outputs to improve the performance of a single classifier
- Intuition: if each classifier makes different errors, then their strategic combination can reduce the total error!
- Need base classifiers whose decision boundaries are adequately different from those of others
  - Such a set of classifiers is said to be ***diverse***
- How to achieve classifier diversity?
  - Use different training sets to train individual classifiers
  - Use different features with each classifier
  - Use Heterogenous vs Homogenous classifiers
- How to obtain different training sets?
  - Resampling techniques: bootstrapping or bagging, training subsets are drawn randomly, usually with replacement, from the entire training set

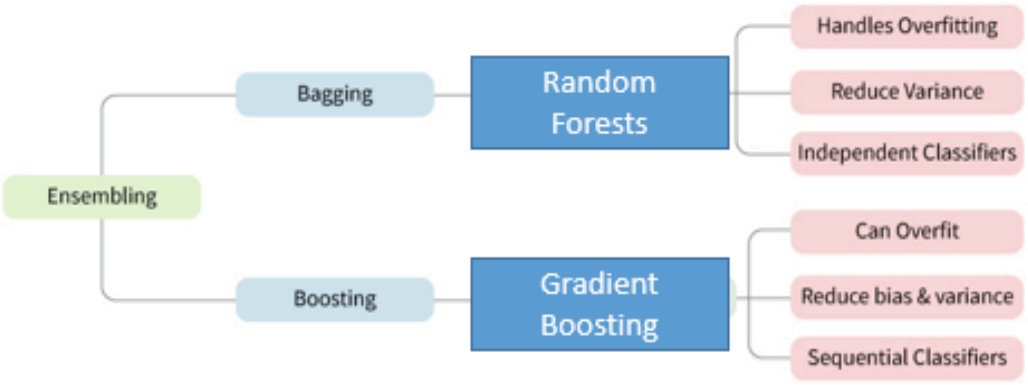
## Types of Ensemble



# Ensembles

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms

- Bagging
- Boosting


Learn different parts of the concept with different models and combine

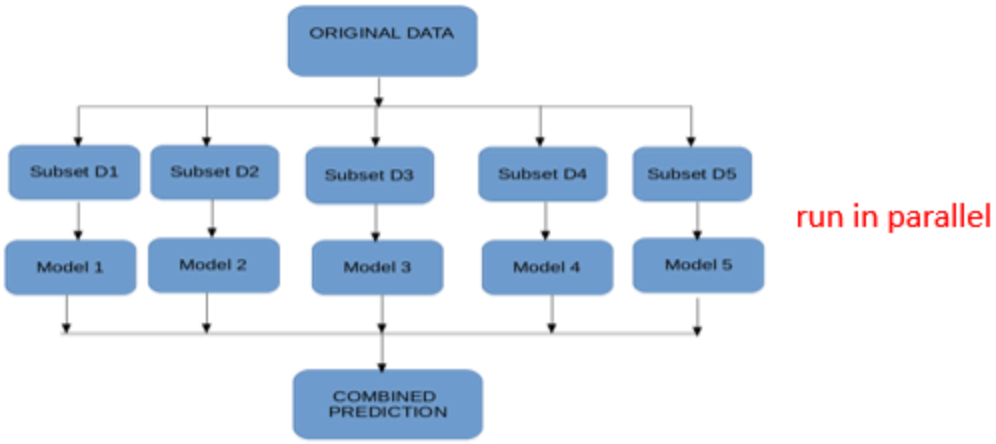
29

# (1) Bagging

- Bagging, short for Bootstrap aggregating, is one of the earliest ensemble based algorithms.
- It is also one of the most intuitive and simplest to implement, with a surprisingly good performance
- Use bootstrapped replicas of the training data; large number of (say 200) training subsets are randomly drawn - with replacement - from the entire training data
- Each resampled training set is used to train a different classifier of the same type (Homogenous)
- Individual classifiers are combined by taking a majority vote of their decisions
- Random Forest is one of the most popular and most powerful machine learning bagging techniques.

# (1) Bagging: Bootstrapping

- Multiple subsets are created from the original dataset, selecting observations with replacement.
- A base model (weak model) is created on each of these subsets.
- The models run in parallel and are independent of each other.
- The final predictions are determined by combining the predictions from all the models.




Def:

Bagging, also known as bootstrap aggregation, is the ensemble learning method that is commonly used to reduce variance within a noisy data set.

## Advantages and Disadvantages of Bagging

The advantages of Bagging are -

**1. Easier for implementation:**

Python libraries, including scikit-examine (sklearn), make it easy to mix the predictions of base beginners or estimators to enhance model performance. Their documentation outlines the available modules you can leverage for your model optimization.

**2. Variance reduction:**

The Bagging can reduce the variance inside a getting-to-know set of rules which is especially helpful with excessive-dimensional facts, where missing values can result in better conflict, making it more liable to overfitting and stopping correct generalization to new datasets.

**Disadvantages of Bagging are -**

**1. Flexible less:**

As a method, Bagging works particularly correctly with algorithms that are much less solid. One that can be more stable or a problem with high amounts of bias does now not provide an awful lot of gain as there is less variation in the dataset of the version. As noted within the hands-On guide for machine learning, "the bagging is a linear regression version will efficaciously just return the original predictions for huge enough b."

**2. Loss of interpretability:**

The Bagging slows down and grows extra in depth because of the quantity of iterations growth. accordingly, it is no longer adequately suitable for actual-time applications. Clustered structures or large processing cores are perfect for quickly growing bagged ensembles on massive look-at units.

**3. Expensive for computation:**

The Bagging is tough to draw unique business insights via Bagging because of the averaging concerned throughout predictions. While the output is more precise than any person's information point, a more accurate or whole dataset may yield greater precision within a single classification or regression model.

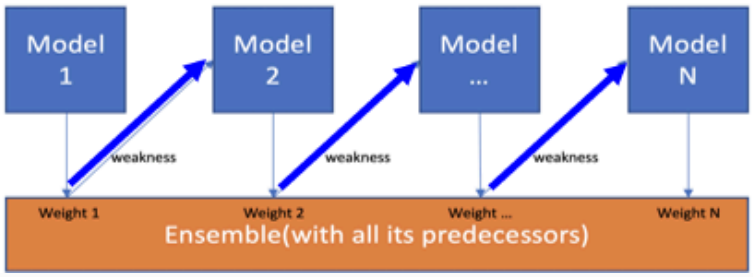
# Boosting

## Def

Boosting is an ensemble learning method that combines a set of weak learners into a strong learner to minimize training errors.

# Recall Boosting

- Boosting algorithms convert weak learners into strong learners.
- Boosting is a sequential process; i.e., trees are grown using the information from a previously grown tree, one after the other.
- This process slowly learns from data and tries to improve its prediction in subsequent iterations.




## Advantages of Boosting

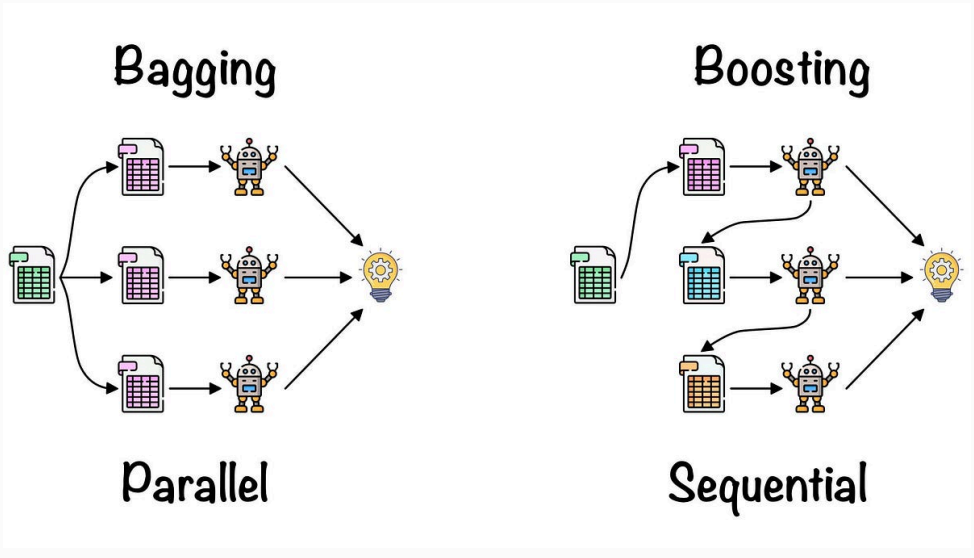
**Improved Accuracy** – Boosting can improve the accuracy of the model by combining several weak models' accuracies and averaging them for regression or voting over them for classification to increase the accuracy of the final model.

**Robustness to Overfitting** – Boosting can reduce the risk of overfitting by reweighting the inputs that are classified wrongly.

**Better handling of imbalanced data** – Boosting can handle the imbalance data by focusing more on the data points that are misclassified

**Better Interpretability** – Boosting can increase the interpretability of the model by breaking the model decision process into multiple processes.

## Difference Bagging vs  Boosting



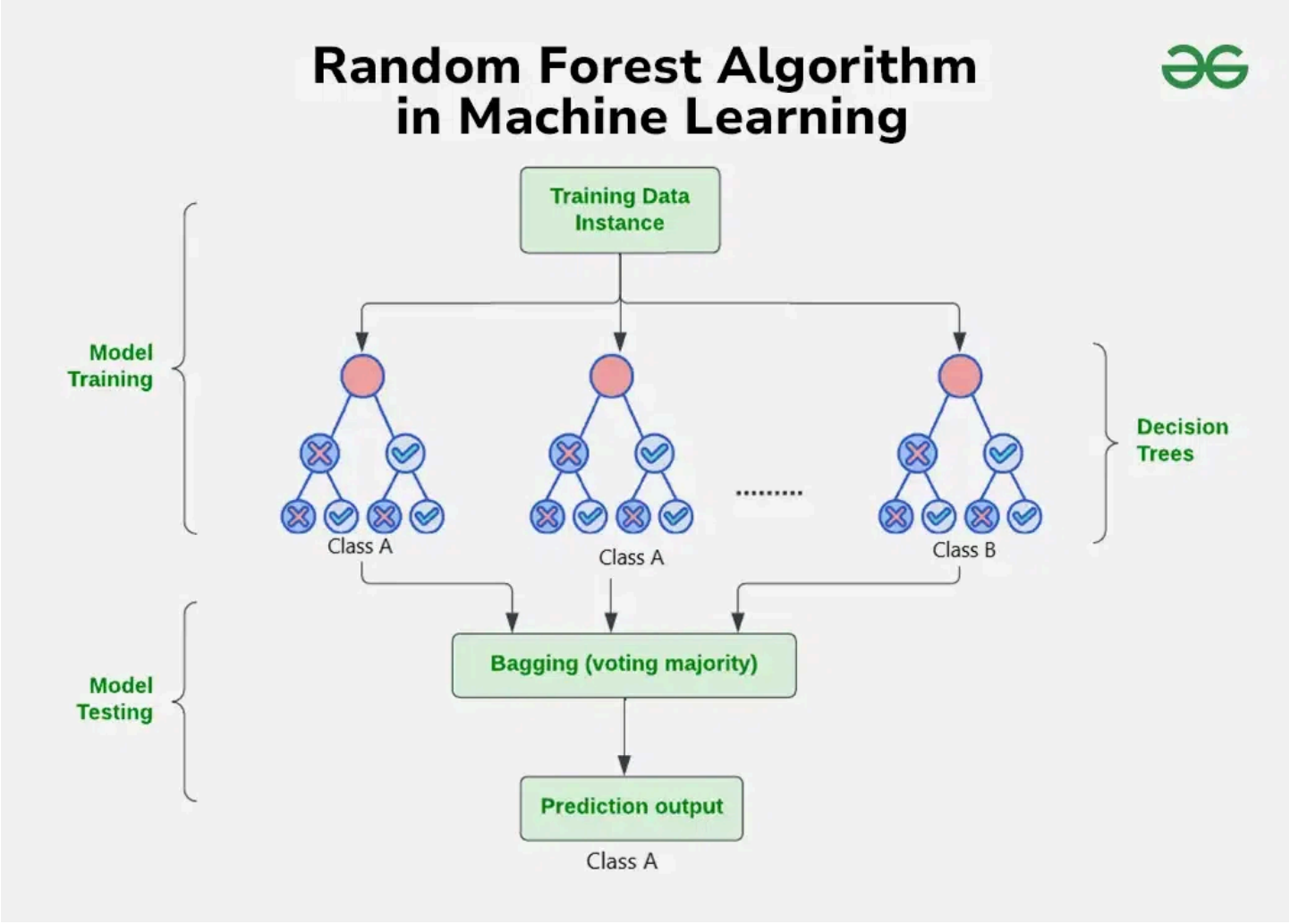
## Difference Between Bagging and Boosting: Bagging vs Boosting

| | Bagging | Boosting |
|---|---|---|
| **Basic Concept** | Combines multiple models trained on different subsets of data. | Train models sequentially, focusing on the error made by the previous model. |
| **Objective** | To reduce variance by averaging out individual model error. | Reduces both bias and variance by correcting misclassifications of the previous model. |
| **Data Sampling** | Use Bootstrap to create subsets of the data. | Re-weights the data based on the error from the previous model, making the next models focus on misclassified instances. |
| **Model Weight** | Each model serves equal weight in the final decision. | Models are weighted based on accuracy, i.e., better-accuracy models will have a higher weight. |
| **Error Handling** | Each model has an equal error rate. | It gives more weight to instances with higher error, making subsequent model focus on them. |
| **Overfitting** | Less prone to overfitting due to average mechanism. | Generally not prone to overfitting, but it can be if the number of the model or the iteration is high. |
| **Performance** | Improves accuracy by reducing variance. | Achieves higher accuracy by reducing both bias and variance. |
| **Common Algorithms** | Random Forest | AdaBoost, XGBoost, Gradient Boosting Mechanism |
| **Use Cases** | Best for high variance, and low bias models. | Effective when the model needs to be adaptive to errors, suitable for both bias and variance errors. |

# Random Forest

Videos: ▶ Random Forest 🌳 in Machine Learning 👩‍👩‍👧‍👧

Random forest is a commonly used machine learning algorithm that combines the output of multiple decision trees to reach a single result




## Why was Random Forest used?

- Reduced overfitting
- High accuracy
- Versatile
- Easy to use
- Fast

# RF Features/Advantages

The advantages of random forest are:

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

# RF: Disadvantages

- Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.
- For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

## Choosing between Random Forest and SVM

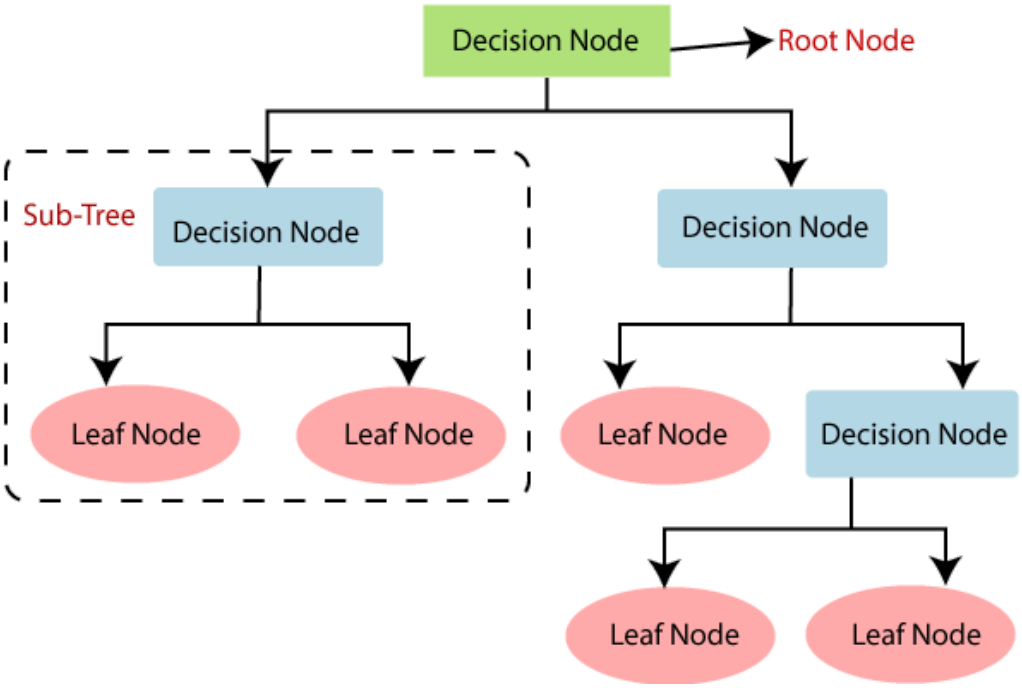
1. **Dataset size and complexity:** Random Forests tend to work well for large datasets with high dimensional data due to their ability to handle substantial amounts of data effectively and the fact that they use feature randomness during tree formation. On the other hand, SVM works well for a well-structured small to medium-sized dataset with low dimensional data.
2. **Dataset type:** Random Forest captures complex non-linear patterns in data easily and can also find correlations between features, while, SVM works well when the classes can be separated linearly but using kernel trick SVMs can handle non-linear data.
3. **Computational Efficiency:** Random Forests are computationally efficient because they allow for the parallel training of several decision trees within the forest.
4. **Margin Considerations:** SVMs optimize for maximal margin, offering a strong and clear decision boundary, if a distinct margin between classes is essential.
5. **Feature Importance Ranking:** The feature importance ranking that Random Forests offers is useful for figuring out how important each feature is about the other in the dataset.
6. **Interpretability:** Random Forests offer an overall model interpretability, but SVMs may be chosen if interpretability is important for your application because of their distinct decision limits.
7. **Hyperparameter Tuning**: In certain situations, Random Forests are more user-friendly than Support Vector Machines (SVMs) because they often require less hyperparameter adjustment.
8. **Training Time Sensitivity**: Take into account the size of your dataset and the parallelization potential of each algorithm if training time is a crucial consideration.
9. **Single vs. Ensemble:** SVMs are single models, but Random Forests are an ensemble of decision trees. Whether or not an ensemble strategy is advantageous for your particular challenge may influence your decision.

# Decision Tree

### Def

A decision tree is a machine-learning model that uses a tree-like structure to make predictions or categorize data

Video for basic: Lec-9: Introduction to Decision Tree 🌲 with Real life examples




## Why use Decision Trees?

I. Decision Trees usually mimic human thinking ability while making a decision, making it easy to understand.
II. The logic behind the decision tree can be easily understood because it shows a tree-like structure.

## Decision Tree Terminologies

**Root Node:** The root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

**Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

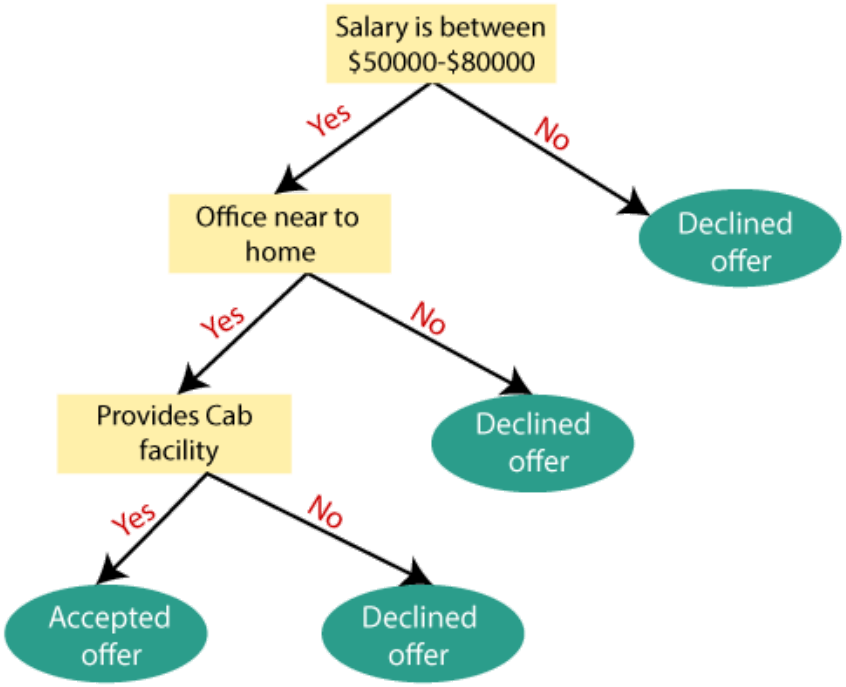
**Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

**Branch/Sub Tree:** A tree formed by splitting the tree.

**Pruning:** Pruning is the process of removing the unwanted branches from the tree.

**Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

## Examples



## Attribute Selection Measures

While implementing a Decision tree, the main issue arises as to how to select the best attribute for the root node and sub-nodes. So, to solve such problems there is a technique which is called an Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are some popular techniques for ASM, which are:

I. Entropy
II. Information Gain
III. Gini Index
IV. Gain Ratio,
V. Reduction in Variance
VI. Chi-Square

### Entropy:

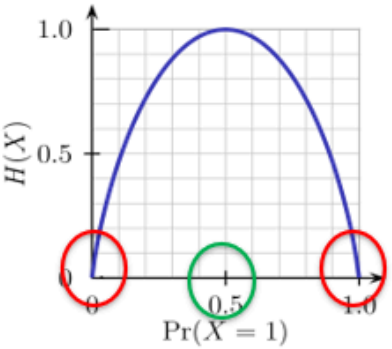
In machine learning, entropy is a metric that measures the level of uncertainty or disorder in a dataset. It's used in decision tree algorithms to help determine the best attribute to split a dataset

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

# ASM "Entropy"

- Entropy is a measure of the randomness in the information being processed.
- The higher the entropy, the harder it is to draw any conclusions from that information.
- Flipping a coin is an example of an action that provides information that is random.




- From the graph, it is quite evident that the entropy H(X) is zero when the probability is either 0 or 1.
- The Entropy is maximum when the probability is 0.5 because it projects perfect randomness in the data and there is no chance if perfectly determining the outcome.

The lower the value the better!!

41

# "Entropy": One Attribute

- Entropy helps in finding the purity of a feature
  - Better distinction of classes (more yes or no in one branch)
- Mathematically Entropy for ONE attribute is represented as:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

| Play Golf | |
|---|---|
| Yes | No |
| 9 | 5 |

P(Play Golf=Yes) = 9/14 = 0.642

P(Play Golf=No) = 5/14 = 0.3571

14 Records In Total

| ID code | Outlook | Temp | Humidity | Windy | Play |
|---|---|---|---|---|---|
| a | Sunny | Hot | High | False | No |
| b | Sunny | Hot | High | True | No |
| c | Overcast | Hot | High | False | Yes |
| d | Rainy | Mild | High | False | Yes |
| e | Rainy | Cool | Normal | False | Yes |
| f | Rainy | Cool | Normal | True | No |
| g | Overcast | Cool | Normal | True | Yes |
| h | Sunny | Mild | High | False | No |
| i | Sunny | Cool | Normal | False | Yes |
| j | Rainy | Mild | Normal | False | Yes |
| k | Sunny | Mild | Normal | True | Yes |
| l | Overcast | Mild | High | True | Yes |
| m | Overcast | Hot | Normal | False | Yes |
| n | Rainy | Mild | High | True | No |

**Entropy(PlayGolf) = Entropy (5,9)**
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

42

# "Entropy": Multiple Attribute

- Mathematically Entropy for Multiple attributes is represented as:

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

| | | Play Golf | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |
| | | | | 14 |

**E**(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

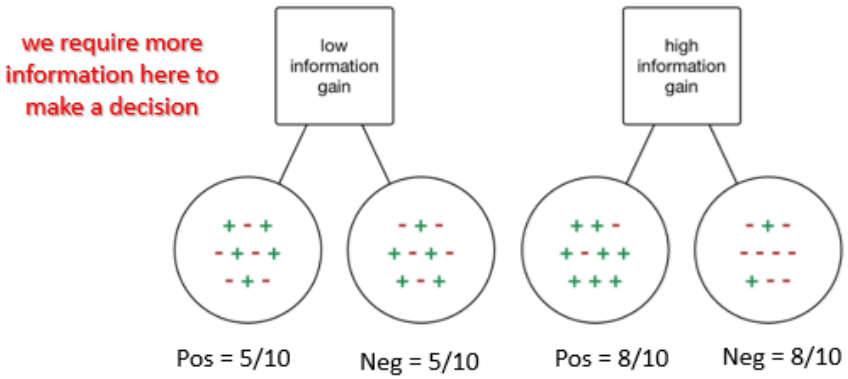
A multi-attribute decision model is a tool used to evaluate and compare different options based on multiple criteria.

