- **Least Squares Methods**

Machine Learning

Dr. Jagendra Singh

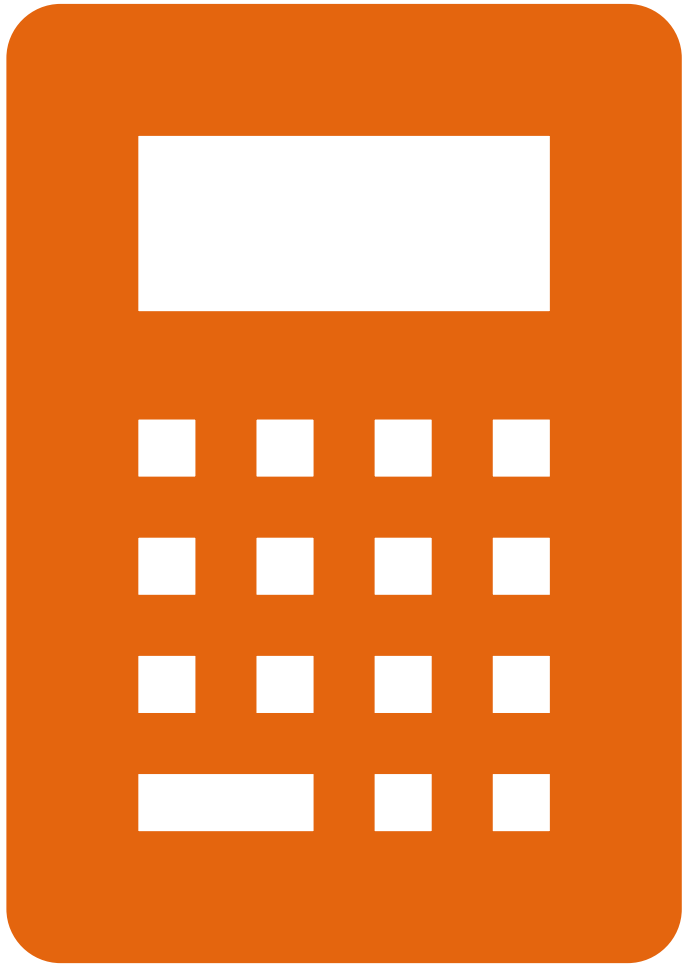
LEAST SQUARES REGRESSION METHOD

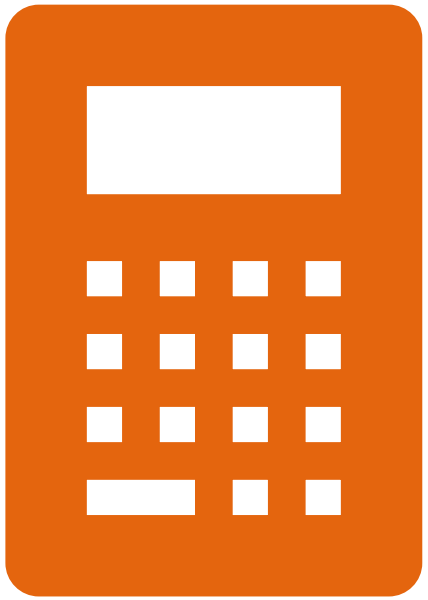
list of topics that will be covered in this session:

1. What Is the Least Squares Method?
2. Line Of Best Fit
3. Steps to Compute the Line Of Best Fit
4. The least-squares regression method with an example
5. A python program to implement Least Squares method

WHAT IS THE LEAST SQUARES METHOD

- The least-squares regression method is a technique commonly used in Regression Analysis.
- It is a mathematical method used to find the best fit line that represents the relationship between an independent and dependent variable.

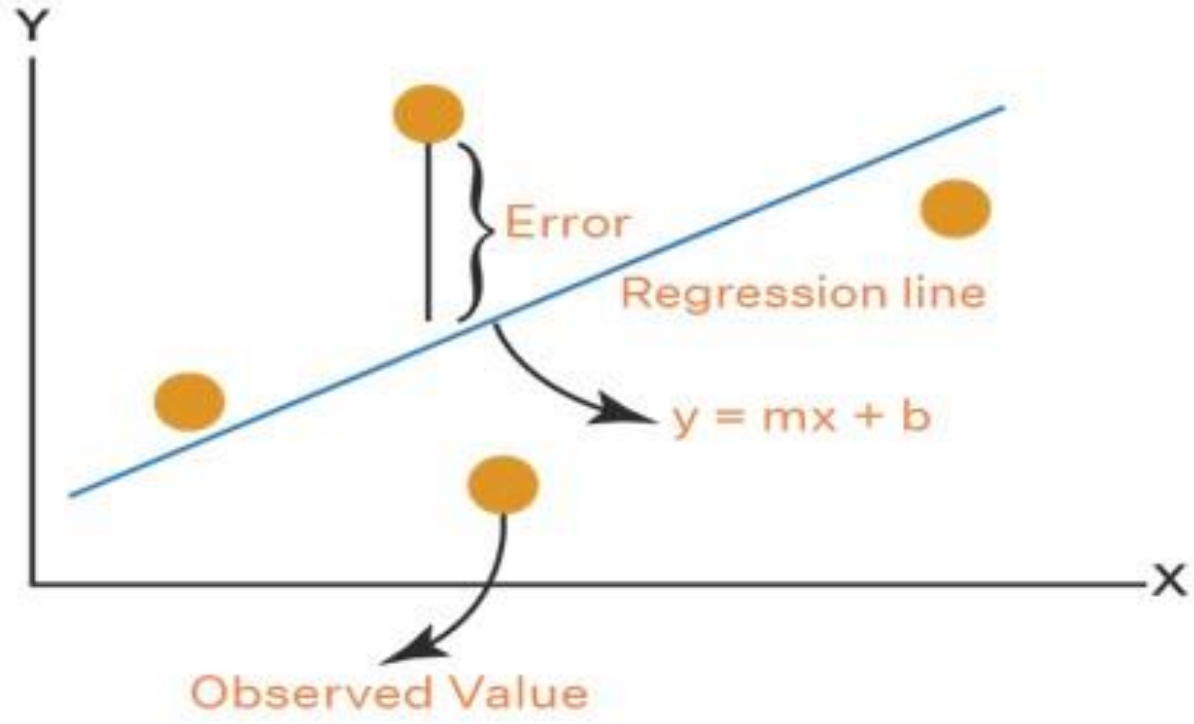




WHAT IS THE LEAST SQUARES METHOD

- The least-squares method is a statistical method used to find the line of best fit of the form of an equation such as y = mx + b to the given data.
- The curve of the equation is called the regression line. Our main objective in this method is to reduce the sum of the squares of errors as much as possible.
- This is the reason this method is called the least-squares method.
- This method is often used in data fitting where the best fit result is assumed to reduce the sum of squared errors that is considered to be the difference between the observed values and corresponding fitted value.
- The sum of squared errors helps in finding the variation in observed data. For example, we have 4 data points and using this method we arrive at the following graph.

LEAST-SQUARE METHOD GRAPH



WHAT IS THE LINE OF BEST FIT?

Line of best fit is drawn to represent the relationship between 2 or more variables.

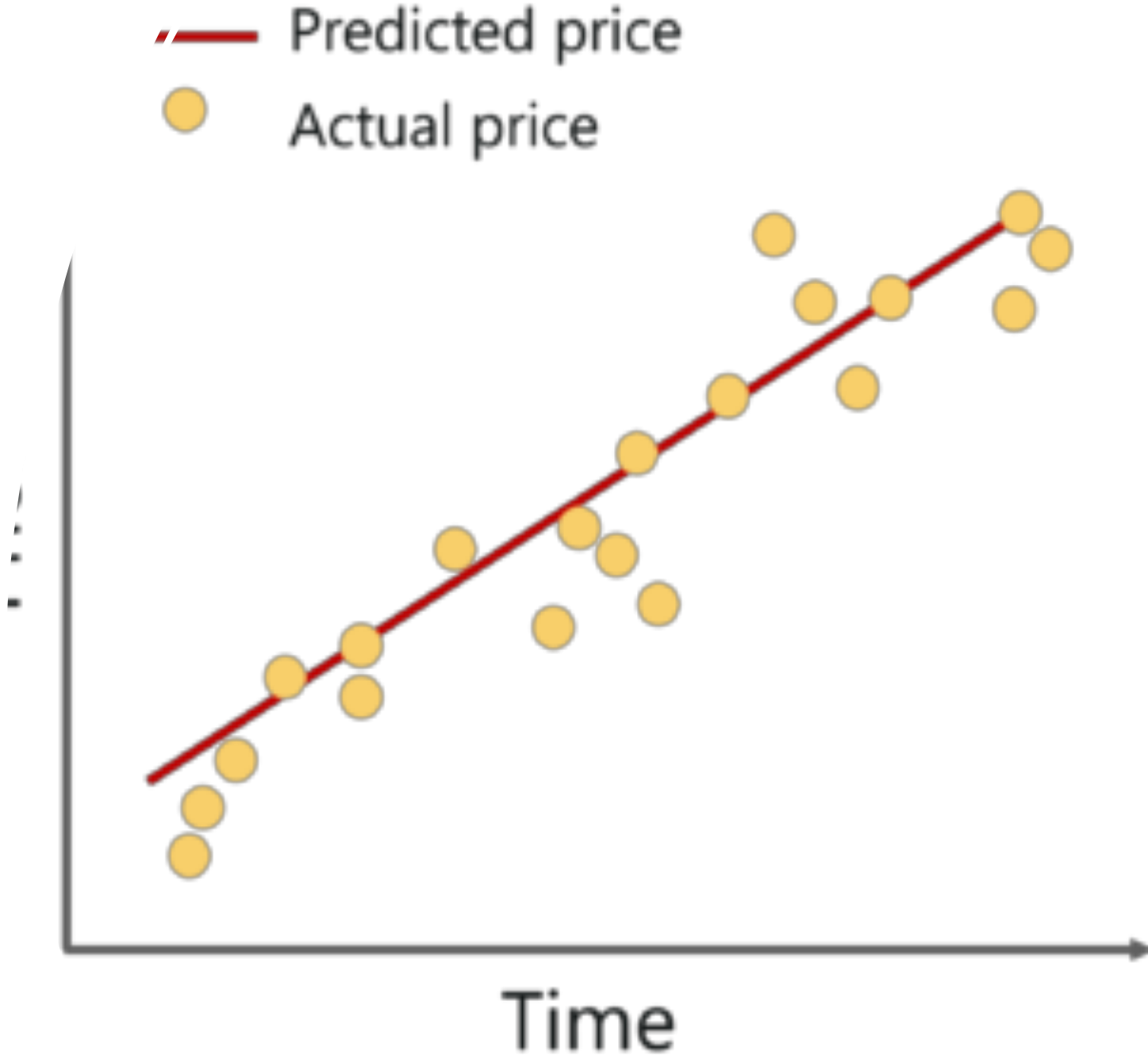
To be more specific, the best fit line is drawn across a scatter plot of data points in order to represent a relationship between those data points.

The least-squares method is one of the most effective ways used to draw the line of best fit. It is based on the idea that the square of the errors obtained must be minimized to the most possible extent and hence the name least squares method.

If we were to plot the best fit line that shows the depicts the sales of a company over a period of time, it would look something like this:

LINE OF BEST FIT

- Notice that the line is as close as possible to all the scattered data points. This is what an ideal best fit line looks like.
- To better understand the whole process let's see how to calculate the line using the Least Squares Regression.



LEAST SQUARE METHOD FORMULA

Least-square method is the curve that best fits a set of observations with a minimum sum of squared residuals or errors.

Let us assume that the given points of data are (x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n) in which all x's are independent variables, while all y's are dependent ones.

This method is used to find a linear line of the form $y = mx + c$, where y and x are variables, m is the slope, and c is the y-intercept.

The formula to calculate slope m and the value of c is given by:

STEPS TO CALCULATE THE LINE OF BEST FIT

To start constructing the line that best depicts the relationship between variables in the data, we first need to get our basics right. Take a look at the equation below:

It is a simple equation that represents a straight line along 2 Dimensional data, i.e. x-axis and y-axis. To better understand this, let's break down the equation: $y = mx + c$

y: dependent variable

m: the slope of the line

x: independent variable

c: y-intercept

STEPS TO CALCULATE THE LINE OF BEST FIT

- As an assumption, let's consider that there are 'n' data points.
- **Step 1:** *Calculate the slope 'm' by using the following formula:*

$$m = \frac{n\sum xy - (\Sigma x)(\Sigma y)}{n\Sigma x^2 - (\Sigma x)^2}$$

- **Step 2:** *Compute the y-intercept (the value of y at the point where the line crosses the y-axis):*

$$c = y - mx$$

- **Step 3:** *Substitute the values in the final equation:*

$$y = mx + c$$

- Now let's look at an example and see how you can use the least-squares regression method to compute the line of best fit.

LEAST SQUARES REGRESSION EXAMPLE

Consider an example. Tom who is the owner of a retail shop, found the price of different T-shirts vs the number of T-shirts sold at his shop over a period of one week.

He tabulated this like shown below:

Price of T-shirts in dollars (x)	# of T-shirts sold (y)
2	4
3	5
5	7
7	10
9	15

Let us use the concept of least squares regression to find the line of best fit for the above data.

LEAST SQUARES CALCULATION

- **Step 1:** Calculate the slope ‘m’ by using the following formula:

$$m = \frac{n\sum xy - (\Sigma x)(\Sigma y)}{n\Sigma x^2 - (\Sigma x)^2}$$

- After you substitute the respective values, m = 1.518 approximately.
- **Step 2:** Compute the y-intercept value

$$c = y - mx$$

- After you substitute the respective values, c = 0.305 approximately.
- **Step 3:** Substitute the values in the final equation

$$y = mx + c$$

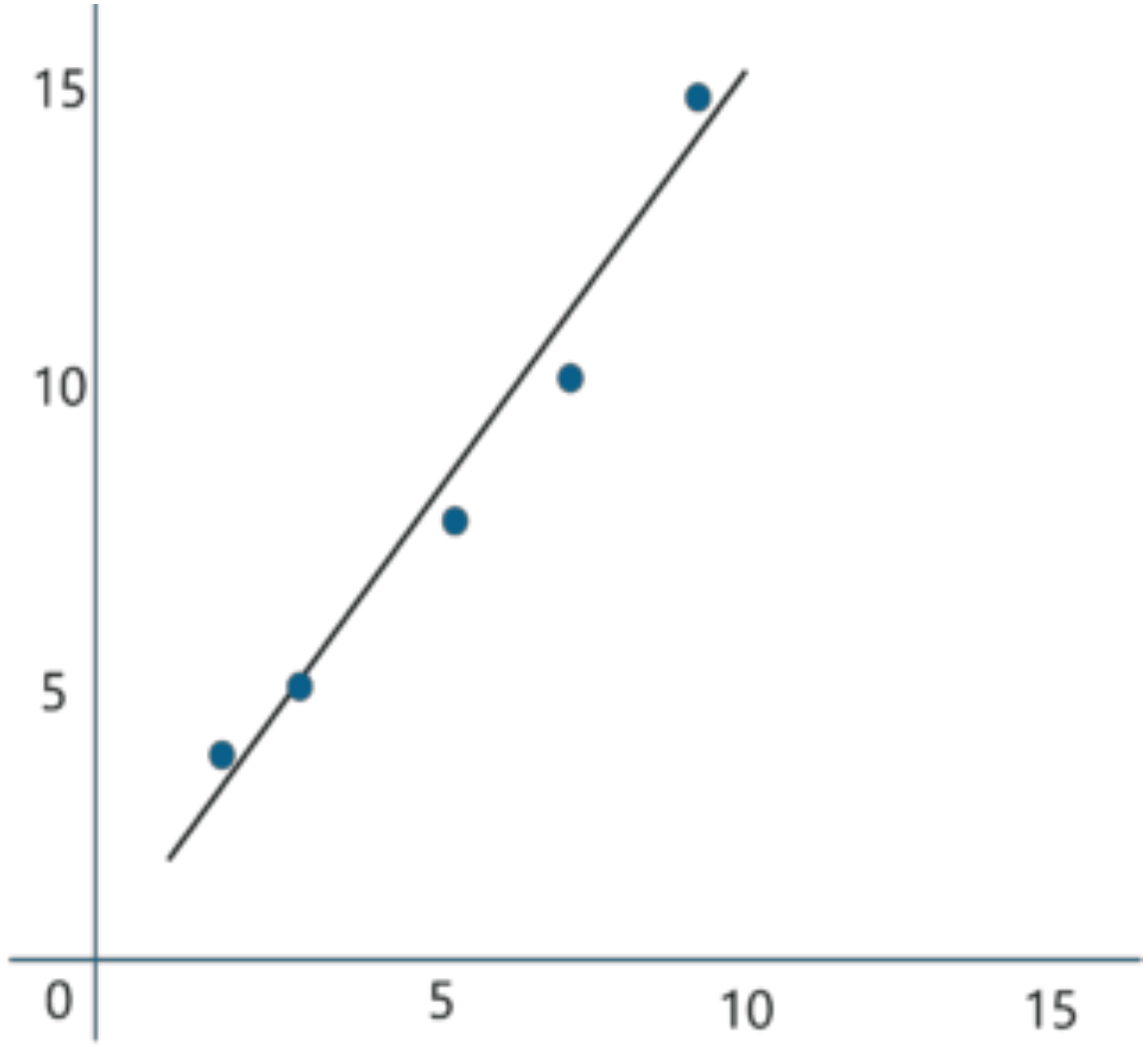
LEAST SQUARES CALCULATION

- Once you substitute the values, it should look something like this:

Price of T-shirts in dollars (x)	# of T-shirts sold (y)	Y=mx+c	error
2	4	3.3	-0.67
3	5	4.9	-0.14
5	7	7.9	0.89
7	10	10.9	0.93
9	15	13.9	-1.03

LEAST SQUARES GRAPH

- Let's construct a graph that represents the **y=mx + c** line of best fit:
- Now Tom can use the above equation to estimate how many T-shirts of price $8 can he sell at the retail shop.
- y = 1.518 x 8 + 0.305 = 12.45 T-shirts
- This comes down to 13 T-shirts! That's how simple it is to make predictions using Linear Regression.



LEAST SQUARES METHOD IN PYTHON

Problem Statement: To apply Linear square method and build a model that studies the relationship between the head size and the brain weight of an individual.

Data Set Description: The data set contains the following variables:

- Gender: Male or female represented as binary variables
- Age: Age of an individual
- Head size in cm^3: An individuals head size in cm^3
- Brain weight in grams: The weight of an individual's brain measured in grams

These variables need to be analyzed in order to build a model that studies the relationship between the head size and brain weight of an individual.

IMPLEMENTATION

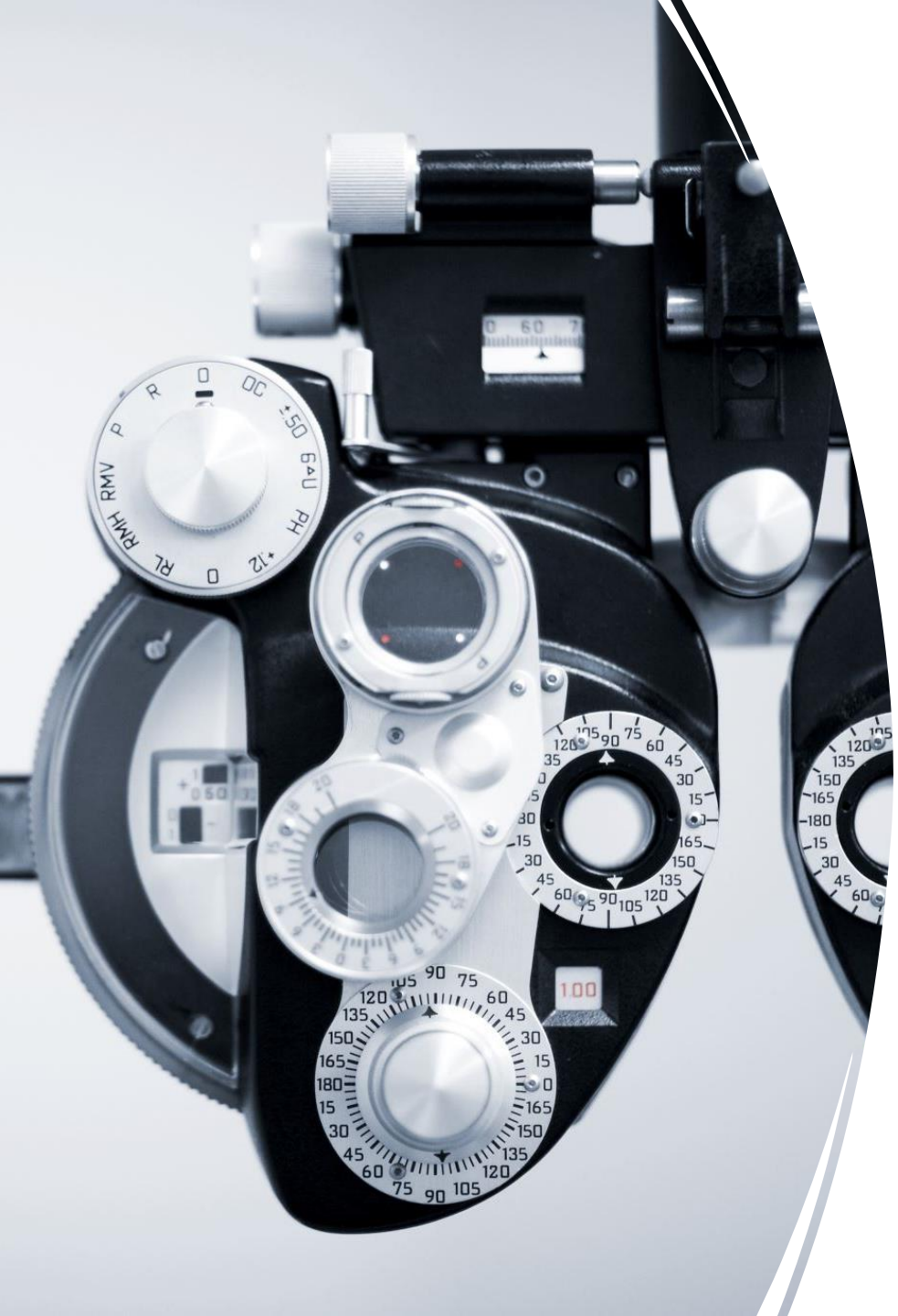
- **Logic:** To implement Least Squares method in order to build a model that studies the relationship between an independent and dependent variable.
- The model will be evaluated by using least square regression method where RMSE and R-squared will be the model evaluation parameters.
- **Step 1: Import the required libraries**
- **Step 2: Import the data set**
- **Step 3: Assigning ‘X’ as independent variable and ‘Y’ as dependent variable**
- **Step 4: Calculate the values of the slope and y-intercept**
- **Step 5: Plotting the line of best fit**
- **Step 6: Model Evaluation**

THANK YOU

- **Bias and Variance**

Machine Learning

Dr. Jagendra Singh

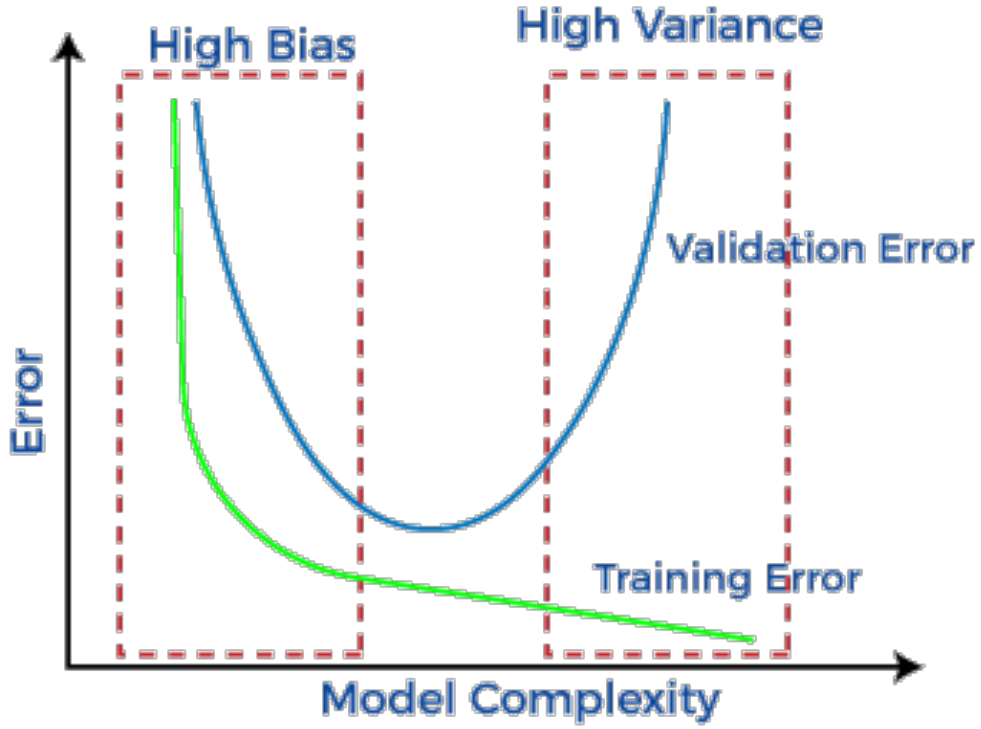


BIAS AND VARIANCE

- Machine learning perform data analysis and make predictions. However, if the machine learning model is not accurate, it can make predictions errors, and these prediction errors are usually known as Bias and Variance.
- In machine learning, these errors will always be present as there is always a slight difference between the model predictions and actual predictions.

BIAS AND VARIANCE

- The main aim of ML/data science analysts is to reduce these errors in order to get more accurate results.
- In this topic, we are going to discuss bias and variance, Bias-variance trade-off, Underfitting and Overfitting. But before starting, let's first understand what errors in Machine learning are?

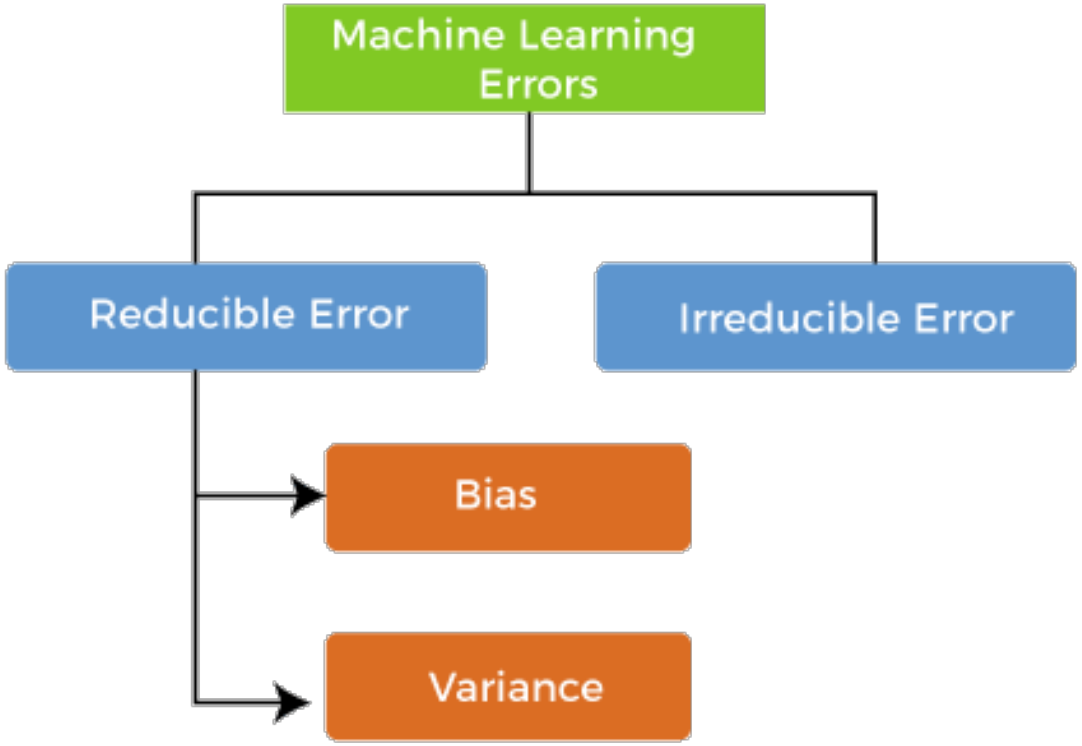


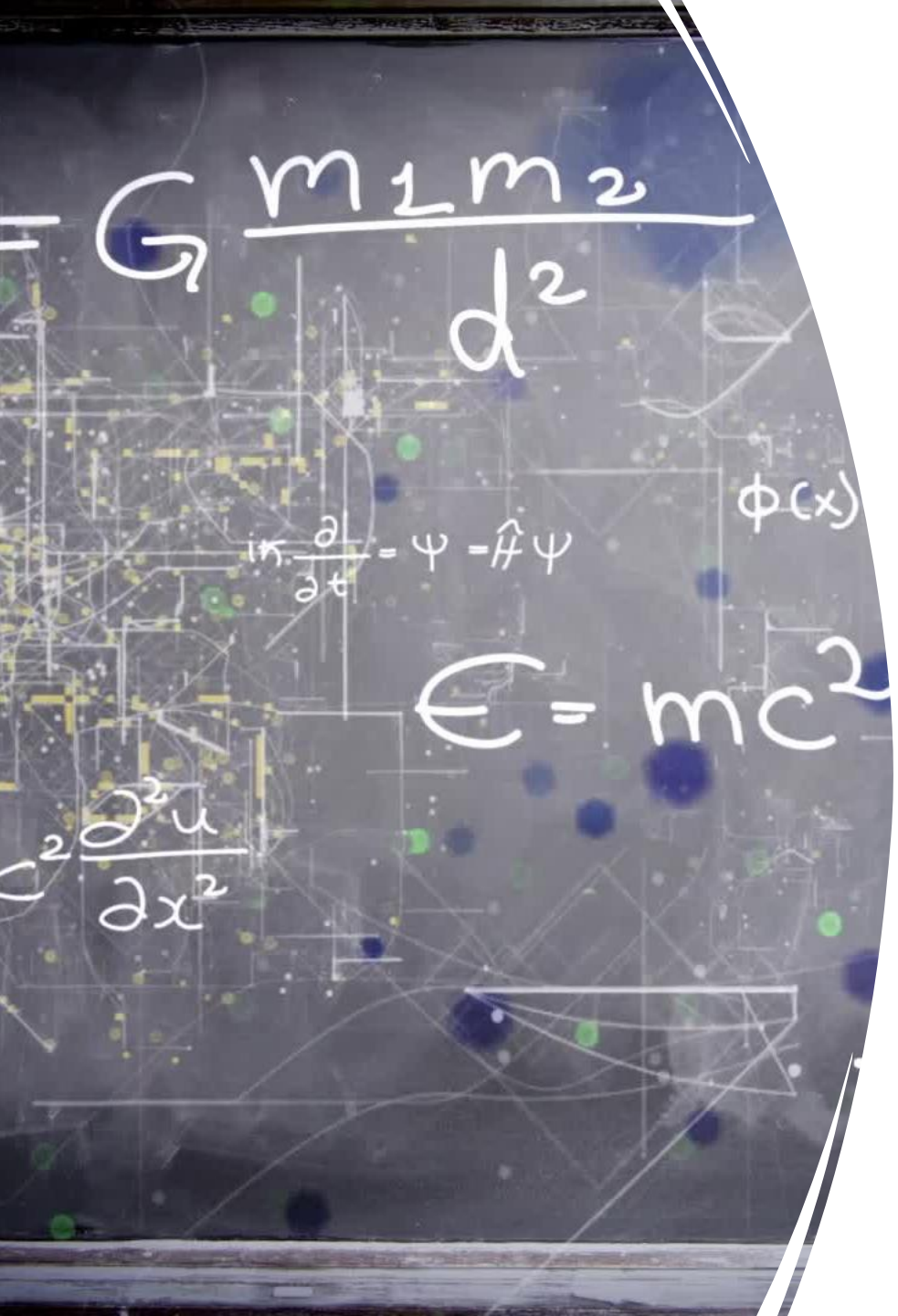
ERRORS IN MACHINE LEARNING

- In machine learning, an error is a measure of how accurately an algorithm can make predictions for the previously unknown dataset.
- On the basis of these errors, the machine learning model is selected that can perform best on the particular dataset. There are mainly two types of errors in machine learning, which are:

REDUCIBLE ERRORS

- These errors can be reduced to improve the model accuracy. Such errors can further be classified into bias and Variance.