## Information Gain:

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.

# Information Gain

- Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification.
- Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.



# Information Gain

- Information gain is a decrease in entropy. It computes the difference between entropy **before split** and average entropy **after split** of the dataset based on given attribute values.
- The **ID3** decision tree algorithm uses information gain.
- Mathematically, IG is represented as:

Information Gain(T,X) = Entropy(T) - Entropy(T, X)

IG(PlayGolf, Outlook) = E(PlayGolf) - E(PlayGolf, Outlook)
= 0.940 - 0.693
= 0.247

Entropy(T)

**Entropy(PlayGolf)** = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Entropy(T,X)

**E**(PlayGolf, Outlook) = **P**(Sunny)*E(3,2) + **P**(Overcast)*E(4,0) + **P**(Rainy)*E(2,3)
= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971
= 0.693

$$Information\ Gain = Entropy(before) - \sum_{j=1}^{K} Entropy(j, after)$$

## Gini Index:

The Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.

An attribute with the low Gini index should be preferred as compared to the high Gini index. It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.

# Gini Index

- The Gini index is a cost function that is also used to evaluate splits in the dataset.
- It is calculated by subtracting the sum of the squared probabilities of each class from one.
- It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

# Gini Index

- Gini Index works with the categorical target variable "Success" or "Failure". It performs only Binary splits.
  - Higher value of Gini index implies higher inequality, higher heterogeneity.
- Steps to Calculate Gini index for a split

1. Calculate Gini for sub-nodes, using the formula below for success(p) and failure(q) ($p^2+q^2$).
2. Calculate the Gini index for split using the weighted Gini score of each node of that split.

CART (Classification and Regression Tree) uses the Gini index method to create split points.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

The Gini index can be calculated using the below formula:

Gini Index= 1- ∑jPj2

## Gain Ratio

In machine learning, the gain ratio (GR) is a method used in decision trees to determine the best feature to split on. It's a modified version of the information gain

### Gain Ratio

- Information Gain (IG) is biased towards choosing attributes with a large number of values as root nodes.
- IG prefers the attribute with a large number of distinct values.
- **C4.5**, an improvement of ID3, uses Gain ratio which is a modification of Information gain that reduces its bias and is usually the best option.
- Gain ratio overcomes the problem with information gain by taking into account the number of branches that would result before making the split.
- It corrects information gain by taking the intrinsic information of a split into account.

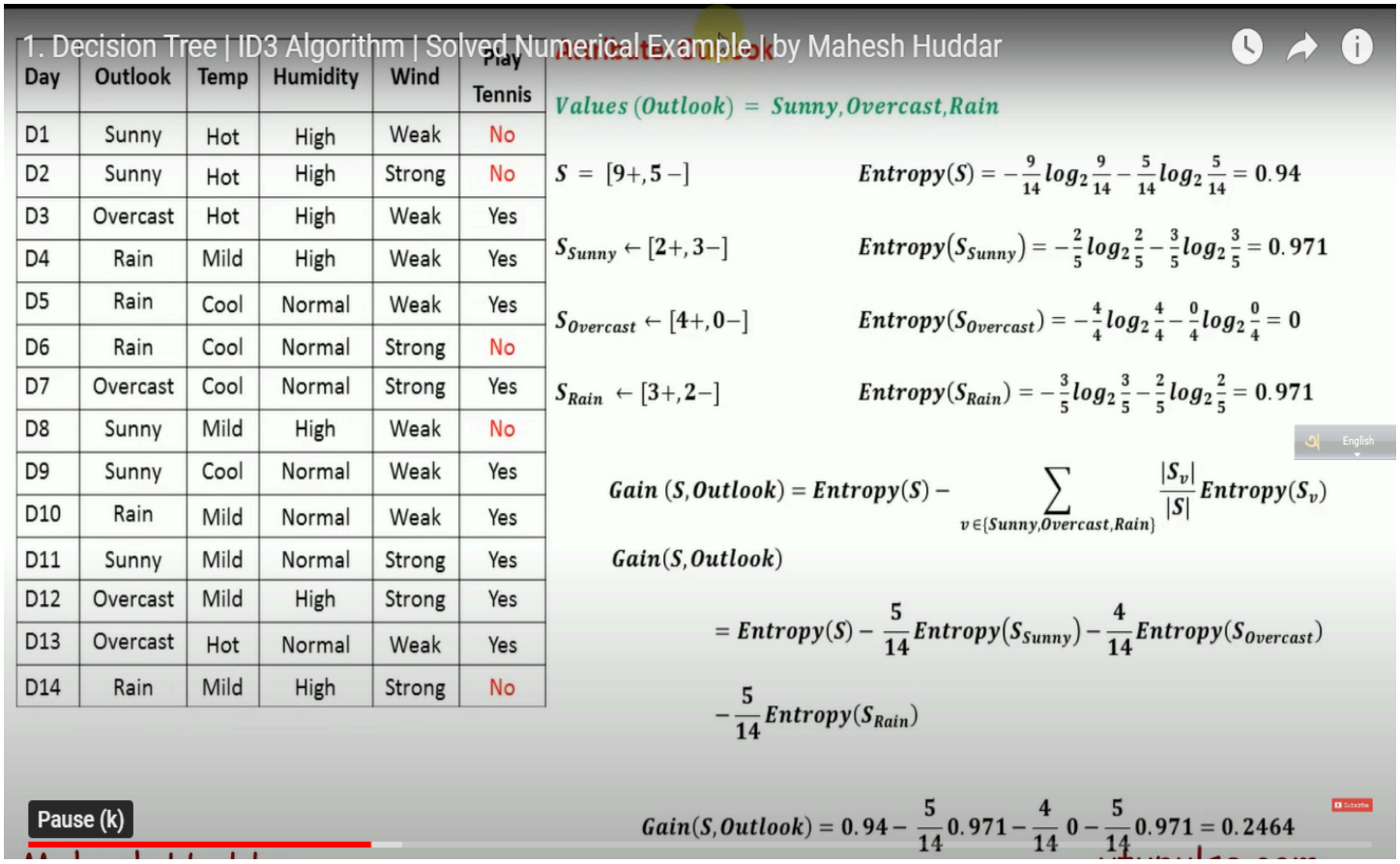
$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j, after)}{\sum_{j=1}^{K} w_j \log_2 w_j}$$

## Mathematical Example

| Day | Outlook | Temp | Humidity | Wind | PlayTennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

Must Watch the video for that  Math:
1. Decision Tree | ID3 Algorithm | Solved Numerical Example | by Mahesh Huddar

NB: Solve boro etar, But Maam bolche eto boro xm e ashbe na. Jekono ekta part ashbe..




| Day | Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

$$Values\,(Outlook) = Sunny, Overcast, Rain$$

$$S = [9+, 5-] \qquad Entropy(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$$S_{Sunny} \leftarrow [2+, 3-] \qquad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.971$$

$$S_{Overcast} \leftarrow [4+, 0-] \qquad Entropy(S_{Overcast}) = -\frac{4}{4}log_2\frac{4}{4} - \frac{0}{4}log_2\frac{0}{4} = 0$$

$$S_{Rain} \leftarrow [3+, 2-] \qquad Entropy(S_{Rain}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.971$$

$$Gain\,(S, Outlook) = Entropy(S) - \sum_{v \in \{Sunny, Overcast, Rain\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Outlook)$$

$$= Entropy(S) - \frac{5}{14}Entropy(S_{Sunny}) - \frac{4}{14}Entropy(S_{Overcast}) - \frac{5}{14}Entropy(S_{Rain})$$

$$Gain(S, Outlook) = 0.94 - \frac{5}{14}0.971 - \frac{4}{14}0 - \frac{5}{14}0.971 = 0.2464$$

| Day | Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

## Attribute: Temp

$$Values\ (Temp) = Hot, Mild, Cool$$

$$S = [9+, 5-] \qquad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$$

$$S_{Hot} \leftarrow [2+, 2-] \qquad Entropy(S_{Hot}) = -\frac{2}{4} log_2 \frac{2}{4} - \frac{2}{4} log_2 \frac{2}{4} = 1.0$$

$$S_{Mild} \leftarrow [4+, 2-] \qquad Entropy(S_{Mild}) = -\frac{4}{6} log_2 \frac{4}{6} - \frac{2}{6} log_2 \frac{2}{6} = 0.9183$$

$$S_{Cool} \leftarrow [3+, 1-] \qquad Entropy(S_{Cool}) = -\frac{3}{4} log_2 \frac{3}{4} - \frac{1}{4} log_2 \frac{1}{4} = 0.8113$$

$$Gain\ (S, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Temp)$$

$$= Entropy(S) - \frac{4}{14} Entropy(S_{Hot}) - \frac{6}{14} Entropy(S_{Mild})$$

$$- \frac{4}{14} Entropy(S_{Cool})$$

$$Gain(S, Temp) = 0.94 - \frac{4}{14} 1.0 - \frac{6}{14} 0.9183 - \frac{4}{14} 0.8113 = 0.0289$$

Mahesh Huddar

1. Decision Tree | ID3 Algorithm | Solved Numerical Example | by Mahesh Huddar

| Day | Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

**Attribute: Humidity**

$Values\ (Humidity) = High, Normal$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{High} \leftarrow [3+, 4-]$ $\quad Entropy(S_{High}) = -\frac{3}{7} log_2 \frac{3}{7} - \frac{4}{7} log_2 \frac{4}{7} = 0.9852$

$S_{Normal} \leftarrow [6+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{6}{7} log_2 \frac{6}{7} - \frac{1}{7} log_2 \frac{1}{7} = 0.5916$

$$Gain\ (S, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Humidity) = Entropy(S) - \frac{7}{14} Entropy(S_{High}) - \frac{7}{14} Entropy(S_{Normal})$$

$$Gain(S, Humidity) = 0.94 - \frac{7}{14} 0.9852 - \frac{7}{14} 0.5916 = 0.1516$$

1. Decision Tree | ID3 Algorithm | Solved Numerical Example | by Mahesh Huddar

| Day | Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

**Attribute: Wind**

$Values\ (Wind) = Strong, Weak$

$S = [9+, 5-]$ $\quad Entropy(S) = -\frac{9}{14} log_2 \frac{9}{14} - \frac{5}{14} log_2 \frac{5}{14} = 0.94$

$S_{Strong} \leftarrow [3+, 3-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [6+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{6}{8} log_2 \frac{6}{8} - \frac{2}{8} log_2 \frac{2}{8} = 0.8113$

$$Gain\ (S, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S, Wind) = Entropy(S) - \frac{6}{14} Entropy(S_{Strong}) - \frac{8}{14} Entropy(S_{Weak})$$

$$Gain(S, Wind) = 0.94 - \frac{6}{14} 1.0 - \frac{8}{14} 0.8113 = 0.0478$$

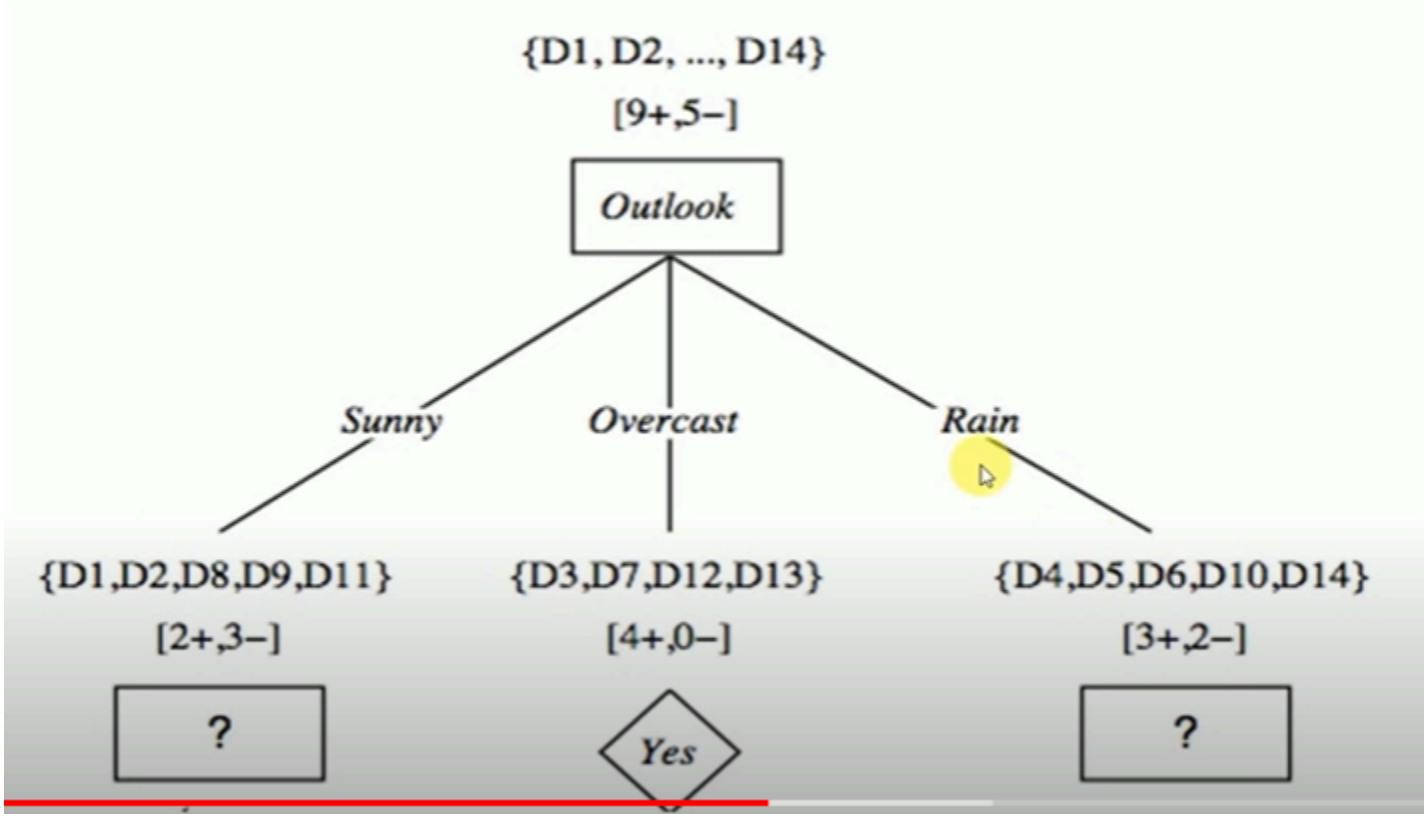
| Day | Outlook | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

$$Gain(S, Outlook) = 0.2464$$

$$Gain(S, Temp) = 0.0289$$

$$Gain(S, Humidity) = 0.1516$$

$$Gain(S, Wind) = 0.0478$$



| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| D1 | Hot | High | Weak | No |
| D2 | Hot | High | Strong | No |
| D8 | Mild | High | Weak | No |
| D9 | Cool | Normal | Weak | Yes |
| D11 | Mild | Normal | Strong | Yes |

## Attribute: Temp

$Values\ (Temp) = Hot, Mild, Cool$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S_{Sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 2-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Mild}) = 1.0$

$S_{Cool} \leftarrow [1+, 0-]$ $\quad Entropy(S_{Cool}) = 0.0$

$$Gain\ (S_{Sunny}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Temp)$$

$$= Entropy(S) - \frac{2}{5}Entropy(S_{Hot}) - \frac{2}{5}Entropy(S_{Mild})$$

$$- \frac{1}{5}Entropy(S_{Cool})$$

$$Gain(S_{sunny}, Temp) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}1 - \frac{1}{5}0.0 = 0.570$$

| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| Dl | Hot | High | Weak | No |
| D2 | Hot | High | Strong | No |
| D8 | Mild | High | Weak | No |
| D9 | Cool | Normal | Weak | Yes |
| Dl1 | Mild | Normal | Strong | Yes |

## Attribute: Humidity

$Values\ (Humidity) = High, Normal$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{high} \leftarrow [0+, 3-]$ $\quad Entropy(S_{High}) = 0.0$

$S_{Normal} \leftarrow [2+, 0-]$ $\quad Entropy(S_{Normal}) = 0.0$

$$Gain\ (S_{Sunny}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Humidity) = Entropy(S) - \frac{3}{5}Entropy(S_{High}) - \frac{2}{5}Entropy(S_{Normal})$$

$$Gain(S_{sunny}, Humidity) = 0.97 - \frac{3}{5}0.0 - \frac{2}{5}0.0 = 0.97$$

| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| Dl | Hot | High | Weak | No |
| D2 | Hot | High | Strong | No |
| D8 | Mild | High | Weak | No |
| D9 | Cool | Normal | Weak | Yes |
| Dl1 | Mild | Normal | Strong | Yes |

## Attribute: Wind

$Values\ (Wind) = Strong, Weak$

$S_{Sunny} = [2+, 3-]$ $\quad Entropy(S) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.97$

$S_{Strong} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Strong}) = 1.0$

$S_{Weak} \leftarrow [1+, 2-]$ $\quad Entropy(S_{Weak}) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = 0.9183$

$$Gain\ (S_{Sunny}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Sunny}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

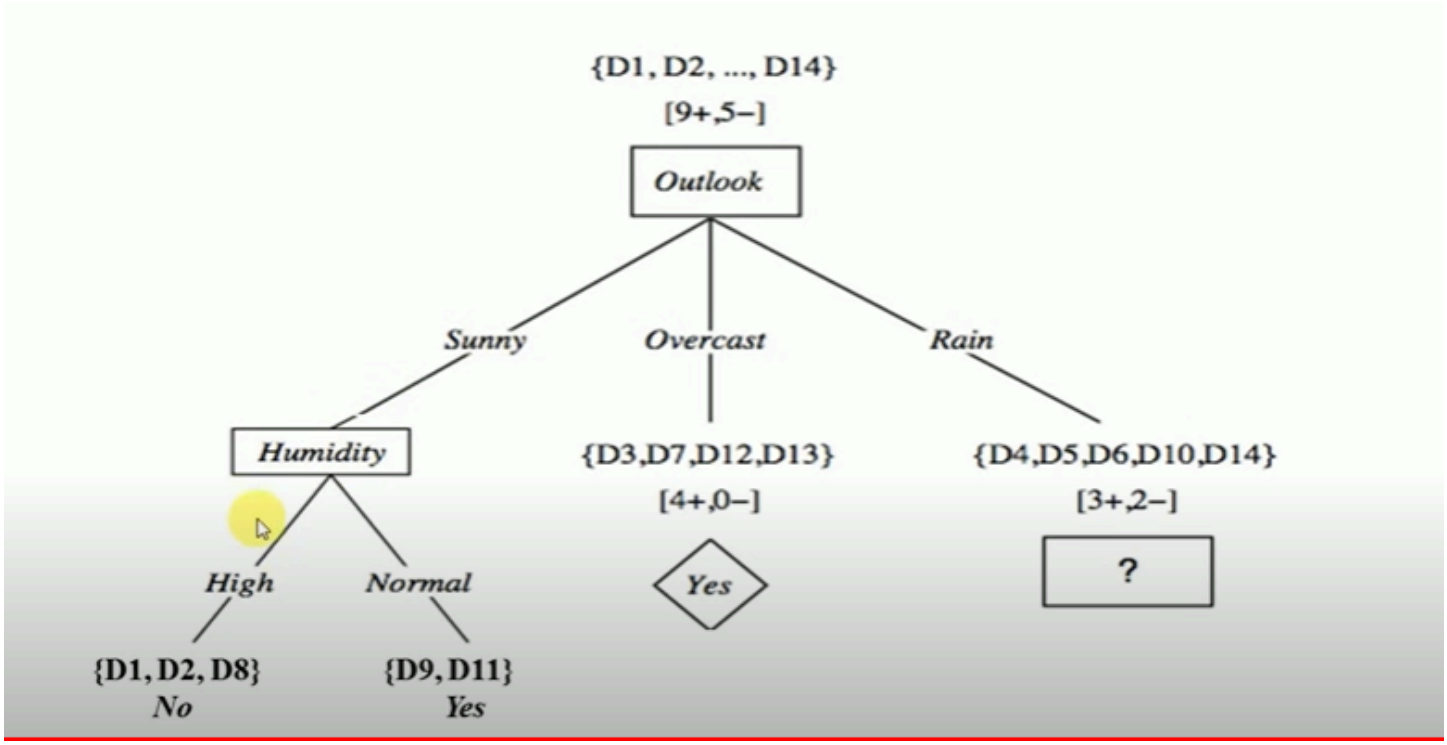
$$Gain(S_{sunny}, Wind) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}\,0.918 = 0.0192$$

| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| D1 | Hot | High | Weak | No |
| D2 | Hot | High | Strong | No |
| D8 | Mild | High | Weak | No |
| D9 | Cool | Normal | Weak | Yes |
| D11 | Mild | Normal | Strong | Yes |

$$Gain(S_{sunny}, Temp) = 0.570$$

$$Gain(S_{sunny}, Humidity) = 0.97$$

$$Gain(S_{sunny}, Wind) = 0.0192$$



| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| D4 | Mild | High | Weak | Yes |
| D5 | Cool | Normal | Weak | Yes |
| D6 | Cool | Normal | Strong | No |
| D10 | Mild | Normal | Weak | Yes |
| D14 | Mild | High | Strong | No |

## Attribute: Temp

$$Values\ (Temp) = Hot, Mild, Cool$$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Hot} \leftarrow [0+, 0-]$ $\quad Entropy(S_{Hot}) = 0.0$

$S_{Mild} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Mild}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$S_{Cool} \leftarrow [1+, 1-]$ $\quad Entropy(S_{Cool}) = 1.0$

$$Gain\ (S_{Rain}, Temp) = Entropy(S) - \sum_{v \in \{Hot, Mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Temp) = Entropy(S) - \frac{0}{5}Entropy(S_{Hot}) - \frac{3}{5}Entropy(S_{Mild}) - \frac{2}{5}Entropy(S_{Cool})$$

$$Gain(S_{Rain}, Temp) = 0.97 - \frac{0}{5}0.0 - \frac{3}{5}0.918 - \frac{2}{5}1.0 = 0.0192$$

Mahesh Huddar vtupulse.com

| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| D4 | Mild | High | Weak | Yes |
| D5 | Cool | Normal | Weak | Yes |
| D6 | Cool | Normal | Strong | No |
| Dl0 | Mild | Normal | Weak | Yes |
| Dl4 | Mild | High | Strong | No |

## Attribute: Humidity

$$Values\ (Humidity) = High, Normal$$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{High} \leftarrow [1+, 1-]$ $\quad Entropy(S_{High}) = 1.0$

$S_{Normal} \leftarrow [2+, 1-]$ $\quad Entropy(S_{Normal}) = -\frac{2}{3}log_2\frac{2}{3} - \frac{1}{3}log_2\frac{1}{3} = 0.9183$

$$Gain\ (S_{Rain}, Humidity) = Entropy(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Humidity) = Entropy(S) - \frac{2}{5}Entropy(S_{High}) - \frac{3}{5}Entropy(S_{Normal})$$

$$Gain(S_{Rain}, Humidity) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.0192$$

| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| D4 | Mild | High | Weak | Yes |
| D5 | Cool | Normal | Weak | Yes |
| D6 | Cool | Normal | Strong | No |
| Dl0 | Mild | Normal | Weak | Yes |
| Dl4 | Mild | High | Strong | No |

## Attribute: Wind

$Values\ (wind) = Strong, Weak$

$S_{Rain} = [3+, 2-]$ $\quad Entropy(S_{Sunny}) = -\frac{3}{5}log_2\frac{3}{5} - \frac{2}{5}log_2\frac{2}{5} = 0.97$

$S_{Strong} \leftarrow [0+, 2-]$ $\quad Entropy(S_{Strong}) = 0.0$

$S_{Weak} \leftarrow [3+, 0-]$ $\quad Entropy(S_{weak}) = 0.0$

$$Gain\ (S_{Rain}, Wind) = Entropy(S) - \sum_{v \in \{Strong, Weak\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$Gain(S_{Rain}, Wind) = Entropy(S) - \frac{2}{5}Entropy(S_{Strong}) - \frac{3}{5}Entropy(S_{Weak})$$

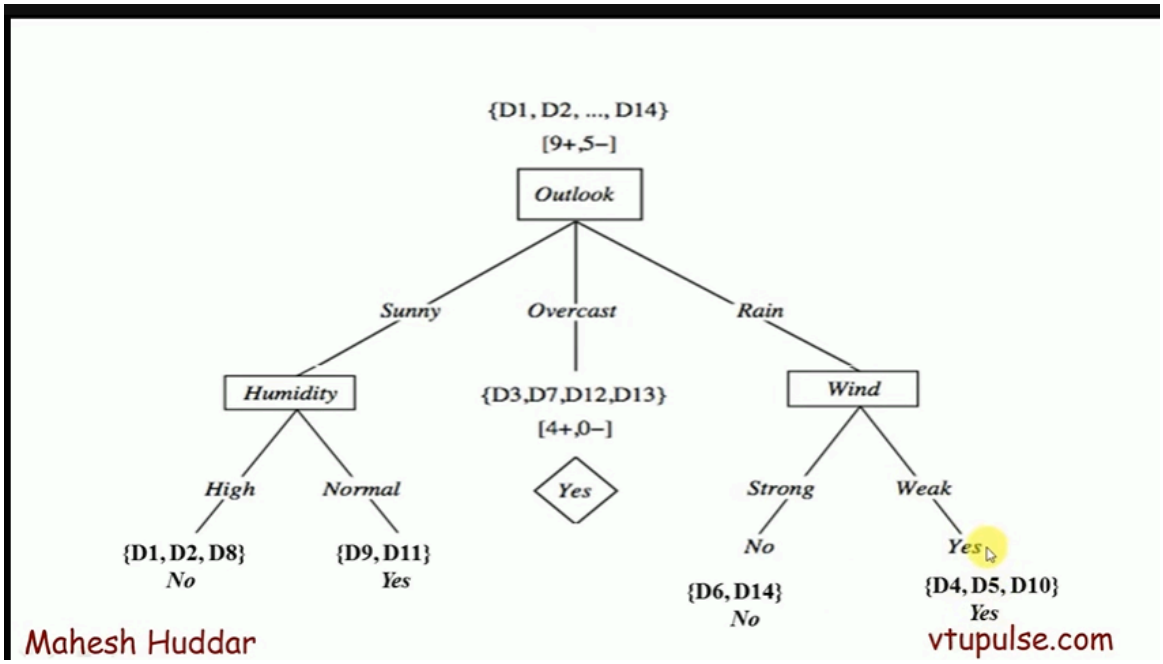
$$Gain(S_{Rain}, Wind) = 0.97 - \frac{2}{5}\,0.0 - \frac{3}{5}0.0 = 0.97$$

| Day | Temp | Humidity | Wind | Play Tennis |
|---|---|---|---|---|
| D4 | Mild | High | Weak | Yes |
| D5 | Cool | Normal | Weak | Yes |
| D6 | Cool | Normal | Strong | No |
| Dl0 | Mild | Normal | Weak | Yes |
| Dl4 | Mild | High | Strong | No |

$$Gain(S_{Rain}, Temp) = 0.0192$$

$$Gain(S_{Rain}, Humidity) = 0.0192$$

$$Gain(S_{Rain}, Wind) = 0.97$$



# Advantages

1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
2. A decision tree does not require normalization of data.
3. A decision tree does not require scaling of data as well.
4. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
5. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

# Disadvantages

1. Overfitting can become a problem if a decision tree's design is too complex.
2. They are not well-suited to continuous variables (i.e. variables which can have more than one value, or a spectrum of values.)
3. They are unstable, meaning that a small change in the data can lead to a large change in the structure of the optimal decision tree.
4. They are often relatively inaccurate.
5. Decision trees often involves higher time to train the model.

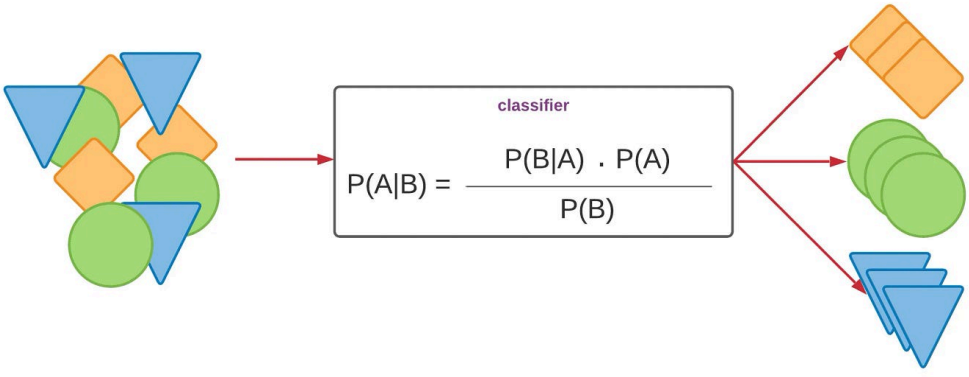
Decision Tree vs Random Forest

| Decision Tree | Random Forest |
|---|---|
| A decision tree is a tree-like model of decisions along with possible outcomes in a diagram. | A classification algorithm consisting of many decision trees combined to get a more accurate result as compared to a single tree. |
| There is always a scope for overfitting, caused due to the presence of variance. | Random forest algorithm avoids and prevents overfitting by using multiple trees. |
| The results are not accurate. | This gives accurate and precise results. |
| Decision trees require low computation, thus reducing time to implement and carrying low accuracy. | This consumes more computation. The process of generation and analyzing is time-consuming. |
| It is easy to visualize. The only task is to fit the decision tree model. | This has complex visualization as it determines the pattern behind the data. |

# Naive Bayes

Naive bayes classifier / algorithm in data mining in bangla/Data mining tutorial in Bangla

The Naive Bayes algorithm is a supervised machine learning classifier that uses probability models to perform classification tasks. It's known for being fast, accurate, and reliable, and is often used for text classification, spam filtering, and recommendation systems.



- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

**Where,**

**P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.

**P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.

**P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.

**P(B) is Marginal Probability**: Probability of Evidence.

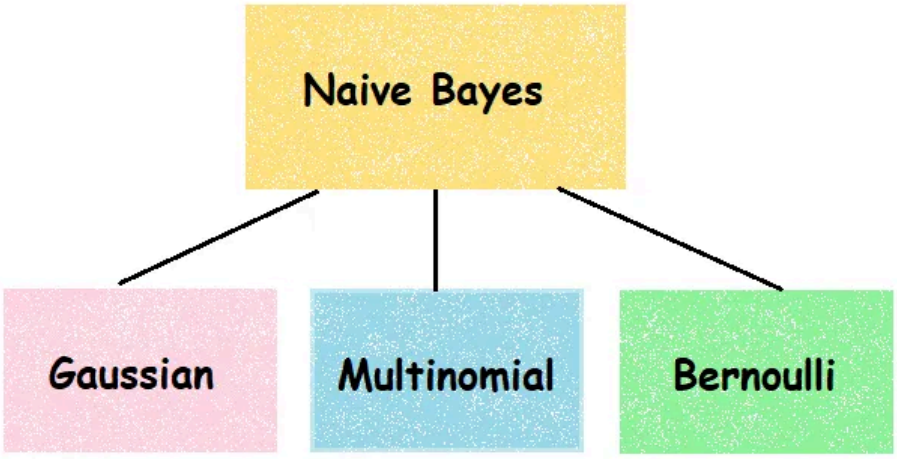
### Why it is important?

Naive Bayes is a significant algorithm in machine learning because it is incredibly fast, easy to implement, performs well with limited training data, and is particularly useful for text classification tasks where features are often assumed to be independent, making it a practical choice for real-time predictions and large datasets; however, its key limitation is the strong assumption of feature independence, which may not always hold true in real-world scenarios.

### Why is the Naïve Bayesian classification called "naïve"?

The naive Bayes algorithm (NB) is a Bayesian graphical model that has nodes corresponding to each of the columns or features. It is called naive because it ignores the prior distribution of parameters and assumes independence of all features and all rows.

## Types of Naive Bayas



1. Gaussian: is a variant of Naive Bayes used in classification and it assumes that features follow a normal distribution.
   - Gaussian Naive Bayes supports continuous data.
2. Multinomial: It is used for discrete counts. ...
   - Multinomial Naive bayes is good at handling discrete values
   - The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification).
3. Bernoulli: The binomial model is useful if your feature vectors are binary (i.e. zeros and ones).
   - It accepts only binary values, i.e., 0 or 1.
   - If the features of the dataset are binary, then we can assume that Bernoulli Naive Bayes is the algorithm to be used.

## Pros and Cons

- The Naïve Bayes' approach is a very popular one, which often works well.
- Simple and easy to implement.
- It doesn't require much training data.
- It is fast and highly scalable with the number of predictors.
- It works well with high-dimensional data such as text classification, email spam detection.

- However, it has a number of potential problems
  - It relies on all attributes being categorical.
  - If the data is less, then it estimates poorly.

## Mathematical Example 1

Given the training data in the table below (Buy Computer data), predict the class of the following 8 new examples using Naïve Bayes classification:
age<=30, income-medium, student-yes, credit- rating-fair

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

Sol:

7. Solved Example Naive Bayes Classification Age Income Student Credit Rating Buys Comp…

| age | income | student | Credit rating | Buys computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31...40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31...40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31...40 | medium | no | excellent | yes |
| 31...40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

- $E_1$ is age<=30,
- E2 is income=medium,
- $E_3$ student=yes,
- $E_4$ is credit-rating=fair
- We need to compute **P(yes|E)** and **P(no|E)** and compare them.

$$P(yes|E) = \frac{P(E_1|yes)\,P(E_2|yes)\,P(E_3|yes)\,P(E_4|yes)\,P(yes)}{P(E)}$$

$$P(no|E) = \frac{P(E_1|no)\,P(E_2|no)\,P(E_3|no)\,P(E_4|no)\,P(no)}{P(E)}$$

| age | income | student | Credit rating | Buys computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31...40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31...40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31...40 | medium | no | excellent | yes |
| 31...40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

$$P(yes|E) = \frac{P(E_1|yes)\,P(E_2|yes)\,P(E_3|yes)\,P(E_4|yes)\,P(yes)}{P(E)}$$

$$P(no|E) = \frac{P(E_1|no)\,P(E_2|no)\,P(E_3|no)\,P(E_4|no)\,P(no)}{P(E)}$$

P(yes)=9/14=0.643 P(no)=5/14=0.357 $E_1$ is age<=30,

$P(E_1|yes)$=2/9=0.222 P(E1|no)=3/5=0.6 E2 is income=medium

$E_3$ student=yes,

$P(E_2|yes)$=4/9=0.444 P(E2|no)=2/5=0.4 $E_4$ is credit-rating=fair

$P(E_3|yes)$=6/9=0.667 P(E3|no)=1/5=0.2

$P(E_4|yes)$=6/9=0.667 P(E4|no)=2/5=0.4

$$P(yes|E) = \frac{0.222 * 0.444 * 0.667 * 0.667 * 0.643}{P(E)} = \frac{0.028}{P(E)}$$

$$P(no|E) = \frac{0.6 * 0.4 * 0.2 * 0.4 * 0.357}{P(E)} = \frac{0.007}{P(E)}$$

Hence, the Naïve Bayes classifier predicts

**buys_computer = yes** for the new example.

## Example-2

# NB Algorithm (Multiple Features)

14 Records, 2 Feature

| Weather | Temperature | Play |
|---|---|---|
| Sunny | Hot | No |
| Sunny | Hot | No |
| Overcast | Hot | Yes |
| Rainy | Mild | Yes |
| Rainy | Cool | Yes |
| Rainy | Cool | No |
| Overcast | Cool | Yes |
| Sunny | Mild | No |
| Sunny | Cool | Yes |
| Rainy | Mild | Yes |
| Sunny | Mild | Yes |
| Overcast | Mild | Yes |
| Overcast | Hot | Yes |
| Rainy | Mild | No |

- **Step 1**: Calculate the Prior Probability for given Class Labels
- **Step 2**: Calculate Conditional Probability with each Attribute for each Class.
- **Step 3**: Multiply Same Class Conditional Probability.
- **Step 4**: Multiply Prior Probability with Step 3 Probability.
- **Step 5**: See which Class has Higher Probability, Higher Probability Class belongs to given input Step.

38

# NB Algorithm (Multiple Features)

Calculate the probability of playing when (a) weather is overcast and (b) temperature is mild

Probability of playing:

P(Play=Yes | W=Overcast, T=Mild) = P(W=Overcast, T=Mild | Play=Yes) * (Play=Yes) ...(1)

P(W=Overcast, T=Mild | Play= Yes)= P(W=Overcast |Play=Yes) * P(T=Mild |Play=Yes) ...(2)

1. Calculate Prior Probabilities:
   P(Play=Yes) = 9/14 = 0.64

2. Calculate Posterior Probabilities:
   P(W=Overcast |Yes) = 4/9 = 0.44
   P(T=Mild | Yes) = 4/9 = 0.44

- Step 1: Calculate the Prior Probability for given Class Labels
- Step 2: Calculate Conditional Probability with each Attribute for each Class.
- Step 3: Multiply Same Class Conditional Probability.
- Step 4: Multiply Prior Probability with Step 3 Probability.
- Step 5: See which Class has Higher Probability, Higher Probability Class belongs to given input Step.

3. Put Posterior Probabilities in Equation (2)
   P(W=Overcast, T=Mild | Play= Yes) = 0.44 * 0.44 = 0.1936

4. Put Prior and Posterior probabilities in Equation (1)
   P (Play=Yes | W=Overcast, T=Mild) = 0.1936 * 0.64 = 0.124

Probability of not playing: 0

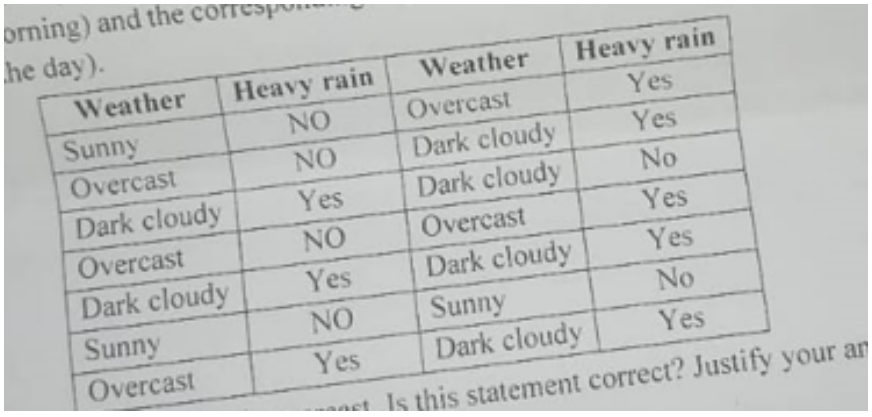
The probability of a 'Yes' class is higher.

So, you can say here that if the weather is overcast than players will play the sport.

39

## Example 3

Consider the table below, the training set of weather (in the morning) and the corresponding target variable 'Heavy rain' (i.e., the possibility of heavy rain in the day).



orning) and the corresp
he day).

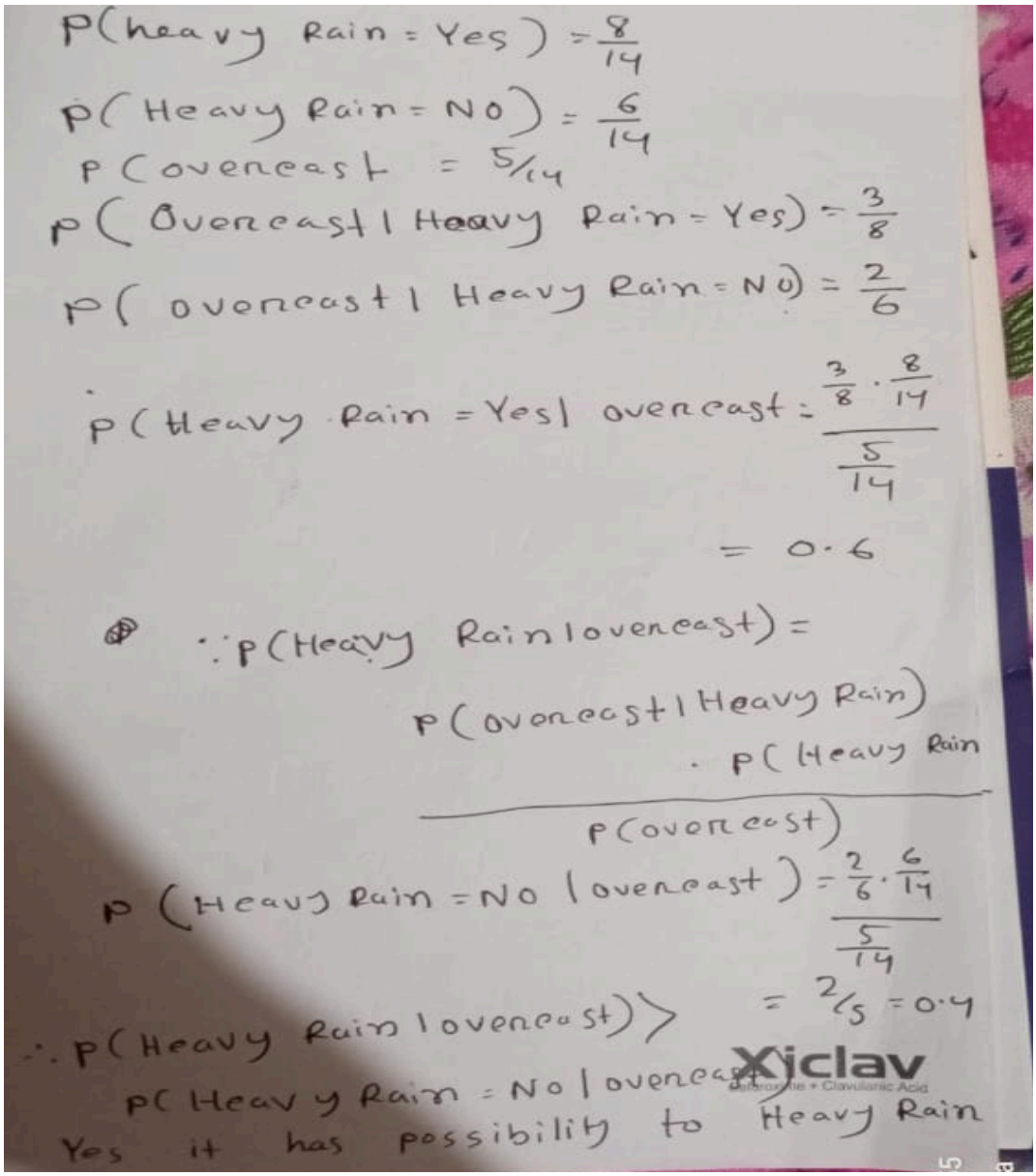
| Weather | Heavy rain | Weather | Heavy rain |
|---|---|---|---|
| Sunny | NO | Overcast | Yes |
| Overcast | NO | Dark cloudy | Yes |
| Dark cloudy | Yes | Dark cloudy | No |
| Overcast | NO | Overcast | Yes |
| Dark cloudy | Yes | Dark cloudy | No |
| Sunny | NO | Sunny | Yes |
| Overcast | Yes | Dark cloudy | |

cast. Is this statement correct? Justify your an

There will be heavy rain if the weather is overcast. Is this statement correct? Justify your answer

Sol:

1. Solved Example Naive Bayes Classifier to classify New Instance PlayTennis Example Mahes…



# Independent Events

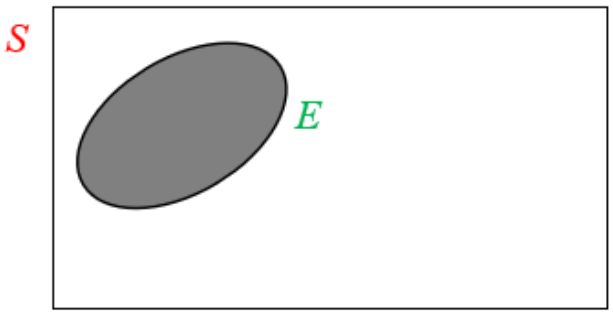
- **Independent events:** Two events are independent if occurrences of one does not alter the occurrence of other.

**Example:** Tossing both coin and rolling a Die together.

(How many events are here?)

# Sample Space & Events

- The **sample space**, $S$, for a random phenomena is the set of all possible outcomes.
- The **event**, $E$, is any subset of the sample space, S. i.e. any set of outcomes (not necessarily all outcomes) of the random phenomena




Events
- Flipping a coin,
- Rolling a dice
- ...

7

# Special Events

**The Null Event, The empty event - $\phi$**

$\phi = \{\ \} =$ the event that contains no outcomes

**The Entire Event, The Sample Space - $S$**

$S =$ the event that contains all outcomes

The empty event, $\phi$, never occurs.

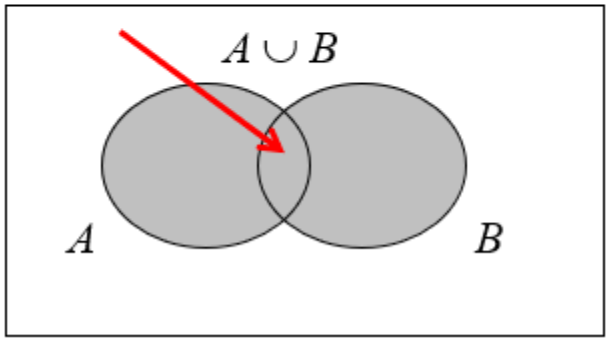
The entire event, $S$, always occurs.