IRREDUCIBLE ERRORS

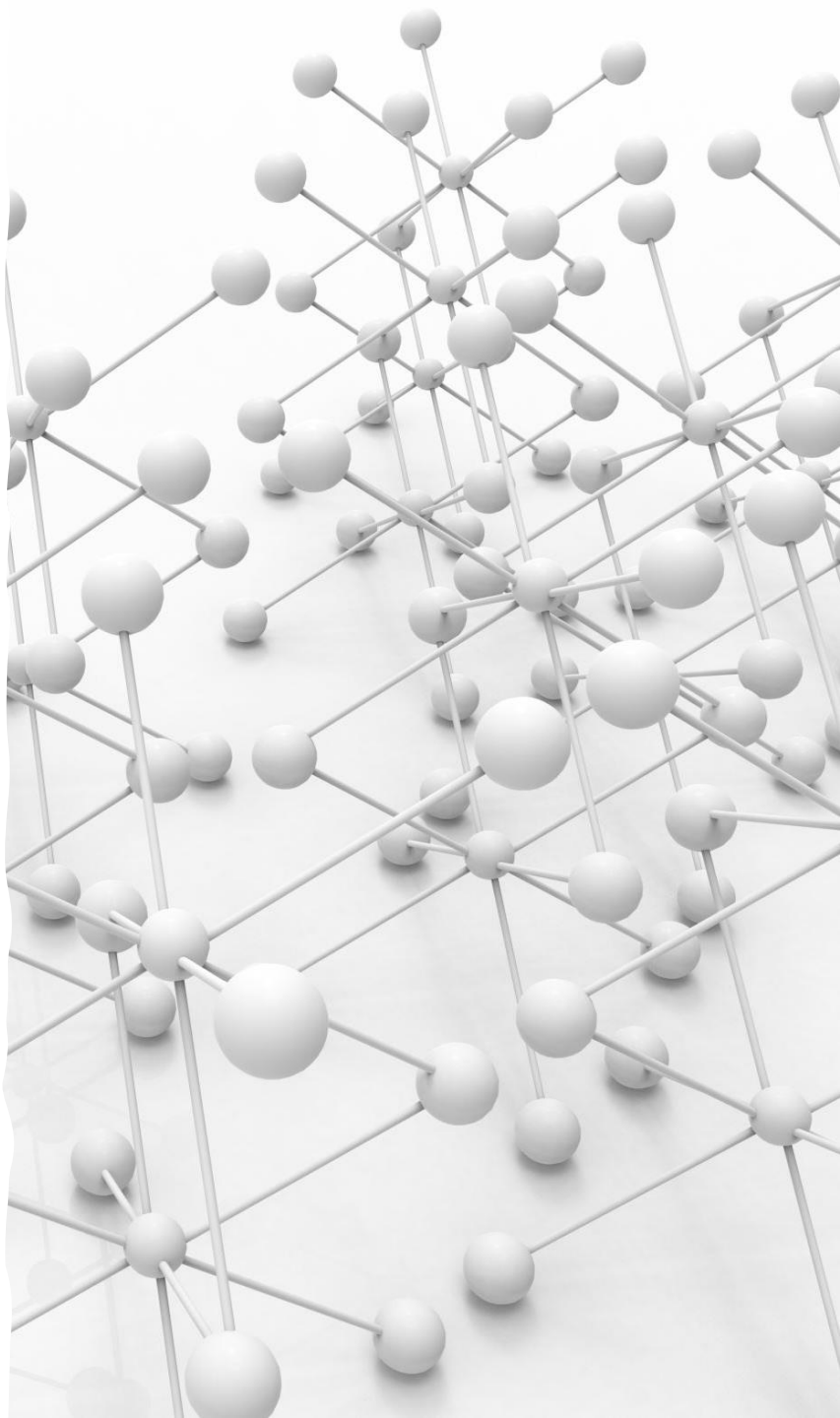
- These errors will always be present in the model regardless of which algorithm has been used. The cause of these errors is unknown variables whose value can't be reduced.

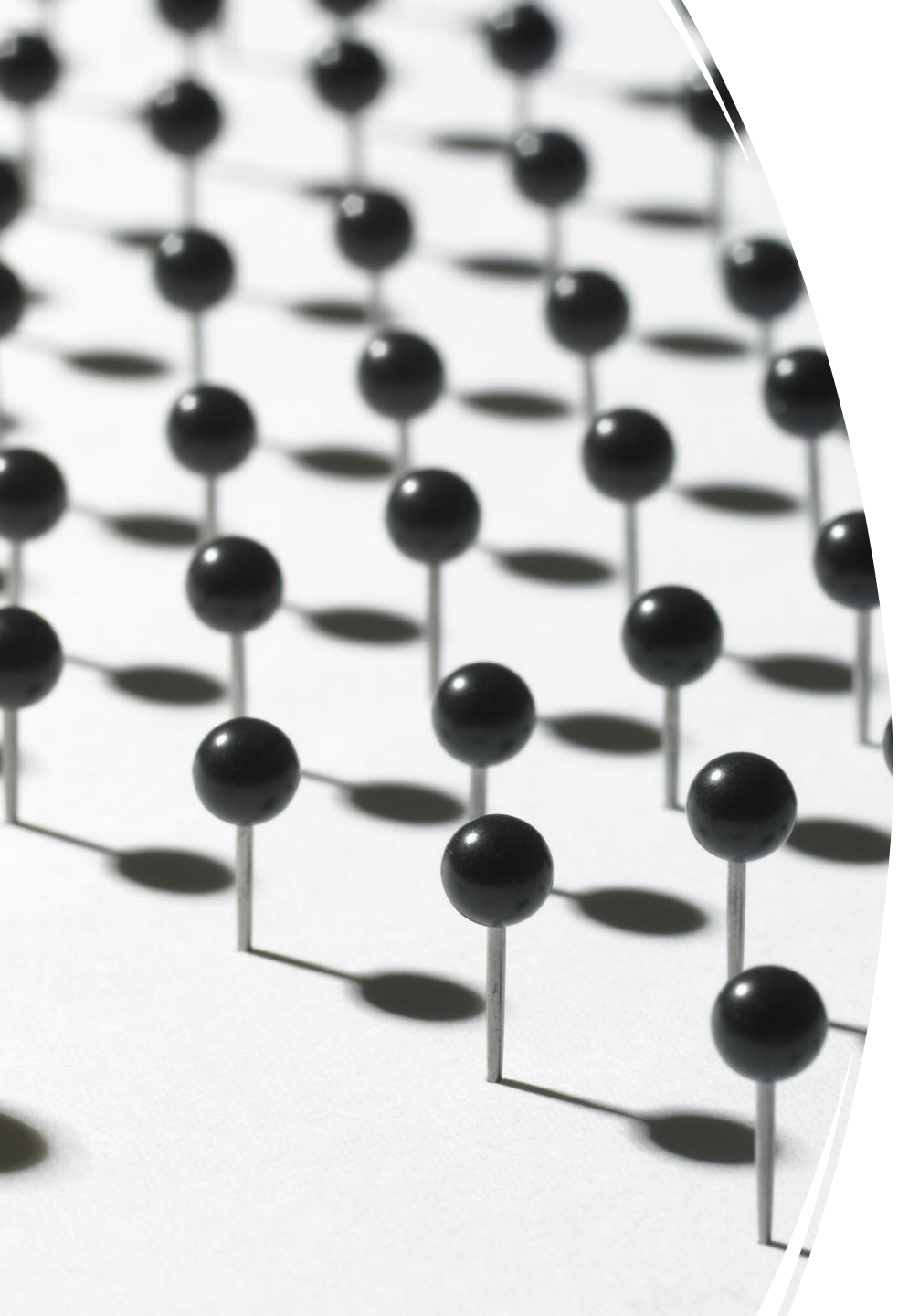
WHAT IS BIAS

- In general, a machine learning model analyses the data, find patterns in it and make predictions. While training, the model learns these patterns in the dataset and applies them to test data for prediction.
- ***While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias***.

BIAS

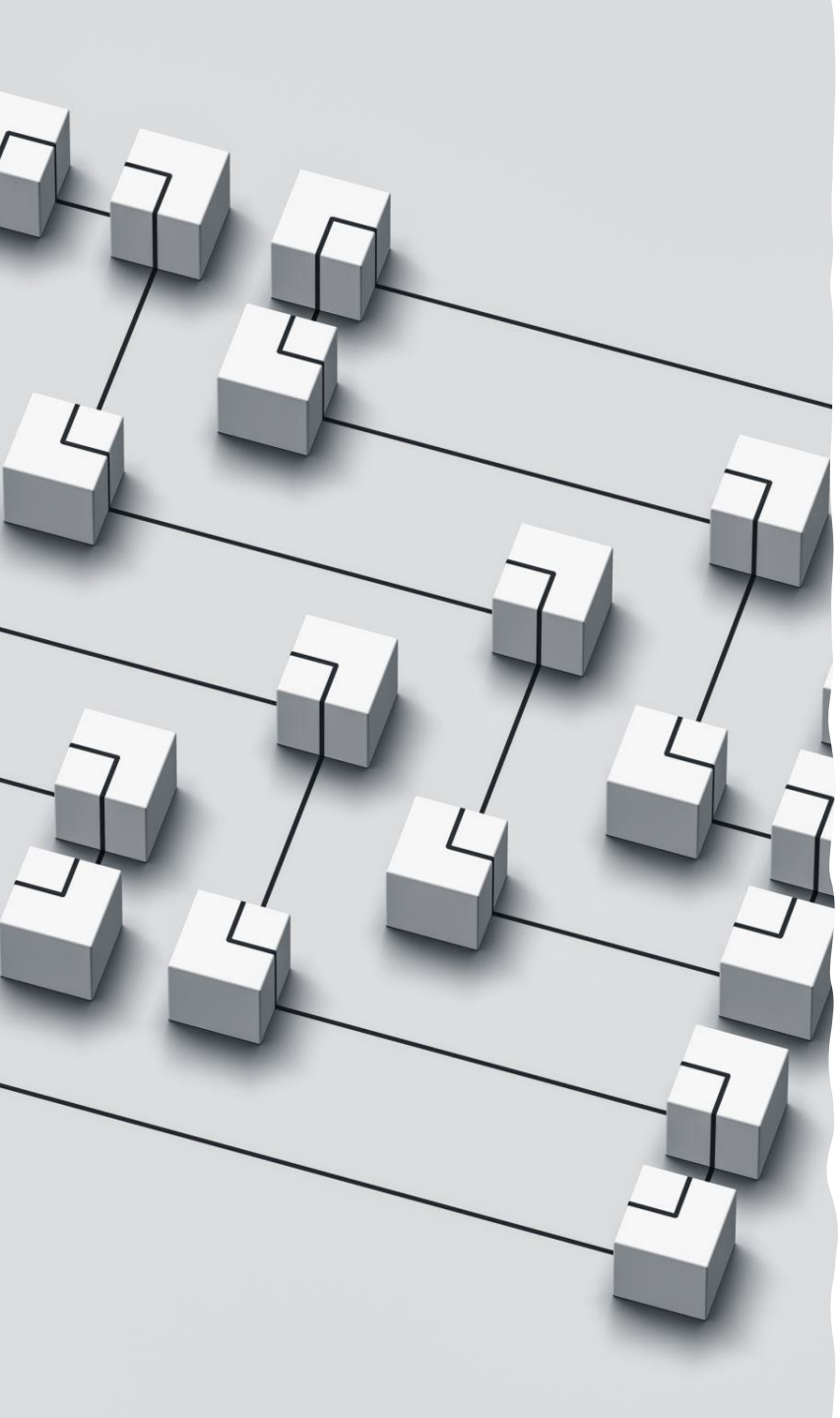
- It can be defined as an inability of machine learning algorithms such as Linear Regression to capture the true relationship between the data points.
- Each algorithm begins with some amount of bias because bias occurs from assumptions in the model, which makes the target function simple to learn. A model has either:





BIAS TYPE

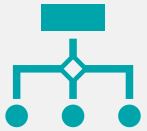
- **Low Bias:** A low bias model will make fewer assumptions about the form of the target function.
- **High Bias:** A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset. **A high bias model also cannot perform well on new data.**



BIAS

- Generally, a linear algorithm has a high bias, as it makes them learn fast. The simpler the algorithm, the higher the bias it has likely to be introduced. Whereas a nonlinear algorithm often has low bias.
- Some examples of machine learning algorithms with low bias **are Decision Trees, k-Nearest Neighbours and Support Vector Machines**.
- At the same time, an algorithm with high bias is **Linear Regression, Linear Discriminant Analysis and Logistic Regression.**

WAYS TO REDUCE HIGH BIAS



Increase the input features as the model is underfitted.

Decrease the regularization term.

Use more complex models, such as including some polynomial features.

VARIANCE ERROR

The variance would specify the amount of variation in the prediction if the different training data was used.

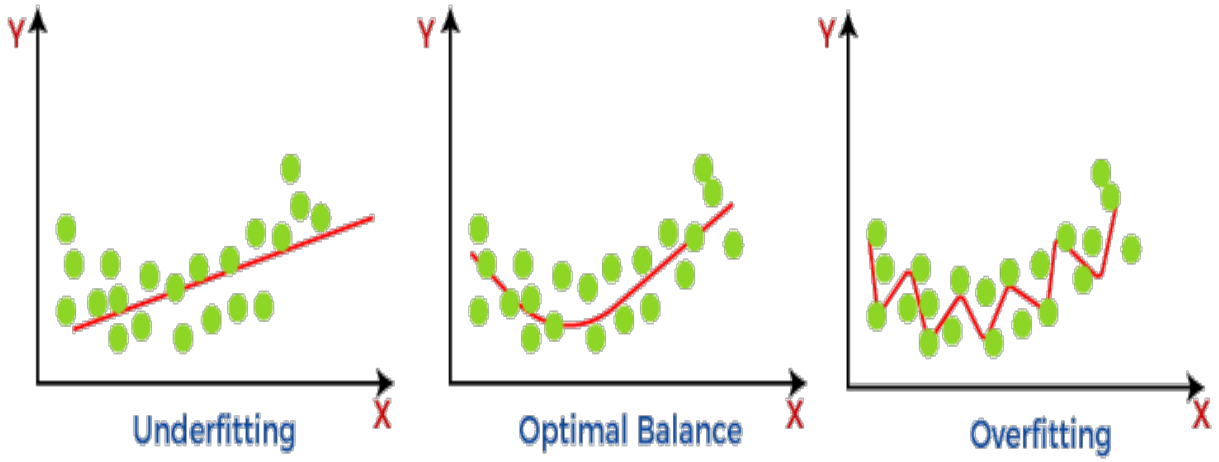
In simple words, variance tells that how much a random variable is different from its expected value.

Ideally, a model should not vary too much from one training dataset to another, which means the algorithm should be good in understanding the hidden mapping between inputs and output variables. Variance errors are either of low variance or high variance.

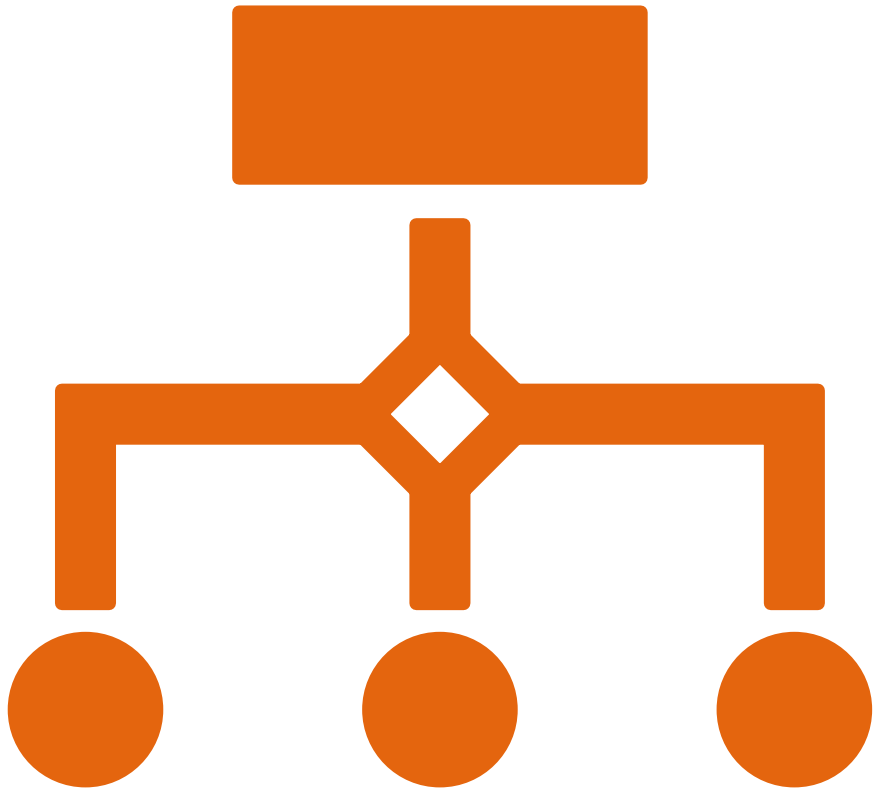
VARIANCE

- **Low variance** means there is a small variation in the prediction of the target function with changes in the training data set.
- At the same time, **High variance** shows a large variation in the prediction of the target function with changes in the training dataset.
- Since, with high variance, the model learns too much from the dataset, it leads to overfitting of the model. A model with high variance has the below problems:

VARIANCE

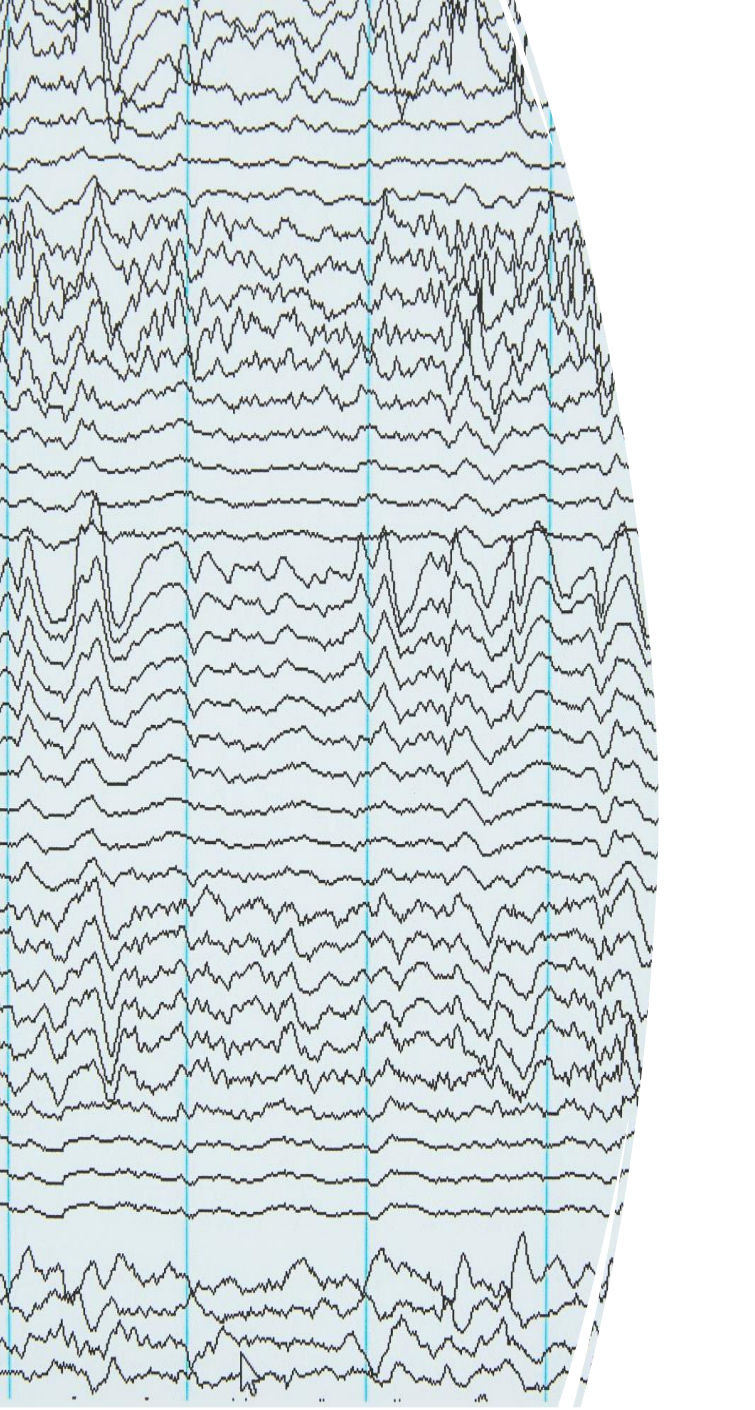


- Some examples of machine learning algorithms with low variance are, **Linear Regression, Logistic Regression, and Linear discriminant analysis**.
- At the same time, algorithms with high variance are **decision tree, Support Vector Machine, and K-nearest neighbours.**



WAYS TO REDUCE HIGH VARIANCE

- Reduce the input features or number of parameters as a model is overfitted.
- Do not use a much complex model.
- Increase the training data.
- Increase the Regularization term.

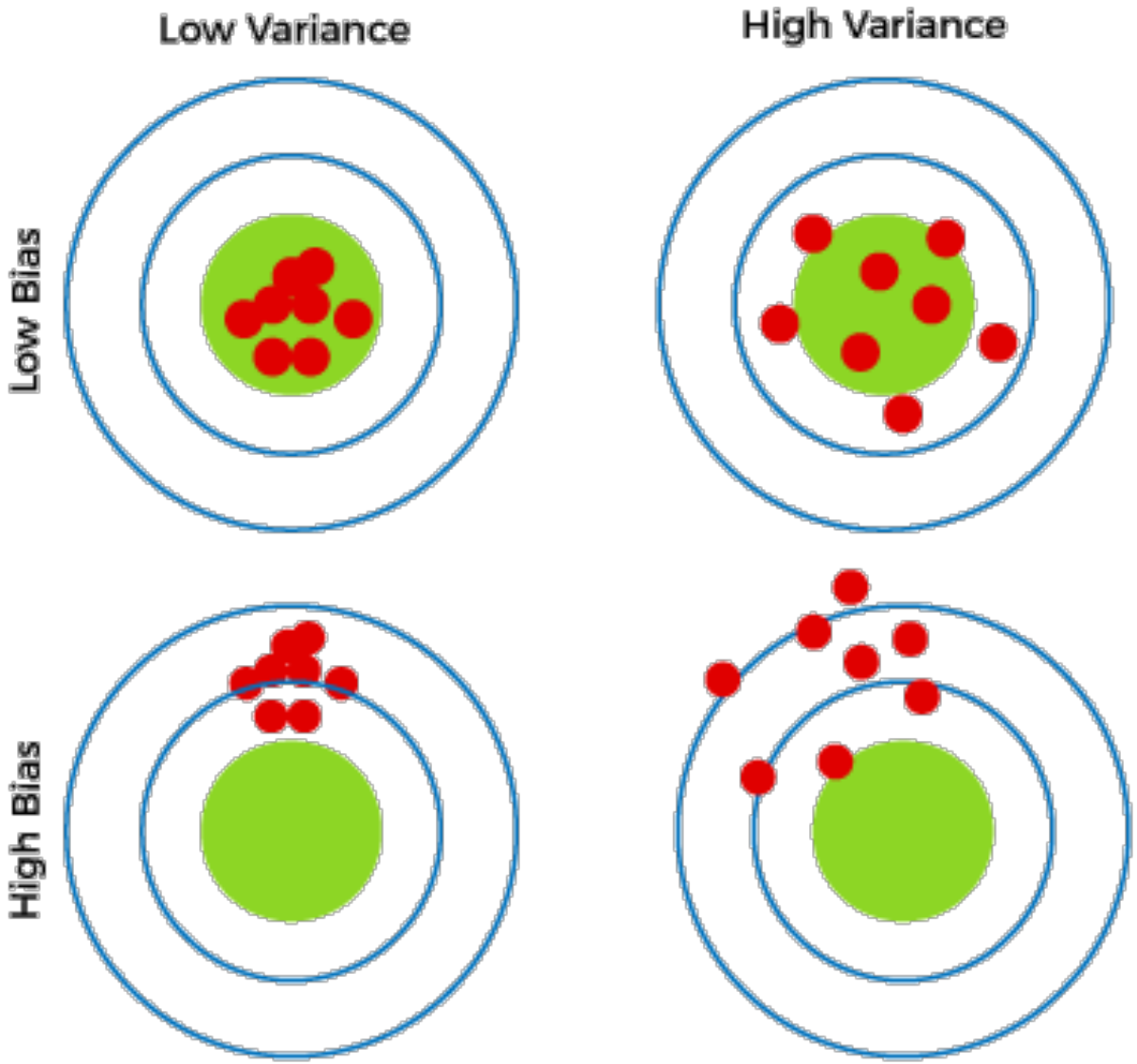


DIFFERENT COMBINATIONS OF BIAS-VARIANCE

- There are four possible combinations of bias and variances, which are represented by the below diagram:

1. **Low-Bias, Low-Variance:**
   The combination of low bias and low variance shows an ideal machine learning model. However, it is not possible practically.
2. **Low-Bias, High-Variance:** With low bias and high variance, model predictions are inconsistent and accurate on average.
3. **High-Bias, Low-Variance:** With High bias and low variance, predictions are consistent but inaccurate on average.
4. **High-Bias, High-Variance:**
   With high bias and high variance, predictions are inconsistent and also inaccurate on average.

DIFFERENT COMBINATIONS OF BIAS-VARIANCE



THANK YOU

- **Overfitting and Underfitting**

Machine Learning

Dr. Jagendra Singh

OVERFITTING AND UNDERFITTING

- Overfitting and Underfitting are the two main problems that occur in machine learning and degrade the performance of the machine learning models.
- The main goal of each machine learning model is **to generalize well**.
- Here **generalization** defines the ability of an ML model to provide a suitable output by adapting the given set of unknown input.
- It means after providing training on the dataset, it can produce reliable and accurate output.

OVERFITTING AND UNDERFITTING

- Hence, the underfitting and overfitting are the two terms that need to be checked for the performance of the model and whether the model is generalizing well or not.
- Before understanding the overfitting and underfitting, let's understand some basic term that will help to understand this topic well:
  - **Signal:** It refers to the true underlying pattern of the data that helps the machine learning model to learn from the data.
  - **Noise:** Noise is unnecessary and irrelevant data that reduces the performance of the model.
  - **Bias**
  - **Variance**

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.**

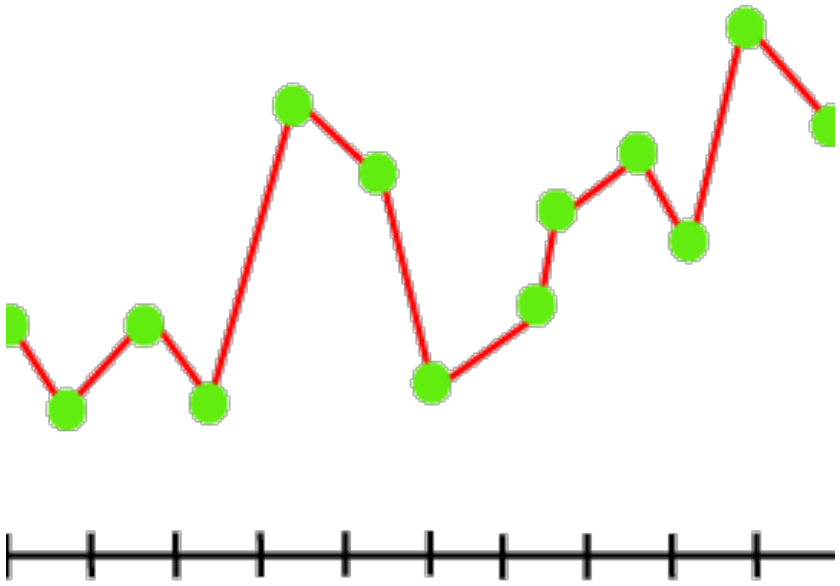
OVERFITTING

- The chances of occurrence of overfitting increase as much we provide training to our model.
- It means the more we train our model, the more chances of occurring the overfitted model.
- Overfitting is the main problem that occurs in supervised learning

OVERFITTING

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

- As we can see from the 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.



WAY TO AVOID THE OVERFITTING IN MODEL

- There are some ways by which we can reduce the occurrence of overfitting in our model.
  - **Cross-Validation**
  - **Training with more data**
  - **Removing features**
  - **Early stopping the training**
  - **Regularization**
  - **Ensembling**

UNDERFITTING

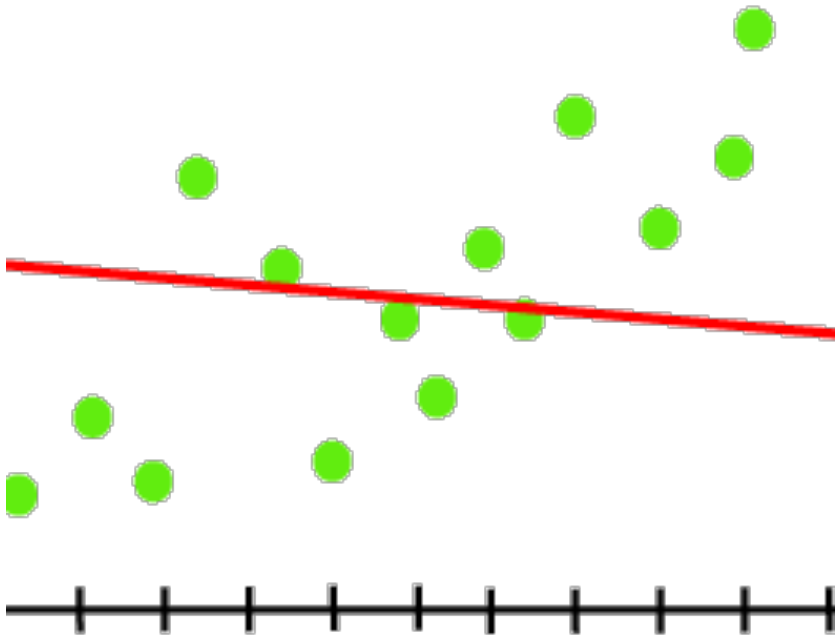
Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data.

To avoid the 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.

UNDERFITTING

- In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.
- 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 diagram, the model is unable to capture the data points present in the plot.



WAYS TO AVOID UNDERFITTING

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

GOODNESS OF FIT

- The "Goodness of fit" term is taken from the statistics, and the goal of the machine learning models to achieve the goodness of fit.
- In statistics modeling, *it defines how closely the result or predicted values match the true values of the dataset.*
- The model with a good fit is between the underfitted and overfitted model, and ideally, it makes predictions with 0 errors, but in practice, it is difficult to achieve it.

GOODNESS OF FIT

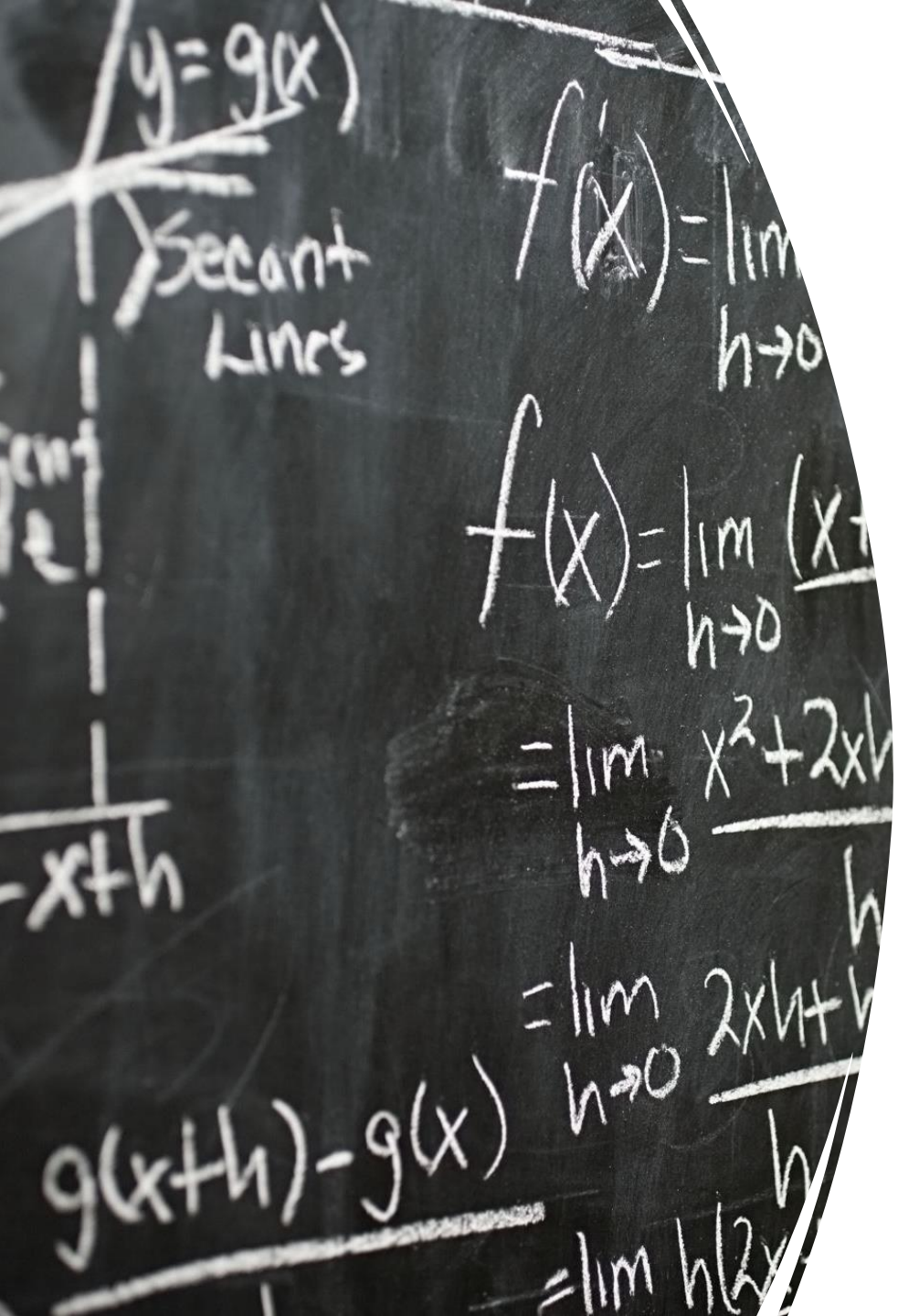
- When we train the model for a long duration, then the performance of the model may decrease due to the overfitting, as the model also learn the noise present in the dataset.
- The errors in the test dataset start increasing, *so the point, just before the raising of errors, is the good point, and we can stop here for achieving a good model.*
- There are two other methods by which we can get a good point for our model, which are the resampling method to estimate model accuracy and validation dataset.

THANK YOU

- **Bayes Theorem Algorithm**

Machine Learning

Dr. Jagendra Singh



BAYES THEOREM

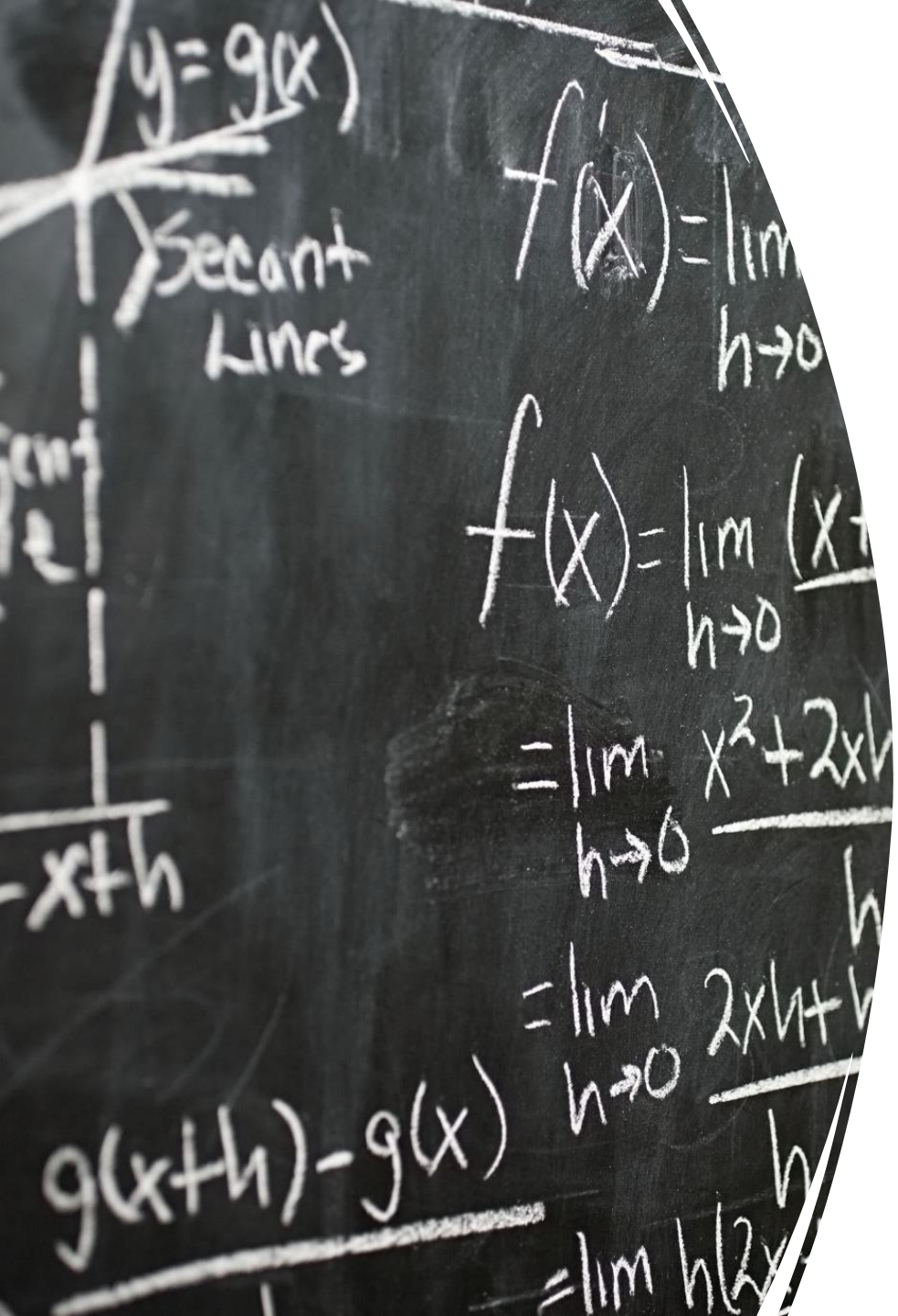
- An important concept of Bayes theorem named **Bayesian method** is used to calculate conditional probability in Machine Learning application that includes classification tasks.
- Further, a simplified version of Bayes theorem (Naïve Bayes classification) is also used to reduce computation time and average cost of the projects.
- Bayes theorem is also known with some other name such as **Bayes rule or Bayes Law.**

BAYES THEOREM

Bayes theorem helps to determine the probability of an event with random knowledge.

It is used to calculate the probability of occurring one event while other one already occurred.

It is a best method to relate the condition probability and marginal probability.



WHAT IS BAYES THEOREM?

- ✓ Bayes theorem is one of the most popular machine learning concepts that helps to calculate the probability of occurring one event with uncertain knowledge while other one has already occurred.
- ✓ Bayes' theorem can be derived using product rule and conditional probability of event X with known event Y:
  - According to the product rule we can express as the probability of event X with known event Y as follows;

$$P(X ? Y) = P(X|Y)\,P(Y) \qquad (1)$$

WHAT IS BAYES THEOREM?

- Further, the probability of event Y with known event X:

  - P(X ? Y)= P(Y|X) P(X) (2)

✓ Mathematically, Bayes theorem can be expressed by combining both equations on right hand side. We will get:

$$P(X|Y) = \frac{P(Y|X).P(X)}{P(Y)}$$

WHAT IS BAYES THEOREM?

Bayes' Theorem Formula

The formula to calculate a posterior probability of A occurring given that B occurred:

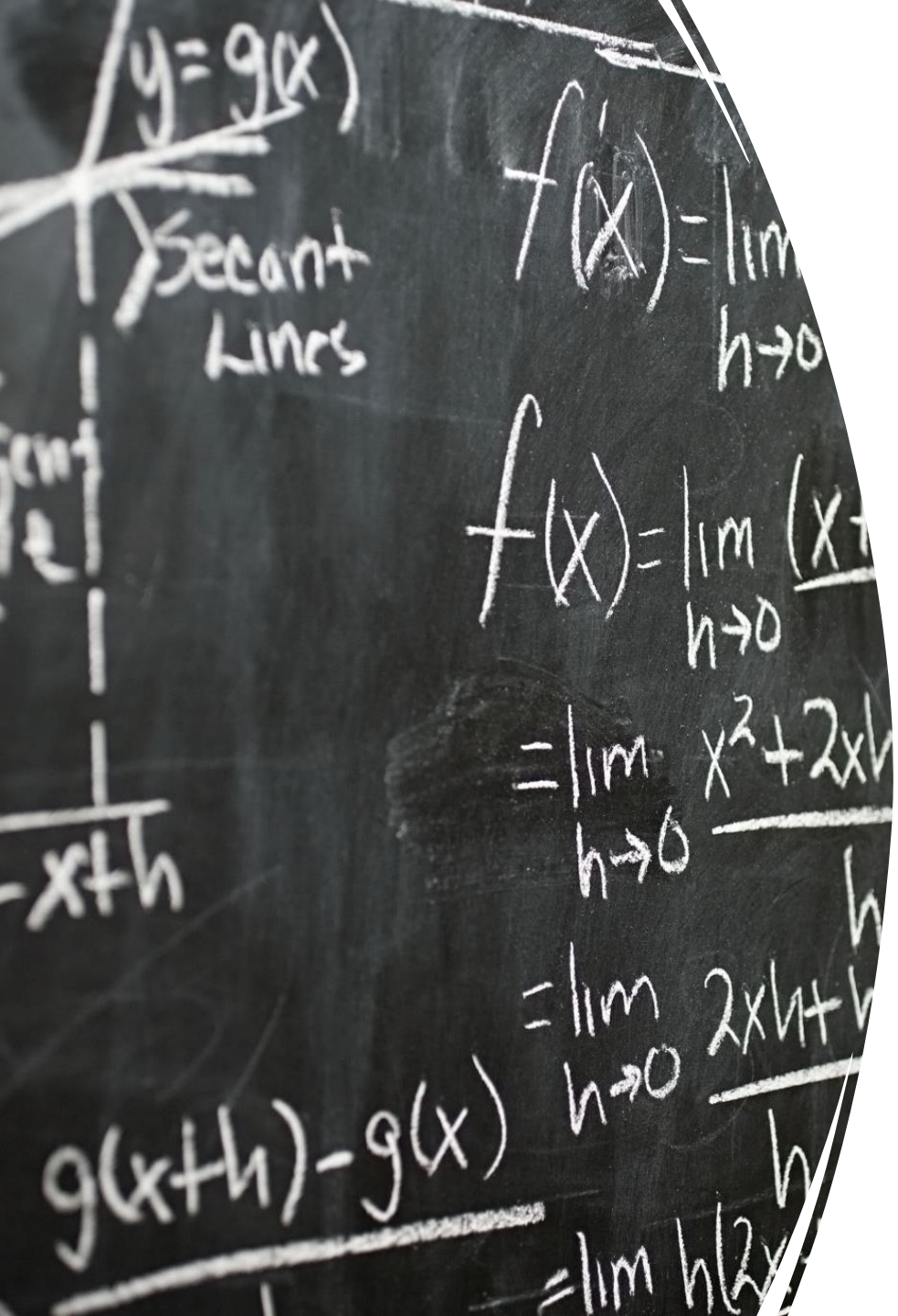
$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \times P(B \mid A)}{P(B)}$$

where:

A, B = Events

$P(B \mid A)$ = The probability of B occurring given that A is true

$P(A)$ and $P(B)$ = The probabilities of A occurring and B occurring independently of each other

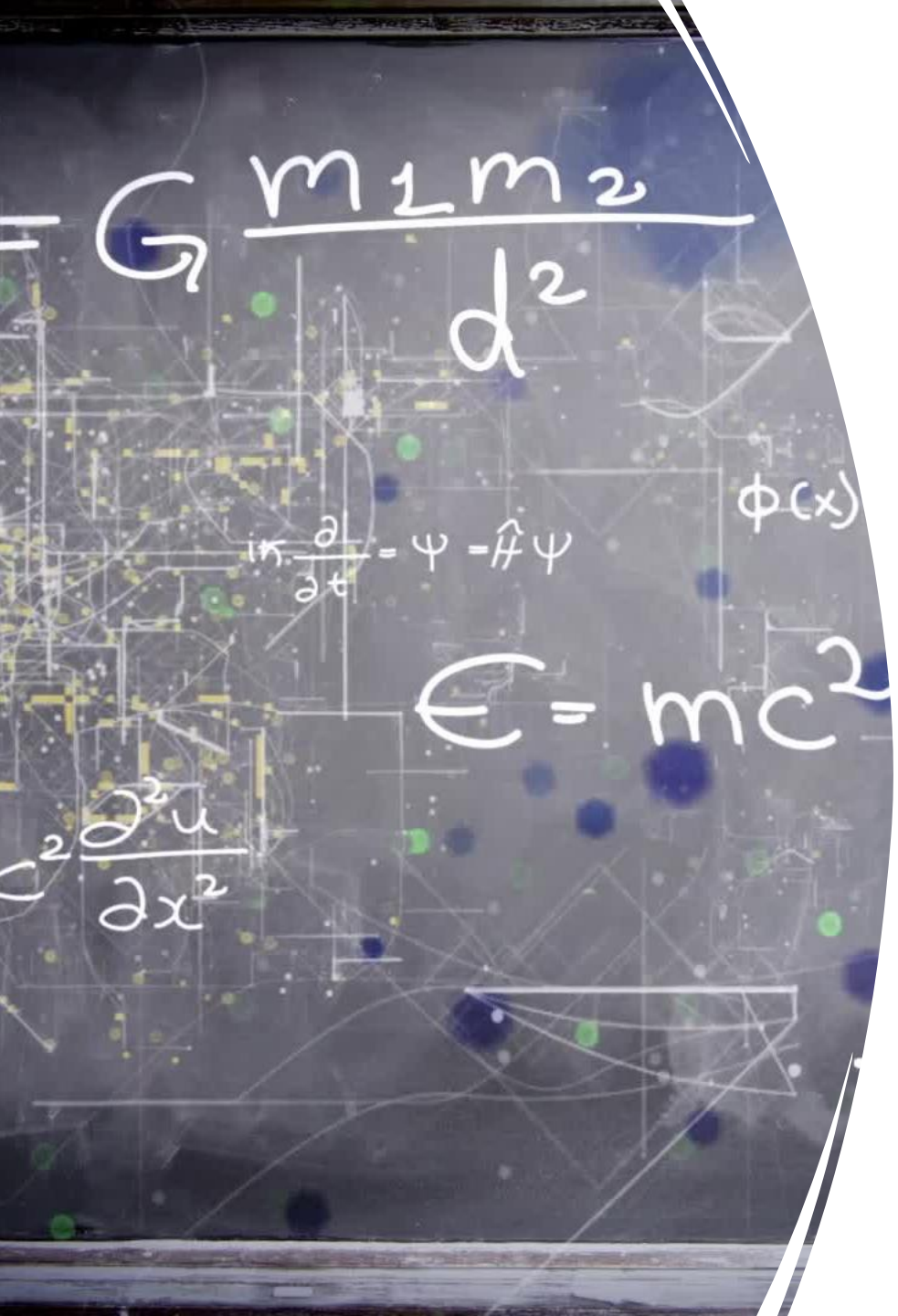


WHAT IS BAYES THEOREM?

- ✓ Here, both events X and Y are independent events which means probability of outcome of both events does not depends one another.
- ✓ The above equation is called as Bayes Rule or Bayes Theorem.

WHERE

- P(X|Y) is called as **posterior**, which we need to calculate. It is defined as updated probability after considering the evidence.
- P(Y|X) is called the **likelihood**. It is the probability of evidence when hypothesis is true.
- P(X) is called the **prior probability**, probability of hypothesis before considering the evidence
- P(Y) is called **marginal probability**. It is defined as the probability of evidence under any consideration.



BAYES THEOREM

- Hence, Bayes Theorem can be written as:
- **posterior = (likelihood * prior) / evidence**

PROBABILITY TERMS

Sample Space

- During an experiment what we get as a result is called as possible outcomes and the set of all possible outcome of an event is known as sample space. For example, if we are rolling a dice, sample space will be:
  - S1 = {1, 2, 3, 4, 5, 6}
- Similarly, if our experiment is related to toss a coin and recording its outcomes, then sample space will be:
  - S2 = {Head, Tail}

PROBABILITY TERMS

Event

- Event is defined as subset of sample space in an experiment. Further, it is also called as set of outcomes.
- Assume in our experiment of rolling a dice, there are two event A and B such that;
- A = Event when an even number is obtained = {2, 4, 6}
- B = Event when a number is greater than 4 = {5, 6}

PROBABILITY TERMS

Event

- Probability of the event A "P(A)"= Number of favourable outcomes / Total number of possible outcomes
  P(E) = 3/6 =1/2 =0.5
  - Similarly, Probability of the event B "P(B)"= Number of favourable outcomes / Total number of possible outcomes
    =2/6
    =1/3
    =0.333
- Union of event A and B:
  A∪B = {2, 4, 5, 6}

PROBABILITY TERMS

Independent Event

- Two events are said to be independent when occurrence of one event does not affect the occurrence of another event.
- In simple words we can say that the probability of outcome of both events does not depends one another.
- Mathematically, two events A and B are said to be independent if:
  - $P(A \cap B) = P(AB) = P(A)*P(B)$

PROBABILITY TERMS

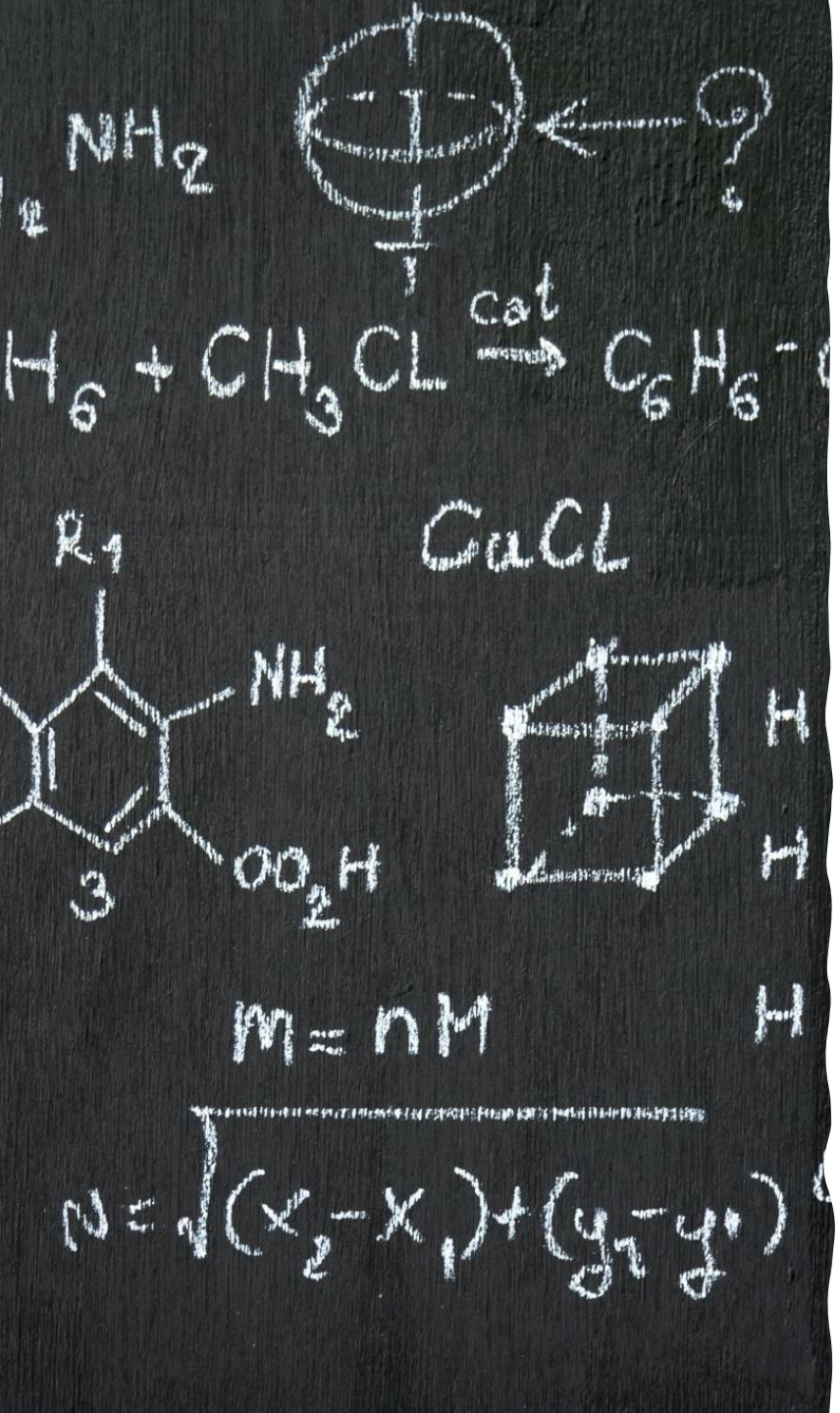
Conditional Probability

- Conditional probability is defined as the probability of an event A, given that another event B has already occurred (i.e. A conditional B). This is represented by P(A|B) and we can define it as:
  - $P(A|B) = P(A \cap B) / P(B)$

PROBABILITY TERMS

Marginal Probability

- Marginal probability is defined as the probability of an event A occurring independent of any other event B. Further, it is considered as the probability of evidence under any consideration.
  - $P(A) = P(A|B)*P(B) + P(A|\sim B)*P(\sim B)$
- Here ~B represents the event that B does not occur.



HOW TO APPLY BAYES THEOREM OR IN MACHINE LEARNING?

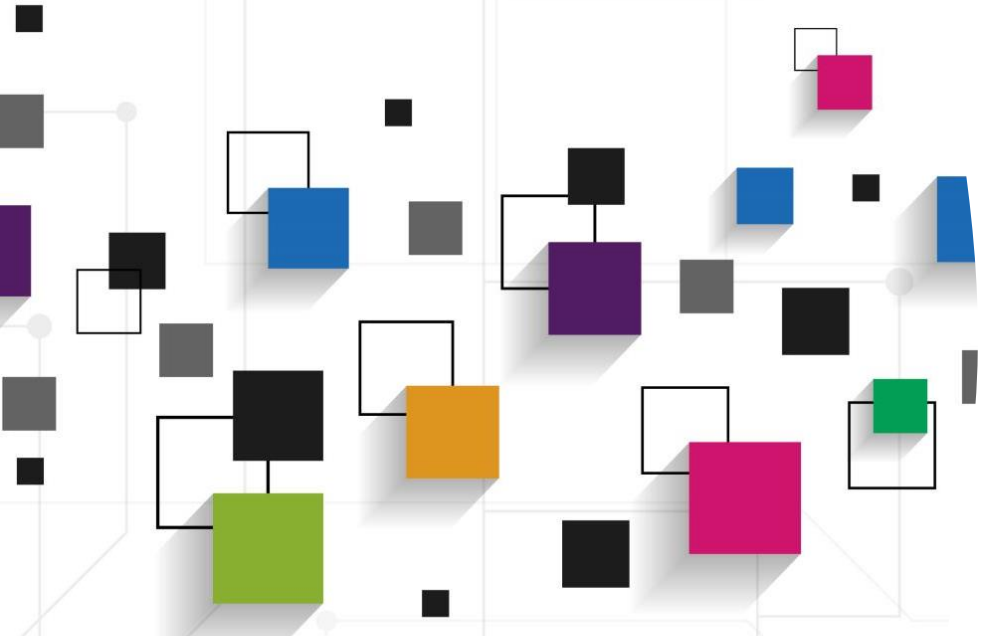
- Bayes theorem helps us to calculate the single term P(B|A) in terms of P(A|B), P(B), and P(A).
- This rule is very helpful in such scenarios where we have a good probability of P(A|B), P(B), and P(A) and need to determine the fourth term.
- Naïve Bayes classifier is one of the simplest applications of Bayes theorem which is used in classification algorithms to isolate data as per accuracy, speed and classes.

THANK YOU

- **Naïve Bayes Classifier Algorithm**

Dr. Jagendra Singh

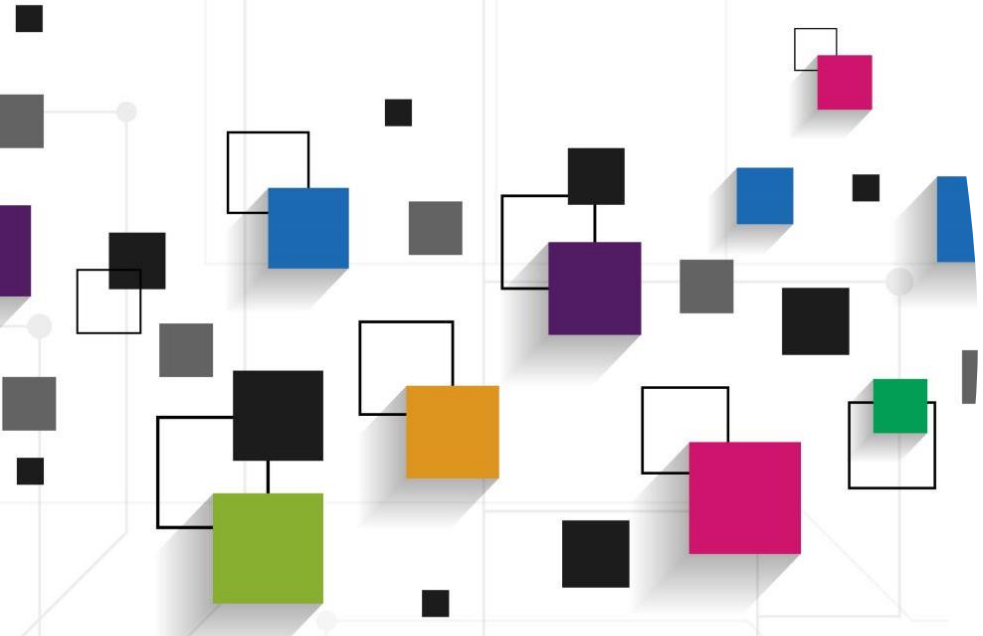
Machine Learning



NAÏVE BAYES CLASSIFIER

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.

- It is mainly used in *text classification* that includes a high-dimensional training dataset.



NAÏVE BAYES CLASSIFIER

- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

NAÏVE BAYES CLASSIFIER

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

WHY CALLED NAÏVE BAYES

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.
- Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple.
- Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes Theorem

BAYES' THEOREM

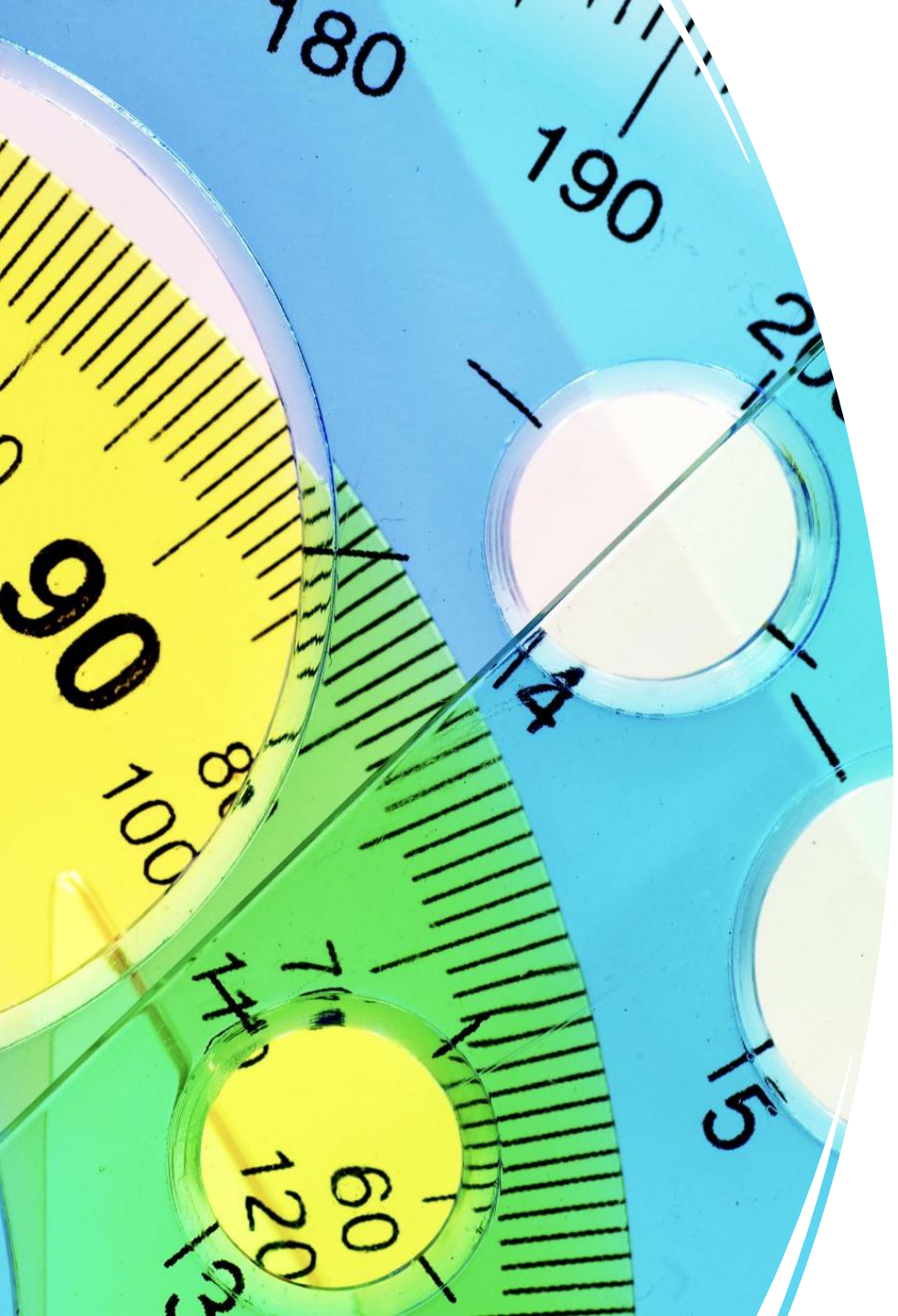
Where,

- **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.

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

NAÏVE BAYES' CLASSIFIER WORKING

- Working of Naïve Bayes' Classifier can be understood with the help of the below example:
- Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**".
- So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions.
- So to solve this problem, we need to follow the below steps:



NAÏVE BAYES' CLASSIFIER WORKING

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

NAÏVE BAYES' CLASSIFIER WORKING

- **Problem**: If the weather is sunny, then the Player should play or not?
- **Solution**: To solve this, first consider the below dataset:

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

FREQUENCY TABLE FOR THE WEATHER CONDITIONS

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

LIKELIHOOD TABLE WEATHER CONDITION

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

APPLYING BAYE'S THEOREM

- **P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)**
- P(Sunny|Yes)= 3/10= 0.3
- P(Sunny)= 0.35
- P(Yes)=0.71
- So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**

APPLYING BAYE'S THEOREM

- **P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)**
- P(Sunny|NO)= 2/4=0.5
- P(No)= 0.29
- P(Sunny)= 0.35
- So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**
- So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**
- **Hence on a Sunny day, Player can play the game.**

ADVANTAGES OF NB CLASSIFIER

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.

ADVANTAGES OF NB CLASSIFIER

- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

DISADVANTAGES OF NB CLASSIFIER

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

APPLICATIONS OF NB CLASSIFIER

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

TYPES OF NAÏVE BAYES MODEL

There are **three types** of Naive Bayes Model, which are given below: (Gaussian, Multinomial, Bernoulli)

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution.
- This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

TYPES OF NAÏVE BAYES MODEL

- For example, suppose the training data contains a continuous attribute x.
- We first segment the data by the class, and then compute the mean and variance of x in each class.
- Let μi be the mean of the values and let σi be the variance of the values associated with the ith class.
- Suppose we have some observation value xi .
- Then, the probability distribution of xi given a class can be computed by the following equation –

$$p(x_i|y_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(x_i-\mu_j)^2}{2\sigma_j^2}}$$

TYPES OF NAÏVE BAYES MODEL

- Formula:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$f(x)$ = probability density function

σ = standard deviation

μ = mean

TYPES OF NAÏVE BAYES MODEL

- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed.
- It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc.
- The classifier uses the frequency of words for the predictors.

TYPES OF NAÏVE BAYES MODEL

- **Multinomial**: The probability mass function of this multinomial distribution is:

$$f(x_1,\ldots,x_k;n,p_1,\ldots,p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

$$= \begin{cases} \dfrac{n!}{x_1!\cdots x_k!} p_1^{x_1} \times \cdots \times p_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

for non-negative integers $x_1, \ldots, x_k$.

The probability mass function can be expressed using the gamma function as:

$$f(x_1,\ldots,x_k;p_1,\ldots,p_k) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_{i=1}^{k} p_i^{x_i}.$$

This form shows its resemblance to the Dirichlet distribution, which is its conjugate prior.

TYPES OF NAÏVE BAYES MODEL

- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables.
- Such as if a particular word is present or not in a document.
- This model is also famous for document classification tasks.

TYPES OF NAÏVE BAYES MODEL

- **Bernoulli**: Formula

$$f(k;p) = pk + (1-p)(1-k)$$

p = probability

k = possible outcomes

f = probability mass function

IMPLEMENTATION OF NB ALGORITHM

- Now we will implement a Naive Bayes Algorithm using Python.
- So for this, we will use the "**user_data**" **dataset**, which we have used in our other classification model.
- Therefore we can easily compare the Naive Bayes model with the other models.

IMPLEMENTATION OF NB ALGORITHM

- **Steps to implement:**
  - Data Pre-processing step
  - Fitting Naive Bayes to the Training set
  - Predicting the test result
  - Test accuracy of the result(Creation of Confusion matrix)
  - Visualizing the test set result.

THANK YOU

Naïve Bayes Classification

Things We'd Like to Do

- Spam Classification
  - Given an email, predict whether it is spam or not
- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has disease X or not
- Weather
  - Based on temperature, humidity, etc… predict if it will rain tomorrow

- The relationship between attribute set and the class variable is non-deterministic.
- Even if the attributes are same, the class label may differ in training set even and hence can not be predicted with certainty.
- Reason: noisy data, certain other attributes are not included in the data.

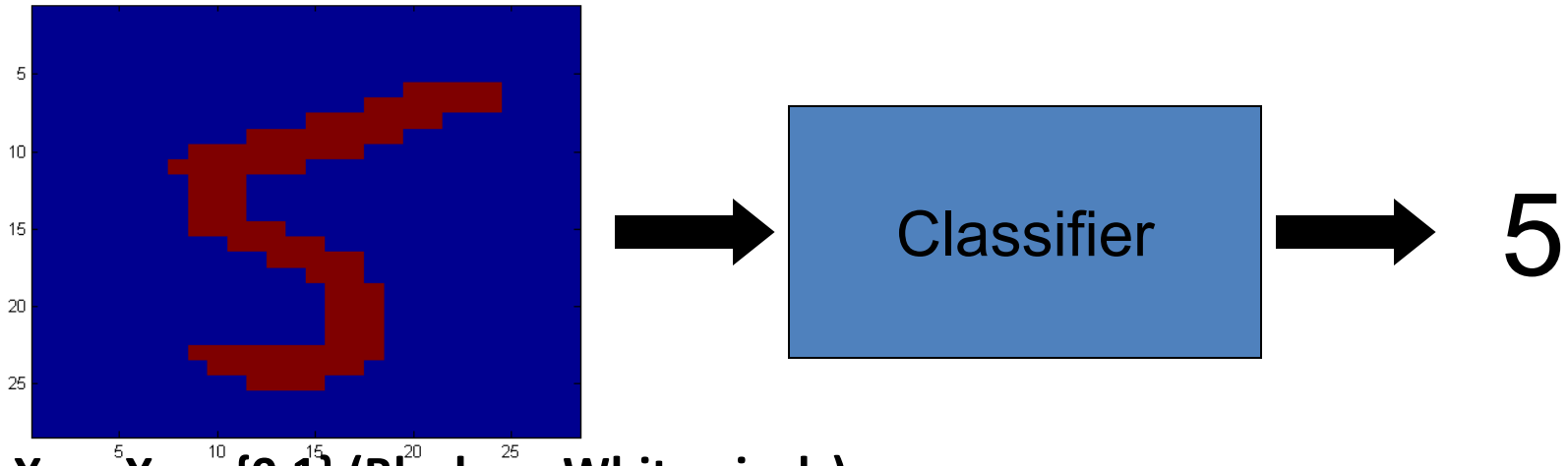
- Example: Task of predicting whether a person is at risk for heart disease based on the person's diet and workout frequency.
- So need an approach to model probabilistic relationship between attribute set and the class variable.