# Set operations on Events

**Union**

Let $A$ and $B$ be two events, then the **union** of $A$ and $B$ is the event (denoted by $A \cup B$) defined by:

$A \cup B = \{e |\ e$ belongs to $A$ **or** $e$ belongs to $B\}$

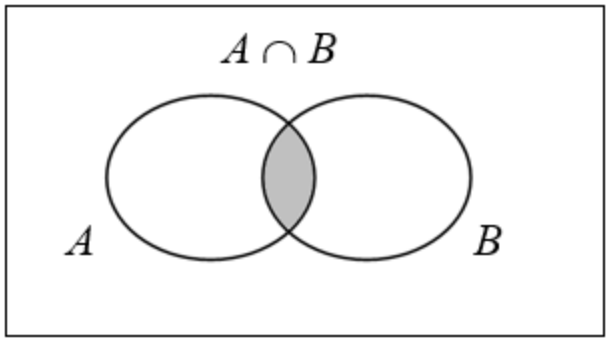



9

## Intersection

Let $A$ and $B$ be two events, then the **intersection** of $A$ and $B$ is the event (denoted by $A \cap B$) defined by:

$$A \cap B = \{e |\ e \text{ belongs to } A \textbf{ and } e \text{ belongs to } B\}$$



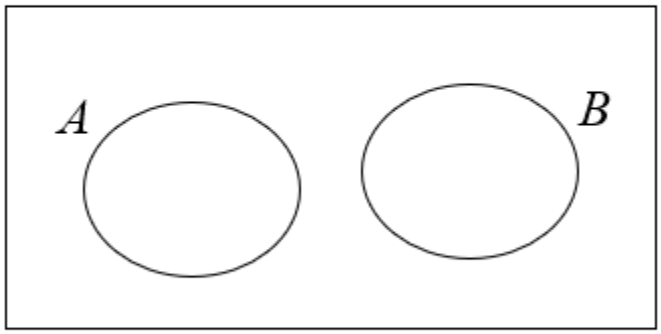

The event $A \cap B$ **occurs** if the event $A$ **occurs and** the event $B$ **occurs** .

10

# Mutually Exclusive

**Definition:**

Two events $A$ and $B$ are called **mutually exclusive** if:




$$A \cap B = \phi$$

- They have no outcomes in common.
- They can't occur at the same time.
- The outcome of the random experiment can not belong to both A and B.

11

# Simple Probability

**Definition : Simple Probability**

If there are $n$ elementary events associated with a random experiment and $m$ of $n$ of them are favorable to an event $A$, then the probability of happening or occurrence of $A$ is

$$P(A) = \frac{m}{n}$$

**Examples**

1. Tossing a coin – outcomes $S = \{$**Head, Tail**$\}$
2. Rolling a die – outcomes

$$S = \{ \boxdot \quad \boxdot \quad \boxdot \quad \boxdot \quad \boxdot \quad \boxdot \}$$

$$= \{1, 2, 3, 4, 5, 6\}$$

# Union & Joint Probability

**Definition: Union and Joint Probability**

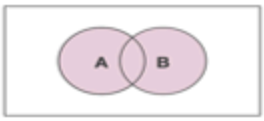
If $P(A)$ and $P(B)$ are the probability of two events, then the **Union**:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

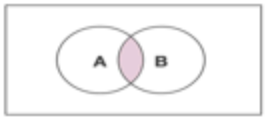
If $A$ and $B$ are mutually exclusive, then $P(A \cap B) = 0$
If $A$ and $B$ are independent events, then $P(A \cap B) = P(A).P(B)$

Thus, for mutually exclusive events the **joint probability** → $A \cap B = \phi$

$$P(A \cup B) = P(A) + P(B)$$



union: A ∪ B



intersection: A ∩ B

# Example

$$P[A \cup B] = P[A] + P[B] - P[A \cap B]$$

**Example:**

Saskatoon and Moncton are two of the cities competing for the World University Games. The organizers are narrowing the competition to the **final 5 cities.**

- There is a 20% chance that Saskatoon will be amongst the **final 5**.
- There is a 35% chance that Moncton will be amongst the **final 5**
- There is an 8% chance that both Saskatoon and Moncton will be amongst the **final 5**.

What is the probability that Saskatoon or Moncton will be amongst the **final 5.?**

# Example: Solution

**Solution:**

Let $A$ = the event that Saskatoon is amongst the **final 5**.

Let $B$ = the event that Moncton is amongst the **final 5**.

Given $P[A] = 0.20$, $P[B] = 0.35$, and $P[A \cap B] = 0.08$

What is $P[A \cup B]$?

**Note:** "and" $\equiv \cap$, "or" $\equiv \cup$ .

$$P[A \cup B] = P[A] + P[B] - P[A \cap B]$$
$$= 0.20 + 0.35 - 0.08 = 0.47$$

# Conditional Probability

Corollary: **Conditional Probability**

$$P(A \cap B) = P(A).P(B|A), \quad if\ P(A) \neq 0$$

or $$P(A \cap B) = P(B).P(A|B), \quad if\ P(B) \neq 0$$

For three events $A$, $B$ and $C$

$$P(A \cap B \cap C) = P(A).P(B).P(C|A \cap B)$$

For $n$ events $A_1, A_2, \ldots, A_n$ and if all events are mutually independent to each other

$$P(A_1 \cap A_2 \cap \quad \cap A_n) = P(A_1).P(A_2) \quad P(A_n)$$

**Note:**

$P(A|B) = 0$ if events are **mutually exclusive**

$P(A|B) = P(A)$ if $A$ and $B$ are **independent**

$P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$ otherwise,

$P(A \cap B) = P(B \cap A)$

# Cond. Probability: Example

The academy awards is soon to be shown.

For a specific married couple the probability that the husband watches the show is 80%, the probability that his wife watches the show is 65%, while the probability that they both watch the show is 60%.

If the husband is watching the show, what is the probability that his wife is also watching the show?

**Solution:**

The academy awards is soon to be shown.

- Let $B$ = the event that the husband watches the show
  $P[B] = 0.80$
- Let $A$ = the event that his wife watches the show
  $P[A] = 0.65$ and $P[A \cap B] = 0.60$

$$P[A|B] = \frac{P[A \cap B]}{P[B]} = \frac{0.60}{0.80} = 0.75$$

# Conditional Probability

- Generalization of Conditional Probability:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)}$$

$$= \frac{P(B|A) \cdot P(A)}{P(B)} \quad \because P(A \cap B) = P(B|A) \cdot P(A) = P(A|B) \cdot P(B)$$

By the law of total probability : $P(B) = P[(B \cap A) \cup (B \cap \bar{A})]$, where $\bar{A}$ denotes the compliment of event A. Thus,

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P[(B \cap A) \cup (B \cap \bar{A})]}$$

$$= \frac{P(B|A) \cdot P(A)}{P(B|A) \cdot P(A) + P(B|\bar{A}) \cdot P(\bar{A})}$$

# Prior and Posterior Probabilities

- P(A) and P(B) are **called prior probabilities**
- P(A|B), P(B|A) are **called posterior probabilities**

**Example: Prior versus Posterior Probabilities**

- This table shows that the event $Y$ has two outcomes namely $A$ and $B$, which is dependent on another event $X$ with various outcomes like $x_1$, $x_2$ and $x_3$.
- **Case1:** Suppose, we don't have any information of the event $A$. Then, from the given sample space, we can calculate $P(Y = A) = \frac{5}{10} = 0.5$
- **Case2:** Now, suppose, we want to calculate $P(X = x_2 | Y = A) = \frac{2}{5} = 0.4$ .

The later is the conditional or posterior probability, where as the former is the prior probability.

| X | Y |
|---|---|
| $x_1$ | A |
| $x_2$ | A |
| $x_3$ | B |
| $x_3$ | A |
| $x_2$ | B |
| $x_1$ | A |
| $x_1$ | B |
| $x_3$ | B |
| $x_2$ | B |
| $x_2$ | A |

# Independent Events

Two events $A$ and $B$ are called **independent** if

$$P[A \cap B] = P[A]P[B]$$

if $P[B] \neq 0$ and $P[A] \neq 0$ then

$$P[A|B] = \frac{P[A \cap B]}{P[B]} = \frac{P[A]P[B]}{P[B]} = P[A]$$

and $$P[B|A] = \frac{P[A \cap B]}{P[A]} = \frac{P[A]P[B]}{P[A]} = P[B]$$

Thus in the case of independence the conditional probability of an event is not affected by the knowledge of the other event

# Avoiding the 0-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be non-zero. Otherwise, the predicted prob. will be zero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

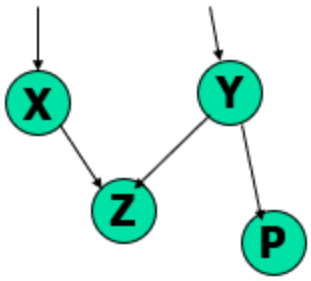
- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10),
- Use Laplacian correction (or Laplacian estimator)
  - Adding 1 to each case
    Prob(income = low) = 1/1003
    Prob(income = medium) = 991/1003
    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

## Bayesian Network

Video: Bayesian Network with Examples | Easiest Explanation

# Bayesian Belief Networks

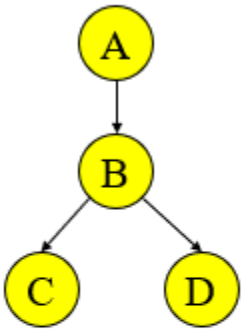
- Bayesian belief network allows a *subset* of the variables conditionally independent
- BBNs are graphical representations of the probabilities
- A graphical model of causal relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution




- ❑ Nodes: random variables
- ❑ Links: dependency
- ❑ X and Y are the parents of Z, and Y is the parent of P
- ❑ No dependency between Z and P
- ❑ Has no loops or cycles

A Bayesian network is made up of:

1. A Directed Acyclic Graph




2. A set of tables for each node in the graph

| A | P(A) |
|---|---|
| false | 0.6 |
| true | 0.4 |

| A | B | P(B\|A) |
|---|---|---|
| false | false | 0.01 |
| false | true | 0.99 |
| true | false | 0.7 |
| true | true | 0.3 |

| B | D | P(D\|B) |
|---|---|---|
| false | false | 0.02 |
| false | true | 0.98 |
| true | false | 0.05 |
| true | true | 0.95 |

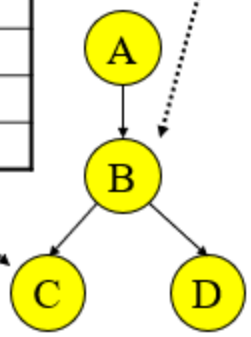
| B | C | P(C\|B) |
|---|---|---|
| false | false | 0.4 |
| false | true | 0.6 |
| true | false | 0.9 |
| true | true | 0.1 |

# A Set of Tables for Each Node

| A | P(A) |
|---|---|
| false | 0.6 |
| true | 0.4 |

| A | B | P(B\|A) |
|---|---|---|
| false | false | 0.01 |
| false | true | 0.99 |
| true | false | 0.7 |
| true | true | 0.3 |

| B | C | P(C\|B) |
|---|---|---|
| false | false | 0.4 |
| false | true | 0.6 |
| true | false | 0.9 |
| true | true | 0.1 |




| B | D | P(D\|B) |
|---|---|---|
| false | false | 0.02 |
| false | true | 0.98 |
| true | false | 0.05 |
| true | true | 0.95 |

Each node $X_i$ has a conditional probability distribution $P(X_i \mid Parents(X_i))$ that quantifies the effect of the parents on the node

The parameters are the probabilities in these conditional probability tables (CPTs)

57

**Sol:**

Using the network in the example, suppose you want to calculate:

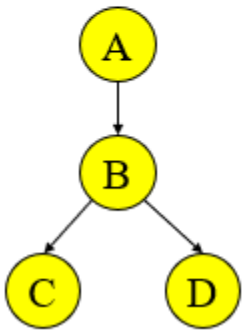
$P(A = true, B = true, C = true, D = true)$

$= P(A = true) * P(B = true \mid A = true) *$
$P(C = true \mid B = true) P(D = true \mid B = true)$

This is from the graph structure

$= (0.4)*(0.3)*(0.1)*(0.95)$

These numbers are from the conditional probability tables



## Naïve Bayesian Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.
      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

More Mathematical Example saw that video:

1. Bayesian Belief Network | BBN | Solved Numerical Example | Burglar Alarm System by Mahesh Huddar

# Evaluation Technique

Ei slide ta ami aga matha kisu bujhi na valo vabe, jototuku sajanor sajaichi. Jodio porbo na eta ami

Model Evaluation is how we quantify the quality of a system's predictions.
To do this, we measure the newly trained model performance on a new and independent dataset. This model will compare labeled data with its predictions. We will cover several metrics for Regression and Classification

- Several metrics are available for:
  - Regression Models
    - R^2, MAE, MSE, RMSE
  - Classification Models
    - Accuracy, Precision, Sensitivity, Recall, ROC, $F_1$ Measure

Which metric should we use? Why? and When?

# Evaluation Criteria

- **Predictive accuracy**

$$\text{Accuracy} = \frac{\text{Number of Correct Classifications}}{\text{Total Number of Test Cases}}$$

- **Efficiency**
  - Time to construct the model ... (Training Time)
  - Time to use the model ... (Inference Time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency in disk-resident databases
- **Interpretability**:
  - understandable and insight provided by the model
- **Compactness of the model**: size of the tree, or number of rules.

or,

**Accuracy**

Use for: Classification (e.g., spam detection)

Why: Measures correct predictions

When: Classes are balanced.

**Precision**

Use for: Classification (e.g., fraud detection)

Why: Measures correct positive predictions

When: False positives are costly.

**Recall**

Use for: Classification (e.g., medical tests)

Why: Measures correctly identified positives

When: False negatives are costly.

When: Large errors should be penalized.

**R-squared ($R^2$)**

Use for: Regression

Why: Explains variance in the target

When: General model fit is needed

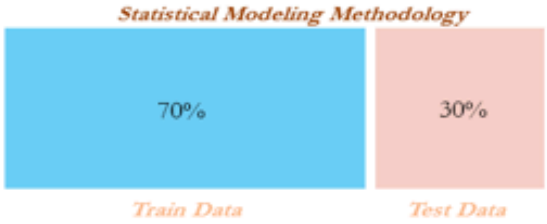
## How to split data

We usually split our data into (a) training and (b) testing

1. Training is responsible for fitting the model to learn a function
2. Testing is used to evaluate the goodness of the model

# Train/Test Split

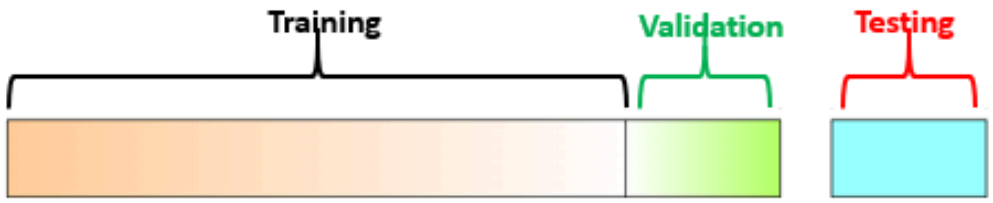
- Learning the parameters of a prediction function and testing it on the same data is a methodological mistake!! Why?
- A model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data.
- This situation is called overfitting (Memorizing).
- To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set
- So, we should divide the dataset into say 70% training data, 30% test data.
  - Use the training data to develop the model
  - Use the test data to evaluate the model performance.




5

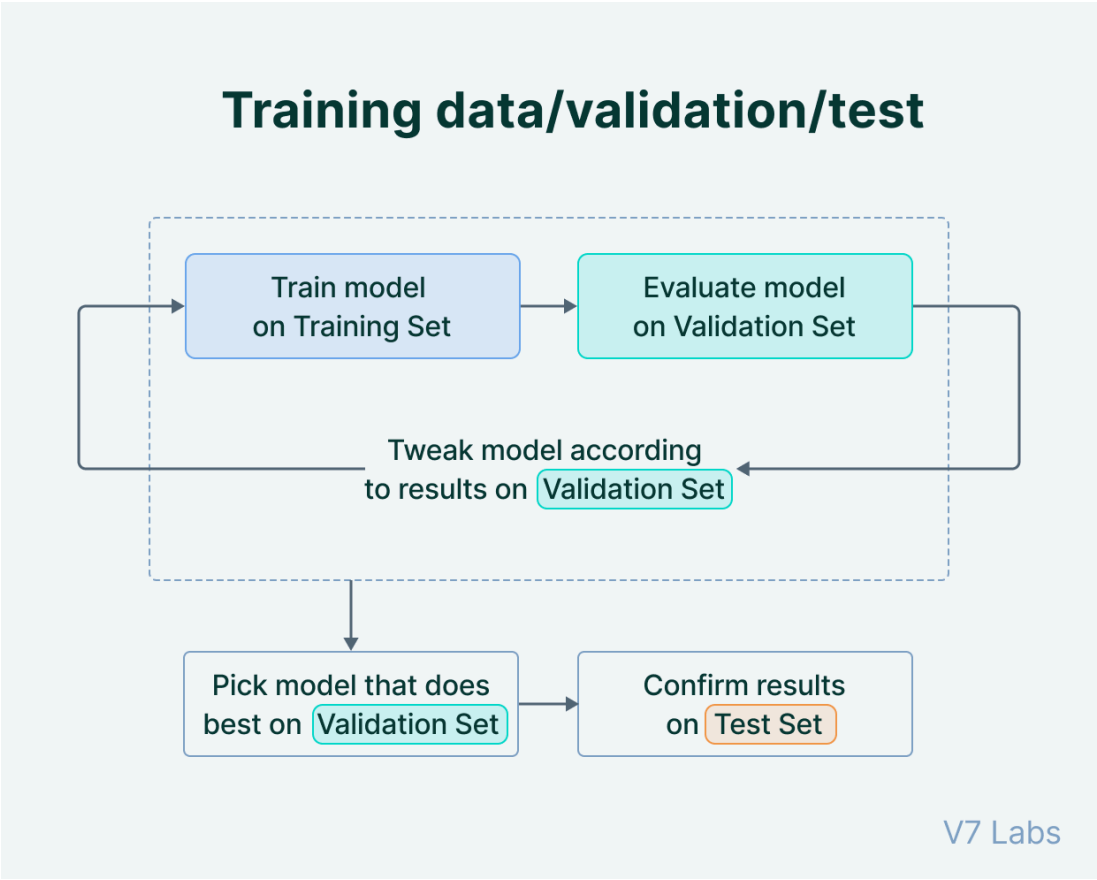
# Train/Validation/Test Split

- When evaluating different settings ("Hyperparameters") for estimators such as the value "k" in K-Means clustering or K-Nearest Neighbors, there is risk of overfitting (i.e. memorizing) on the test set because the parameters can be tweaked until the estimator performs optimally.
- To solve this problem, yet another part of the dataset can be held out as a so called "validation set": training proceeds on the training set, after which evaluation is done on the validation set, and when the experiment seems to be successful, final evaluation can be done on the test set.
- Divide the dataset into 70% training data, 15 % Validation, 15% test data.
  - Use the training data to develop the model
  - Use the validation data to tune the parameters of the model
  - Use the test data to evaluate the model performance.



# Train/Validation/Test Split

- What is the disadvantage of Train/Validate/Test??
- By partitioning the available data into three sets, we drastically reduce the number of samples which can be used for learning the model.
- Also, the results can depend on a particular random choice for the pair of (train, validation) sets!
- A solution to this problem is a procedure called Cross Validation.



### When to Use the Train-Test Split

A "sufficiently large" dataset means having enough data to split into training and test sets that reflect real-world cases. For small datasets, use **k-fold cross-validation** for better evaluation.

The **train-test split** is faster for large datasets or complex models, like deep learning. Random splitting ensures both sets represent the original data well.

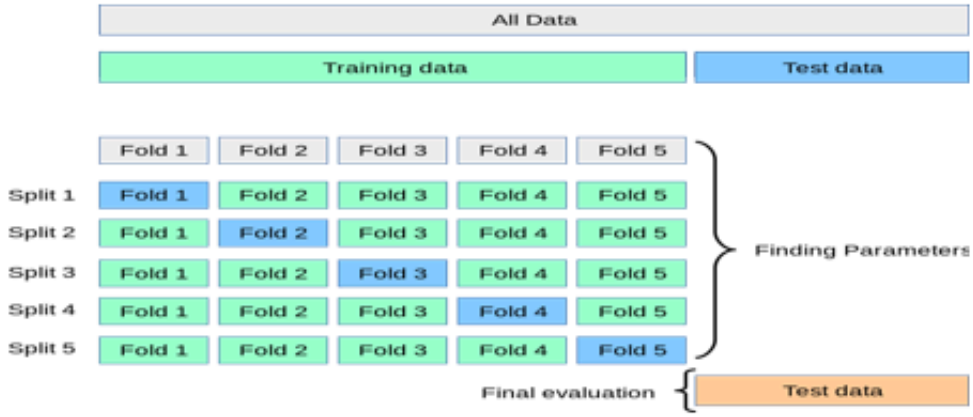
Choose the method based on your dataset size and the need for speed or accuracy.

## Cross Validation

Videos: Cross Validation in Machine Learning with Examples

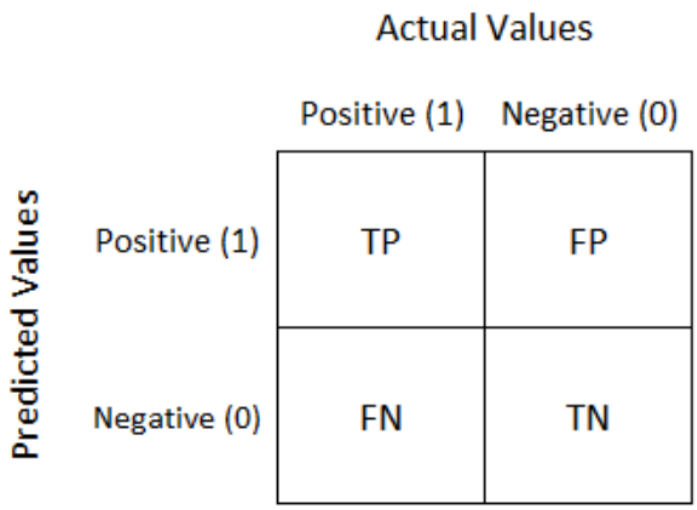
# K-Fold Cross Validation

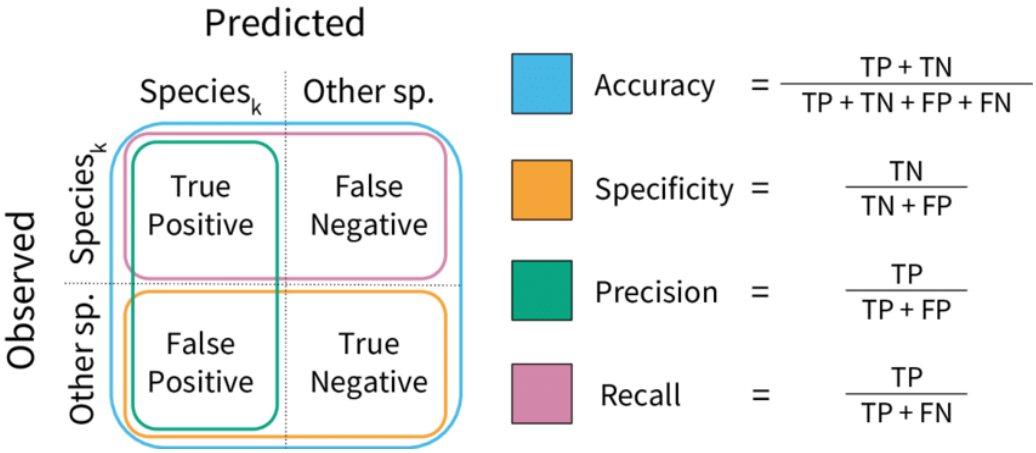
- In K-fold cross validation, a test set should still be held out for final evaluation, but the validation set is no longer needed when doing Cross Validation.
- The training set is split into k smaller sets.
- The value of k could be 5, 10 ...
- The model is trained using k-1 of the folds as training data
- The resulting model is validated on the remaining part (fold) of the data.




11

## Classification of Evaluation Matrics




I. **True Positives (TP):** The number of positive instances correctly classified as positive.
E.g., predicting an email as spam when it actually is spam.