Bayesian Classification

- Problem statement:
  - Given features $X_1, X_2, \ldots, X_n$
  - Predict a label Y

Another Application

- **Digit Recognition**



- **$X_1,\ldots,X_n \in \{0,1\}$ (Black vs. White pixels)**
- **$Y \in \{5,6\}$ (predict whether a digit is a 5 or a 6)**

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Example of Bayes Theorem

- Given:
  - A doctor knows that Cold causes fever 50% of the time
  - Prior probability of any patient having cold is 1/50,000
  - Prior probability of any patient having fever is 1/20

- If a patient has fever, what's the probability he/she has cold?

$$P(C \mid F) = \frac{P(F \mid C)P(C)}{P(F)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2,…,A_n)

- Can we estimate P(C| A_1, A_2,…,A_n) directly from data?

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

  - Choose value of C that maximizes
    $P(C \mid A_1, A_2, \ldots, A_n)$

  - Equivalent to choosing value of C that maximizes
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - Can estimate $P(A_i \mid C_j)$ for all A_i and C_j.
  - New point is classified to C_j if $P(C_j) \prod P(A_i \mid C_j)$ is maximum.

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$
  - e.g., P(No) = 7/10,
    P(Yes) = 3/10

- For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}| / N_c$$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
  - Examples:

    P(Status=Married|No) = 4/7
    P(Refund=Yes|Yes)=0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - Two-way split: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

k

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (A_i, c_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No)
  × P(Married| Class=No)
  × P(Income=120K| Class=No)
  = 4/7 × 4/7 × 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  × P(Married| Class=Yes)
  × P(Income=120K| Class=Yes)
  = 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original} : P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace} : P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m-estimate} : P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability

m: parameter

Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(A|M)P(M) > P(A|N)P(N)

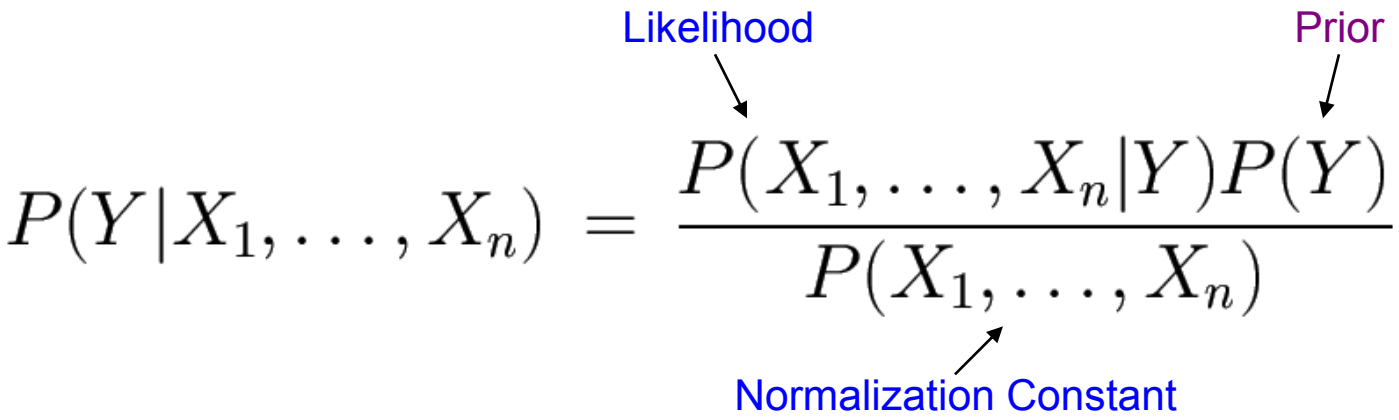
=> Mammals

Naïve Bayes (Summary)

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)

The Bayes Classifier

- Use Bayes Rule!



- Why did this help? Well, we think that we might be able to specify how features are "generated" by the class label

The Bayes Classifier

- Let's expand this for our digit recognition task:

$$P(Y=5|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=5)P(Y=5)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

$$P(Y=6|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=6)P(Y=6)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

- To classify, we'll simply compute these two probabilities and predict based on which one is greater

Model Parameters

- For the Bayes classifier, we need to “learn” two functions, the likelihood and the prior

- How many parameters are required to specify the prior for our digit recognition example?

Model Parameters

- The problem with explicitly modeling $P(X_1,\ldots,X_n|Y)$ is that there are usually way too many parameters:
  - We'll run out of space
  - We'll run out of time
  - And we'll need tons of training data (which is usually not available)

The Naïve Bayes Model

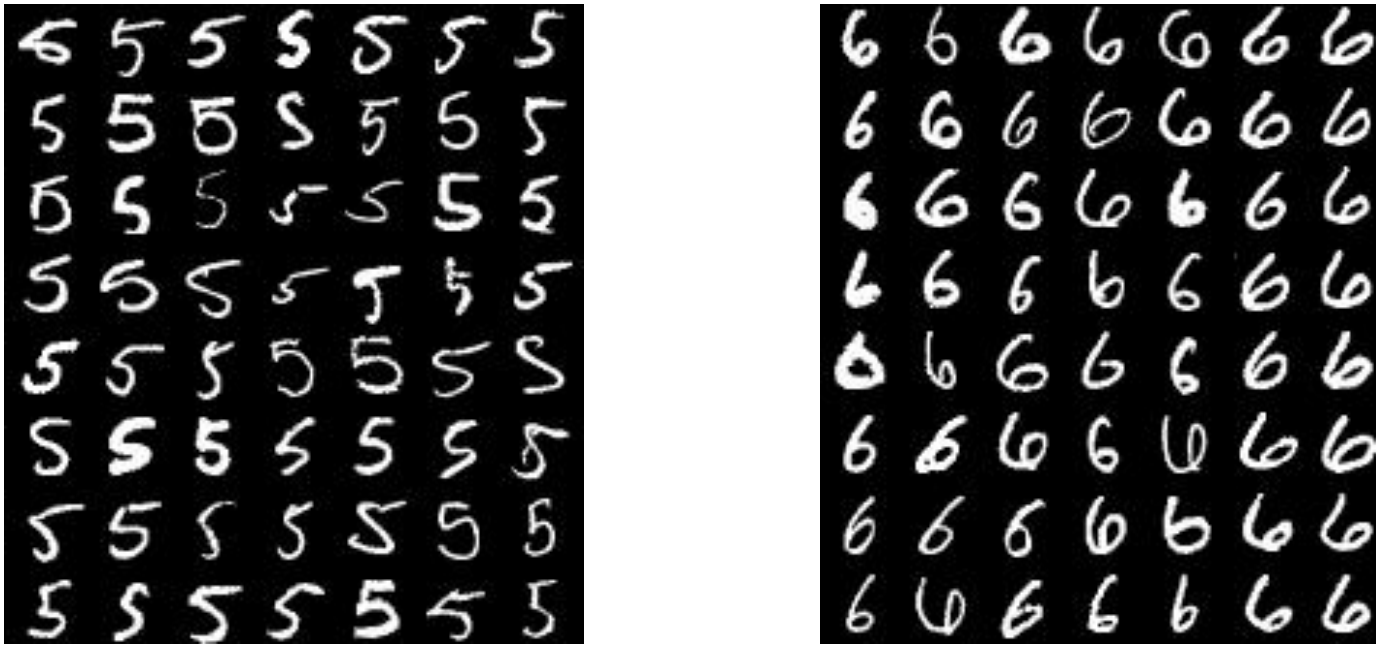
- The *Naïve Bayes Assumption*: Assume that all features are independent **given the class label Y**
- Equationally speaking:

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

- (We will discuss the validity of this assumption later)

Naïve Bayes Training

- Now that we've decided to use a Naïve Bayes classifier, we need to train it with some data:



MNIST Training Data

Naïve Bayes Training

- Training in Naïve Bayes is **easy**:
  - Estimate P(Y=v) as the fraction of records with Y=v

$$P(Y = v) = \frac{Count(Y = v)}{\#\ records}$$

  - Estimate P(X_i=u|Y=v) as the fraction of records with Y=v for which X_i=u

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v)}{Count(Y = v)}$$

Naïve Bayes Training

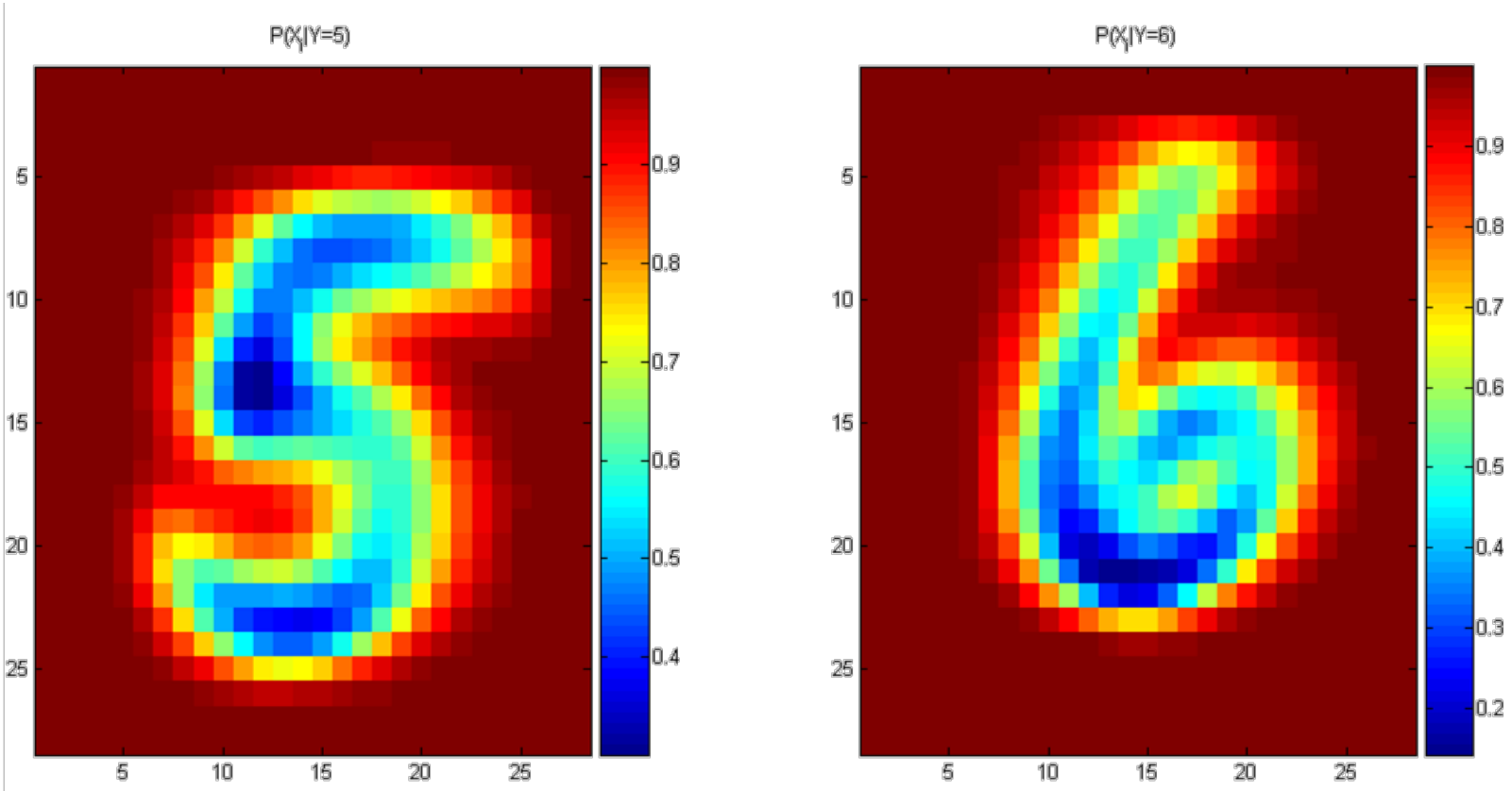
- In practice, some of these counts can be zero
- Fix this by adding "virtual" counts:

$$P(X_i = u|Y = v) = \frac{Count(X_i = u \wedge Y = v) + 1}{Count(Y = v) + 2}$$

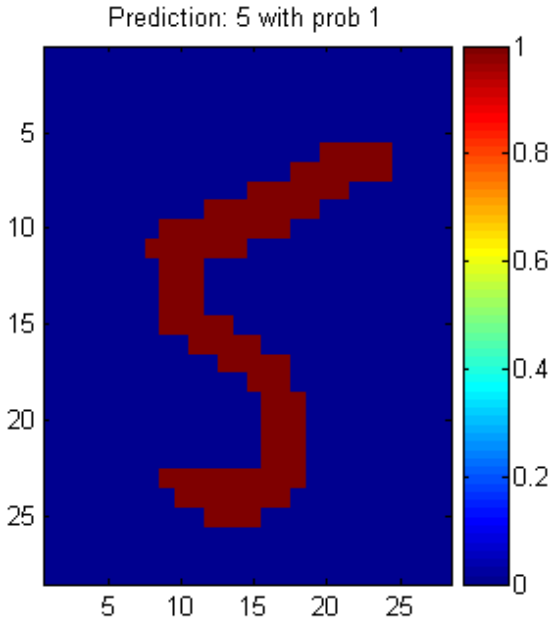
  - This is called *Smoothing*

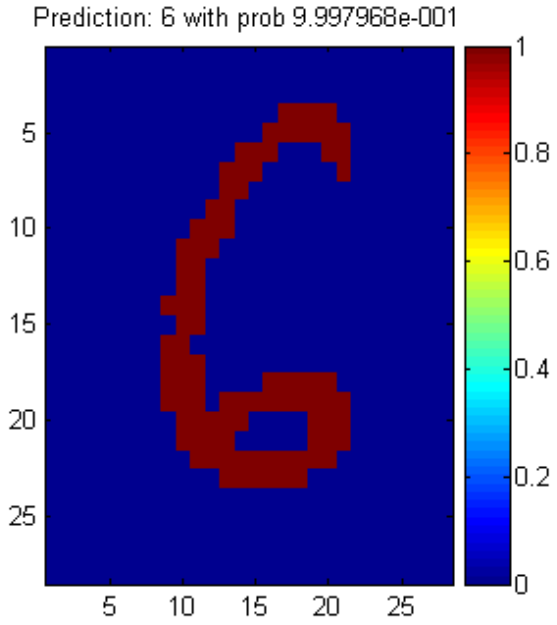
Naïve Bayes Training

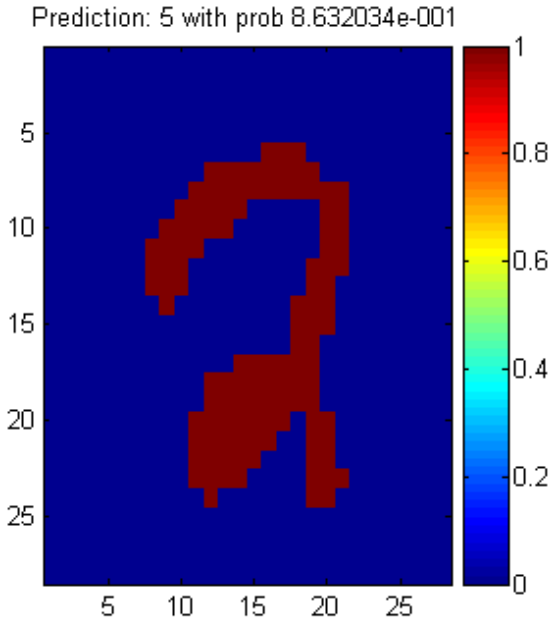
- For binary digits, training amounts to averaging all of the training fives together and all of the training sixes together.



Naïve Bayes Classification







Another Example of the Naïve Bayes Classifier

The weather data, with counts and probabilities

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

A new day

outlook	temperature	humidity	windy	play
sunny	cool	high	true	?

- Likelihood of yes

$$= \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0.0053$$

- Likelihood of no

$$= \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0206$$

- Therefore, the prediction is No

The Naive Bayes Classifier for Data Sets with Numerical Attribute Values

- One common practice to handle numerical attribute values is to assume normal distributions for numerical attributes.

The numeric weather data with summary statistics

outlook			temperature		humidity		windy			play	
	yes	no	yes	no	yes	no		yes	no	yes	no
sunny	2	3	83	85	86	85	false	6	2	9	5
overcast	4	0	70	80	96	90	true	3	3		
rainy	3	2	68	65	80	70					
			64	72	65	95					
			69	71	70	91					
			75		80						
			75		70						
			72		90						
			81		75						
sunny	2/9	3/5	mean 73	74.6	mean 79.1	86.2	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	std dev 6.2	7.9	std dev 10.2	9.7	true	3/9	3/5		
rainy	3/9	2/5									

- Let x_1, x_2, …, x_n be the values of a numerical attribute in the training data set.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

$$f(w) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(w-\mu)^2}{\sigma^2}}$$

- For examples,

$$f(\text{temperature} = 66 \mid \text{Yes}) = \frac{1}{\sqrt{2\pi}(6.2)} e^{-\frac{(66-73)^2}{2(6.2)^2}} = 0.0340$$

- Likelihood of Yes = $\frac{2}{9} \times 0.0340 \times 0.0221 \times \frac{3}{9} \times \frac{9}{14} = 0.000036$

- Likelihood of No = $\frac{3}{5} \times 0.0291 \times 0.038 \times \frac{3}{5} \times \frac{5}{14} = 0.000136$

Outputting Probabilities

- What's nice about Naïve Bayes (and generative models in general) is that it returns probabilities
  - These probabilities can tell us how confident the algorithm is
  - So… don't throw away those probabilities!

Recap

- We defined a *Bayes classifier* but saw that it's intractable to compute $P(X_1,\ldots,X_n|Y)$
- We then used the *Naïve Bayes assumption* – that everything is independent given the class label Y

- A natural question: is there some happy compromise where we only assume that *some* features are conditionally independent?

Conclusions

- Naïve Bayes is:
  - Really easy to implement and often works well
  - Often a good first thing to try
  - Commonly used as a “punching bag” for smarter algorithms

- **Maximum Likelihood Estimation (MLE)**

Machine Learning

Dr. Jagendra Singh

MAXIMUM LIKELIHOOD ESTIMATION

- Maximum Likelihood Estimation (MLE) is a probabilistic based approach to determine values for the parameters of the model.
- MLE is a widely used technique in machine learning, time series, panel data and discrete data.

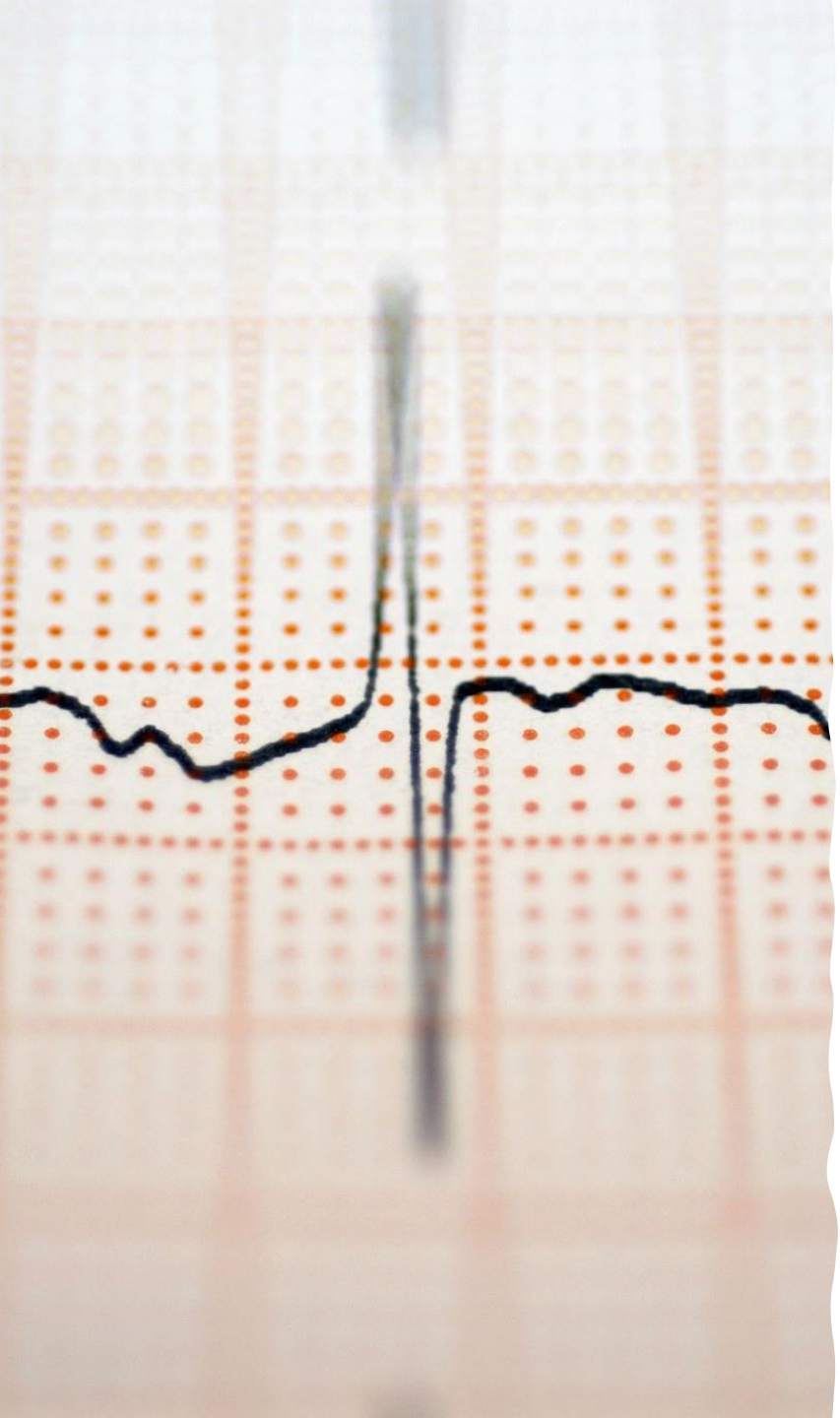
MAXIMUM LIKELIHOOD

- The motive of MLE is to maximize the likelihood of values for the parameter to get the desired outcomes.
- Following are the topics to be covered.

- What is the likelihood?
- Working of Maximum Likelihood Estimation
- Maximum likelihood estimation in machine learning

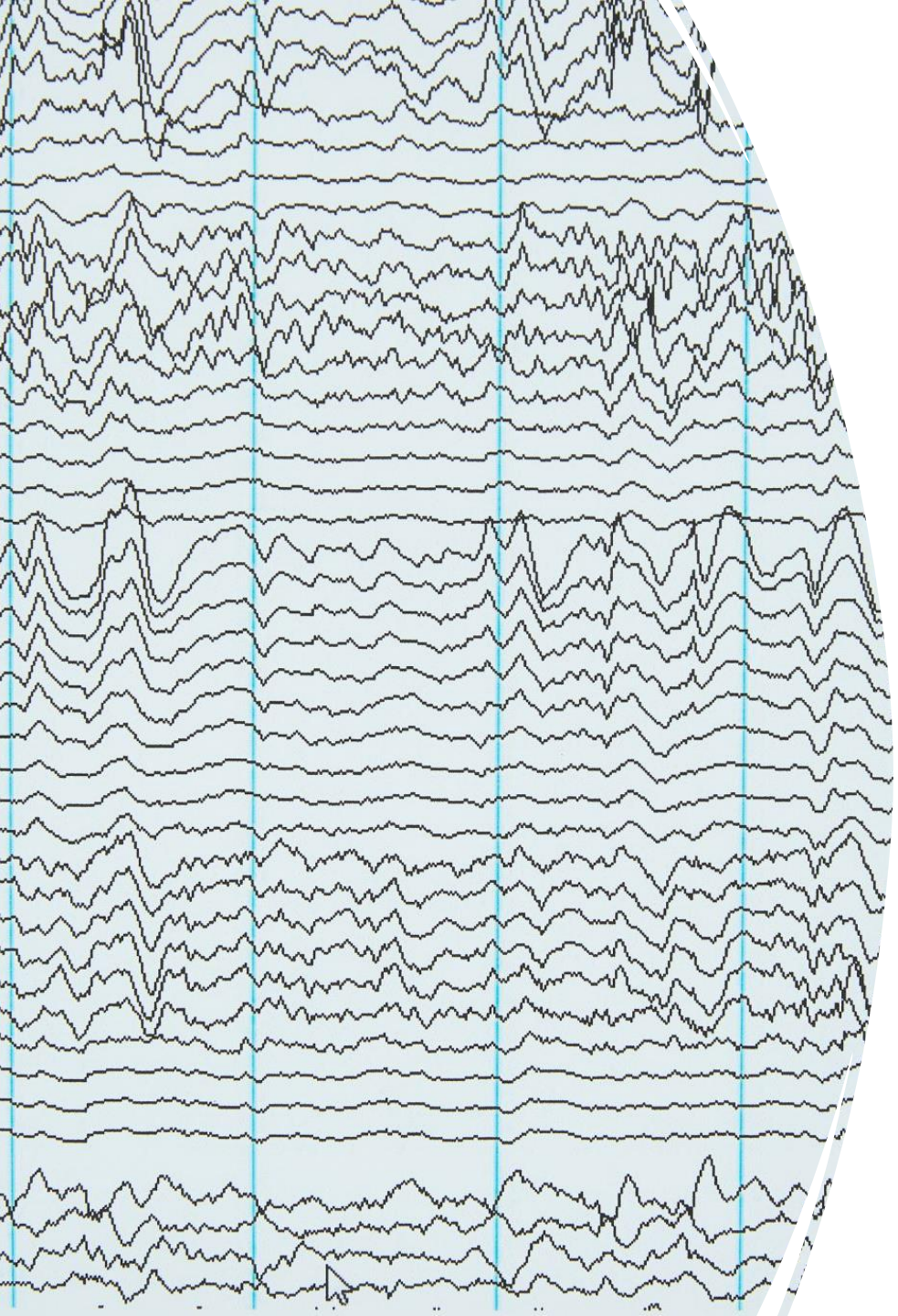
MAXIMUM LIKELIHOOD

- For understanding the concept of Maximum Likelihood Estimation (MLE) we need to understand the concept of Likelihood first and how it is related to probability.



WHAT IS THE LIKELIHOOD?

- The likelihood function measures the extent to which the data provide support for different values of the parameter.
- It indicates how likely it is that a particular population will produce a sample.

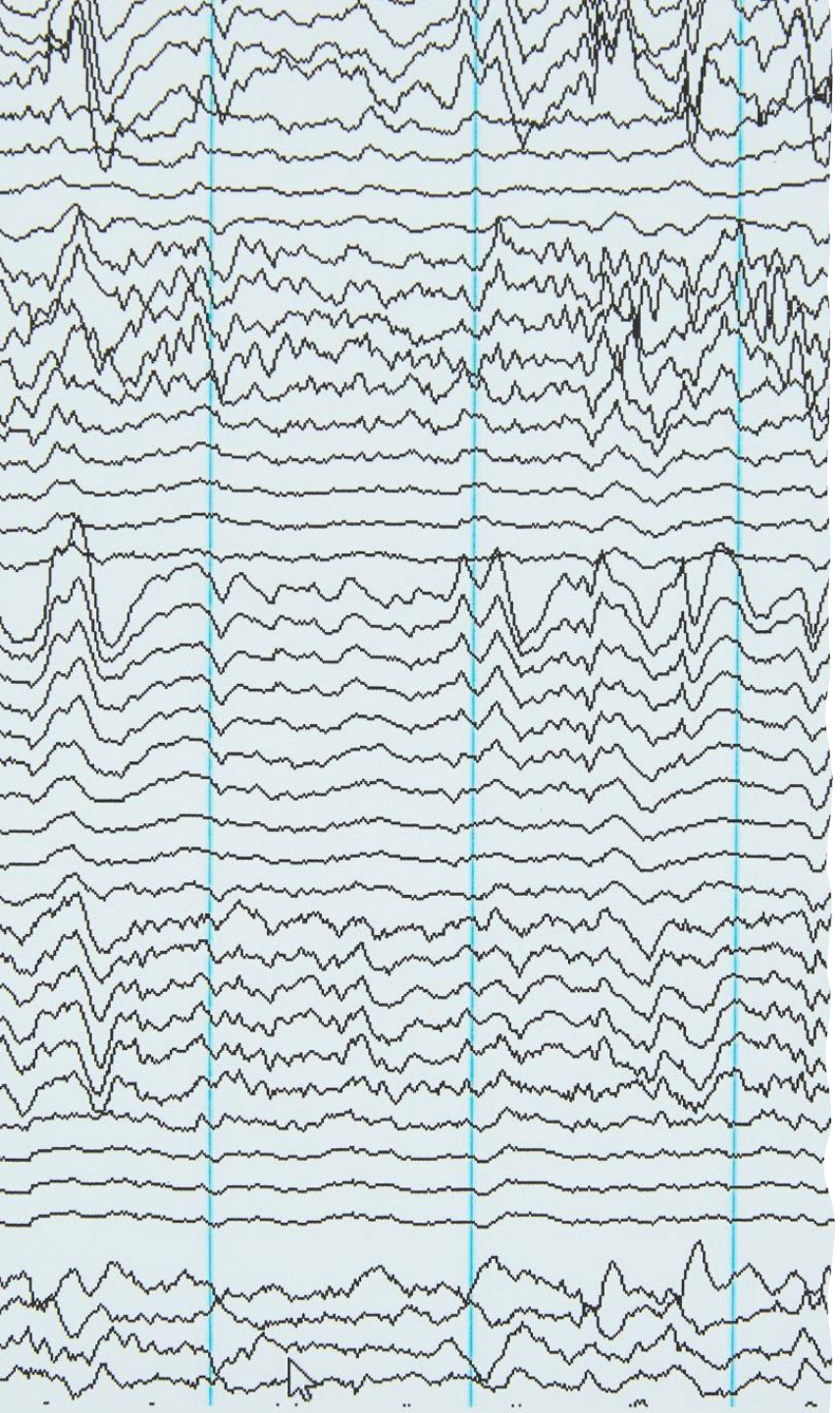


MAXIMUM LIKELIHOOD

- The likelihood function is different from the probability density function.
- Likelihood describes how to find the best distribution of the data for some feature or some situation.

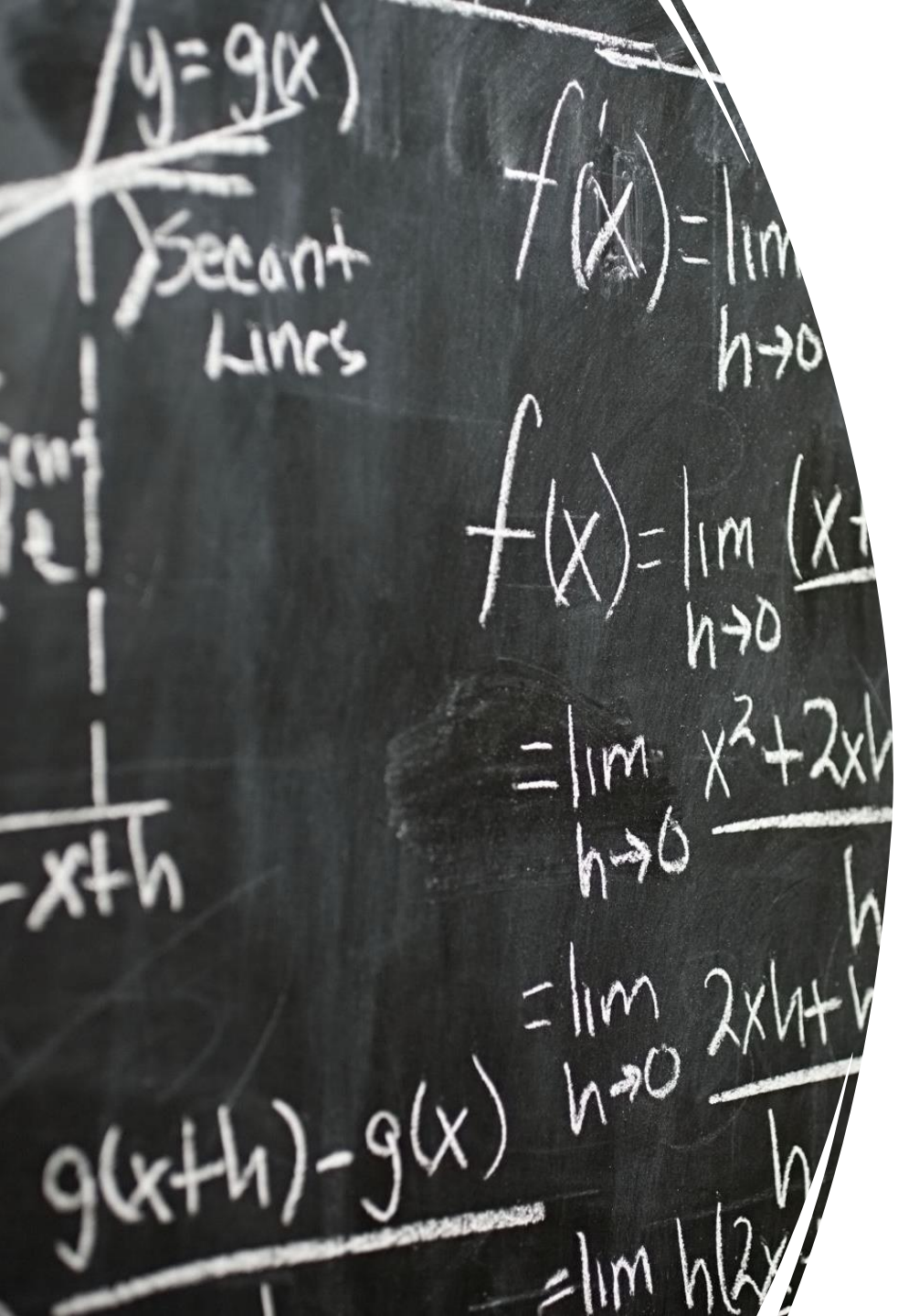
MAXIMUM LIKELIHOOD

- Probability describes how to find the chance of something given a sample distribution of data.
- Let's understand the difference between the likelihood and probability density function with the help of an example.
- Consider a dataset containing the weight of the customers. Let's say the mean of the data is 70 & the standard deviation is 2.5.



MAXIMUM LIKELIHOOD

- When Probability has to be calculated for any situation using this dataset, then the mean and standard deviation of the dataset will be constant.
- But in the case of Likelihood, mean and standard deviation of the dataset will be varied to get the maximum likelihood for weight > 70 kg.



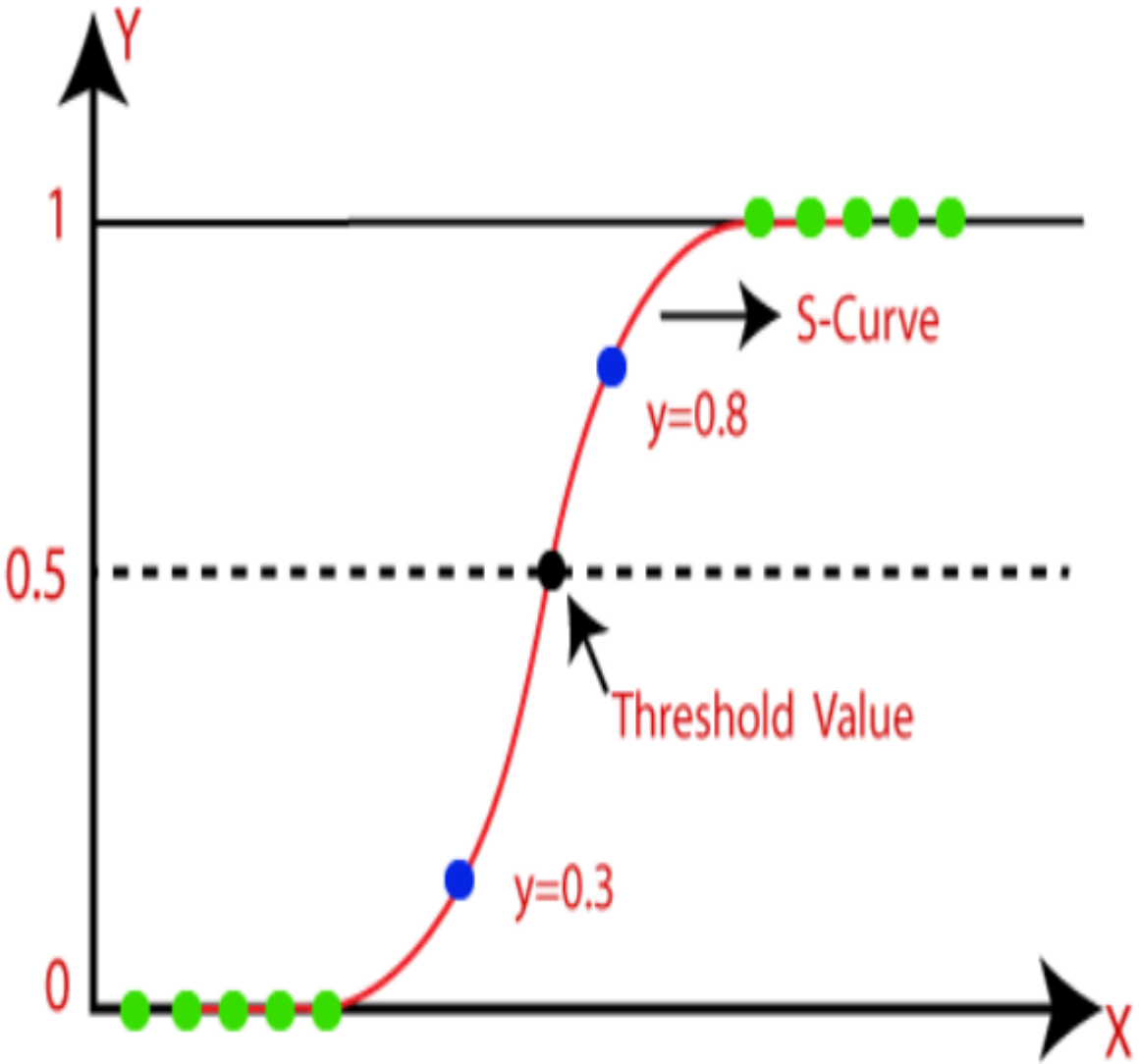
WORKING OF MAXIMUM LIKELIHOOD ESTIMATION

- The maximization of the likelihood estimation is the main objective of the MLE.
- Let’s understand this with an example.
- Consider there is a binary classification problem in which we need to classify the data into two categories either 0 or 1 based on a feature called “salary”.

WORKING OF MAXIMUM LIKELIHOOD ESTIMATION

- So MLE will calculate the possibility for each data point in salary and then by using that possibility, it will calculate the likelihood of those data points to classify them as either 0 or 1.

- It will repeat this process of likelihood until the learner line is best fitted. This process is known as the maximization of likelihood.



WORKING OF MAXIMUM LIKELIHOOD ESTIMATION

- The above explains the scenario, as we can see there is a threshold of 0.5 so if the possibility comes out to be greater than that it is labelled as 1 otherwise 0.
- Let's see how MLE could be used for classification.

MAXIMUM LIKELIHOOD ESTIMATION IN MACHINE LEARNING

- MLE is the base of a lot of supervised learning models, one of which is Logistic regression.
- Logistic regression maximum likelihood technique to classify the data.
- Let's see how Logistic regression uses MLE.
- MLE procedures have the advantage that they can exploit the properties of the estimation problem to deliver better efficiency and numerical stability.

MAXIMUM LIKELIHOOD ESTIMATION IN MACHINE LEARNING

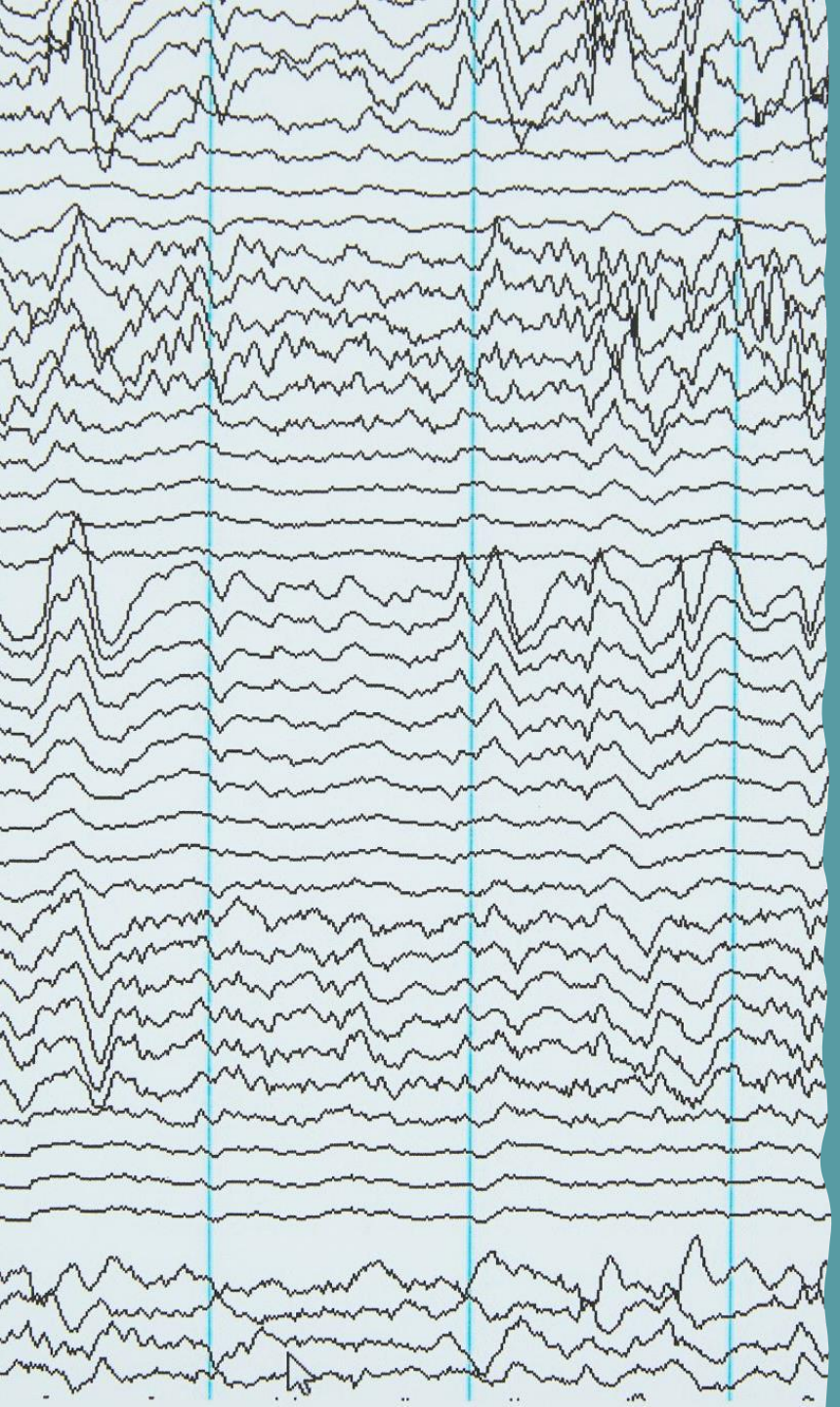
- These methods can often calculate explicit confidence intervals.
- The parameter “solver” of the logistic regression is used for selecting different solving strategies for classification for better MLE formulation.

THANK YOU

- **Expectation-Maximization Algorithm**

Machine Learning

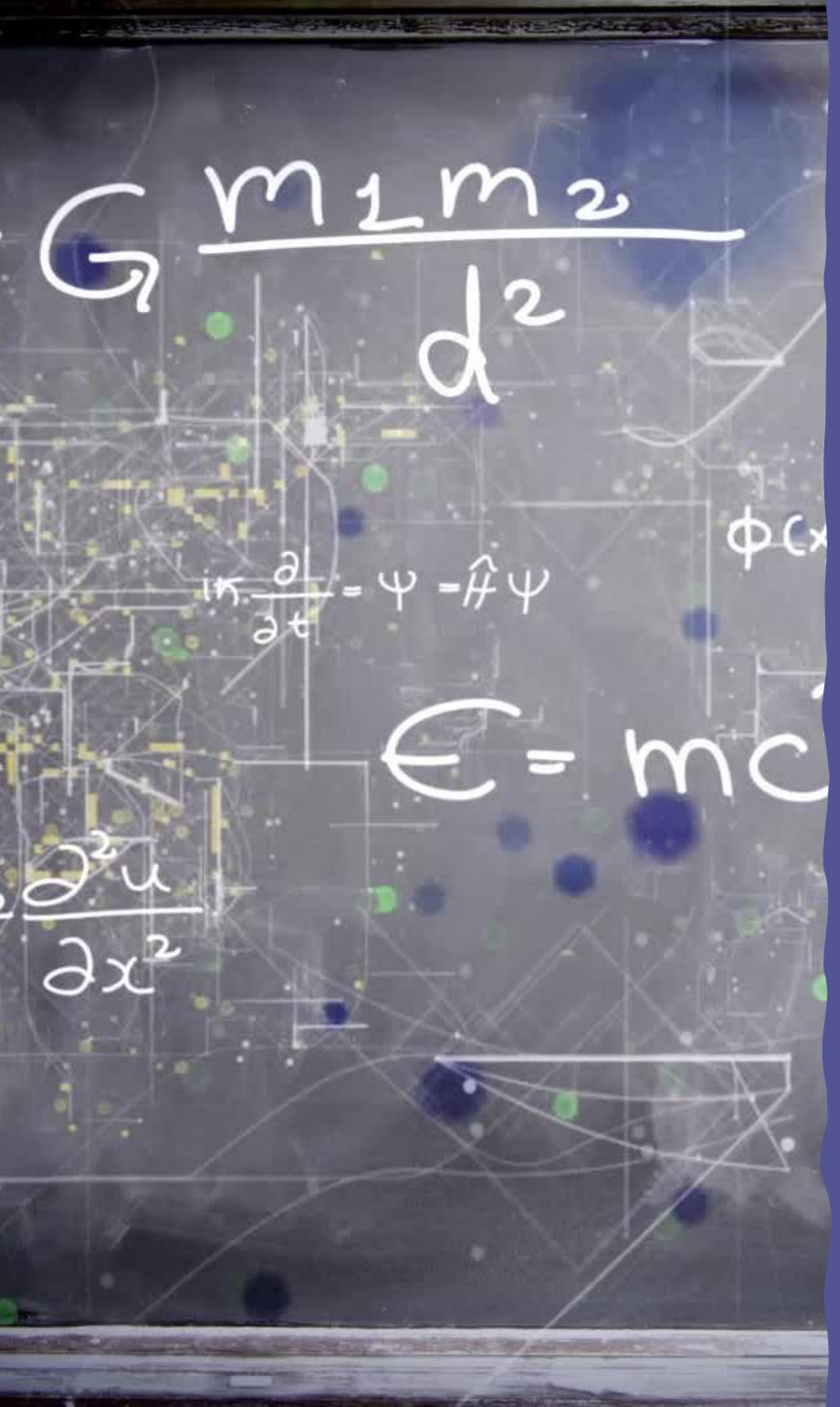
Dr. Jagendra Singh



WHAT IS AN EM ALGORITHM?

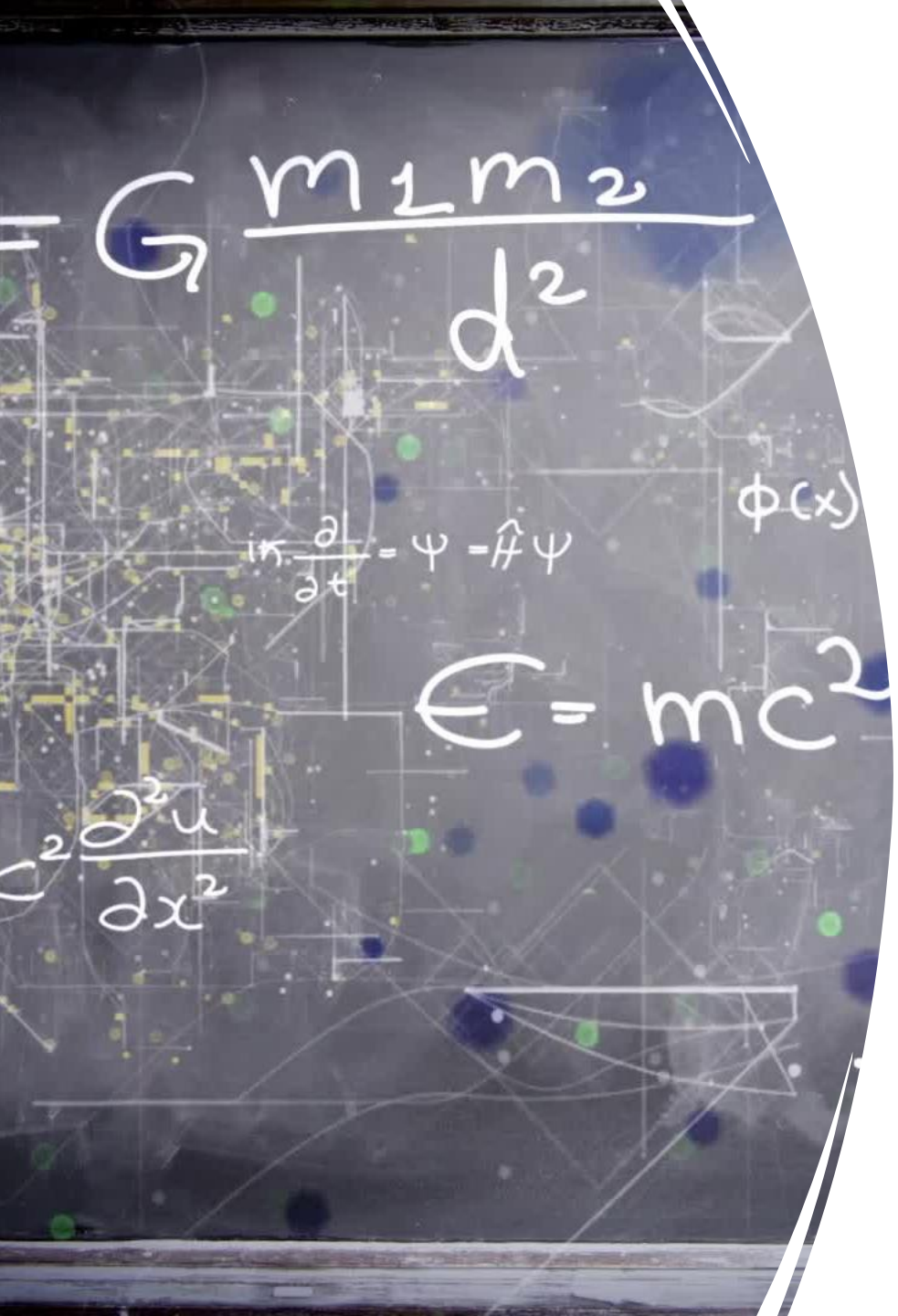
- The Expectation-Maximization (EM) algorithm is defined as the combination of various unsupervised machine learning algorithms, which is used to determine the **local maximum likelihood estimates (MLE)** or **maximum a posteriori estimates (MAP)** for unobservable variables in statistical models.

- It is a technique to find maximum likelihood estimation when the latent variables are present. It is also referred to as the **latent variable model.**



WHAT IS AN EM ALGORITHM?

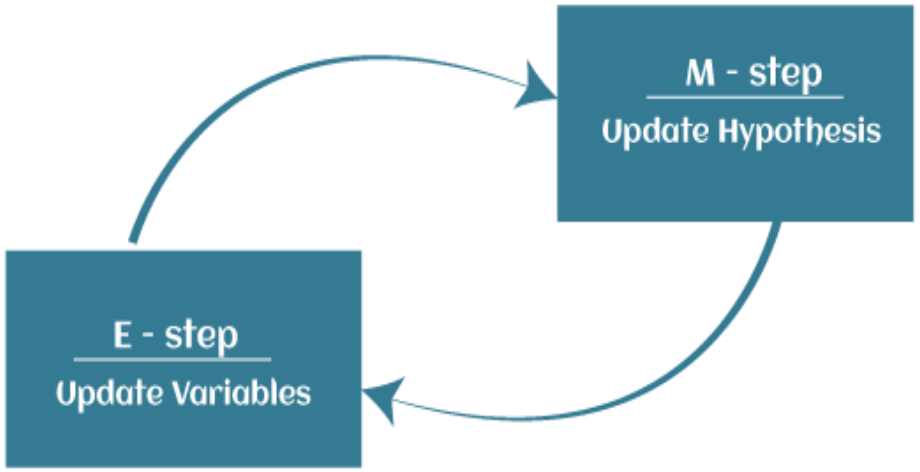
- A latent variable model consists of both observable and unobservable variables where observable can be predicted while unobserved are inferred from the observed variable.
- These unobservable variables are known as latent variables.
- It is used to predict values of parameters in instances where data is missing or unobservable for learning, and this is done until convergence of the values occurs.



EM ALGORITHM

- The EM algorithm is the combination of various unsupervised ML algorithms, such as the **k-means clustering algorithm**.
- Being an iterative approach, it consists of two modes. In the first mode, we estimate the missing or latent variables. Hence it is referred to as the **Expectation/estimation step (E-step**).
- Further, the other mode is used to optimize the parameters of the models so that it can explain the data more clearly. The second mode is known as the **maximization-step or M-step.**

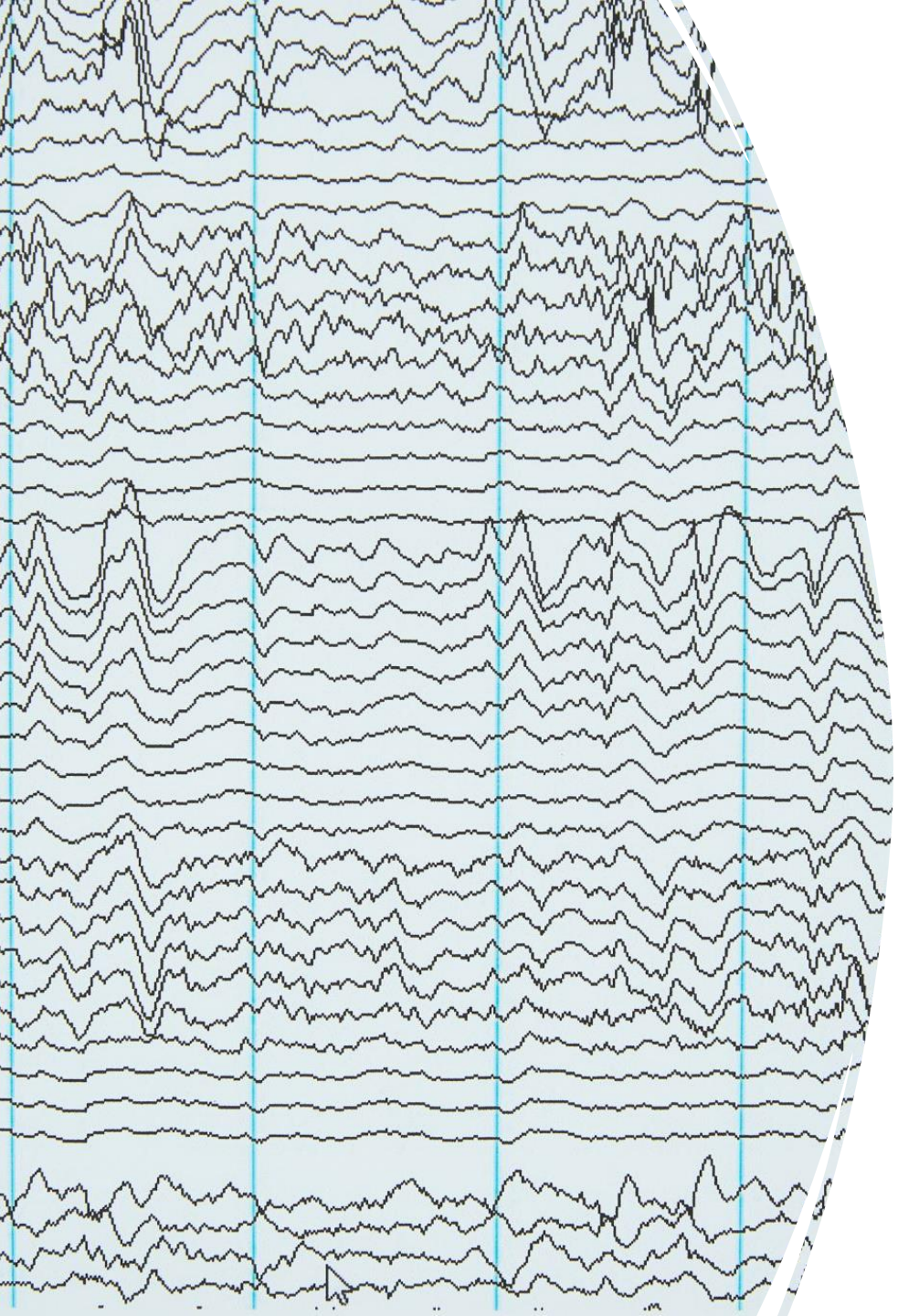
EM ALGORITHM



- **Expectation step (E - step):** It involves the estimation (guess) of all missing values in the dataset so that after completing this step, there should not be any missing value.
- **Maximization step (M - step):** This step involves the use of estimated data in the E-step and updating the parameters.
- **Repeat E-**step and M-step until the convergence of the values occurs.

EM ALGORITHM

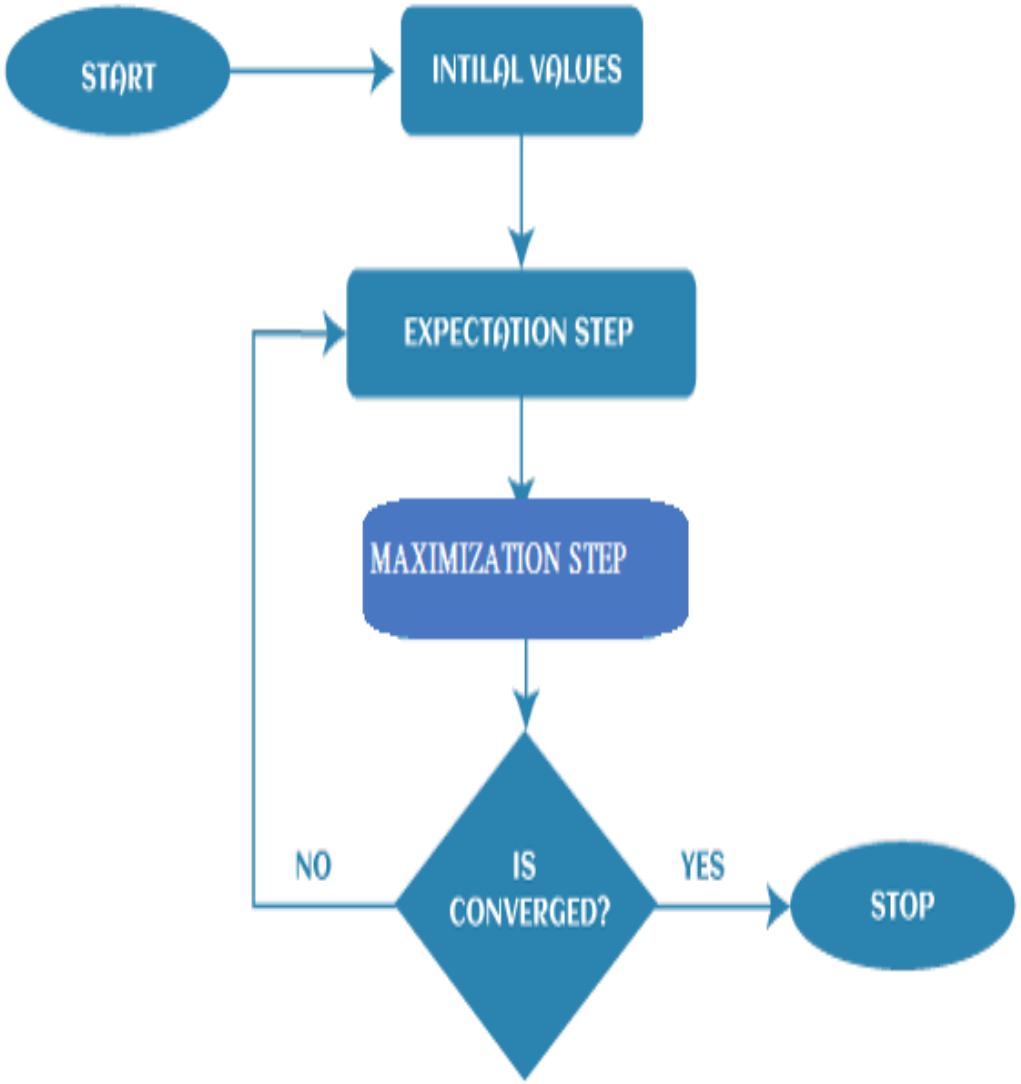
- The primary goal of the EM algorithm is to use the available observed data of the dataset to estimate the missing data of the latent variables.
- And then use that data to update the values of the parameters in the M-step.



WHAT IS CONVERGENCE IN THE EM ALGORITHM?

- ***Convergence is defined as the specific situation in probability based on intuition***, e.g., if there are two random variables that have very less difference in their probability, then they are known as converged.

- In other words, whenever the values of given variables are matched with each other, it is called convergence.



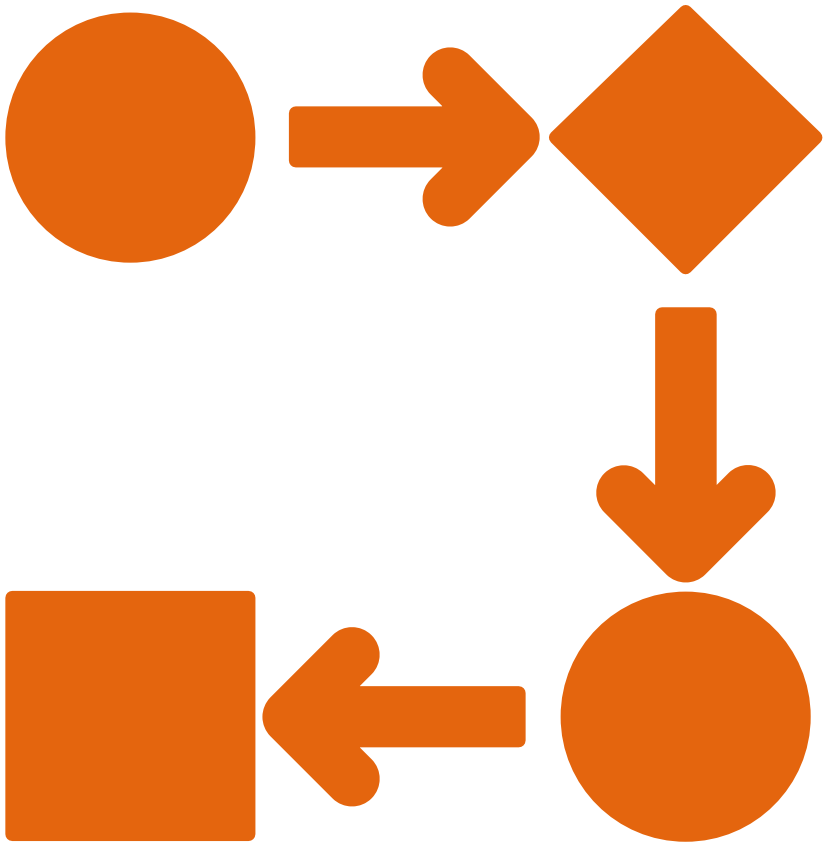
STEPS IN EM ALGORITHM

- The EM algorithm is completed mainly in 4 steps, which include ***Initialization Step, Expectation Step, Maximization Step, and convergence Step***.
- These steps are explained as follows:

STEPS IN EM ALGORITHM

- **1st Step:** The very first step is to initialize the parameter values. Further, the system is provided with incomplete observed data with the assumption that data is obtained from a specific model.

- **2nd Step:** This step is known as Expectation or E-Step, which is used to estimate or guess the values of the missing or incomplete data using the observed data. Further, E-step primarily updates the variables.

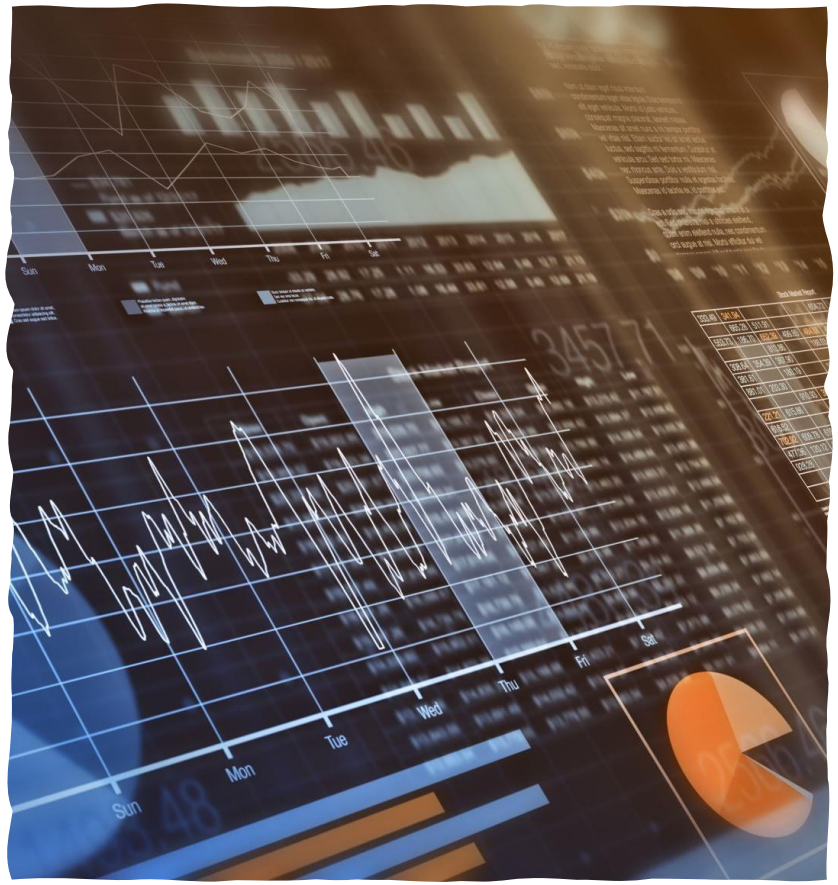


STEPS IN EM ALGORITHM

- **3rd Step:** This step is known as Maximization or M-step, where we use complete data obtained from the 2nd step to update the parameter values. Further, M-step primarily updates the hypothesis.

- **4th step:** The last step is to check if the values of latent variables are converging or not. If it gets "yes", then stop the process; else, repeat the process from step 2 until the convergence occurs.