II. **True Negatives (TN):** The number of negative instances correctly classified as negative.
E.g., predicting an email is not spam when it actually is not spam.

III. **False Positives (FP)**: The number of negative instances incorrectly classified as positive.
E.g., predicting an email is spam when it actually is not spam.

IV. **False Negatives (FN):** The number of positive instances incorrectly classified as negative.
E.g., predicting an email is not spam when it actually is spam.

**Mane**

TP= positive k positive predict korte parce

FP= positive k negative predict korce.

TN= negative k negative predict korce

FN= negative k positive predict korce

## Confusion Matrix Example -1 (from youtube)

- **Accuracy:** Overall, how often is the classifier correct?
- $Accuraccy = \frac{TN+TP}{TN+FP+FN+TP}$

$$= \frac{45+95}{150} = 93.33\%$$

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- **Misclassification Rate:** Overall, how often is it wrong?
- $Missclassification\ Rate = \frac{FN+FP}{TN+FP+FN+TP}$

$$= \frac{5+5}{150} = 6.67\%$$

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- **True Positive Rate:** When it's actually yes, how often does it predict yes?
- also known as "Sensitivity" or "Recall"
- $True\ Positive\ rate = \frac{TP}{Actual\ Yes}$

$$= \frac{95}{100} = 95\%$$

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- **False Positive Rate:** When it's actually no, how often does it predict yes?
- $False\ Positive\ rate = \frac{FP}{Actual\ No}$

$$= \frac{5}{50} = 10\%$$

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- **True Negative Rate:** When it's actually no, how often does it predict no?
- also known as "Specificity"

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- $True\ Negative\ rate = \frac{TN}{Actual\ No}$

$$= \frac{45}{50} = 90\%$$

- **Precision:** When it predicts yes, how often is it correct?

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- $Precision = \frac{TP}{Predicted\ Yes}$

$$= \frac{95}{100} = 95\%$$

- **Prevalence:** How often does the yes condition actually occur in our sample?

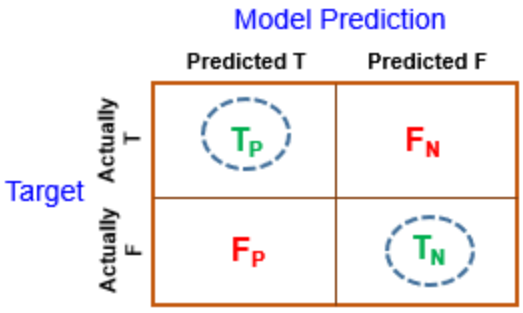
| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN = 45 | FP = 5 |
| Actual Yes | FN = 5 | TP = 95 |

- $Prevalence = \frac{Actual\ Yes}{Total}$

$$= \frac{100}{150} = 66.67\%$$

# Classification Evaluation Metrics

- A confusion matrix is an N x N matrix, where N is the number of target labels (classes)
- It shows the number of correct and incorrect predictions made by the classifier compared to the actual outcomes (target labels) in the actual data
- E.g., binary classification problem (e.g., two classes 0|1 or T|F)




Accuracy: the proportion of the total number of predictions that are correct

$T_N + T_P / (T_N + T_P + F_P + F_N)$

# Confusion Matrix Example

| Classification Confusion Matrix | | |
|---|---|---|
| | Predicted Class | |
| Actual Class | 1 | 0 |
| 1 | 201 | 85 |
| 0 | 25 | 2689 |

**201** 1's correctly classified as "1"

**85** 1's incorrectly classified as "0"

**25** 0's incorrectly classified as "1"

**2689** 0's correctly classified as "0"

Accuracy = (201 + 2689)/(201+85+25+2689) = 96.33%

# Confusion Matrix Example

| Classification Confusion Matrix | | |
|---|---|---|
| | Predicted Class | |
| Actual Class | 1 | 0 |
| 1 | 201 | 85 |
| 0 | 25 | 2689 |

**Total Samples = 3000**

Accuracy = (**201** + **2689**)/3000 = 96.33%

Overall Error Rate = (**25**+**85**)/3000 = 3.67%

Accuracy = 1 – Overall Error Rate = 96.33%

# Recall Class Imbalance

- For classification problems, we often use accuracy as evaluation metric.
- It is easy to calculate and intuitive:
  - Accuracy = # of correct predictions / # of total predictions
- But, it is misleading for highly imbalanced datasets!!.
- For example, in **credit card fraud detection**, we can set a model to always classify new transactions as legit. The accuracy could be high at 99.9% if 99.9% in the dataset is all legit.
- The same applies for imbalance that may occur in cancer classification problems. The majority are cancer free and the minority are cancer positive. Again, our goal is to detect cancer, so such a model is useless.
- That is why **we need other metrics** such as Precision, Recall, F1-Score, Sensitivity, Specificity and many others.

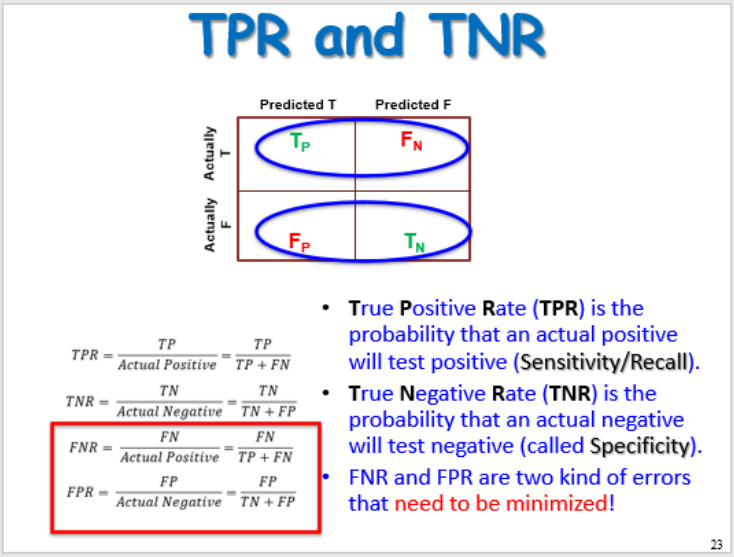
| Acquired Class \ True Class | T | F |
|---|---|---|
| Y | True Positives (TP) | False Positives (FP) |
| N | False Negatives (FN) | True Negatives (TN) |

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$

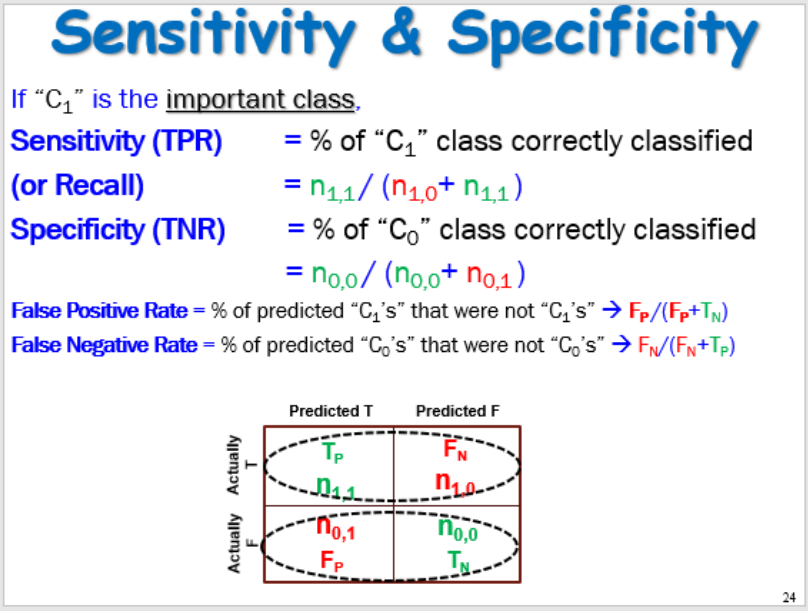
$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

$$\text{Accuracy (ACC)} = \frac{TP + TN}{TP + FP + TN + FN}$$

21

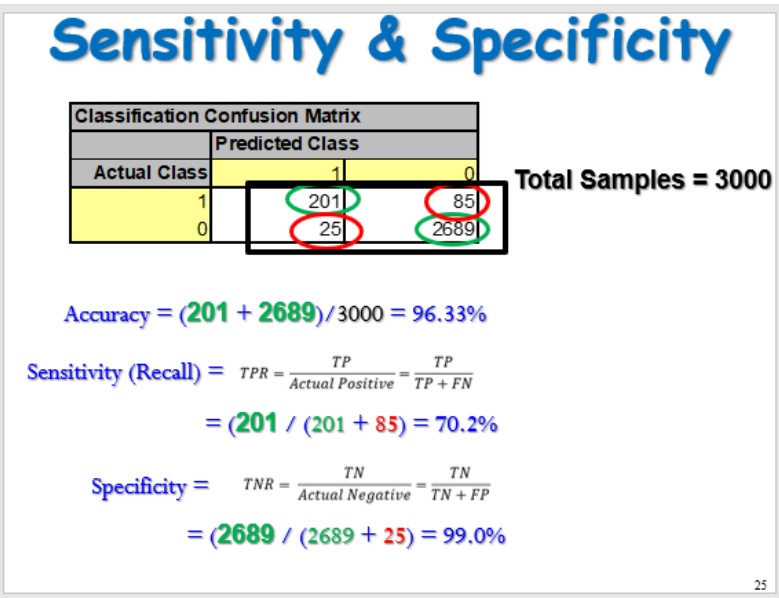



- The True Positive Rate (**TPR**, also called **sensitivity/Recall**) is calculated as TP/TP+FN.
  - **TPR** is the probability that an actual positive will test positive.
  - Indicates a test's **ability to detect the minority class** (detect spam, detect disease …)
- The True Negative Rate (**TNR**, also called **specificity**), is calculated as TN/TN+FP
  - **TNR** is the probability that an actual negative will test negative.
- Both **FNR and FPR are errors** that need to be minimized



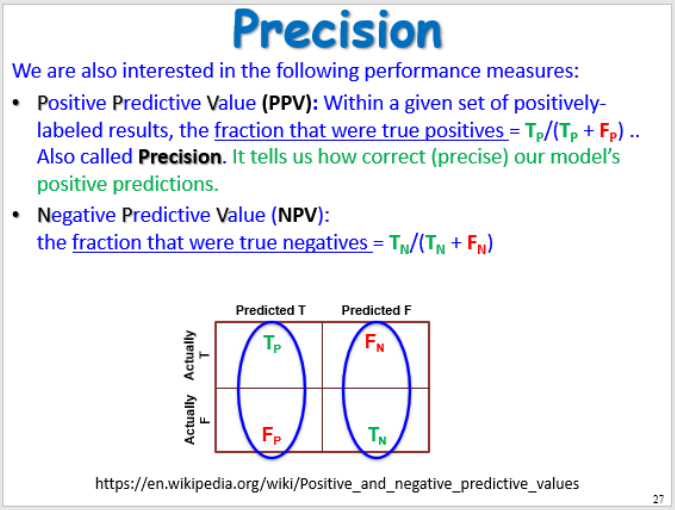

Depending on the important **class** we seek

- We either **improve** TPR or improve the TNR
- In all cases we seek to reduce the FPR and FNR




If we make a **comparison between** Accuracy, Recall and Specificity we notice

The **Negative class is the majority** and therefore the Overall Accuracy leans towards the majority class

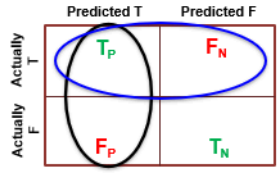



Beside TPR and TNR we might be interested in the following measures and metrics

1) **PPV** (Positive Predictive Value) or "**Precision**" → out of all the records that were **predicted to be Positive** or Class "1" what is the fraction that were **truly Positive**
2) **NPV** (Negative Predictive Value) → i.e., the **truly Negative records** out of all the records that were predicted to be negative

# Precision and Recall

- **Precision:** is the ratio of correctly predicted positive examples divided by the tot.# of positive examples that were predicted (calculates the accuracy of minority class).
- **Recall (Sensitivity):** is the ratio of correctly predicted positive classes to all items that are actually positive




- Precision $P = T_P/(T_P + F_P)$
- Recall $R = T_P/(T_P + F_N)$

28

There is also a **tradeoff between Precision and Recall** ...

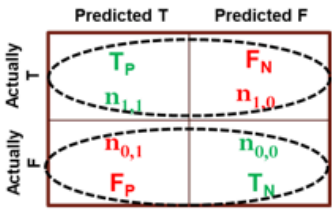
- There are **cases you mostly care about** the precision and **in other context you mostly care about** the recall.
- Which one is more important depends on
  - **Improving Precision** → thus reducing False Positive
  - Or **improving Recall** → thus reducing False Negative

# Introduction to ROC Curves

- *ROC = Receiver Operating Characteristic curves*
- *The **ROC** is another common tool used for evaluation.*
- It plots out the sensitivity and specificity for every possible decision rule cutoff between 0 and 1 for a model.
- Has become very popular in biomedical applications, particularly radiology and imaging
- Also used in machine learning applications to assess classifiers
- Can be used to compare tests/procedures

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

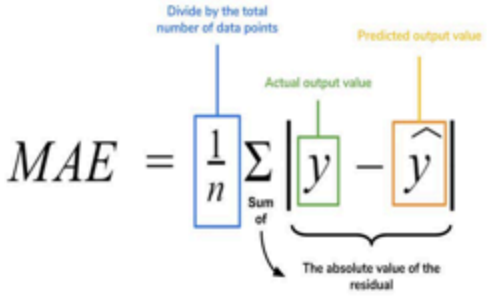


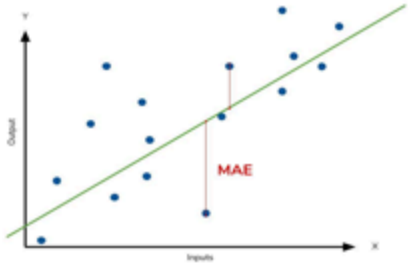

33

The **ROC curve** shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR).

# Regression Evaluation "MAE"

- The **M**ean **A**bsolute **E**rror (**MAE**) is the simplest regression error metric to understand.
- The absolute value of the residual is taken so that negative and positive residuals do not cancel out.
- The average of all residuals is then taken only the absolute value of each
- A small MAE suggests the model is great at prediction
- A MAE of 0 means that your model is **a perfect predictor**



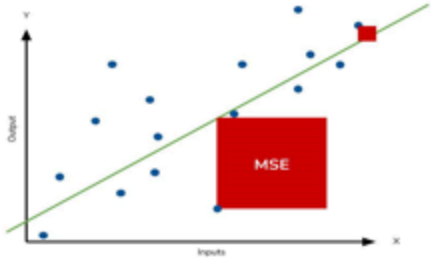



# Regression Evaluation "MSE"

- The **M**ean **S**quare **E**rror (MSE) is just like the MAE, but squares the difference before summing them all instead of using the absolute value.
- Because we are squaring the difference, the MSE will almost be bigger than the MAE.
- It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors.
- The effect of the square term in the MSE equation is most apparent with the presence of outliers in our data.
- Our model will be penalized more for making predictions that differ greatly from the corresponding actual value.
- The large differences between actual and predicted are punished more in MSE than in MAE.

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)}^{2}$$

The square of the difference between actual and predicted



# Regression Evaluation "RMSE"

- The **R**oot **M**ean **S**quare **E**rror (RMSE) is the most widely used metric for regression tasks and is the square root of the averaged squared difference between the target value and the value predicted by the model
- It is preferred more in some cases because the errors are first squared before averaging which poses a high penalty on large errors.
- The RMSE is useful when large errors are undesired.
- However, the range of the dataset you're working with is important in determining whether or not a given RMSE value is "low" or not.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

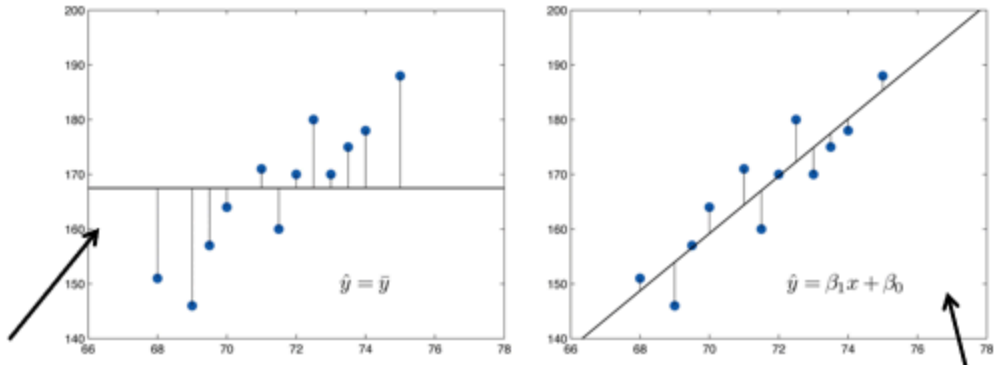
# Regression Evaluation "$R^2$"

- The $R^2$ Error (also called Coefficient of Determination) is another metric used for evaluating the performance of a regression model.
- The most common interpretation of the coefficient of determination is how well the regression model fits the observed data.
- $R^2$ will always be less than or equal to 1.
- $R^2$ is used to explain how much variability of one factor can be caused by its relationship to another factor.
- If the $R^2$ of a model is 0.75, then approximately 75% of the observed variation can be explained by the model's features.

$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

# Regression Evaluation "$R^2$"

- The constant baseline is chosen by taking the mean of the data and drawing a line at the mean.




MSE (Baseline)

(a): Set I

(b): Set II

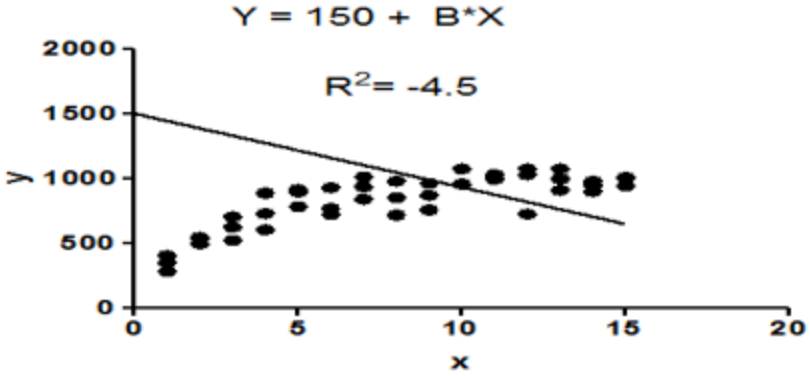
MSE (Model)

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

# Negative "$R^2$" !!

- $R^2$ compares the fit of the chosen model with that of a horizontal straight line (the null hypothesis).
- If the chosen model fits worse than a horizontal line, then $R^2$ is **negative.**
- $R^2$ is negative only when the chosen model does not follow the trend of the data, so fits worse than a horizontal line.

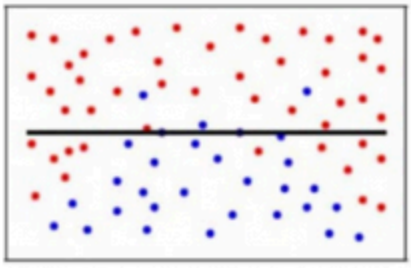
One of the drawbacks of $R^2$ is that the more features are added to a model, the more the $R^2$ increases. This happens even though the features added to the model are not intrinsically predictive.




10

# What is Bias?

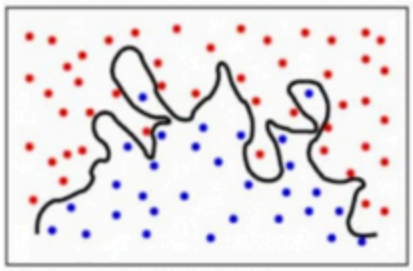
- Bias is the difference between the Predicted Value and Expected Value.
- The model makes certain assumptions when it trains on the data provided.
- When the model is introduced to the testing/validation data, these assumptions may not always be correct.
- In a nutshell,
  - The model used is simple (i.e., predicting a simple relationship when the data points indicate a more complex relationship).
  - The model does not consider the variations very well and does not learn the training data very well



The bias is a measure of how close the model can capture the mapping function between inputs and outputs.

# What is Variance?

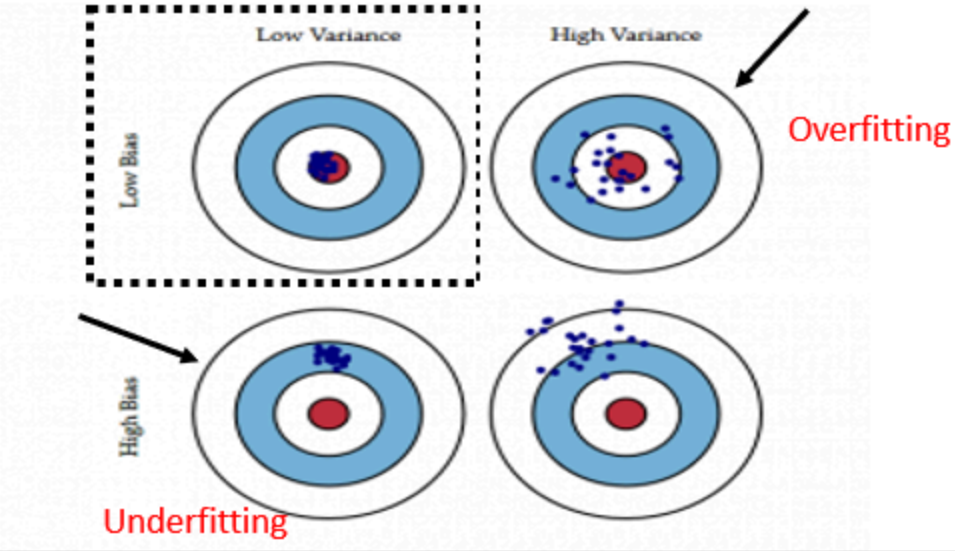
- The variance is when the model takes into account the fluctuations in the data i.e. the noise as well
- When a model has a high variance it learns too much from the training data, (it **memorizes** rather than learns), so much, so that when it is confronted with new (testing) data, it is unable to predict accurately based on it.
- The ML Model used is **too complex** for the task.
- The model **predicts very complex relationships** between the outcome and the input features when for example a quadratic equation would have sufficed.



14

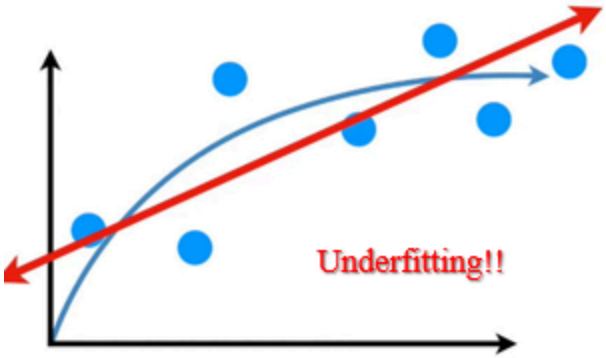
# Bias-Variance Tradeoff

- The following bulls-eye diagram explains the tradeoff:
- Ideally, we would prefer a model with low bias and low variance (challenging!!)
- A model with low bias and high variance predicts points that are around the center generally, but pretty far away from each other.
- A model with high bias and low variance is pretty far away from the bull's eye, but since the variance is low, the predicted points are closer to each other (it is more consistent).