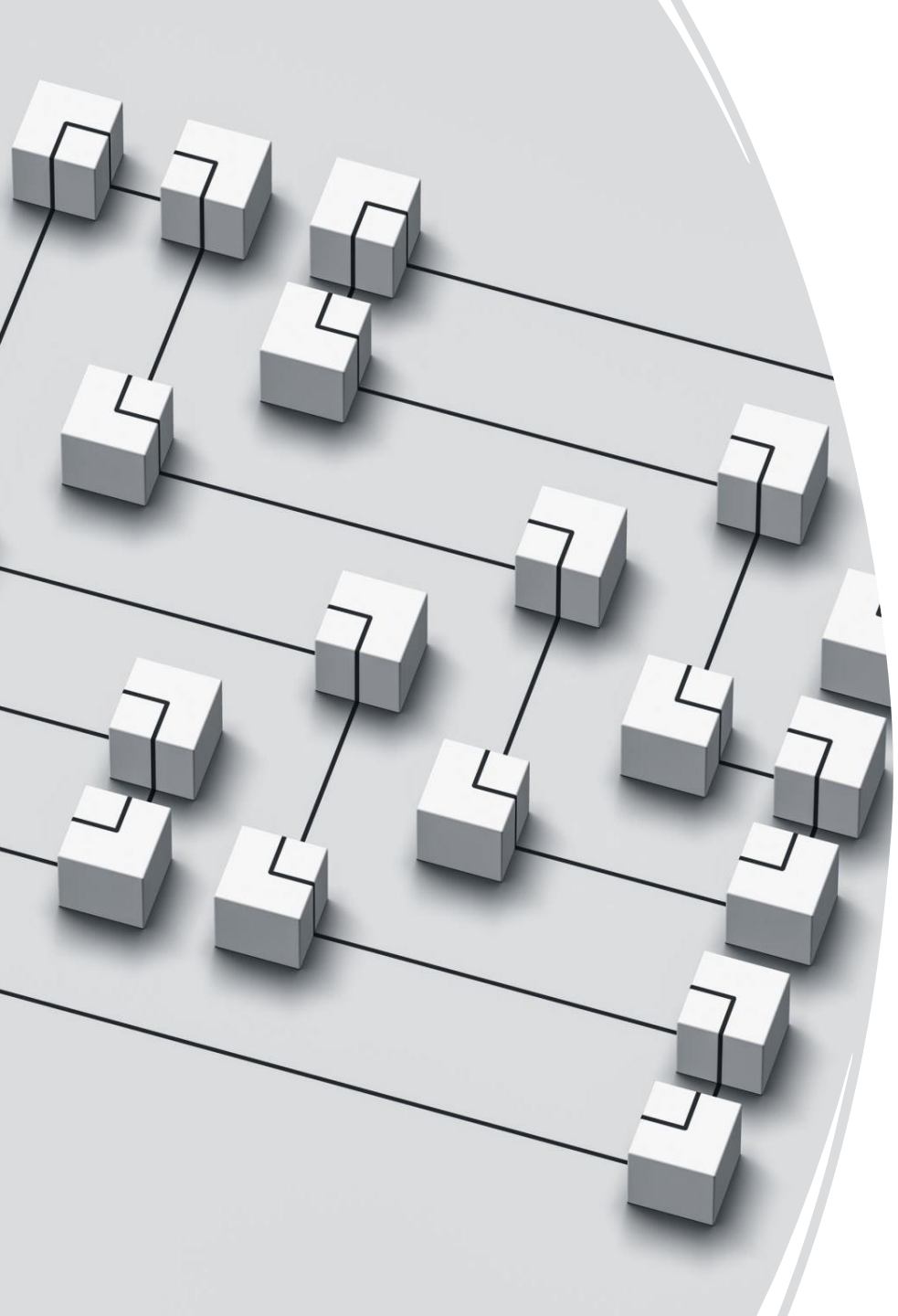
APPLICATIONS OF EM ALGORITHM



1. The primary aim of the EM algorithm is to estimate the missing data in the latent variables through observed data in datasets.
2. The EM algorithm or latent variable model has a broad range of real-life applications in machine learning. These are as follows:
   - The EM algorithm is applicable in data clustering in machine learning.
   - It is often used in computer vision and NLP (Natural language processing).
   - It is used to estimate the value of the parameter in mixed models such as the **Gaussian Mixture Model** and quantitative genetics.

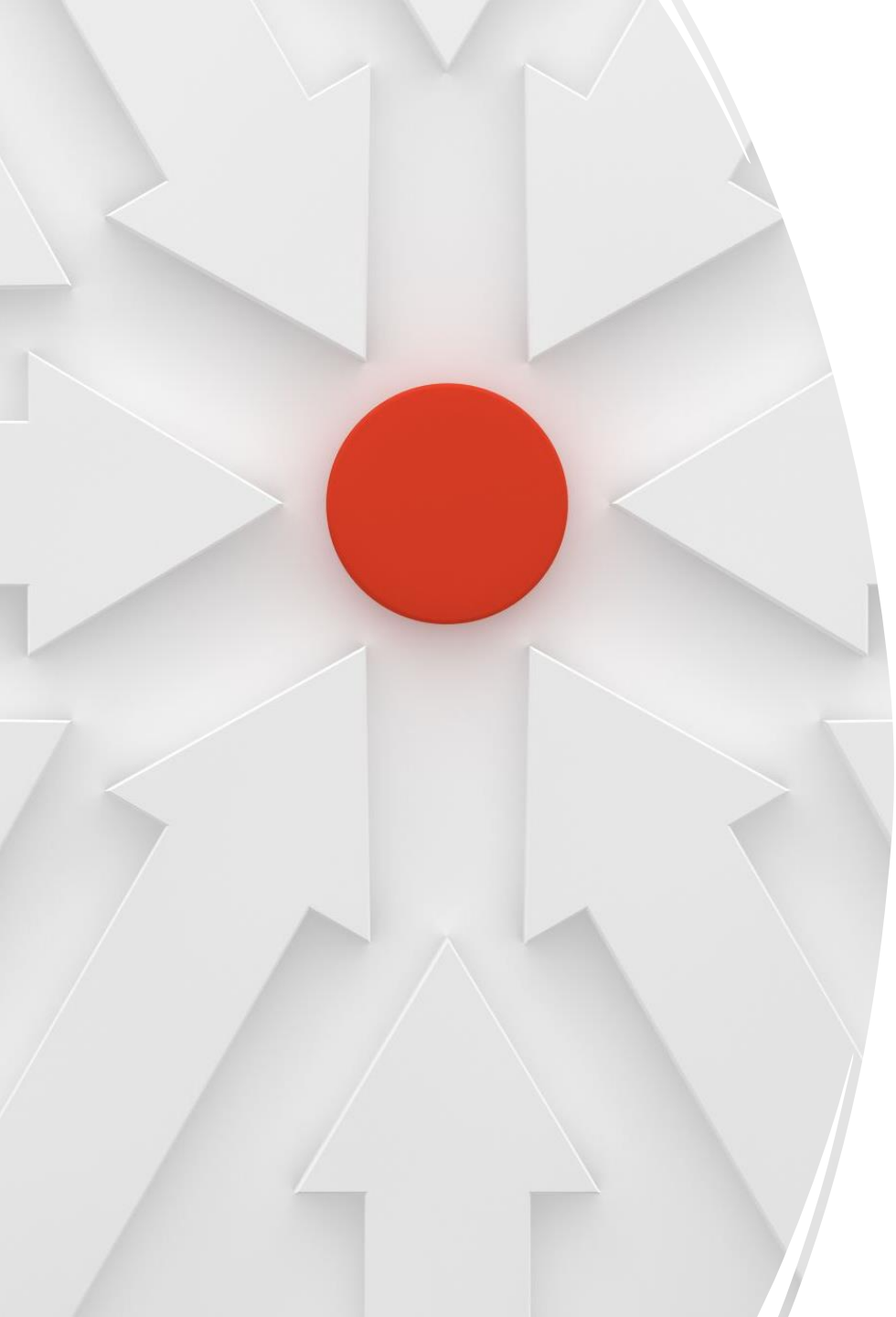
APPLICATIONS OF EM ALGORITHM

- It is also used in psychometrics for estimating item parameters and latent abilities of item response theory models.
- It is also applicable in the medical and healthcare industry, such as in image reconstruction and structural engineering.
- It is used to determine the Gaussian density of a function.



ADVANTAGES OF EM ALGORITHM

- It is very easy to implement the first two basic steps of the EM algorithm in various machine learning problems, which are E-step and M- step.
- It is mostly guaranteed that likelihood will enhance after each iteration.
- It often generates a solution for the M-step in the closed form.



DISADVANTAGES OF EM ALGORITHM

- The convergence of the EM algorithm is very slow.
- It can make convergence for the local optima only.
- It takes both forward and backward probability into consideration. It is opposite to that of numerical optimization, which takes only forward probabilities.

THANK YOU

- **Parametric Methods vs Non-Parametric Methods**

Machine Learning

Dr. Jagendra Singh

PARAMETRIC METHODS

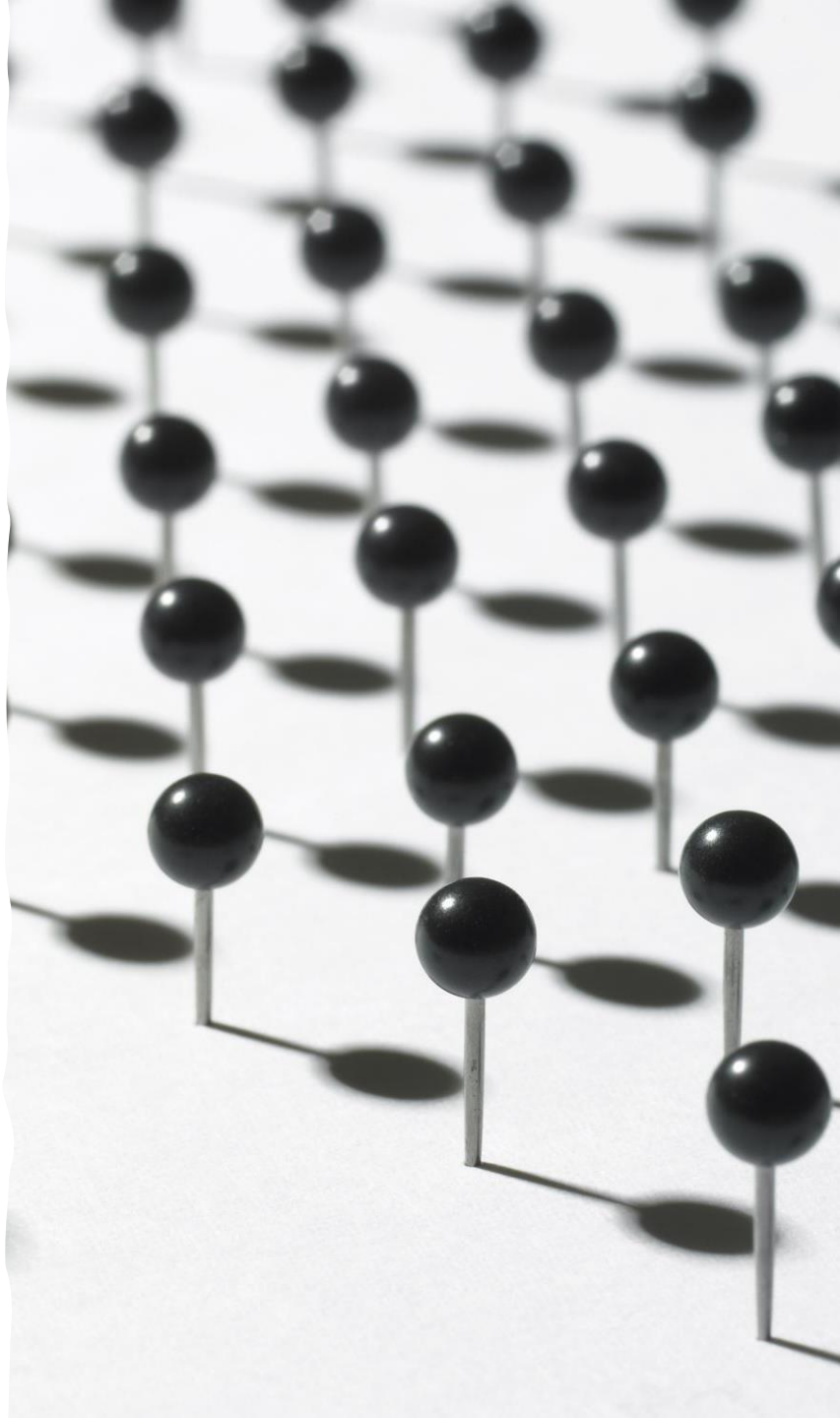
- The basic idea behind the parametric method is that there is a set of fixed parameters that uses to determine a probability model that is used in Machine Learning as well.
- Parametric methods are those methods for which we priory knows that the population is normal.
- Parameters for using the normal distribution is as follows:

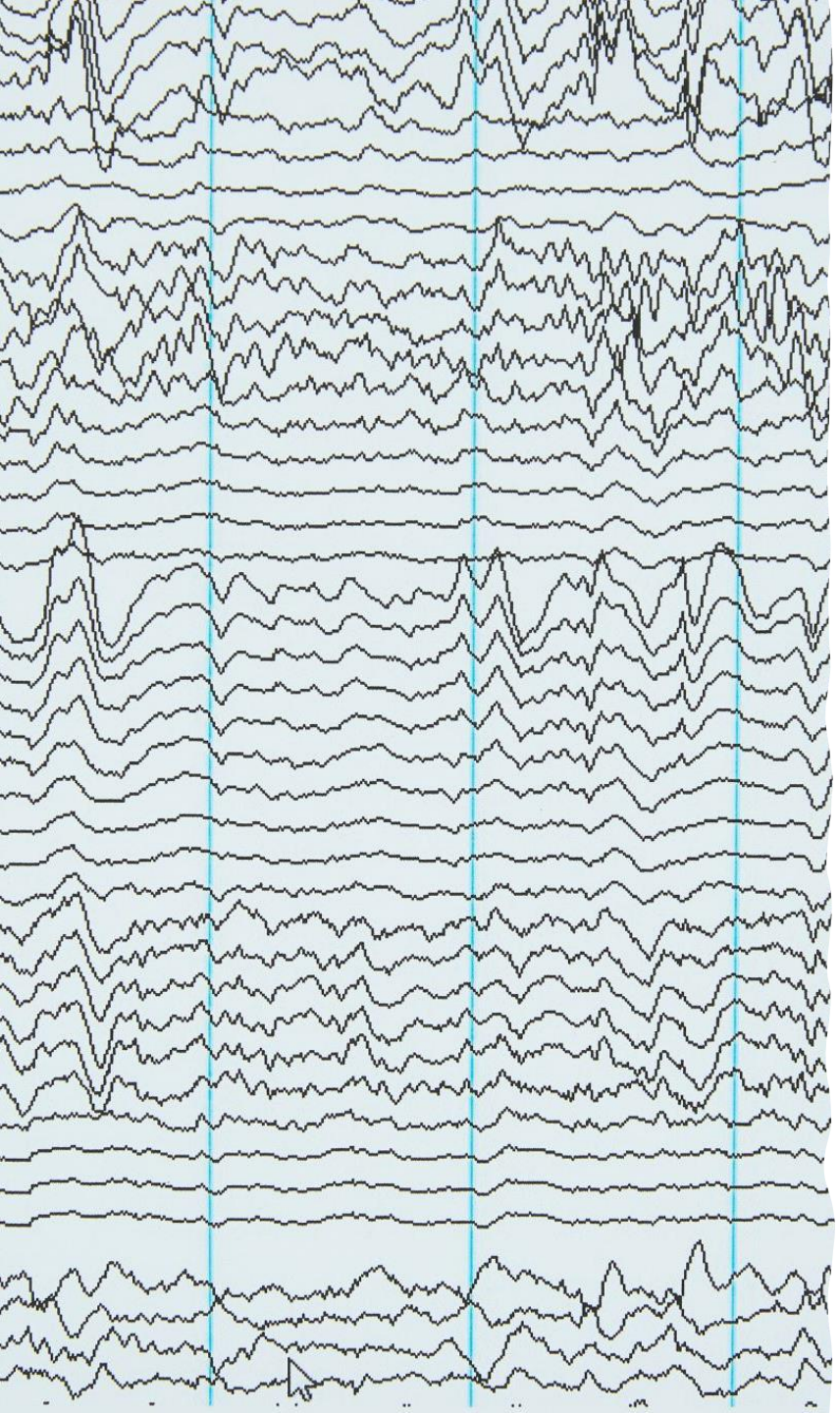
. Mean

. Standard Deviation

PARAMETRIC METHODS

- Eventually, the classification of a method to be parametric is completely depends on the presumptions that are made about a population.





PARAMETRIC METHODS

- There are many parametric methods available some of them are:
- Confidence interval used for – population mean along with known standard deviation.
- The confidence interval is used for – population means along with the unknown standard deviation.
- The confidence interval for population variance.
- The confidence interval for the difference of two means, with unknown standard deviation.

PARAMETRIC METHODS: MEAN

- Suppose that the entire population of interest is eight students in a particular class.
- For a finite set of numbers, the population standard deviation is found by taking the square root of the average of the squared deviations of the values subtracted from their average value.
- The marks of a class of eight students (that is, a statistical population) are the following eight values:

PARAMETRIC METHODS: MEAN,

$$2,\ 4,\ 4,\ 4,\ 5,\ 5,\ 7,\ 9.$$

These eight data points have the mean (average) of 5:

$$\mu = \frac{2+4+4+4+5+5+7+9}{8} = \frac{40}{8} = 5.$$

First, calculate the deviations of each data point from the mean, and square the result of each:

$$(2-5)^2 = (-3)^2 = 9 \qquad (5-5)^2 = 0^2 = 0$$
$$(4-5)^2 = (-1)^2 = 1 \qquad (5-5)^2 = 0^2 = 0$$
$$(4-5)^2 = (-1)^2 = 1 \qquad (7-5)^2 = 2^2 = 4$$
$$(4-5)^2 = (-1)^2 = 1 \qquad (9-5)^2 = 4^2 = 16.$$

VARIANCE, STANDARD DEVIATION

The variance is the mean of these values:

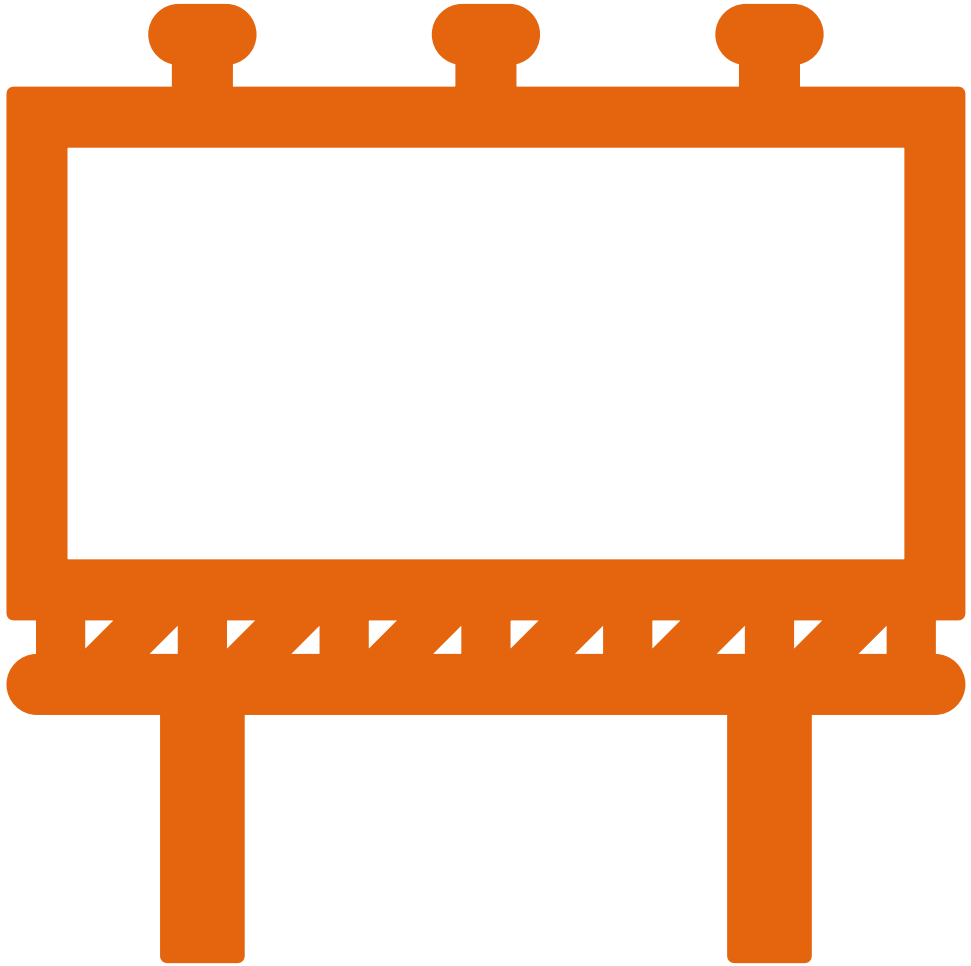
$$\sigma^2 = \frac{9+1+1+1+0+0+4+16}{8} = \frac{32}{8} = 4.$$

and the *population* standard deviation is equal to the square root of the variance:

$$\sigma = \sqrt{4} = 2.$$

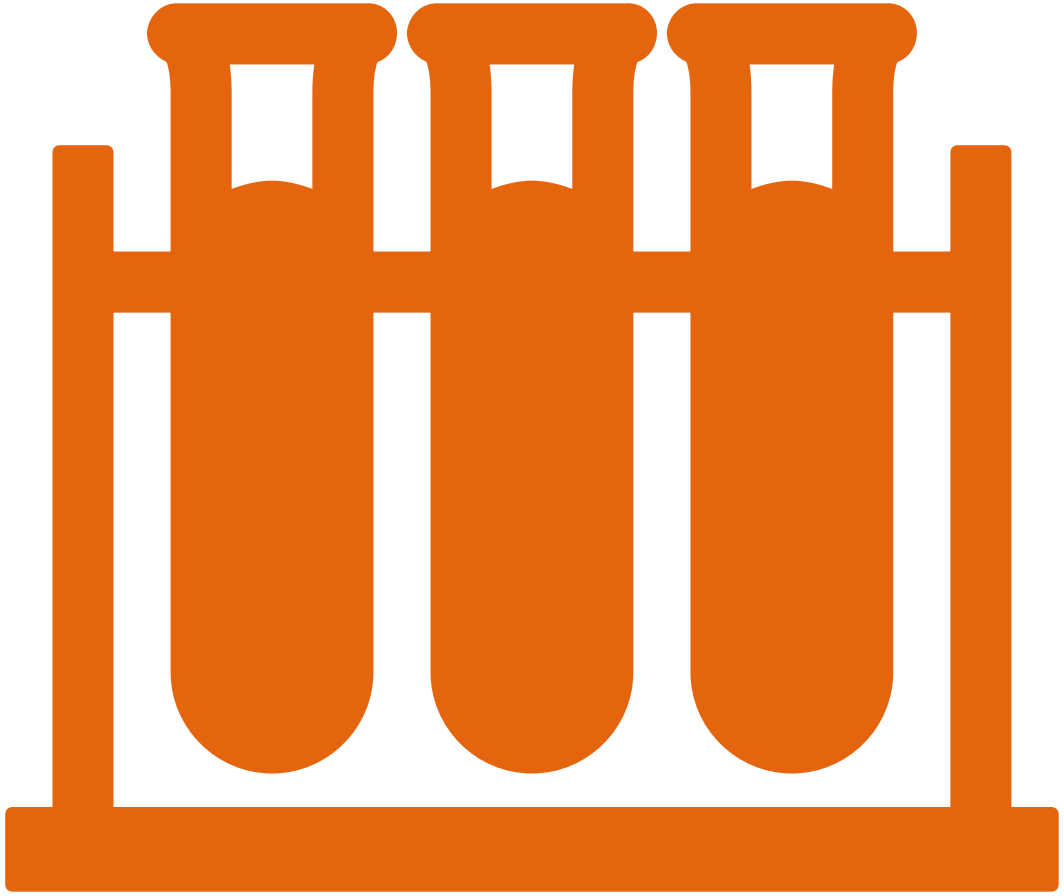
NONPARAMETRIC METHODS

- The basic idea behind the parametric method is no need to make any assumption of parameters for the given population or the population we are studying.
- In fact, the methods don't depend on the population. Here there is no fixed set of parameters are available, and also there is no distribution (normal distribution, etc.) of any kind is available for use.
- This is also the reason that nonparametric methods are also referred to as distribution-free methods.



NONPARAMETRIC METHODS

- Nowadays Non-parametric methods are gaining popularity and an impact of influence some reasons behind this fame is:
- The main reason is that there is no need to be mannered while using parametric methods.
- The second important reason is that we do not need to make more and more assumptions about the population given (or taken) on which we are working on.
- Most of the nonparametric methods available are very easy to apply and to understand also i.e. the complexity is very low.



NONPARAMETRIC METHODS

- There are many nonparametric methods are available today but some of them are as follows:

- Spearman correlation test
- Sign test for population means
- U-test for two independent means

Parametric Methods	Non-Parametric Methods
Parametric Methods uses a fixed number of parameters to build the model.	Non-Parametric Methods use the flexible number of parameters to build the model.
Parametric analysis is to test group means.	A non-parametric analysis is to test medians.
It is applicable only for variables.	It is applicable for both – Variable and Attribute.
It always considers strong assumptions about data.	It generally fewer assumptions about data.
Parametric Methods require lesser data than Non-Parametric Methods.	Non-Parametric Methods requires much more data than Parametric Methods.
Parametric methods assumed to be a normal distribution.	There is no assumed distribution in non-parametric methods.

DIFFERENCE

- Difference between Parametric and Non-Parametric Methods are as:

Here when we use parametric methods then the result or outputs generated can be easily affected by outliers.	When we use non-parametric methods then the result or outputs generated cannot be seriously affected by outliers.
Parametric Methods can perform well in many situations but its performance is at peak (top) when the spread of each group is different.	Similarly, Non-Parametric Methods can perform well in many situations but its performance is at peak (top) when the spread of each group is the same.
Parametric methods have more statistical power than Non-Parametric methods.	Non-parametric methods have less statistical power than Parametric methods.

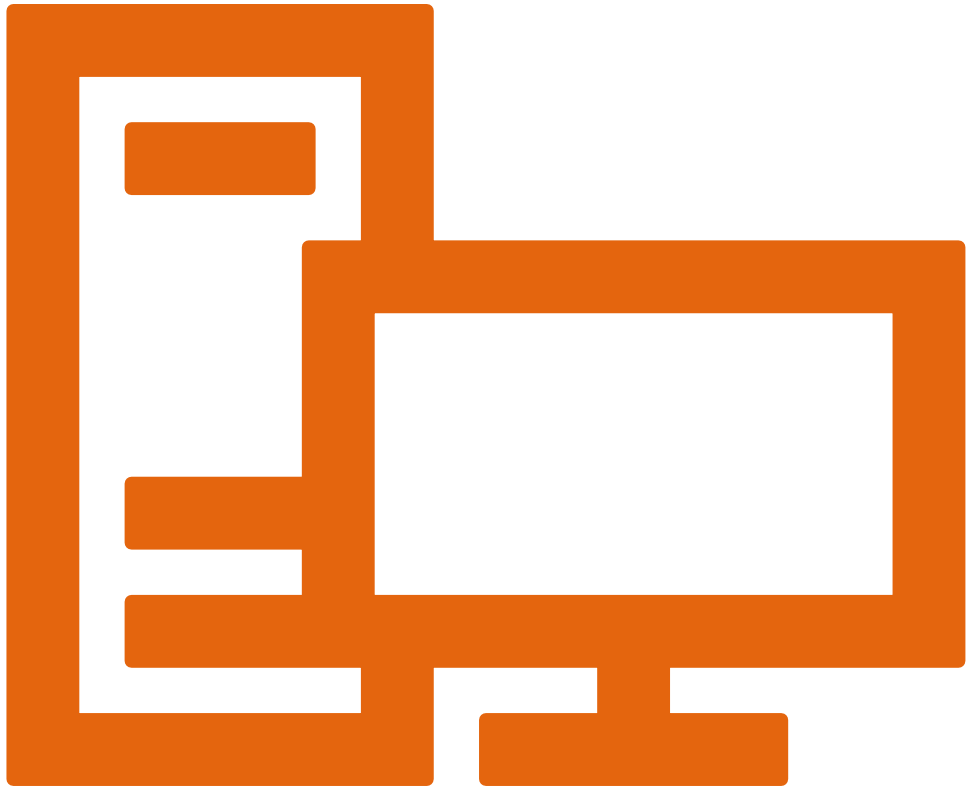
DIFFERENCE

- Difference between Parametric and Non-Parametric Methods are as:

THANK YOU

- **Classification Algorithm in Machine Learning**

Dr. Jagendra Singh

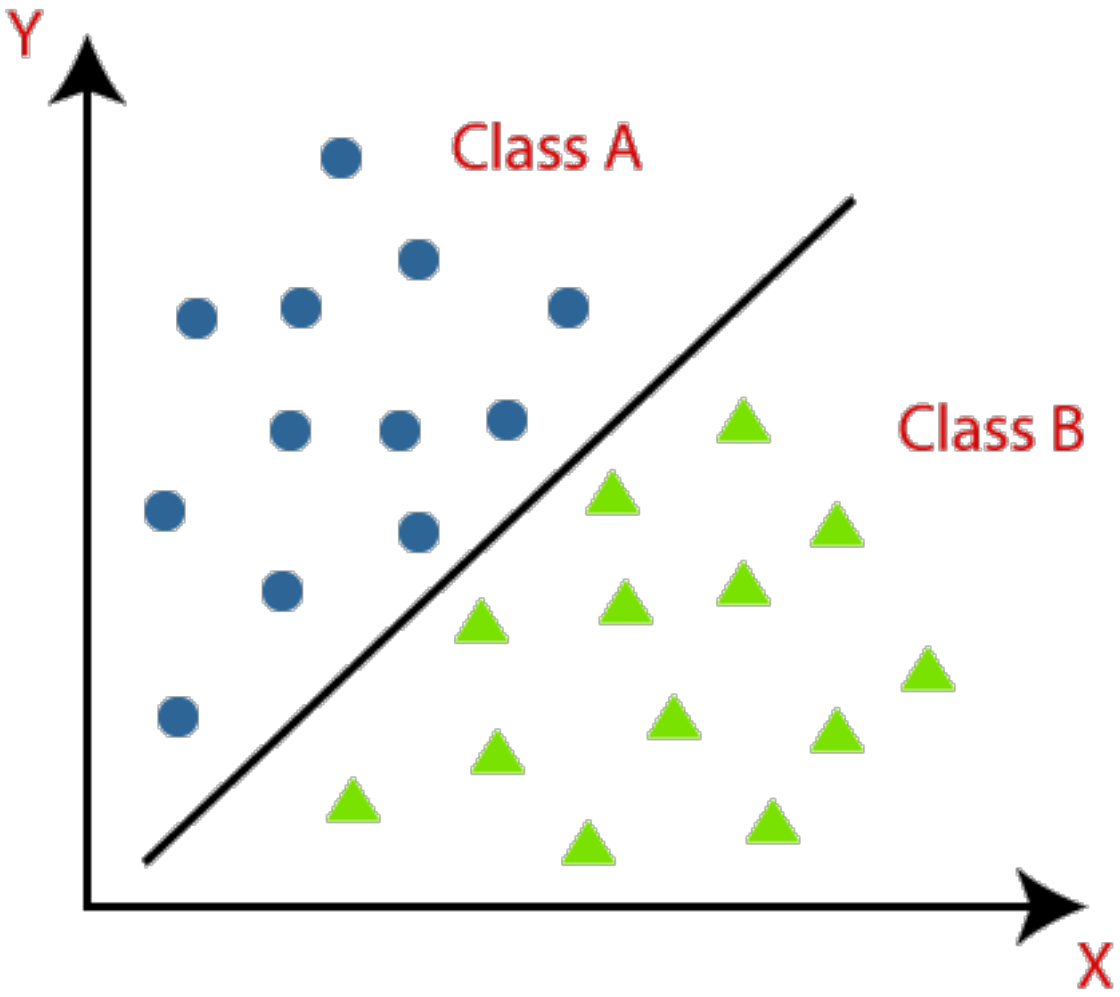


Classification Algorithm

- As we know, the Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms.
- In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

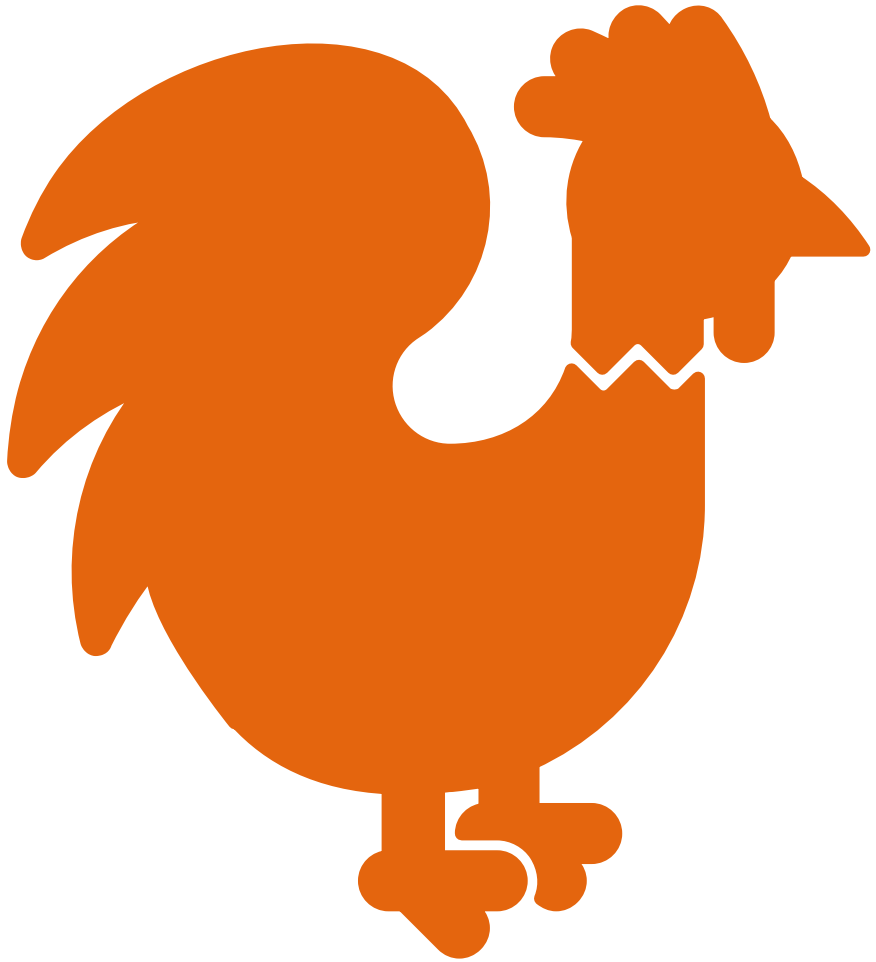
CLASSIFICATION ALGORITHM

- In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog,** etc.
- Classes can be called as targets/labels or categories
- Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc.



CLASSIFICATION ALGORITHM

- The best example of an ML classification algorithm is **Email Spam Detector**.
- Classification algorithms can be better understood using the below diagram.
- In this diagram, there are two classes, class A and Class B. These classes have features that are similar to each other and dissimilar to other classes.



CLASSIFICATION ALGORITHM

- The algorithm which implements the classification on a dataset is known as a classifier. There are two types of Classifications:
  - **Binary Classifier:** If the classification problem has only two possible outcomes, then it is called as Binary Classifier.
    Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
  - **Multi-class Classifier:** If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
    Example: Classifications of types of crops, Classification of types of music.

LEARNERS IN CLASSIFICATION PROBLEMS

In the classification problems, there are two types of learners:

- Lazy Learners: Lazy Learner firstly stores the training dataset and wait until it receives the test dataset. In Lazy learner case, classification is done on the basis of the most related data stored in the training dataset. It takes less time in training but more time for predictions.
  Example: K-NN algorithm, Case-based reasoning
- Eager Learners: Eager Learners develop a classification model based on a training dataset before receiving a test dataset. Opposite to Lazy learners, Eager Learner takes more time in learning, and less time in prediction. Example: Decision Trees, Naïve Bayes, ANN.

TYPES OF ML CLASSIFICATION ALGORITHMS

Classification Algorithms can be further divided into the Mainly two category:

Linear Models

- Logistic Regression
- Support Vector Machines

Non-linear Models

- K-Nearest Neighbours
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

EVALUATING A CLASSIFICATION MODEL

Once our model is completed, it is necessary to evaluate its performance; either it is a Classification or Regression model. So for evaluating a Classification model, we have the following ways:

1. Log Loss or Cross-Entropy Loss:

- It is used for evaluating the performance of a classifier, whose output is a probability value between the 0 and 1.
- For a good binary Classification model, the value of log loss should be near to 0.
- The value of log loss increases if the predicted value deviates from the actual value.
- The lower log loss represents the higher accuracy of the model.

EVALUATING A CLASSIFICATION MODEL

- Cross-entropy loss is used when adjusting model weights during training. The aim is to minimize the loss, i.e, the smaller the loss the better the model. A perfect model has a cross-entropy loss of 0.
- For Binary classification, cross-entropy can be calculated as:

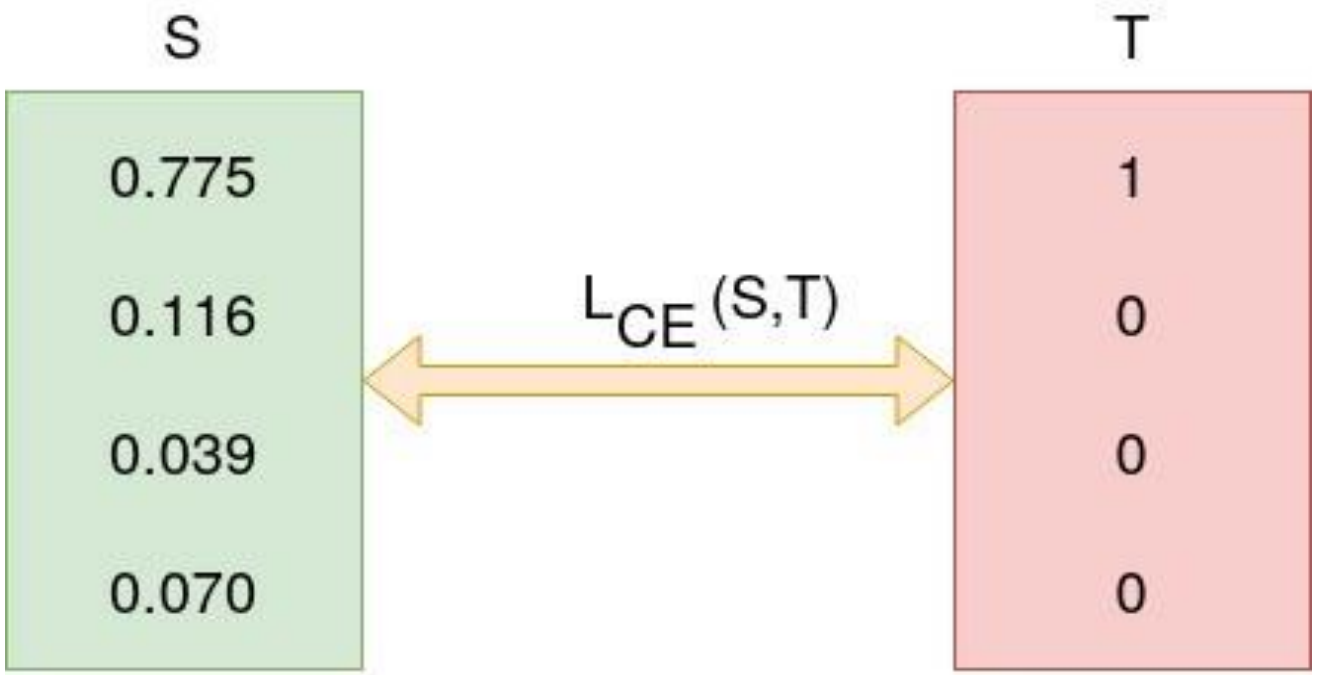
$$L_{\text{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

EVALUATING A CLASSIFICATION MODEL

- **Example**

- Consider the classification problem with the following Softmax probabilities (S) and the labels (T). The objective is to calculate for cross-entropy loss given these information.



EVALUATING A CLASSIFICATION MODEL

- The categorical cross-entropy is computed as follows:

$$
\begin{aligned}
L_{CE} &= -\sum_{i=1} T_i \log(S_i) \\
&= -\left[1 \log_2(0.775) + 0 \log_2(0.126) + 0 \log_2(0.039) + 0 \log_2(0.070)\right] \\
&= -\log_2(0.775) \\
&= 0.3677
\end{aligned}
$$

EVALUATING A CLASSIFICATION MODEL: CONFUSION MATRIX

	Actual Positive	**Actual Negative**
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

$$\text{Accuracy} = \frac{TP+TN}{\text{Total Population}}$$

- The confusion matrix provides us a matrix/table as output and describes the performance of the model.
- It is also known as the error matrix.
- The matrix consists of predictions result in a summarized form, which has a total number of correct predictions and incorrect predictions. The matrix looks like as this table:

EVALUATING A CLASSIFICATION MODEL: AUC-ROC CURVE

- ROC curve stands for **Receiver Operating Characteristics Curve** and AUC stands for **Area Under the Curve**.
- It is a graph that shows the performance of the classification model at different thresholds.
- To visualize the performance of the multi-class classification model, we use the AUC-ROC Curve.
- The ROC curve is plotted with TPR and FPR, where TPR (True Positive Rate) on Y-axis and FPR(False Positive Rate) on X-axis.

USE CASES OF CLASSIFICATION ALGORITHMS

- Classification algorithms can be used in different places. Below are some popular use cases of Classification Algorithms:
  - Email Spam Detection
  - Speech Recognition
  - Identifications of Cancer tumor cells.
  - Drugs Classification
  - Biometric Identification, etc.

THANK YOU

- **MinMax Algorithm**

Machine Learning

Dr. Jagendra Singh

MINI-MAX ALGORITHM

- Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory.
- It provides an optimal move for the player assuming that opponent is also playing optimally.
- Mini-Max algorithm uses recursion to search through the game-tree.
- Min-Max algorithm is mostly used for game playing in AI. Such as Chess, Checkers, tic-tac-toe, go, and various tow-players game.
- This Algorithm computes the minimax decision for the current state.

MINI-MAX ALGORITHM

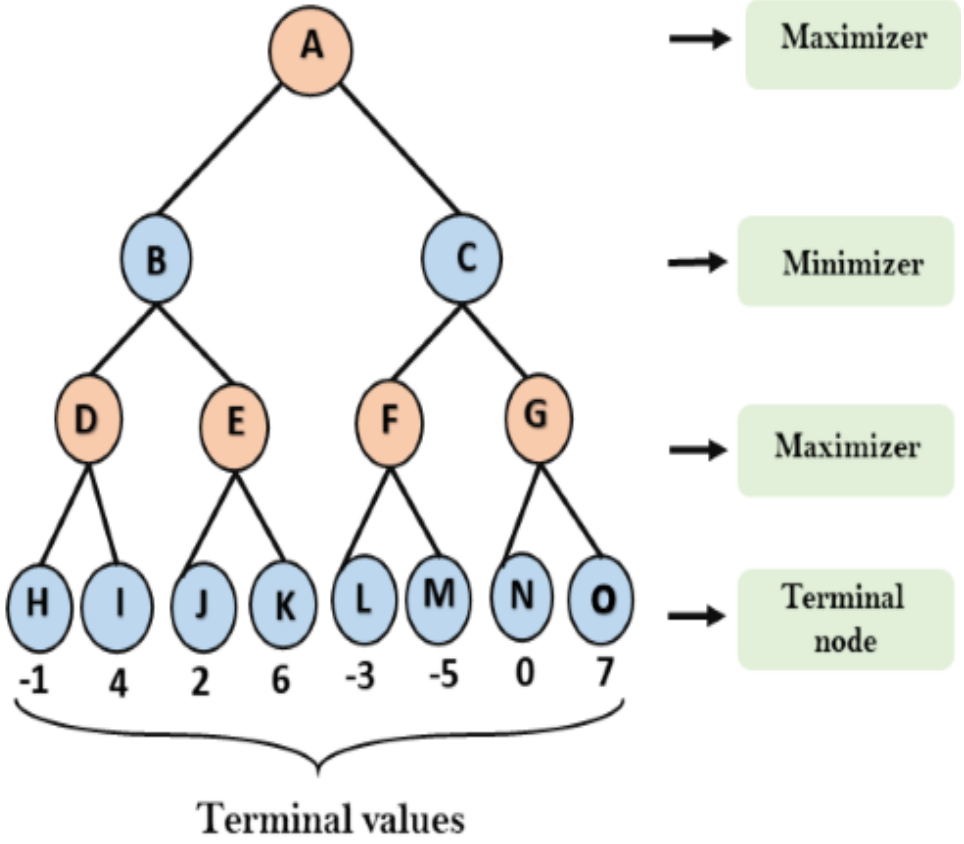
- In this algorithm two players play the game, one is called MAX and other is called MIN.
- Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit.
- Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value.

MINI-MAX ALGORITHM

- The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree.
- The minimax algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.

WORKING OF MIN-MAX ALGORITHM

- The working of the minimax algorithm can be easily described using an example. Below we have taken an example of game-tree which is representing the two-player game.
- In this example, there are two players one is called Maximizer and other is called Minimizer.
- Maximizer will try to get the Maximum possible score, and Minimizer will try to get the minimum possible score.



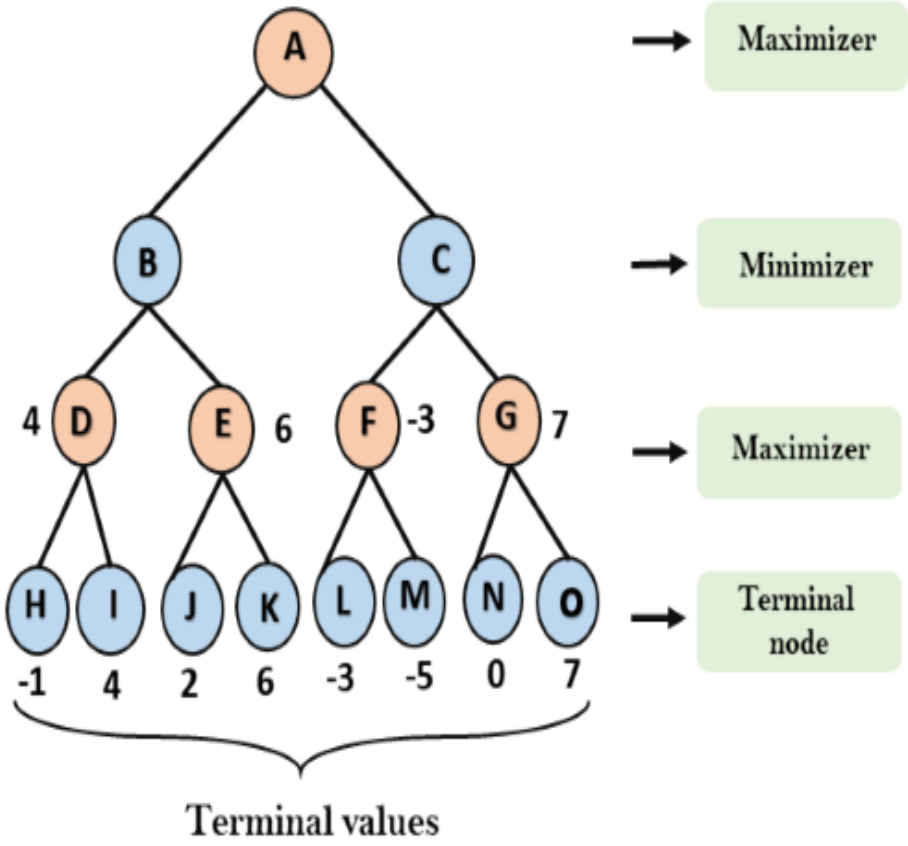
WORKING OF MIN-MAX ALGORITHM

Following are the main steps involved in solving the two-player game tree:

- **Step-1:** In the first step, the algorithm generates the entire game-tree and apply the utility function to get the utility values for the terminal states.
- In the below tree diagram, let's take A is the initial state of the tree.
- Suppose maximizer takes first turn which has worst-case initial value =- infinity, and minimizer will take next turn which has worst-case initial value = +infinity.

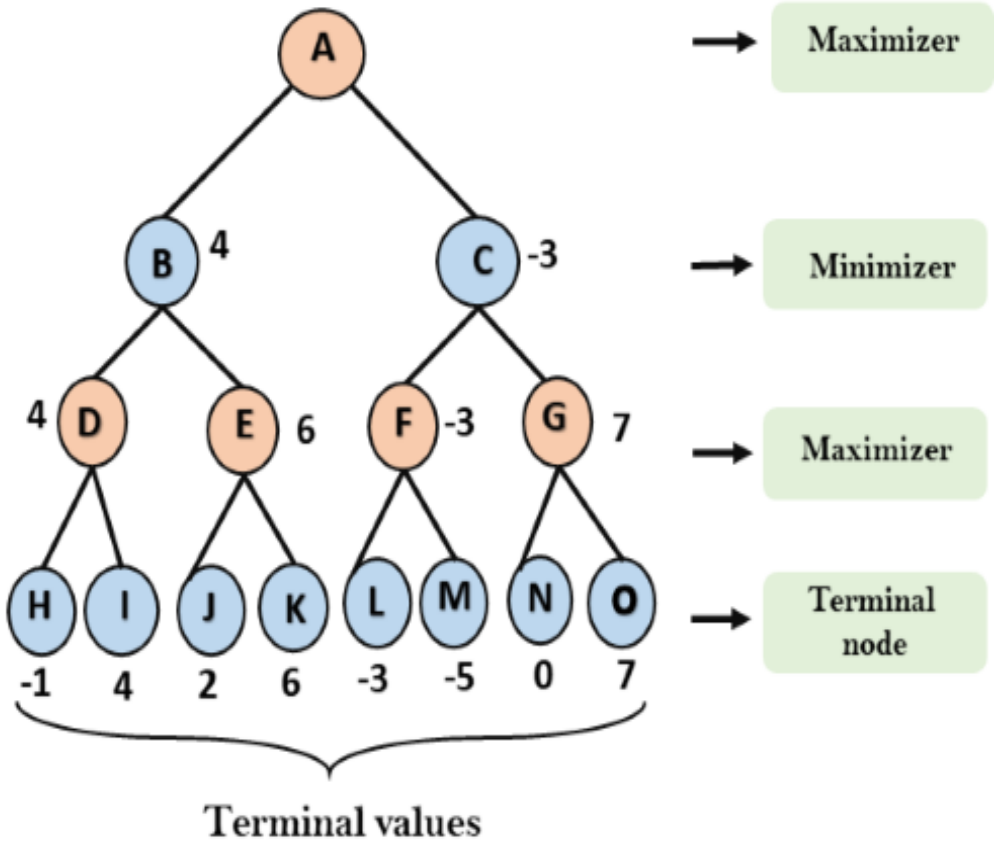
WORKING OF MIN-MAX ALGORITHM

- **Step 2:** Now, first we find the utilities value for the Maximizer, its initial value is -∞, so we will compare each value in terminal state with initial value of Maximizer and determines the higher nodes values. It will find the maximum among the all.
  - For node D max(-1,- -∞) => max(-1,4)= 4
  - For Node E max(2, -∞) => max(2, 6)= 6
  - For Node F max(-3, -∞) => max(-3,-5) = -3
  - For node G max(0, -∞) = max(0, 7) = 7



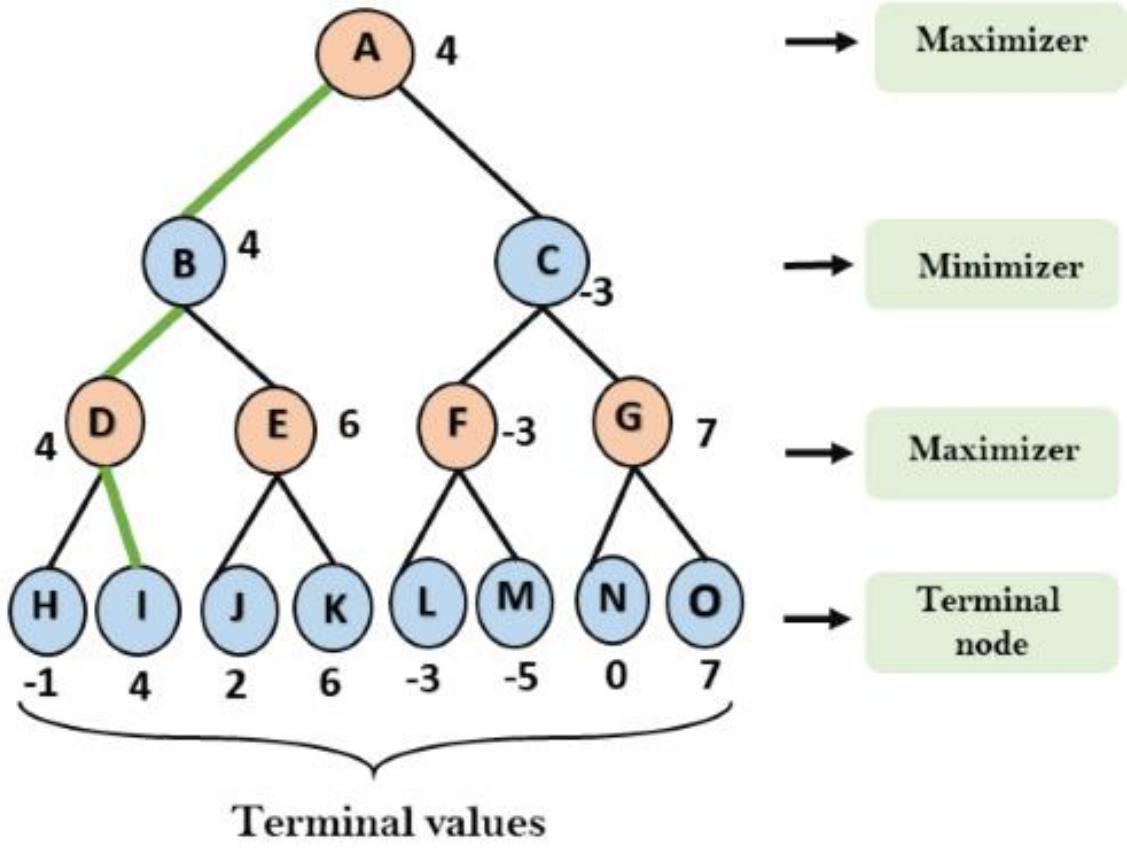
WORKING OF MIN-MAX ALGORITHM

- **Step 3:** In the next step, it's a turn for minimizer, so it will compare all nodes value with +∞, and will find the 3rd layer node values.
  - For node B= min(4,6) = 4
  - For node C= min (-3, 7) = -3



WORKING OF MIN-MAX ALGORITHM

- **Step 4:** Now it's a turn for Maximizer, and it will again choose the maximum of all nodes value and find the maximum value for the root node.
- In this game tree, there are only 4 layers, hence we reach immediately to the root node, but in real games, there will be more than 4 layers.
  - For node A max(4, -3)= 4



LIMITATION OF THE MINIMAX ALGORITHM

- The main drawback of the minimax algorithm is that it gets really slow for complex games such as Chess, go, etc.
- This type of games has a huge branching factor, and the player has lots of choices to decide
- This limitation of the minimax algorithm can be improved from **alpha-beta pruning** which we have discussed in the next topic.

THANK YOU

- **Support Vector Machine Algorithm**

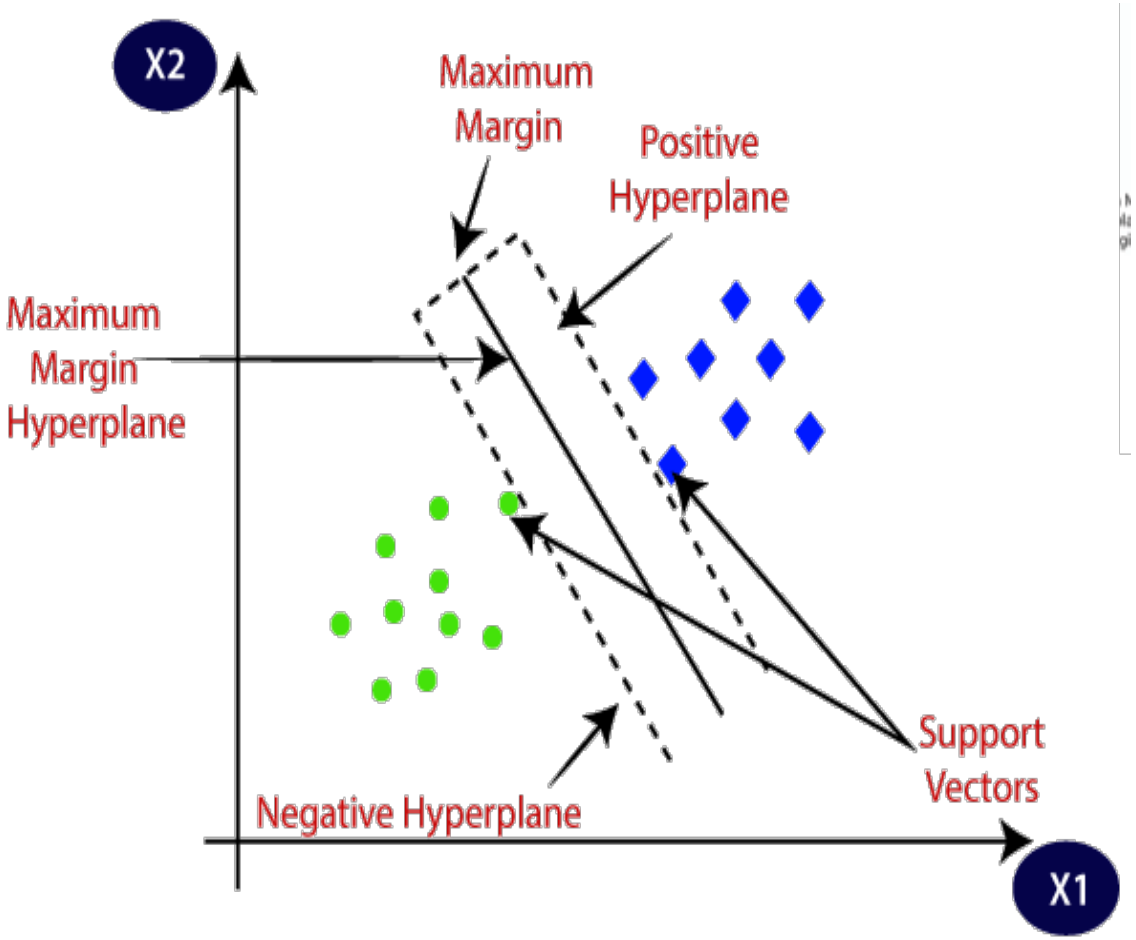
Machine Learning

Dr. Jagendra Singh

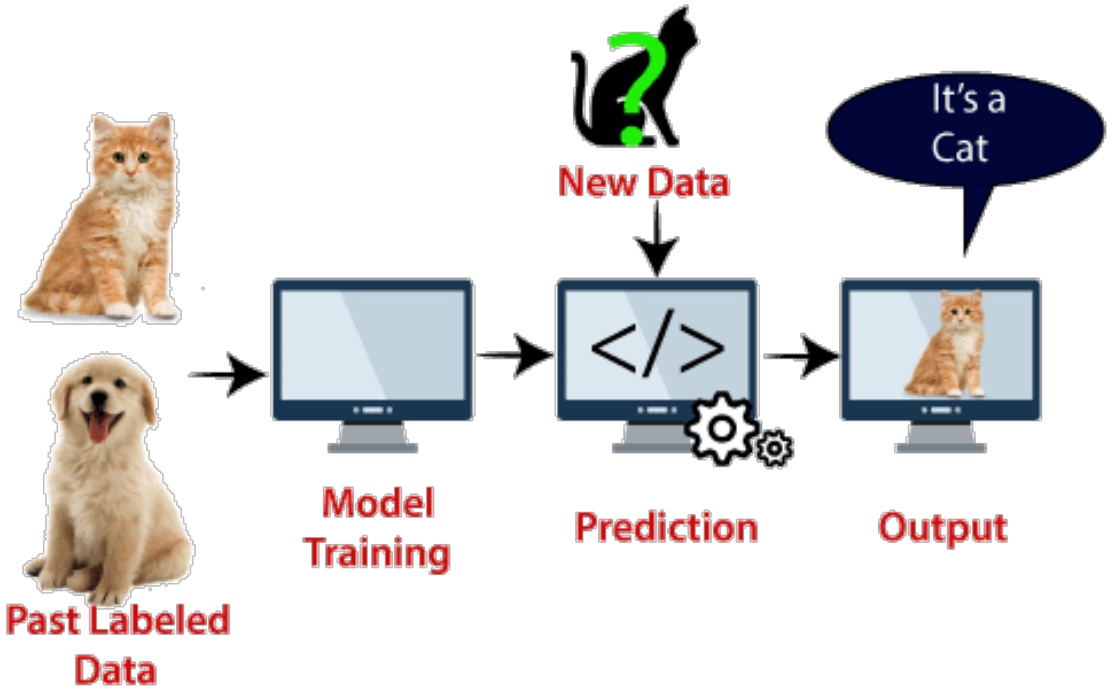
SUPPORT VECTOR MACHINE ALGORITHM

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

SUPPORT VECTOR MACHINE ALGORITHM



- This best decision boundary is called a hyperplane.
- SVM chooses the extreme points/vectors that help in creating the hyperplane.
- These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.
- Consider this diagram in which there are two different categories that are classified using a decision boundary or hyperplane

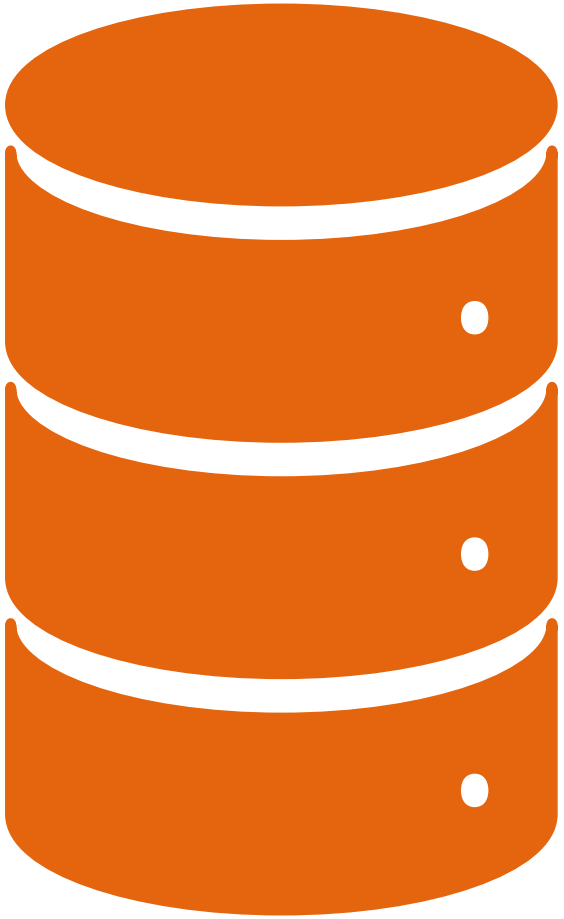


SVM: EXAMPLE

- SVM can be understood with the example:
- Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm.
- We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature.

SVM: EXAMPLE

- So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog.
- On the basis of the support vectors, it will classify it as a cat. Considering diagram
- SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.



TYPES OF SVM

SVM can be of two types:

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

HYPERPLANE AND SUPPORT VECTORS IN THE SVM ALGORITHM

Hyperplane:

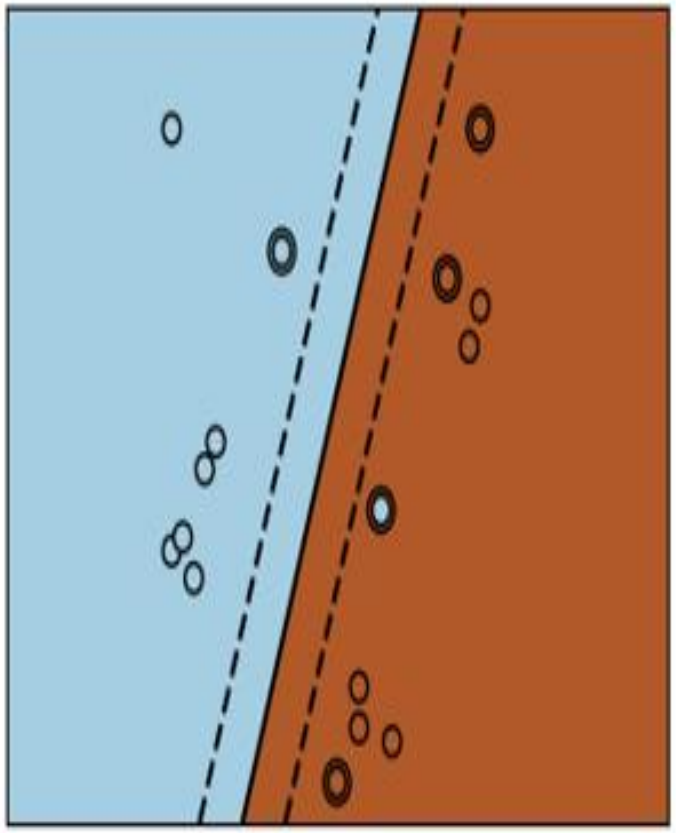
- There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points.
- This best boundary is known as the hyperplane of SVM.
- The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line.
- And if there are 3 features, then hyperplane will be a 2-dimension plane.
- We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

HYPERPLANE AND SUPPORT VECTORS IN THE SVM ALGORITHM

Support Vectors:

- The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector.
- These vectors support the hyperplane, hence called a Support vector.

TYPE OF KERNAL: LINEAR KERNEL

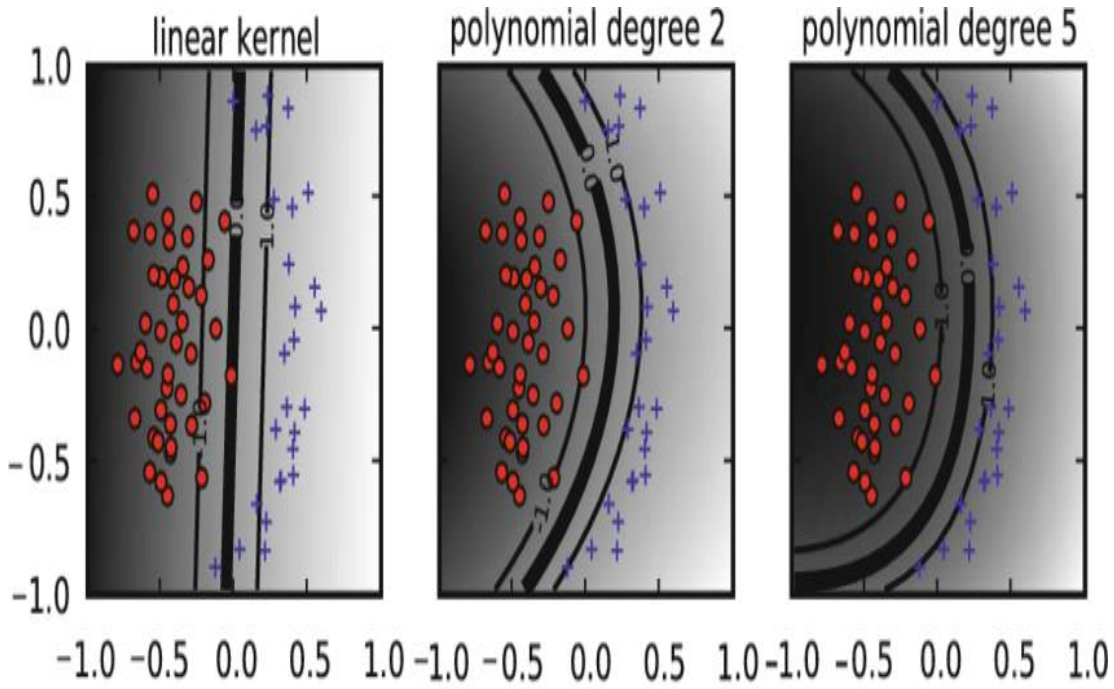


In linear kernel, the kernel function takes the form of a linear function as follows-

- **linear kernel : K(xi , xj) = xiT xj**
- Linear kernel is used when the data is linearly separable.
- It means that data can be separated using a single line. It is one of the most common kernels to be used.
- It is mostly used when there are large number of features in a dataset.
- Linear kernel is often used for text classification purposes.
- Training with a linear kernel is usually faster, because we only need to optimize the C regularization parameter.