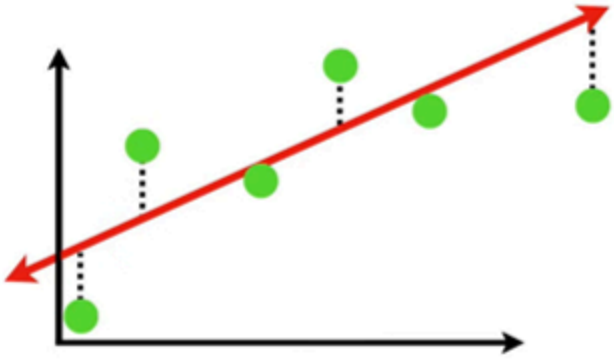



15

# Bias/Variance Tradeoff

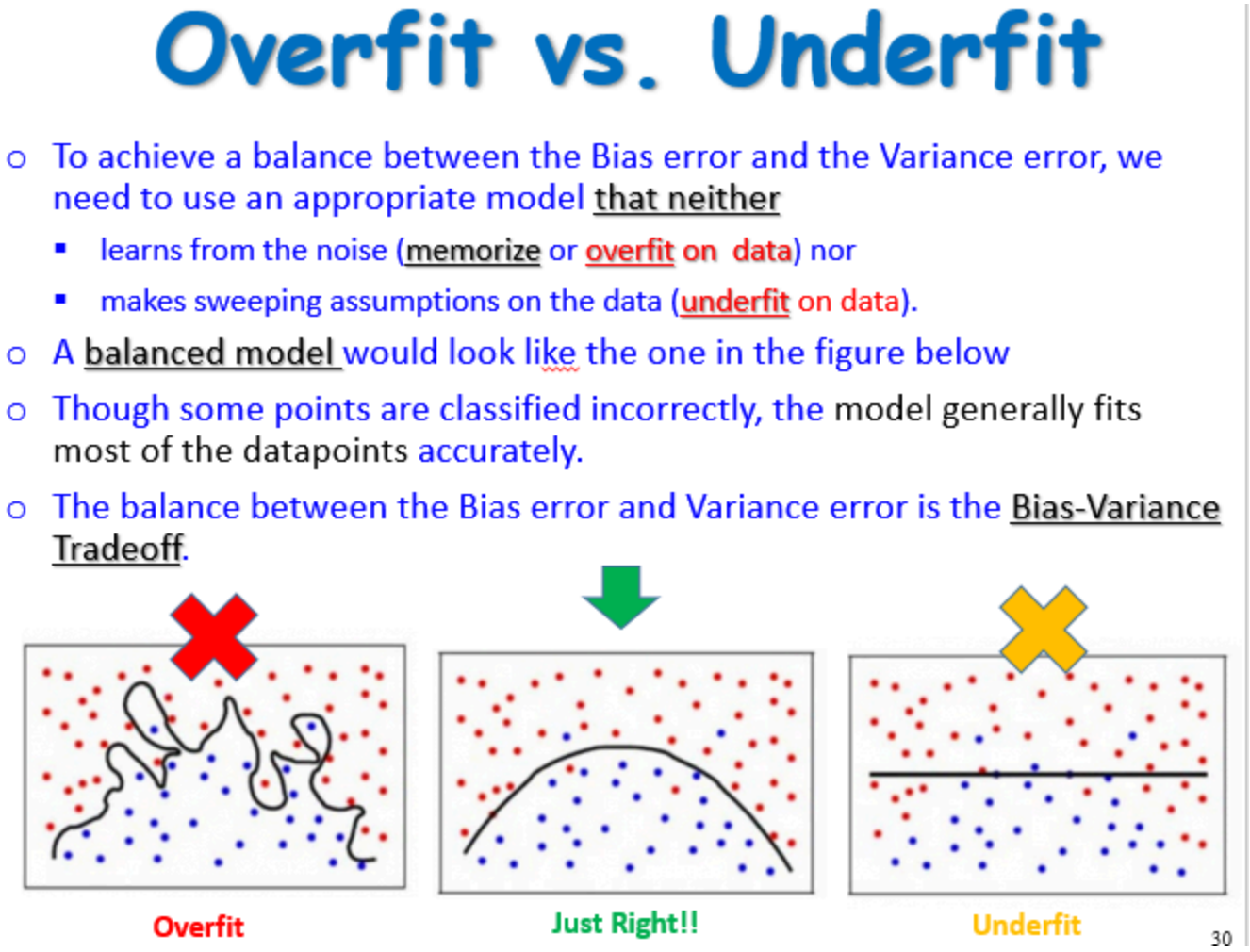



In contrast, the Straight Line has relatively high bias since it can not capture the curve in the relationship between weight and height …



… but the Straight Line has relatively low variance, because the Sums of Squares are very similar for different datasets.

In other words, the Straight Line might only give good predictions, And not great predictions … But they will be **consistently good predictions**.



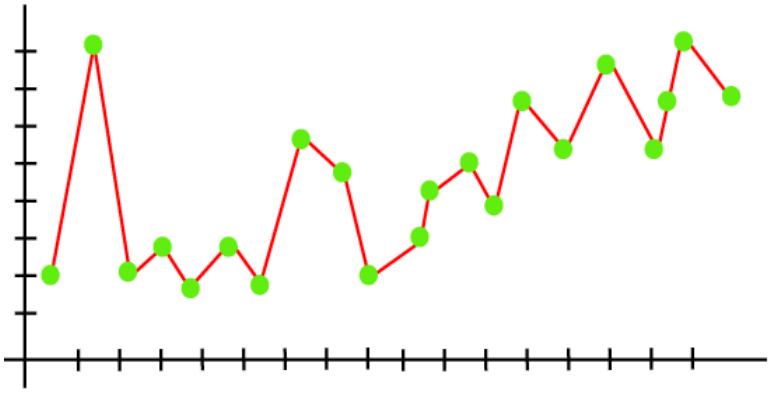

In machine learning, overfitting and underfitting are both common issues that can cause a model to perform poorly. The main difference between the two is that an overfitted model performs well on training data but poorly on unseen data, while an under fitted model performs poorly on both training and unseen data

To learn more about overfilling and underfitting

**Overfitting**

Overfitting occurs when our machine learning model tries to cover all the data points or more than the required data points present in the given dataset. Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model. The overfitted model has low bias and high variance.

Example: The concept of overfitting can be understood by the below graph of the linear regression output:



As we can see from the above graph, the model tries to cover all the data points present in the scatter plot. It may look efficient, but in reality, it is not so. Because the goal of the regression model to find the best fit line, but here we have not got any best fit, so, it will generate the prediction errors.

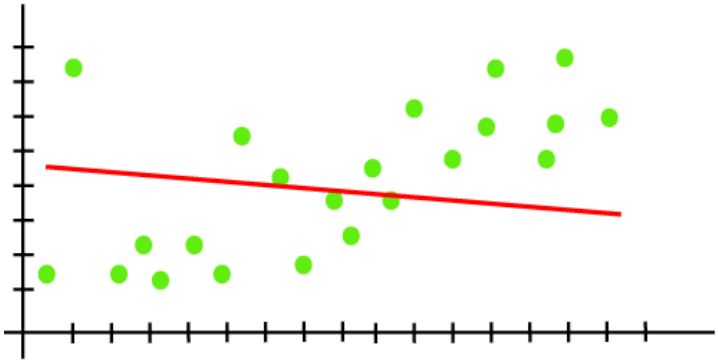
How to avoid the Overfitting in Model

- Cross-Validation
- Training with more data
- Removing features
- Early stopping the training
- Regularization
- Ensembling

**Underfitting**

Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data. To avoid overfitting in the model, the fed of training data can be stopped at an early stage, due to which the model may not learn enough from the training data. As a result, it may fail to find the best fit of the dominant trend in the data.An underfitted model has high bias and low variance.

Example: We can understand the underfitting using below output of the linear regression model:

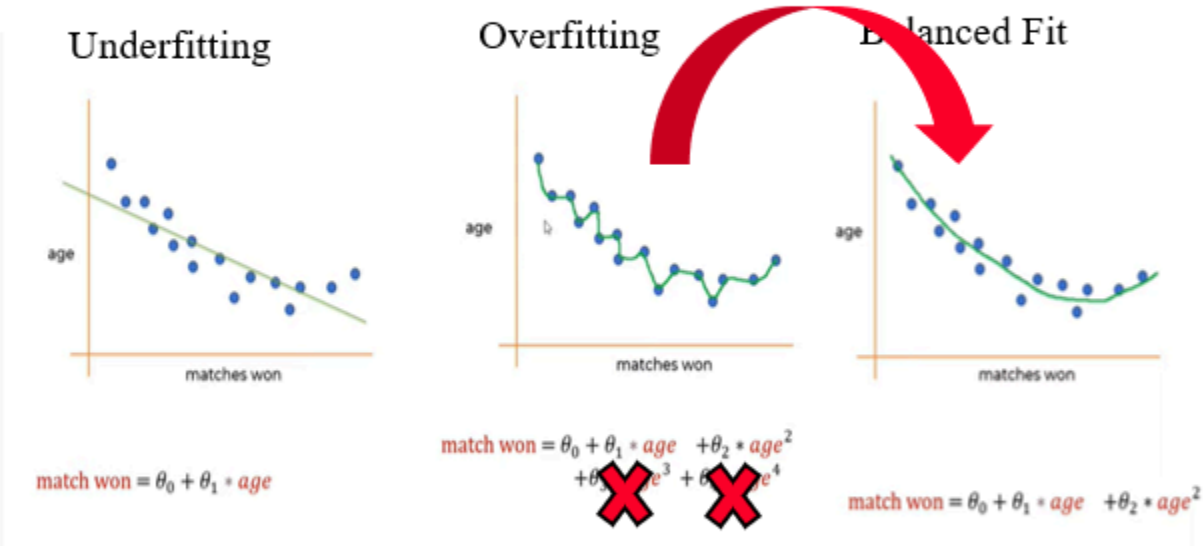


As we can see from the above diagram, the model is unable to capture the data points present in the plot.

How to avoid underfitting:

- By increasing the training time of the model.
- By increasing the number of features.

# Regularization




- Regularization in simple words is adding a penalty on say weights of the model to the objective function such that the **model becomes simpler** by making these weights smaller and thus reduce overfitting

Regularization in machine learning is a set of techniques that help prevent overfitting in models. It involves normalizing the weights associated with features or neurons so that algorithms don't rely too heavily on a few of them to make predictions.

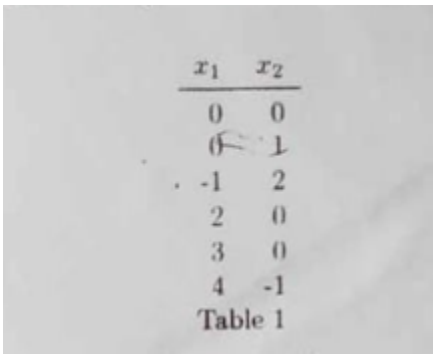
# Ques Solve Part

5th final —-> 5(C), 7(b),

4th final —–>  2(c),

# 6thBatch Mid

1. Compute any three-distance matrix from the table data in Table 1.



| $x_1$ | $x_2$ |
|---|---|
| 0 | 0 |
| 0 | 1 |
| -1 | 2 |
| 2 | 0 |
| 3 | 0 |
| 4 | -1 |

Table 1

Sol: See K means Clustering Distance measure part for solving that

2. Describe the different methods used to split the dataset in machine learning.
Sol: See Evaluation part

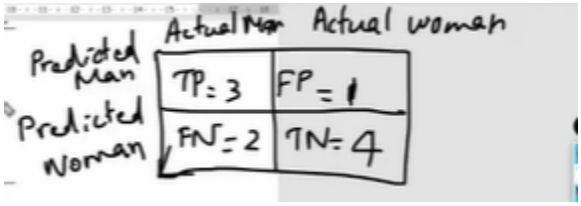
3. Distinguish Lazy and Eager Learning.

| Feature | Lazy Learning | Eager Learning |
|---|---|---|
| Generalization | Adapts quickly | Less flexible |
| Model complexity | Less complex | More complex |
| Training time | Minimal | Longer |
| Prediction time | Slower | Faster |
| Memory usage | Higher | Lower |
| Interpretability | More interpretable | Varies |
| Online Learning | Well-suited | Less suitable |
| Robustness | Less robust | More robust |

4. Compute the accuracy, precision, recall, sensitivity, and specificity of the data.

| | Expected/Actual | Predicted |
|---|---|---|
| 1 | Man | Woman |
| 2 | Man | Man |
| 3 | Woman | Woman |
| 4 | Man | Man |
| 5 | Woman | Man |
| 6 | Woman | Woman |
| 7 | Woman | Woman |
| 8 | Man | Man |
| 9 | Man | Woman |
| 10 | Woman | Woman |

**Sol:**

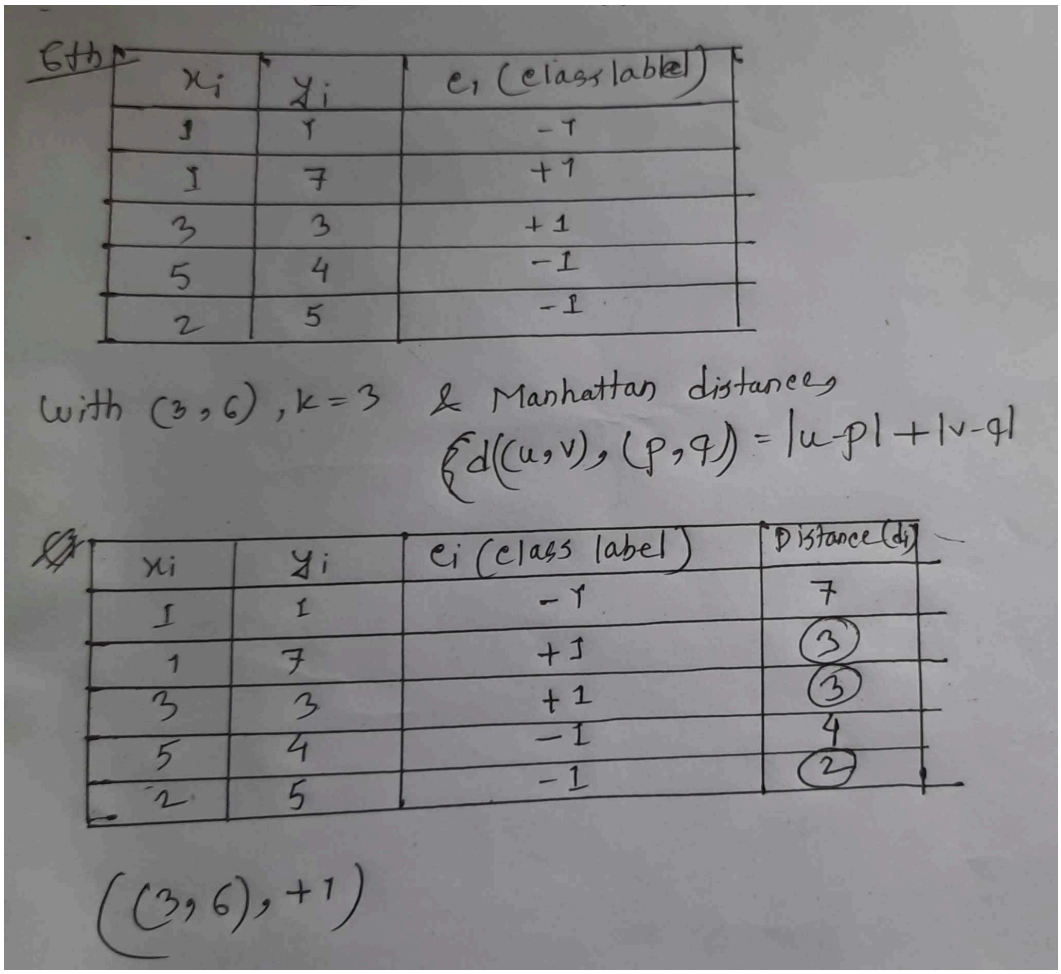


$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{Specificity} = \frac{TN}{TN + FP}$$

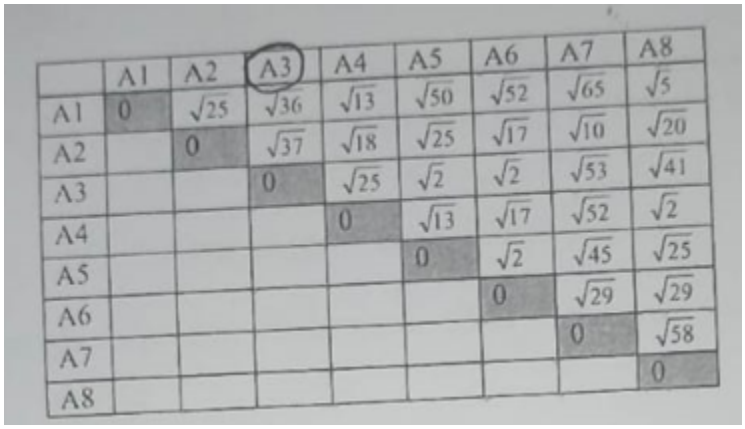
5. Consider a set of five training examples given as ((xi, yi), ci) values, where xi and yi are the two attribute values (positive integers) and ci is the binary class label: (((1, 1), -1), ((1, 7), +1), ((3, 3), +1), ((5, 4), -1), ((2, 5), -1)}. Classify a test example at coordinates (3, 6) using a k-NN classifier with k = 3 and Manhattan distance defined by d((u, v), (p, q)) = l u - p l+ |v-q |. Your answer should be either +1 or -1.

Sol:



# 4th Batch Mid

1. Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters; A1-(2,10), A2-(2,5), A3-(8,4). A4-(5,8), A5-(7,5), A6 (6,4), A7-(1,2), A8-(4,9)
The distance matrix based on the Euclidean distance is given below:



| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 | $\sqrt{25}$ | $\sqrt{36}$ | $\sqrt{13}$ | $\sqrt{50}$ | $\sqrt{52}$ | $\sqrt{65}$ | $\sqrt{5}$ |
| A2 | | 0 | $\sqrt{37}$ | $\sqrt{18}$ | $\sqrt{25}$ | $\sqrt{17}$ | $\sqrt{10}$ | $\sqrt{20}$ |
| A3 | | | 0 | $\sqrt{25}$ | $\sqrt{2}$ | $\sqrt{2}$ | $\sqrt{53}$ | $\sqrt{41}$ |
| A4 | | | | 0 | $\sqrt{13}$ | $\sqrt{17}$ | $\sqrt{52}$ | $\sqrt{2}$ |
| A5 | | | | | 0 | $\sqrt{2}$ | $\sqrt{45}$ | $\sqrt{25}$ |
| A6 | | | | | | 0 | $\sqrt{29}$ | $\sqrt{29}$ |
| A7 | | | | | | | 0 | $\sqrt{58}$ |
| A8 | | | | | | | | 0 |

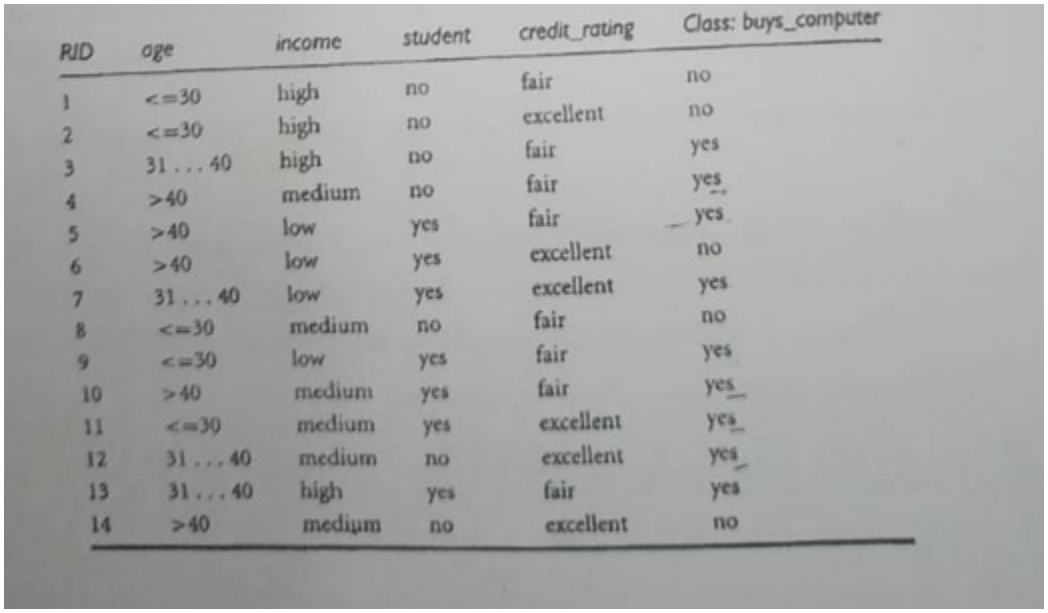
Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7. Run the k-means algorithm for
1 epoch only. At the end of this epoch show:
a. The new clusters (i.e. the examples belonging to each cluster)
b. The centers of the new clusters
c. Draw a 10 by 10 space with all the 8 points and show the clusters after the first epoch and the new centroids.
d. How many more iterations are needed to converge? Draw the result for each epoch.

Sol: See K-Means Clustering Part

2. a. Why is the Naïve Bayesian classification called "naïve"?
Sol: see Naive Bayas part

b.Given the training data in the table below (Buy Computer data), predict the class of the following 8 new examples using Naïve Bayes classification:

age<=30, income-medium, student-yes, credit- rating-fair



| RID | age | income | student | credit_rating | Class: buys_computer |
|---|---|---|---|---|---|
| 1 | <=30 | high | no | fair | no |
| 2 | <=30 | high | no | excellent | no |
| 3 | 31 . . . 40 | high | no | fair | yes |
| 4 | >40 | medium | no | fair | yes |
| 5 | >40 | low | yes | fair | yes |
| 6 | >40 | low | yes | excellent | no |
| 7 | 31 . . . 40 | low | yes | excellent | yes |
| 8 | <=30 | medium | no | fair | no |
| 9 | <=30 | low | yes | fair | yes |
| 10 | >40 | medium | yes | fair | yes |
| 11 | <=30 | medium | yes | excellent | yes |
| 12 | 31 . . . 40 | medium | no | excellent | yes |
| 13 | 31 . . . 40 | high | yes | fair | yes |
| 14 | >40 | medium | no | excellent | no |

Sol: See Naive Bayas Part

# 4th Batch Final

# 1.

a) What is machine learning? Write down the relation between artificial intelligence (AI) and Machine Learning (ML). [2.5]

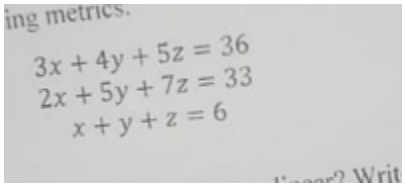
Sol: See Introduction Part

b) What is the difference between supervised and unsupervised machine learning? Give example.[3]
Sol: See Introduction Part

c) How can you define deep learning, and how does it contrast with other machine learning algorithms? [2.5]

Sol: See Introduction Part

d) Solve the equations given below using metrics.[4]




Sol:
To solve the system of equations using matrices, we can express the system as:

$$\begin{aligned} 3x + 4y + 5z &= 36 \\ 2x + 5y + 7z &= 33 \\ x + y + z &= 6 \end{aligned}$$

Step 1: Write in Matrix Form

The system of linear equations can be written in matrix form as:

$$AX = B$$

Where:

$$A = \begin{bmatrix} 3 & 4 & 5 \\ 2 & 5 & 7 \\ 1 & 1 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad B = \begin{bmatrix} 36 \\ 33 \\ 6 \end{bmatrix}$$

Step 2: Solve the Matrix Equation

The solution to the system is given by:

$$X = A^{-1}B$$

Where $A^{-1}$ is the inverse of matrix A.

To calculate the inverse of a matrix A, follow these steps:

Step 1: Write Matrix A

Matrix A is given as:

$$A = \begin{bmatrix} 3 & 4 & 5 \\ 2 & 5 & 7 \\ 1 & 1 & 1 \end{bmatrix}$$

Step 2: Check if A is Invertible

For a matrix to have an inverse, its determinant must be non-zero. The determinant of matrix A is:

$$\det(A) = 3(5-7) - 4(2-7) + 5(2-5) = 3(-2) - 4(-5) + 5(-3) = -6 + 20 - 15 = -1$$

Since the determinant is non-zero (det(A) = −1), matrix A is invertible.

Step 3: Adjoint of Matrix A

The inverse of a matrix A can be calculated using the formula:

$$A^{-1} = \frac{1}{\det(A)} \times \text{Adj}(A)$$

Where Adj(A) is the adjugate (or adjoint) of matrix A, which is the transpose of the cofactor matrix.

Step 3.1: Calculate Cofactor Matrix

To calculate the cofactor matrix, you must find the cofactor for each element of matrix A.

$$\text{Cofactor of } a_{11}: \text{Cofactor}(3) = 5 \times 1 - 7 \times 1 = -2$$

$$\text{Cofactor of } a_{12}: \text{Cofactor}(4) = 2 \times 1 - 7 \times 1 = -5$$

$$\text{Cofactor of } a_{13}: \text{Cofactor}(5) = 2 \times 1 - 5 \times 1 = -3$$

Continue this for all elements:

$$\text{Cofactor Matrix} = \begin{bmatrix} -2 & 5 & -3 \\ 1 & -1 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

Step 3.2: Find the Adjugate Matrix

The adjugate matrix is the transpose of the cofactor matrix:

$$\text{Adj}(A) = \begin{bmatrix} -2 & 1 & -1 \\ 5 & -1 & 2 \\ -3 & 0 & -1 \end{bmatrix}$$

Step 4: Calculate the Inverse

Now, calculate the inverse using the formula:

$$A^{-1} = \frac{1}{\det(A)} \times \text{Adj}(A)$$

Substitute the values:

$$A^{-1} = \frac{1}{-1} \times \begin{bmatrix} -2 & 1 & -1 \\ 5 & -1 & 2 \\ -3 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 1 \\ -5 & 1 & -2 \\ 3 & 0 & 1 \end{bmatrix}$$

So,

$X = A^{-1}B$

$X = \begin{bmatrix} 2 & -1 & 1 \\ -5 & 1 & -2 \\ 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 36 \\ 33 \\ 6 \end{bmatrix}$

2. a) How can you convert the non-linear decision boundary to linear? Write with an example.

Sol: See SVM part

or,

When faced with a non-linearly separable dataset, a linear classifier like logistic regression or linear SVM will struggle to accurately separate the classes. This is because the decision boundary required to separate the classes is not a straight line but a more complex curve.

**The Solution: Kernel Trick**

The kernel trick is a mathematical technique that allows us to implicitly map data points into a higher-dimensional feature space without explicitly computing the new representations. This higher-dimensional space often makes the data linearly separable, allowing a linear classifier to perform well.

**How it Works**

1. **Kernel Function:** A kernel function is a similarity measure between pairs of data points. It computes the inner product of the mapped data points in the higher-dimensional space without actually performing the mapping.
2. **Dual Representation:** Instead of working with the original data points, the algorithm works with a dual representation, which involves computing the inner products between all pairs of data points using the kernel function.
3. **Linear Classifier in the Dual Space:** A linear classifier is trained on the dual representation, effectively learning a linear decision boundary in the higher-dimensional space.

**Example: Support Vector Machines (SVMs) with Kernels**

Consider a dataset that is not linearly separable in the original 2D space. By using a kernel function like the Radial Basis Function (RBF) kernel, we can implicitly map the data points into a higher-dimensional space where they become linearly separable.

**RBF Kernel:**

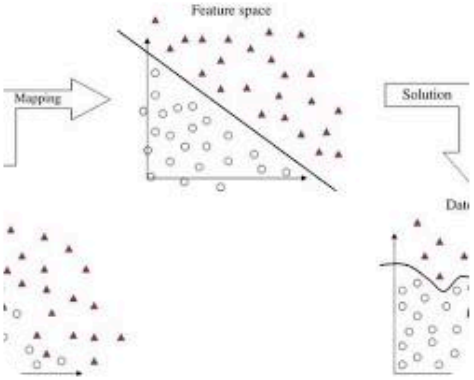
```
K(x, y) = exp(-gamma * ||x - y||^2)
```

Here, `gamma` is a hyperparameter that controls the width of the RBF kernel.

**Steps:**

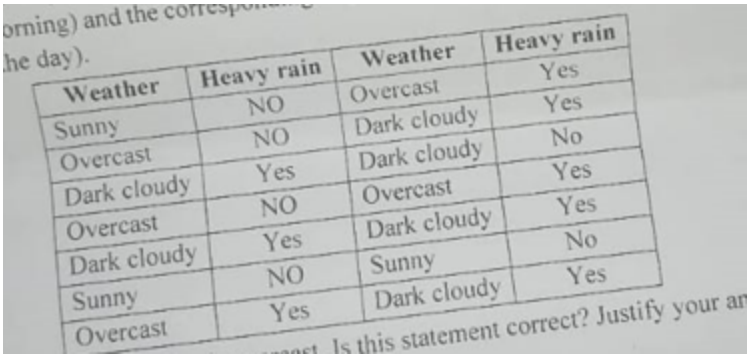
1. **Choose a Kernel:** Select an appropriate kernel function based on the characteristics of the data.
2. **Train the SVM:** Train the SVM using the kernel function. The SVM will learn a linear decision boundary in the higher-dimensional space.
3. **Make Predictions:** To classify new data points, compute their kernel values with all training points and use the learned weights to make predictions.

**Visual Representation:**




nonlinearly separable dataset in 2D space, mapped into a higherdimensional space using an RBF kernel, becoming linearly separable

b) Discuss the Naïve Bayes' classification algorithm. Consider the table below, the training set of weather (in the morning) and the corresponding target variable 'Heavy rain' (i.e., the possibility of heavy rain in the day).



orning) and the correspon
he day).

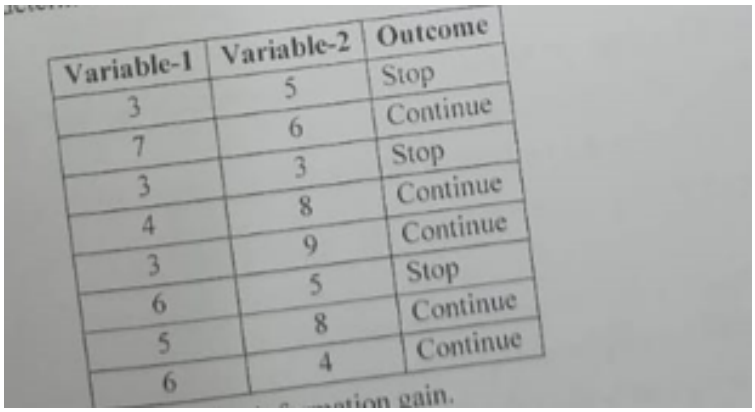
| Weather | Heavy rain | Weather | Heavy rain |
|---|---|---|---|
| Sunny | NO | Overcast | Yes |
| Overcast | NO | Dark cloudy | Yes |
| Dark cloudy | Yes | Dark cloudy | No |
| Overcast | NO | Overcast | Yes |
| Dark cloudy | Yes | Dark cloudy | Yes |
| Sunny | NO | Sunny | No |
| Overcast | Yes | Dark cloudy | Yes |

cast. Is this statement correct? Justify your ar

There will be heavy rain if the weather is overcast. Is this statement correct? Justify your answer

Sol: See Naive Bayes part

c) What is information gain? Consider the example in the following table where variable-1 and variable-2 are used to determine whether to continue with the experiment or to stop the experiment.



| Variable-1 | Variable-2 | Outcome |
|---|---|---|
| 3 | 5 | Stop |
| 7 | 6 | Continue |
| 3 | 3 | Stop |
| 4 | 8 | Continue |
| 3 | 9 | Continue |
| 6 | 5 | Stop |
| 5 | 8 | Continue |
| 6 | 4 | Continue |

mation gain.

Sol:

Whether to continue the experiment or to stop the experiment:

$I.G.(\text{outcome}) = \frac{-3}{8}\log_2\left(\frac{3}{8}\right) - \frac{5}{8}\log_2\left(\frac{5}{8}\right)$ continue = 5+ stop = 3-

$= 0.9544$

For variable 1:

$3[1+, 2-] \Rightarrow IG(3) = \frac{-1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)$

$E(3) = 0.918 \times \frac{3}{8} = 0.3444$

$7[1+, 0] \Rightarrow E(7) = 0$

$4[1+, 0] \Rightarrow E(4) = 0$

$5[1+, 0] \Rightarrow E(5) = 0$

$6[1+, 1-] \Rightarrow IG.(6) = \frac{-1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)$

$E(6) = 1 \times \frac{2}{8} = 0.25$

$E(\text{variable 1}) = (0.3444 + 0.25) = 0.5944$

$\text{Gain}(\text{variable 1}) = (0.9544 - 0.5944) = 0.36$

For variable 2:

$5[0+, 2-] \Rightarrow E(5) = 0$ $\quad 9[1+, 0-] = 0$

$6[1+, 0] \Rightarrow E(6) = 0$ $\quad 4[1+, 0-] = 0$

$8[2+, 0] \Rightarrow E(8) = 0$ $\quad \therefore \text{Gain}(\text{variable 2}) = 0.954$

$3[0+, 1-] \Rightarrow E(3) = 0$

∴ Based on the first decision variable 2 should be choosen for stop or continue.

3. a) What do you mean by regression? When should multiple regression analysis be used?
Sol: see Regression part
b) How do you perform a linear regression?
Soll: See regression part

c) The sales of a company (in million dollars) for each year are shown in the table below.

| x (year) | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| y (sales) | 12 | 19 | 29 | 37 | 45 |

i) Find the least square regression line y = ax + b.
ii) Use the least squares regression line as a model to estimate the sales of the company in year 7.

Sol: See Regression Part

4. Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:
A1 (2,10), A2 = (2,5), A3 = (8,4), A4 = (5,8), A5 = (7,5), A6 = (6,4), A7 = (1,2), A8 = (4,9)
The distance matrix based on the Euclidean distance is given below:

| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 | $\sqrt{25}$ | $\sqrt{36}$ | $\sqrt{13}$ | $\sqrt{50}$ | $\sqrt{52}$ | $\sqrt{65}$ | $\sqrt{5}$ |
| A2 | | 0 | $\sqrt{37}$ | $\sqrt{18}$ | $\sqrt{25}$ | $\sqrt{17}$ | $\sqrt{10}$ | $\sqrt{20}$ |
| A3 | | | 0 | $\sqrt{25}$ | $\sqrt{2}$ | $\sqrt{2}$ | $\sqrt{53}$ | $\sqrt{41}$ |
| A4 | | | | 0 | $\sqrt{13}$ | $\sqrt{17}$ | $\sqrt{52}$ | $\sqrt{2}$ |
| A5 | | | | | 0 | $\sqrt{2}$ | $\sqrt{45}$ | $\sqrt{25}$ |
| A6 | | | | | | 0 | $\sqrt{29}$ | $\sqrt{29}$ |
| A7 | | | | | | | 0 | $\sqrt{58}$ |
| A8 | | | | | | | | 0 |

Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7. Run the k-means algorithm for 1 epoch only. At the end of this epoch show:
a) The new clusters (i.e., the examples belonging to each cluster)
b) The centers of the new clusters

c) Draw a 10 by 10 space with all the 8 points and show the clusters after the first epoch and the new centroids.

Sol: See K Means Clustering Part

5. a) What is the basic difference of Recurrent Neural Network (RNN) from Artificial Neural Network? Give some examples of RNN applications.
Sol: RNN nai Syllabus e amader

b. What is the vanishing Gradient Problem of RNN?
Sol: RNN nai Syllabus e amader

c) Shortly describe Long-Term Memory (LSTM) networks and explain how it can avoid long term dependency problems.
Sol: Maybe syllabus e nai taw ditechi solve

LSTM stands for "Long Short-Term Memory," a type of recurrent neural network (RNN) designed to effectively handle long-term dependencies in sequential data by using a special "memory cell" and "gates" to selectively store and retrieve information over extended periods, overcoming the vanishing gradient problem that plagues traditional RNNs; essentially, it can remember relevant information from earlier parts of a sequence when making predictions about later parts, making it ideal for tasks like natural language processing and time series analysis.

6. a) What is the basic principle of a Support Vector Machine?
Sol; See SVM parts

b) Why would you use the Kernel Trick in the Support Vector Machine? Explain with example.
See SVM part

c) What happens when there is no clear Hyperplane in SVM?
See SVM part

d) When would you use SVMs over Random Forest and vice-versa?
See SVM part

7.

a) What are the differences between supervised and unsupervised methods?
See Introduction of ML
b) How does supervised learning work?
See Introduction of ML
c) Briefly describe the steps involved in Supervised Learning.
See Introduction of ML
d) Briefly describe the types of unsupervised learning algorithms.
See Introduction of ML

8.
a) What does machine learning life cycle?

Sol: see Introduction Part

b) Describe the steps involved in the ML life cycle.
Sol: see Introduction Part

c) (True or False?) Justify your answer.
i) If you are given m data points, and use half for training and a half for testing, the difference between training error and test error decreases as m increases.
ii) Overfitting is more likely when the set of training data is small
iii) Overfitting is more likely when the hypothesis space is small
Sol:
**i) True**
With more data points, the model generalizes better, reducing the difference between training and test errors.

**ii) True**
Small training datasets lead to overfitting because the model can memorize specific details rather than general patterns.

**iii) False**
Overfitting happens with a large hypothesis space, as the model becomes too flexible and fits noise. A small hypothesis space prevents this.

d) Describe the methods used to split the dataset in machine learning.
Sol: See The evaluation part

# 5th Batch Final

1. a) What are the differences between supervised and unsupervised methods? [2]
Sol: See Introduction Part

b) How does supervised learning work? [4]
Sol: See Introduction Part

c) Describe the steps involved in the ML life cycle.[6]
Sol: See Introduction Part

2. a) Describe the methods used to split the dataset in machine learning.[3]

Sol: See Evaluation Part

b) Why is data preprocessing important in machine learning? Describe three popular data preprocessing techniques in machine learning.[4]
Sol:
Data preprocessing is crucial in machine learning because it enhances data quality, improves model accuracy, reduces computational cost, facilitates better feature extraction, and ensures data compatibility with algorithms.

Three popular data preprocessing techniques in machine learning are: handling missing values (imputation), outlier detection and removal, and feature scaling (normalization or standardization);

which involve filling in missing data, identifying and addressing extreme data points, and scaling features to a consistent range, respectively, to prepare data for machine learning models.

Explanation:

Handling missing values (imputation):

When data is incomplete, missing values are often replaced with a calculated value like the mean, median, or mode depending on the data type, ensuring the model can still process the information without errors.

Outlier detection and removal:

Outliers are data points significantly different from the rest of the data and can negatively impact model performance. Techniques like z-score analysis can identify outliers, which can then be removed or adjusted to bring them closer to the normal range.

Feature scaling (normalization or standardization):

This technique scales features to a similar range, typically between 0 and 1 (normalization) or with a mean of 0 and standard deviation of 1 (standardization), which is crucial for algorithms that are sensitive to feature magnitudes.

c) Describe the different performance metrics to evaluate the algorithms. Explain with numerical examples.[5]
Sol: See Evaluation Part

3. a) What do you mean by Convolutional Neural Network?[3]
Sol: out of our syllabus

b) Why do we prefer Convolutional Neural networks (CNN) over Artificial Neural networks (ANN) for image data as input?[3]
Sol: out of our syllabus

c) Explain the different layers in CNN.[6]

Sol: out of our syllabus

4. a)
What are Support Vectors in SVMs? Why is SVM an example of a large-margin classifier? [2]
Sol: See SVM part

b) What are Hard-Margin and Soft-Margin SVMs? [3]
Sol: See SVM part

c) Use a support vector machine to classify the following dataset:[7]

| Class | X1 | X2 |
|---|---|---|
| + | 1 | 1 |
| + | 2 | 2 |
| + | 2 | 0 |
| - | 0 | 0 |
| - | 1 | 0 |
| - | 0 | 1 |

i) Plot the six training sets on the X1-X2 axis.
ii) Separate the classes with maximum margin separator and give your choice for vector w and intercept b.
Sol:

$S_1 = \begin{pmatrix}1\\1\\1\end{pmatrix}$ $S_2 = \begin{pmatrix}0\\1\\1\end{pmatrix}$ $S_3 = \begin{pmatrix}1\\0\\1\end{pmatrix}$

$S_1 = (1,1)$

$S_2 = (0,1)$

$S_3 = (1,0)$

$$\alpha_1\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = +1$$

$$\alpha_1\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix} + \alpha_2\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix} + \alpha_3\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix} = -1$$

$$\alpha_1\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_3\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = -1 \quad S_1$$

$$\alpha_1(1+1+1) + \alpha_2(0+1+1) + \alpha_3(1+0+1) = +1$$

$$\alpha_1(0+1+1) + \alpha_2(0+1+1) + \alpha_3(0+0+1) = -1$$

$$\alpha_1(1+0+1) + \alpha_2(0+0+1) + \alpha_3(1+0+1) = -1$$

$$3\alpha_1 + 2\alpha_2 + 2\alpha_3 = +1$$
$$2\alpha_1 + 2\alpha_2 + \alpha_3 = -1$$
$$2\alpha_1 + \alpha_2 + 2\alpha_3 = -1$$

$\alpha_1 = 7$

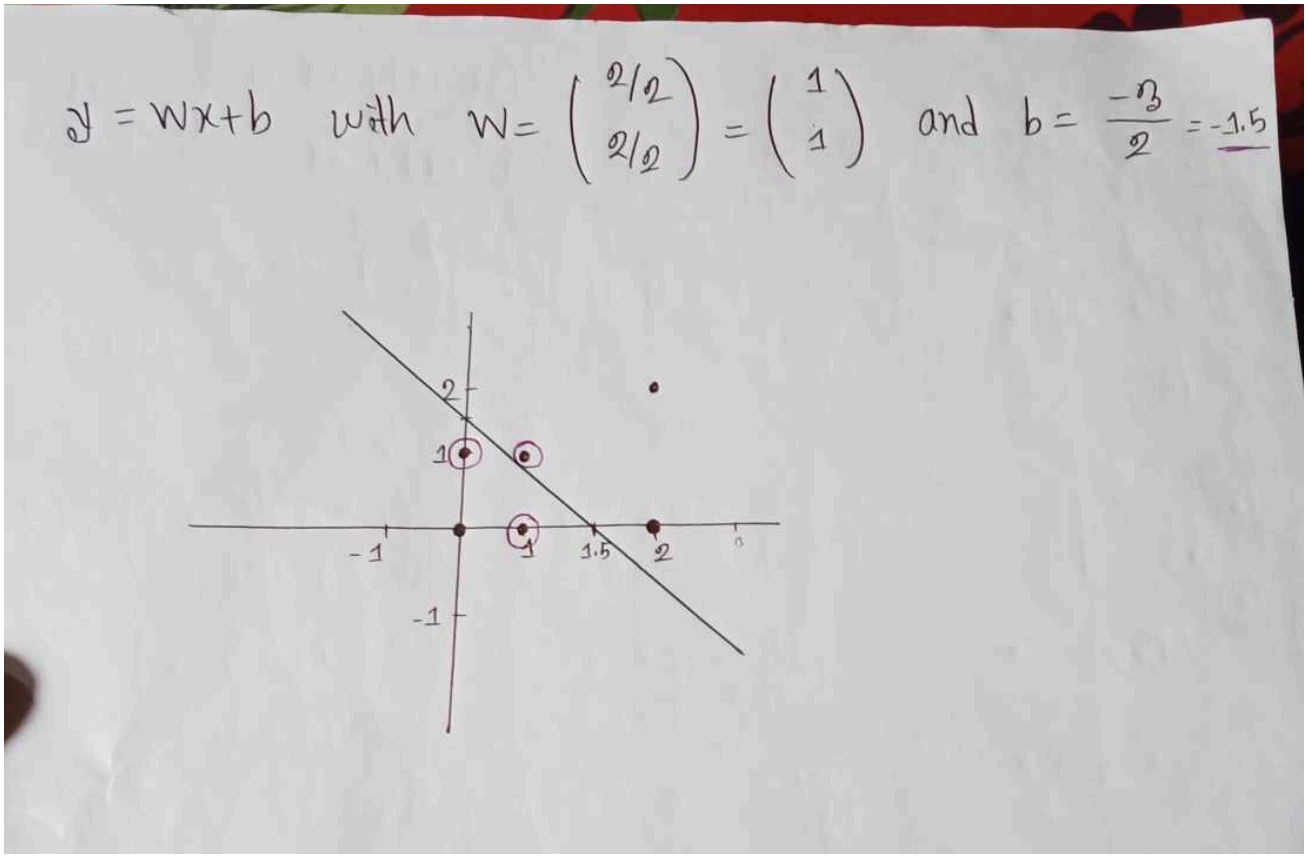
$\alpha_2 = -5$

$\alpha_3 = -5$

$$7\begin{pmatrix}1\\1\\1\end{pmatrix} + (-5)\begin{pmatrix}0\\1\\1\end{pmatrix} + (-5)\begin{pmatrix}1\\0\\1\end{pmatrix}$$

$$= \begin{pmatrix}7\\7\\7\end{pmatrix} + \begin{pmatrix}0\\-5\\-5\end{pmatrix} + \begin{pmatrix}-5\\0\\-5\end{pmatrix} = \begin{pmatrix}2\\2\\-3\end{pmatrix} \quad W = \begin{pmatrix}2\\2\\-3\end{pmatrix}$$

$= 2$




5. a) What is the K-Nearest Neighbor (KNN) Algorithm for Machine Learning? How is KNN different from k-means clustering?[3]

Sol: See K Means Clustering part

b) Discuss maximizing the margin process in Support Vector machine (SVM) with necessary figures and equations.[4]

Sol: See SVM Part

C. For a SunBurn dataset given below, construct a decision tree.

| Name | Hair | Height | Weight | Location | Class |
|---|---|---|---|---|---|
| Amir | Blonde | Average | Light | No | No |
| Babar | Brown | Tall | Average | Yes | Yes |
| Sakib | Blonde | Short | Heavy | Yes | No |
| Mushfik | Red | Short | Average | No | Yes |
| Hadi | Brown | Average | Heavy | No | No |
| Jeleous | Brown | Tall | Light | No | Yes |
| Helal | Blonde | Tall | Heavy | Yes | No |
| Akbar | Red | Short | Heavy | No | No |

Sol:

Sunburn / Total = 8

Step 1: Target attribute - class.