TYPE OF KERNAL: POLYNOMIAL KERNEL

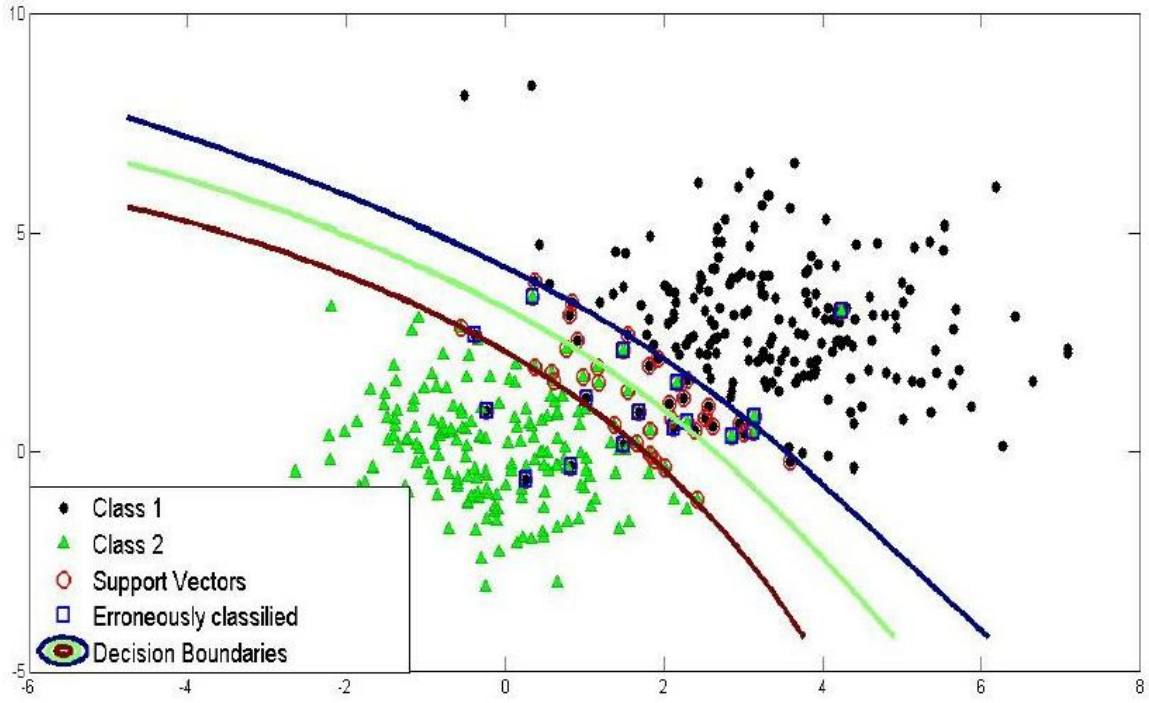
- The polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of the input samples.
- For degree-d polynomials, the polynomial kernel is defined as follows –
- Polynomial kernel : $K(xi , xj) = (\gamma.xiT\ xj + r)d , \gamma > 0$
- Polynomial kernel is very popular in Natural Language Processing.
- The most common degree is d = 2 (quadratic), since larger degrees tend to overfit on NLP problems. It can be visualized with this diagram.



TYPE OF KERNAL: RADIAL BASIC KERNEL

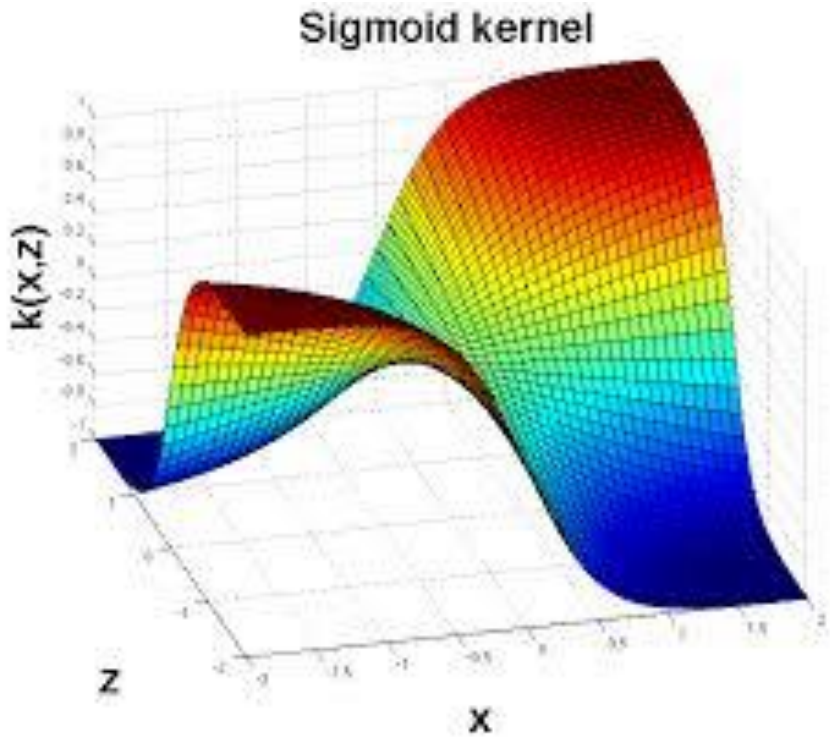
- Radial basis function kernel is a general purpose kernel.
- It is used when we have no prior knowledge about the data.
- The RBF kernel on two samples x and y is defined by the following equation -

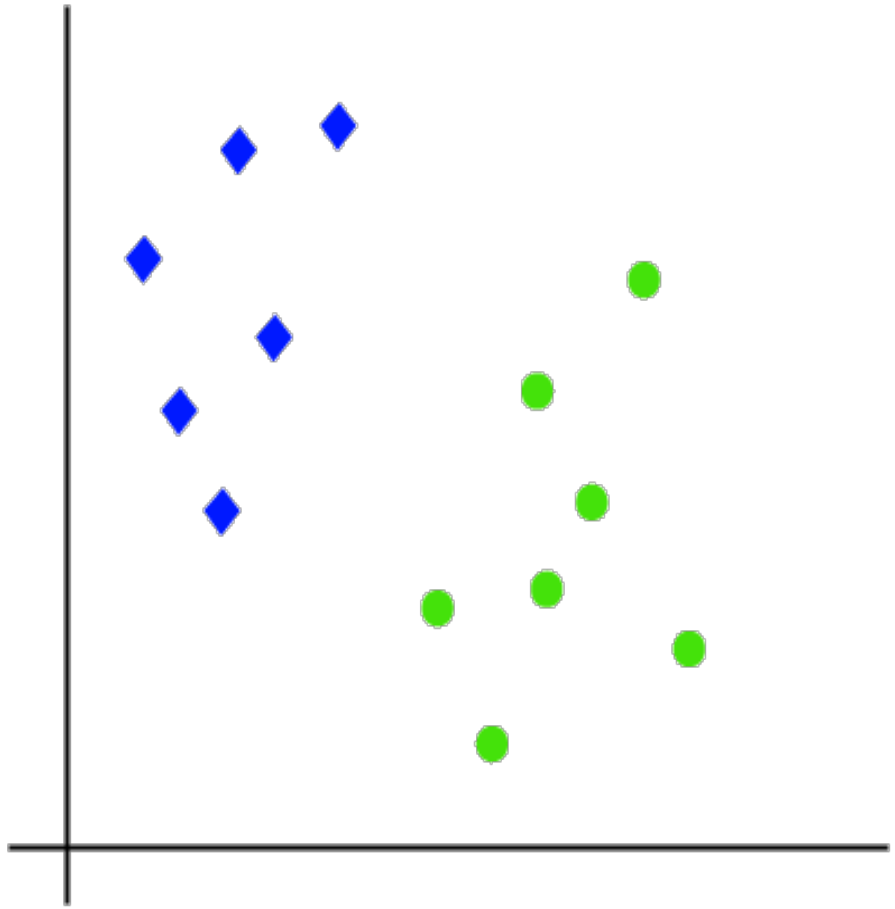
$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$



TYPE OF KERNAL: SIGMOID KERNEL

- Sigmoid kernel has its origin in neural networks.
- We can use it as the proxy for neural networks. Sigmoid kernel is given by the following equation –
- **sigmoid kernel : k (x, y) = tanh(α.xTy + c)**
- Sigmoid kernel can be visualized with the following diagram-

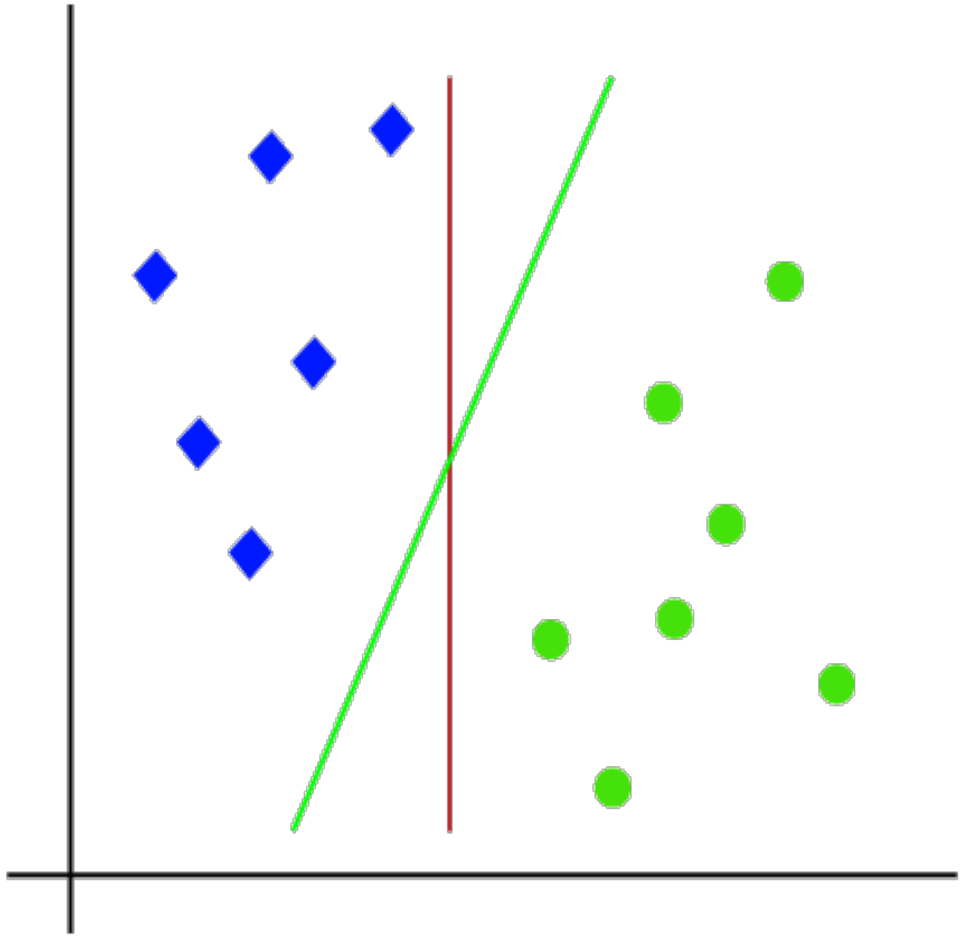




HOW DOES SVM WORKS?

Linear SVM:

- The working of the SVM algorithm can be understood by using an example.
- Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2.
- We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider this image:

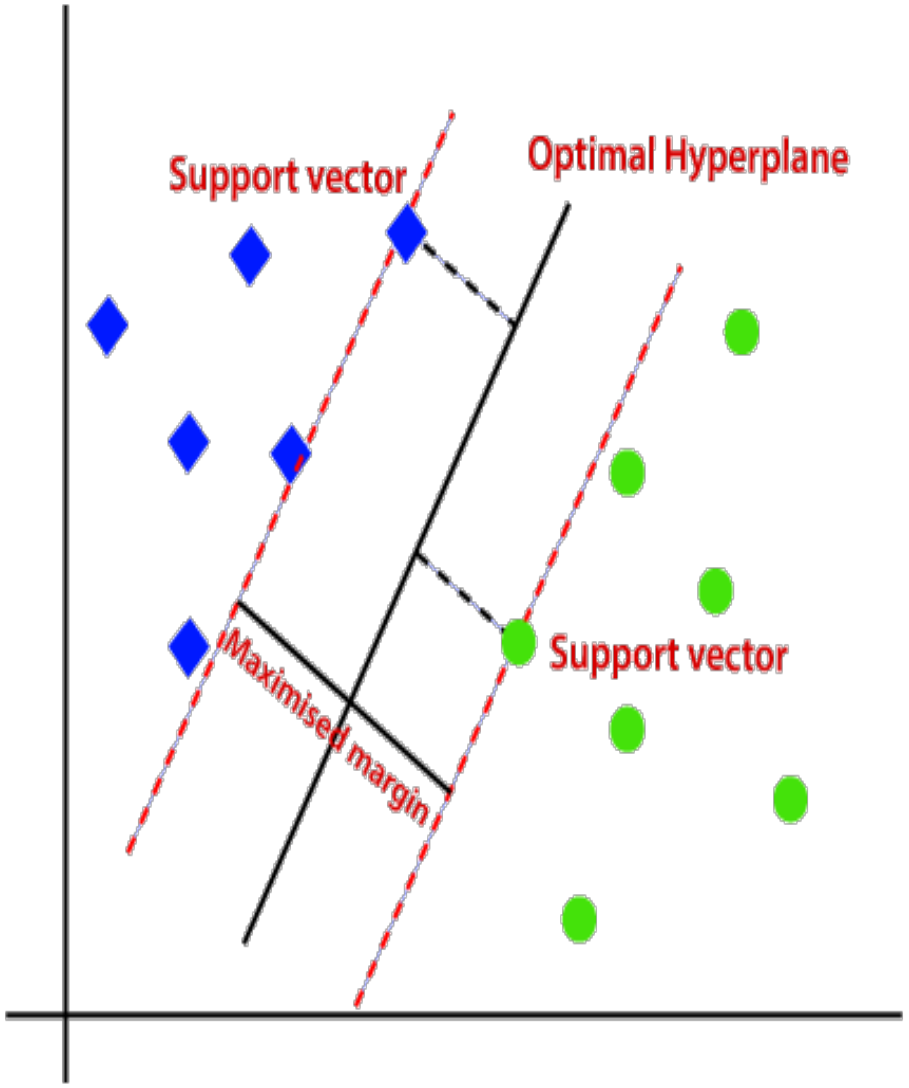


HOW DOES SVM WORKS?

Linear SVM:

- So as it is 2-d space so by just using a straight line, we can easily separate these two classes.

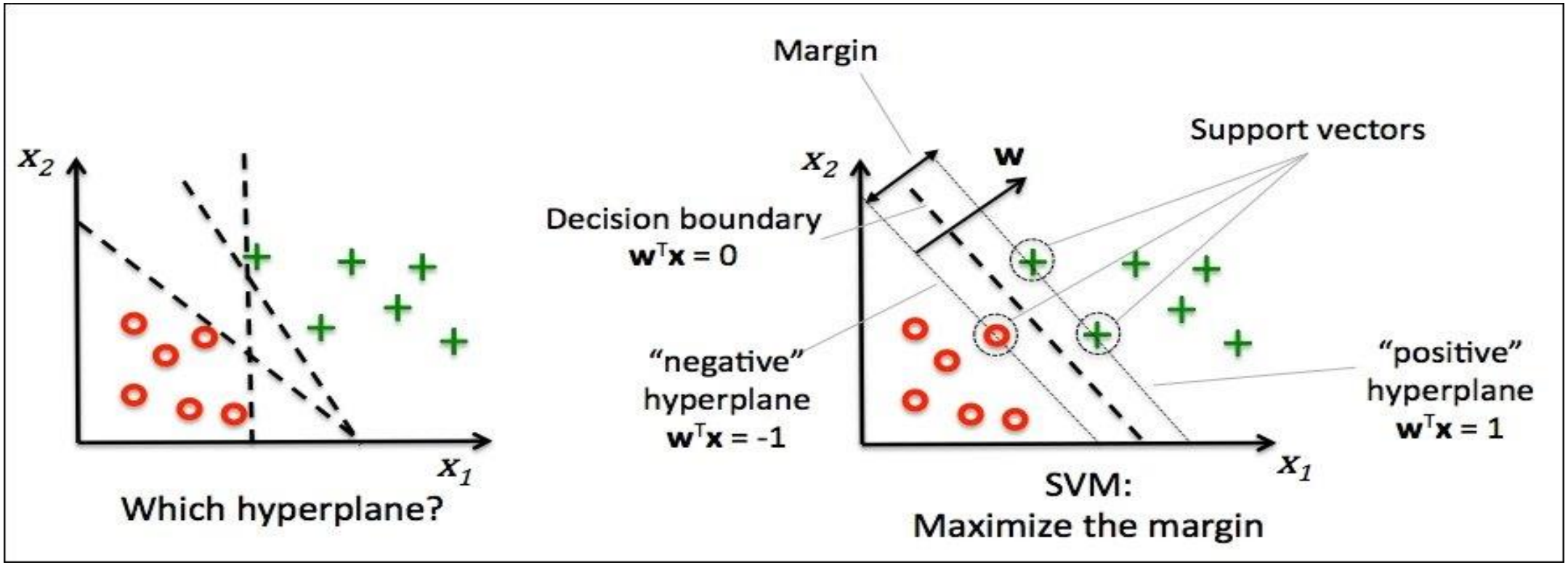
- But there can be multiple lines that can separate these classes. Consider this image.

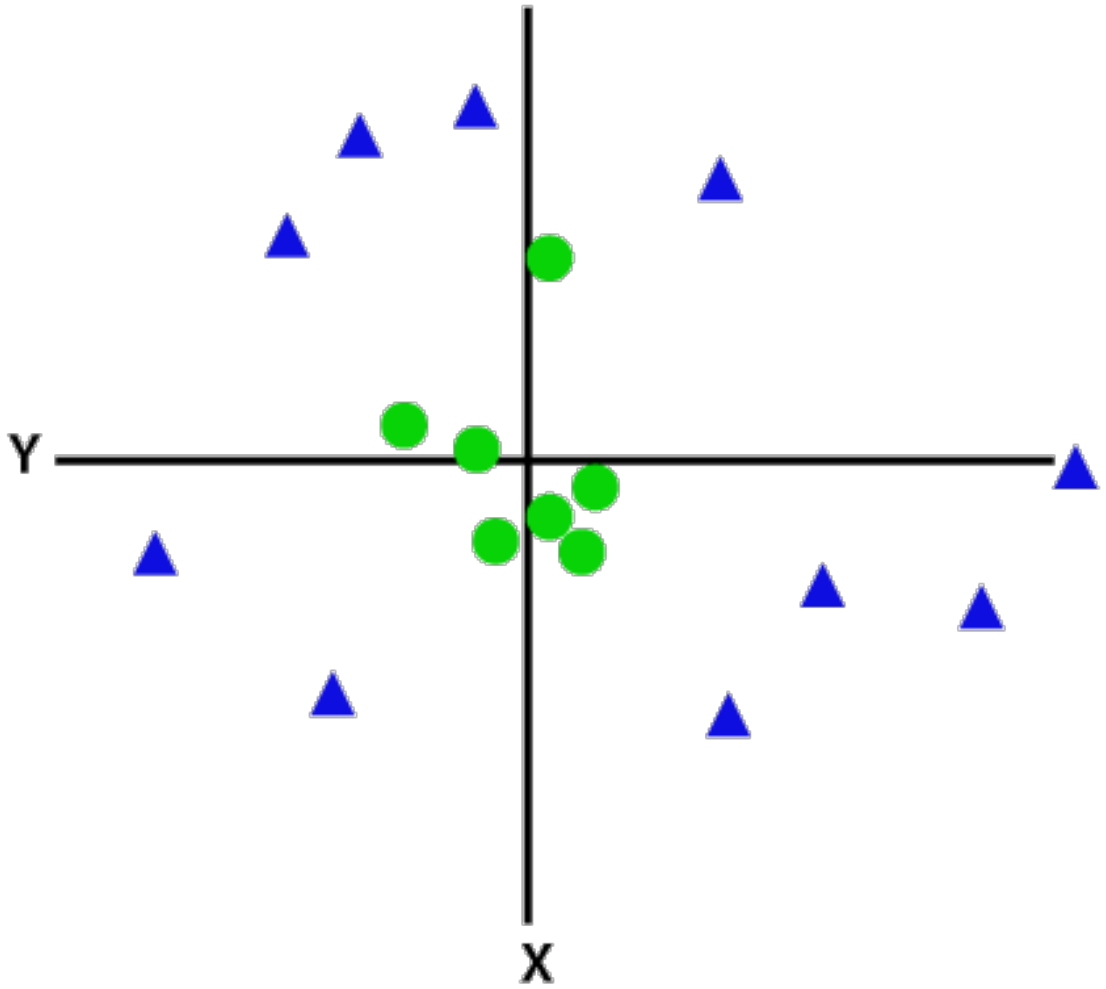


HOW DOES SVM WORKS?

- The SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**.
- SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors.
- The distance between the vectors and the hyperplane is called as **margin**.
- And the goal of SVM is to maximize this margin.
- The **hyperplane** with maximum margin is called the **optimal hyperplane**.

MAXIMUM MARGIN HYPERPLANE

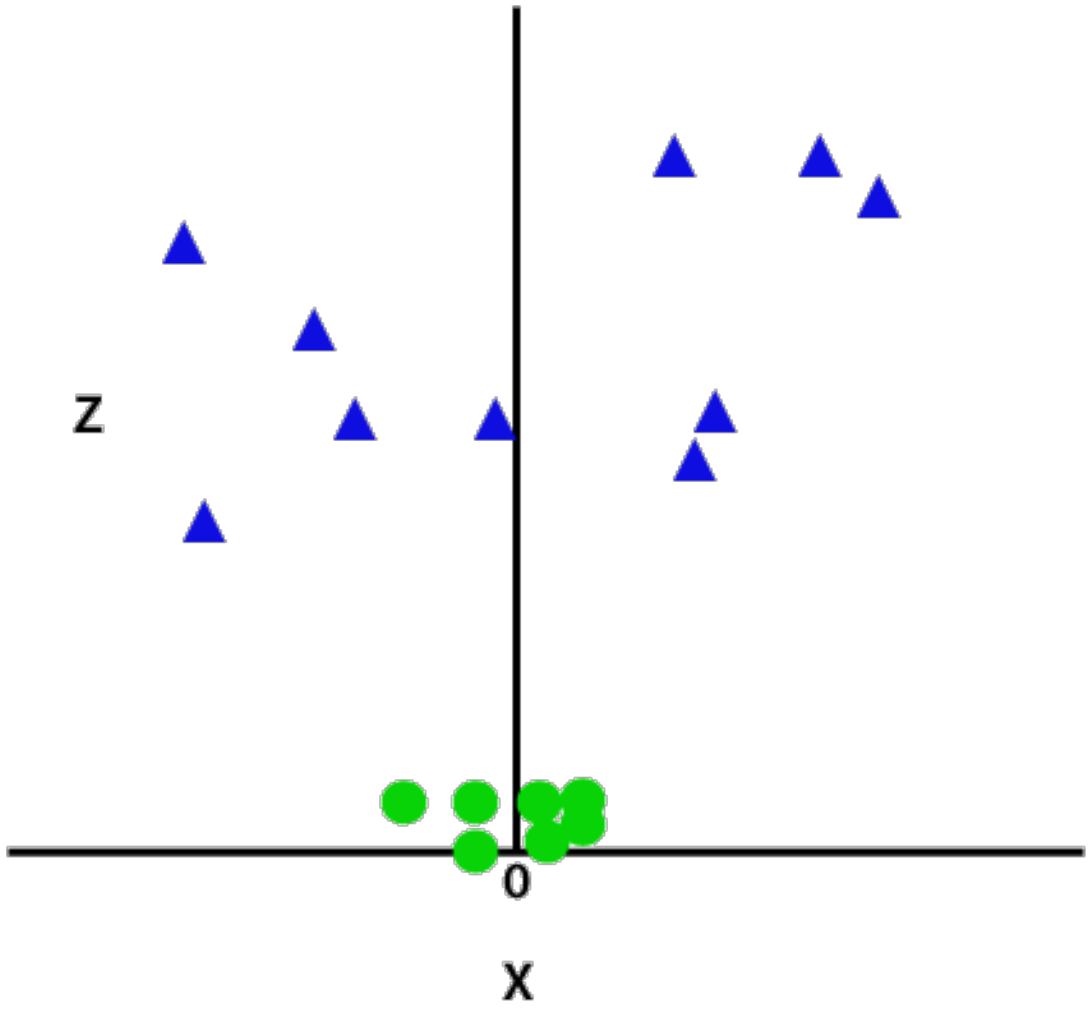




HOW DOES SVM WORKS?

Non-Linear SVM:

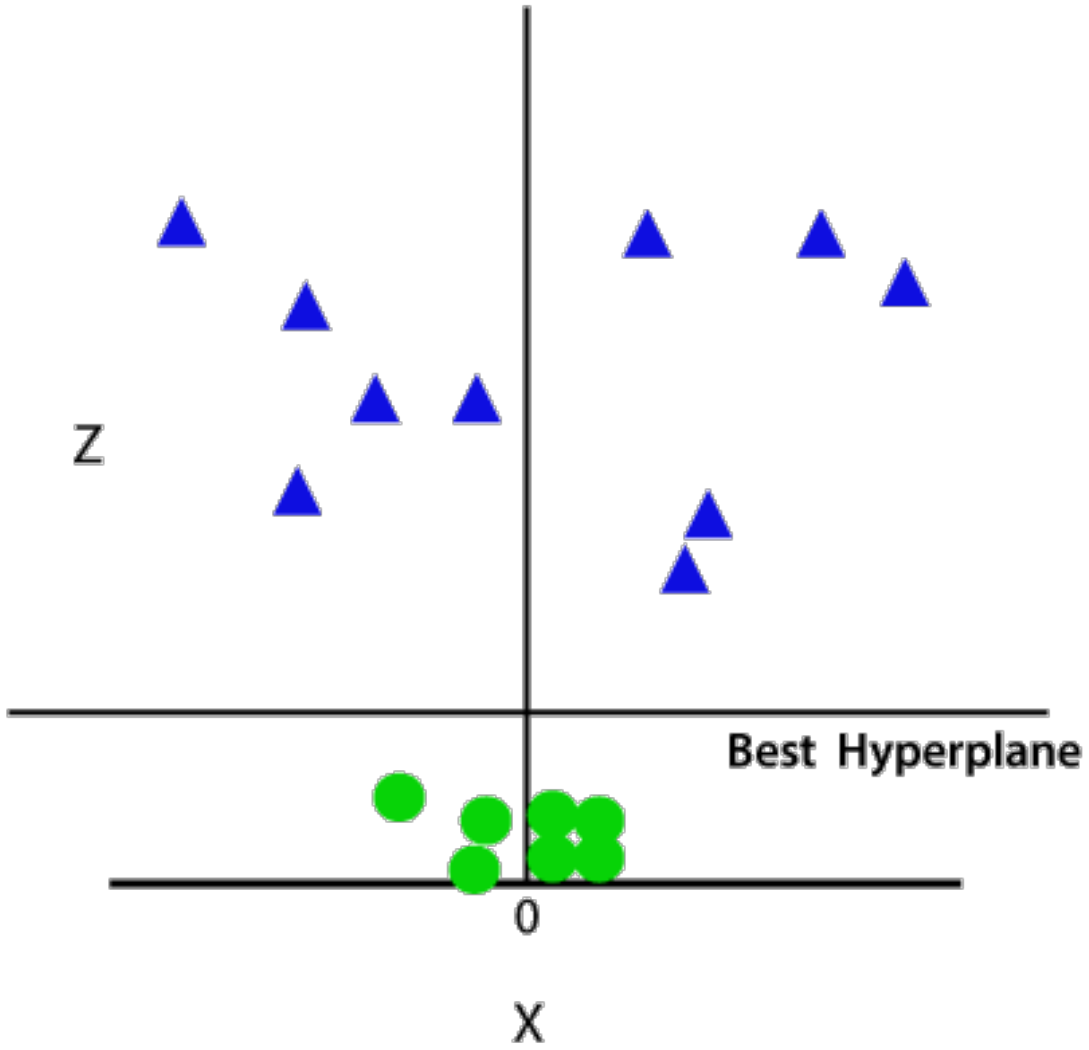
- If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.
- Consider this image
- So to separate these data points, we need to add one more dimension.
- For linear data, we have used two dimensions x and y, so for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



HOW DOES SVM WORKS?

Non-Linear SVM:

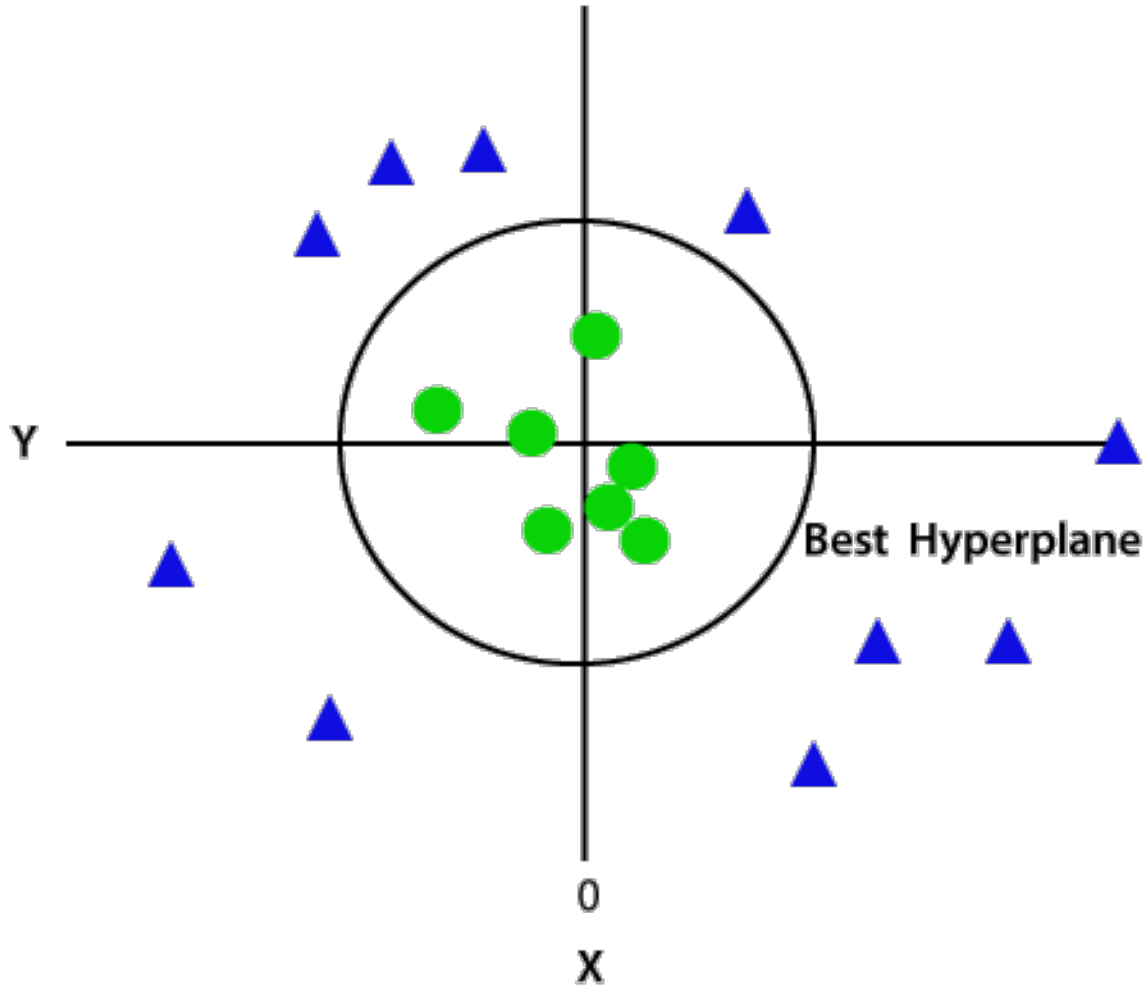
- By adding the third dimension, the sample space will become as this image
- So now, SVM will divide the datasets into classes in the following way.



HOW DOES SVM WORKS?

Non-Linear SVM:

- Consider this image
- Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis.



HOW DOES SVM WORKS?

Non-Linear SVM:

- If we convert it in 2d space with z=1, then it will become as:

- Hence we get a circumference of radius 1 in case of non-linear data.

PYTHON IMPLEMENTATION OF SUPPORT VECTOR MACHINE

- Now we will implement the SVM algorithm using Python.
- Here we will use the same dataset **user_data**, which we have used in Logistic regression.

THANK YOU