Step 2: Information gain of Target Attribute (class)

No = 5 = P
Yes = 3 = N

$$IG(Class) = \frac{P}{P+N}\log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)$$

$$= \frac{5}{8}\log_2\left(\frac{5}{8}\right) - \frac{3}{8}\log_2\left(\frac{3}{8}\right)$$

$$= 0.9544$$

Step 3:

For Hair,

| Hair | No | Yes |
|---|---|---|
| Blonde | 3 | 0 |
| Brown | 4 | 2 |
| Red | 1 | 1 |

$$I.G.(Blonde) = \frac{-3}{3}\log_2\left(\frac{3}{3}\right) - \frac{0}{3}\log_2\left(\frac{0}{3}\right)$$

Entropy (Blonde) = 0

$$I.G.(Brown) = \left(-\frac{1}{2}\right)\log_2\left(\frac{1}{2}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)$$

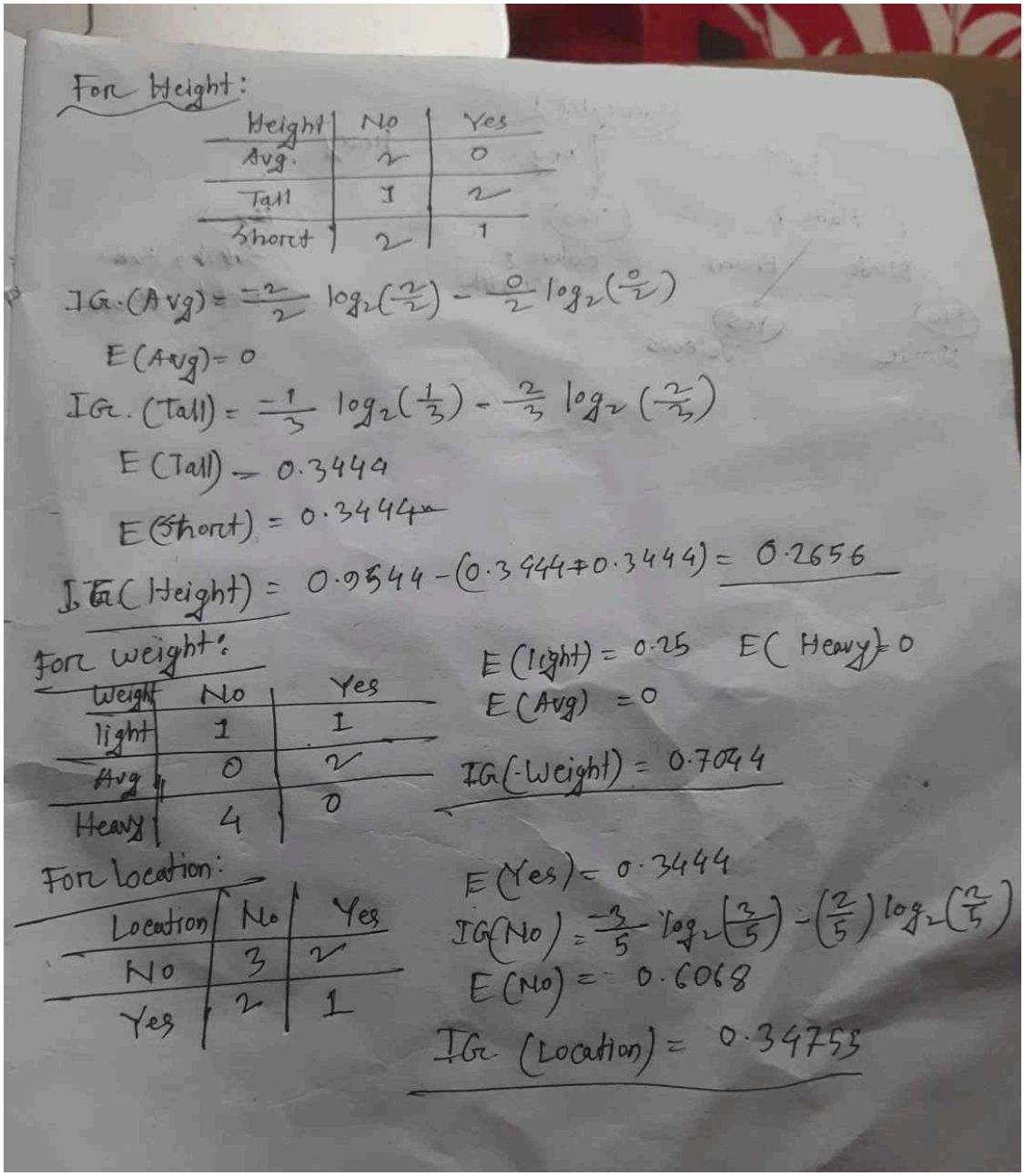
$$E(Brown) = 0.918 \times \frac{3}{8} = 0.3443$$

$$I.G.(Red) = \frac{-1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)$$

$$E(Red) = 1 \times \frac{2}{8} = 0.25$$

$$E(Hair) = (0 + 0.3444 + 0.25) = 0.5944$$

$$I.G.(Hair) = 0.9544 - 0.5944 = 0.36$$



For Height:

| Height | No | Yes |
|---|---|---|
| Avg. | 2 | 0 |
| Tall | 1 | 2 |
| Short | 2 | 1 |

$IG.(Avg) = -\frac{2}{2}\log_2(\frac{2}{2}) - \frac{0}{2}\log_2(\frac{0}{2})$

$E(Avg) = 0$

$IG.(Tall) = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})$

$E(Tall) = 0.3444$

$E(Short) = 0.3444$

$IG(Height) = 0.9544 - (0.3444 + 0.3444) = 0.2656$

For weight:

| Weight | No | Yes |
|---|---|---|
| light | 1 | 1 |
| Avg | 0 | 2 |
| Heavy | 4 | 0 |

$E(light) = 0.25$ $E(Heavy) = 0$

$E(Avg) = 0$

$IG(Weight) = 0.7044$

For location:

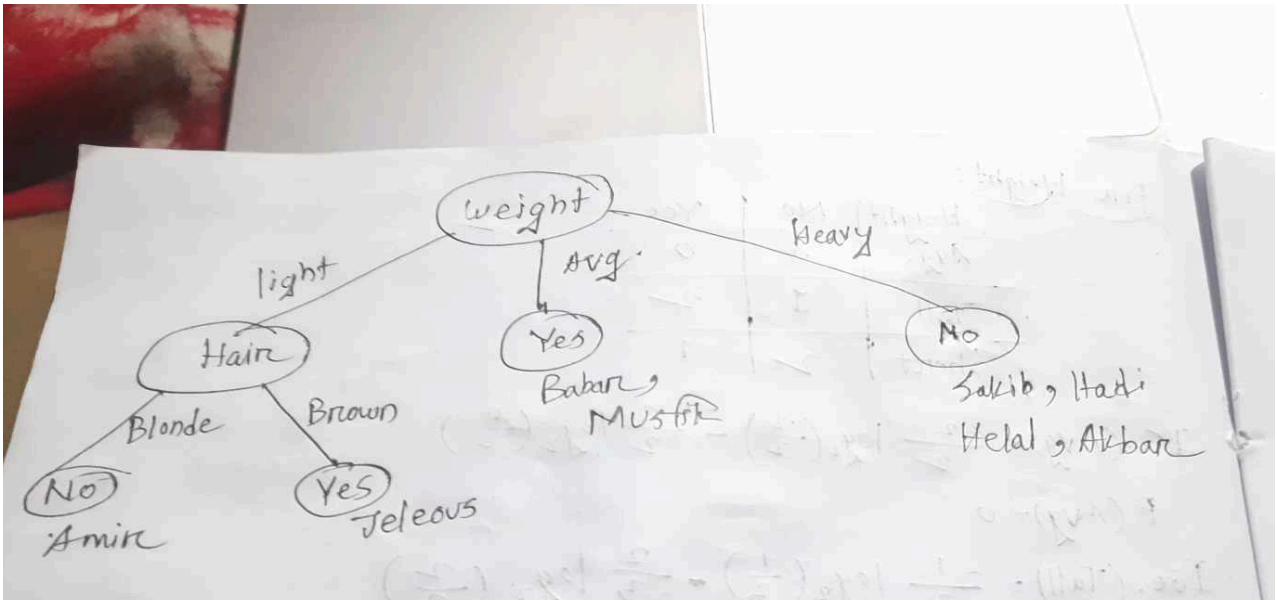
| Location | No | Yes |
|---|---|---|
| No | 3 | 2 |
| Yes | 2 | 1 |

$E(Yes) = 0.3444$

$IG(No) = -\frac{3}{5}\log_2(\frac{3}{5}) - (\frac{2}{5})\log_2(\frac{2}{5})$

$E(No) = 0.6068$

$IG(Location) = 0.34755$




6. a) What is the basic difference of Recurrent Neural Network (RNN) from Artificial Neural Network? Give some examples of RNN application.[4]

Sol: Out of Syllabus

b) What is the vanishing Gradient Problem of RNN?[3]

Sol: Out of Syllabus

c) Shortly describe Long-Term Memory (LSTM) networks and explain how it can avoid long-term dependency problems.[5]

Sol: Out of Syllabus

7. a) What is the difference between Random Forests and Decision Trees?

Sol: See the Decision tree part

b) Consider the below sample data set. In this data set, we have four predictor variables, namely: [6]

*Weight, Blood flow, Blocked Arteries, Chest Pain*

| Blood Flow | Blocked Arteries | Chest Pain | Weight | Heart Disease |
|---|---|---|---|---|
| Abnormal | No | No | 130 | No |
| Normal | Yes | Yes | 195 | Yes |
| Normal | No | Yes | 218 | No |
| Abnormal | Yes | Yes | 180 | Yes |

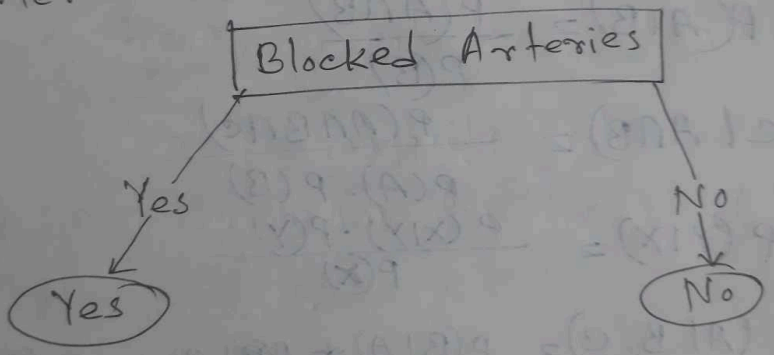
Creating A Random Forest for the above data and check the following query.

| Blood Flow | Blocked Arteries | Chest Pain | Weight | Heart Disease |
|---|---|---|---|---|
| Normal | No | Yes | 218 | No |

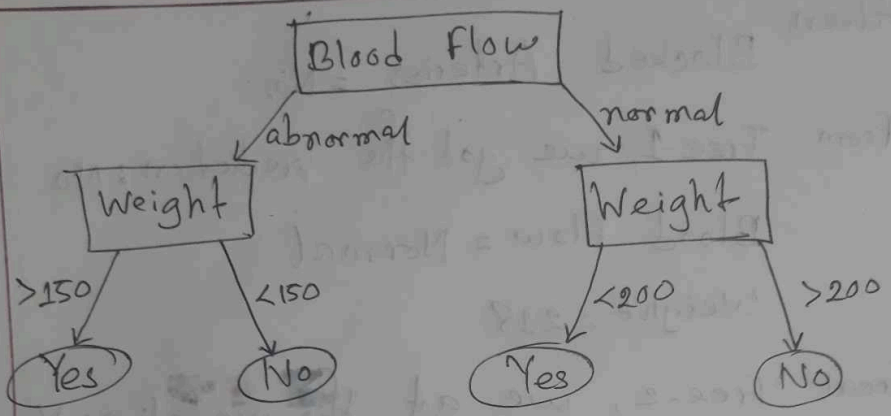
7(b) Random forest involves creating multiple Decision trees and provides the output or decision after merging the performance of all these tree.

Let us constract three DTs by spliting the dataset.

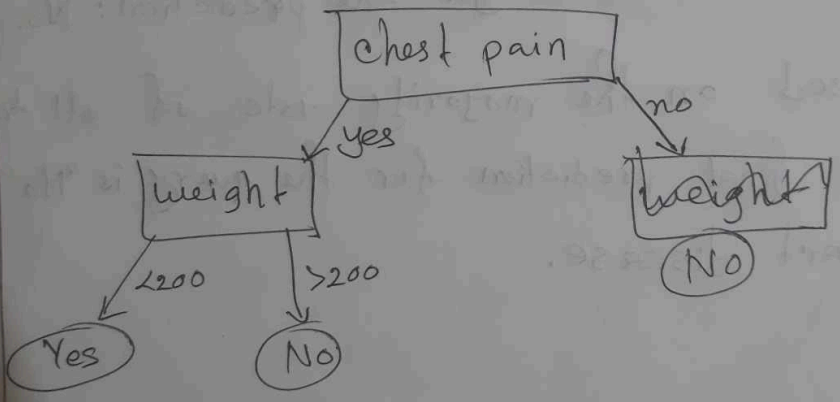
First DT: This tree is based on Blocked Arteries column.



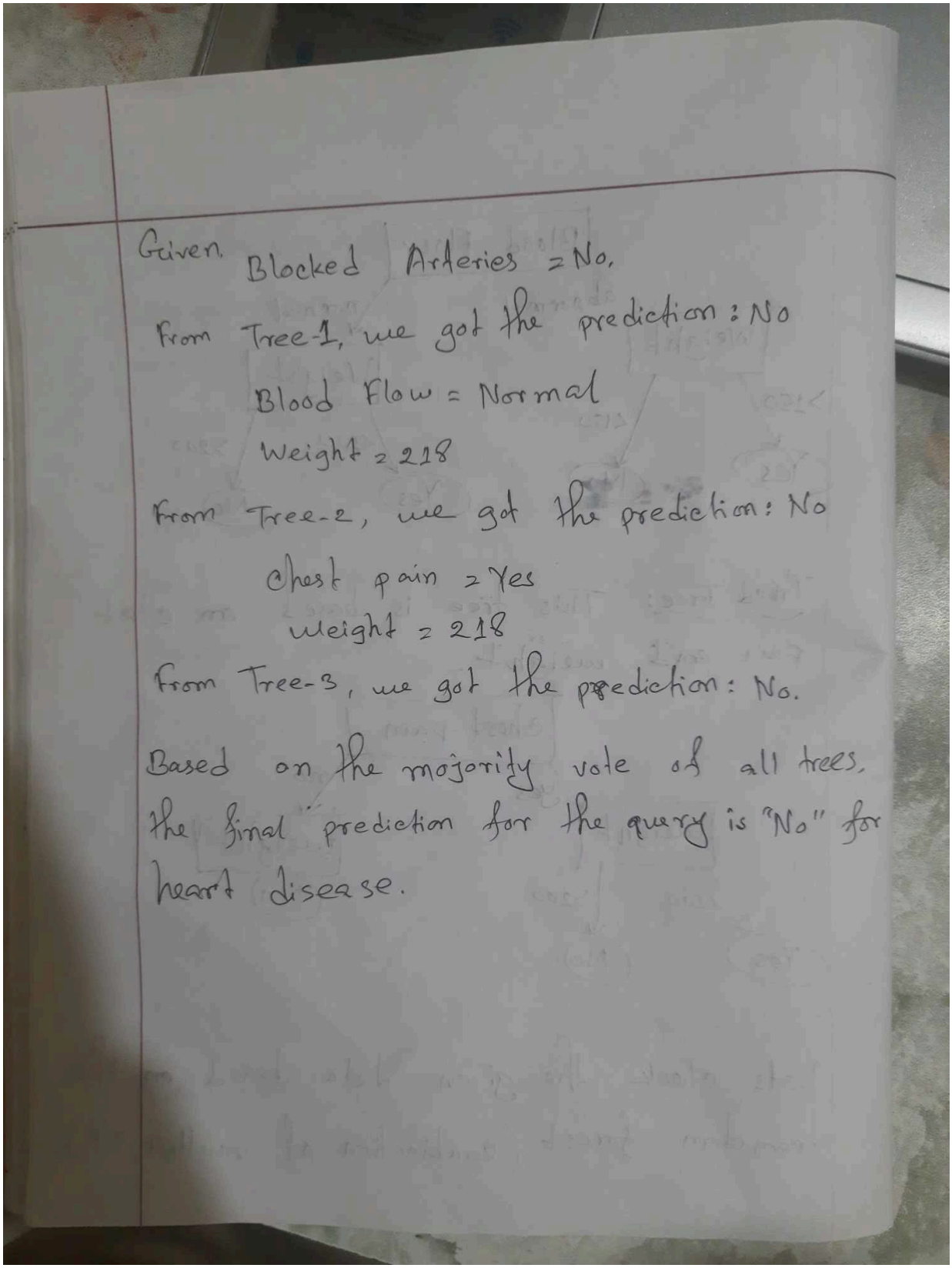

Second DT: This tree is based on Blood Flow and weight.




Third Tree: This tree is based on chest pain and weight.




Lets check the given data based on this random forest (combination of multiple DTs).

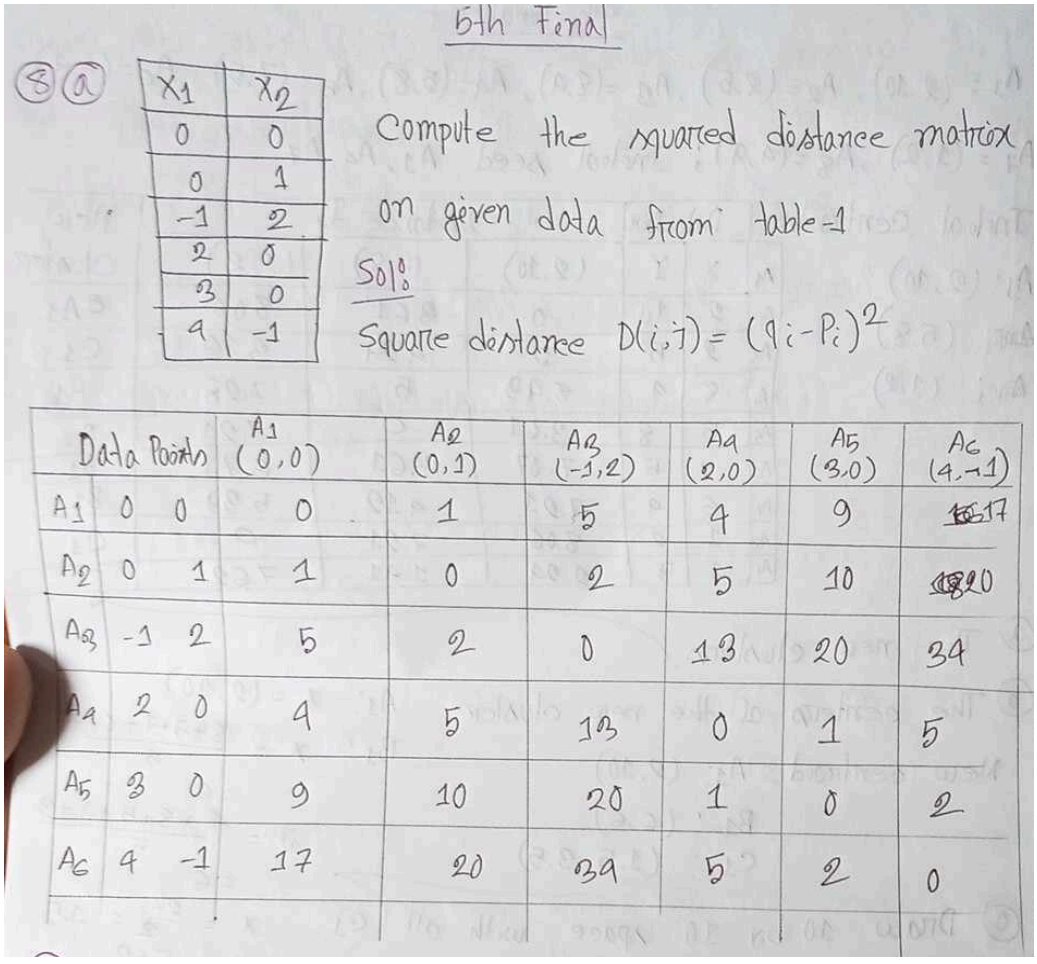



8. a) Compute the squared distance matrix on given the data from Table-1

| | $X_1$ | $X_2$ |
|---|---|---|
| A | 0 | 0 |
| B | 0 | 1 |
| C | -1 | 2 |
| D | 2 | 0 |
| E | 3 | 0 |
| F | 4 | -1 |

Table-1

Sol:



5th Final

(3) (a)

| $X_1$ | $X_2$ |
|---|---|
| 0 | 0 |
| 0 | 1 |
| -1 | 2 |
| 2 | 0 |
| 3 | 0 |
| 4 | -1 |

Compute the squared distance matrix on given data from table-1

Sol:

Square distance $D(i,j) = (p_i - p_j)^2$

| Data Points | | $A_1$ (0,0) | $A_2$ (0,1) | $A_3$ (-1,2) | $A_4$ (2,0) | $A_5$ (3,0) | $A_6$ (4,-1) |
|---|---|---|---|---|---|---|---|
| $A_1$ | 0 0 | 0 | 1 | 5 | 4 | 9 | 17 |
| $A_2$ | 0 1 | 1 | 0 | 2 | 5 | 10 | 20 |
| $A_3$ | -1 2 | 5 | 2 | 0 | 13 | 20 | 34 |
| $A_4$ | 2 0 | 4 | 5 | 13 | 0 | 1 | 5 |
| $A_5$ | 3 0 | 9 | 10 | 20 | 1 | 0 | 2 |
| $A_6$ | 4 -1 | 17 | 20 | 34 | 5 | 2 | 0 |

b) Perform K-means clustering on the dataset from Table 1. Use the first and last data points as initial centers (K = 2). Given the final parameters, which cluster would belong to x(1,1)

Sol:

Initial centroid: $c_1 = (0,0)$, $c_2 = (4,-1)$

Iteration 1:

| Data Point | | Distance (0,0) | Distance (4,-1) | cluster |
|---|---|---|---|---|
| 0 | 0 | 0 | 4.12 | c1 |
| 0 | 1 | 1 | 4.47 | c1 |
| -1 | 2 | 2.23 | 5.83 | c1 |
| 2 | 0 | 2 | 2.23 | c1 |
| 3 | 0 | 3 | 1.41 | c2 |
| 4 | -1 | 4.12 | 0 | c2 |

New centroid:

$c_1: \left(\frac{0+0-1+2}{4}, \frac{0+1+2+0}{4}\right)$

$= (0.25, 0.75)$

$c_2: \left(\frac{3+4}{2}, \frac{0-1}{2}\right)$

$= (3.5, -0.5)$

Iteration :2

| Data Point | | Distance (0.25, 0.75) | Distance (3.5, -0.5) | Clu |
|---|---|---|---|---|
| 0 | 0 | 0.79 | 3.53 | c1 |
| 0 | 1 | 0.35 | 3.80 | c1 |
| -1 | 2 | 1.76 | 5.14 | c1 |
| 2 | 0 | 1.90 | 1.58 | c2 |
| 3 | 0 | 2.85 | 0.707 | c2 |
| 4 | -1 | 4.13 | 0.707 | c2 |

New centroid:

$c_1: \left(\frac{0+0-1}{3}, \frac{0+1+2}{3}\right)$

$= (0.33, 1)$

$c_2: \left(\frac{2+3+4}{3}, \frac{0+0-1}{3}\right)$

$= (3, -0.33)$

এইভাবে iteration করতে যতক্ষণ না previous cluster আর সমান থাকে, ততক্ষণ চলবে। Final centroid, $c_1 = (0.25, 0.75)$

$c_2 = (3.5, -0.5)$

Iteration: 3

| | | (0.33, 1) | (3, -.33) | cluster |
|---|---|---|---|---|
| 0 | 0 | 1.05 | 3.01 | c1 |
| 0 | 1 | 0.33 | 3.28 | c1 |
| -1 | 2 | 1.20 | 4.62 | c1 |
| 2 | 0 | 2.53 | 1.05 | c2 |
| 3 | 0 | 3.47 | 0.33 | c2 |
| 4 | -1 | 4.76 | 1.20 | c2 |

Now, There is change in cluster after the third iteration.

So Final centroid,

$c_1: (-0.33, 1)$

$c_2: (3, -0.33)$