Machine Learning

What Is Machine Learning?

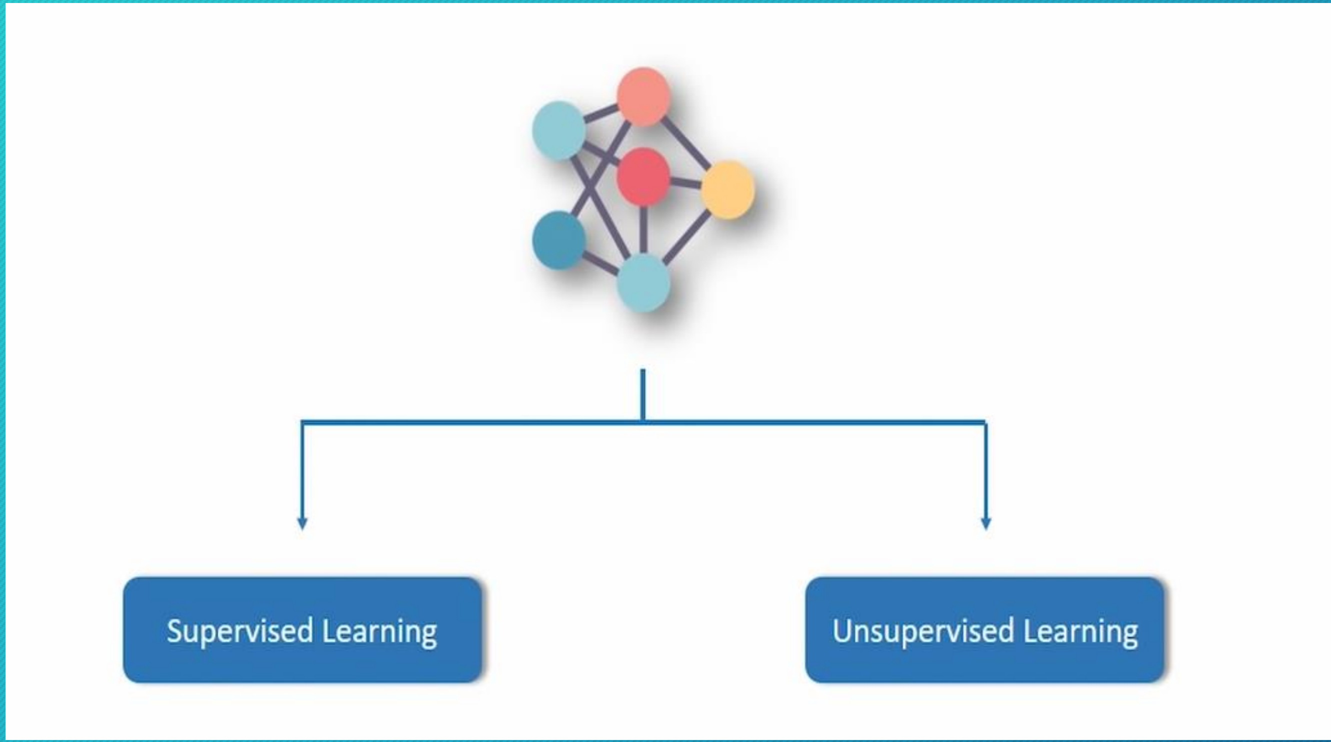
Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

Why Machine Learning Algorithm:

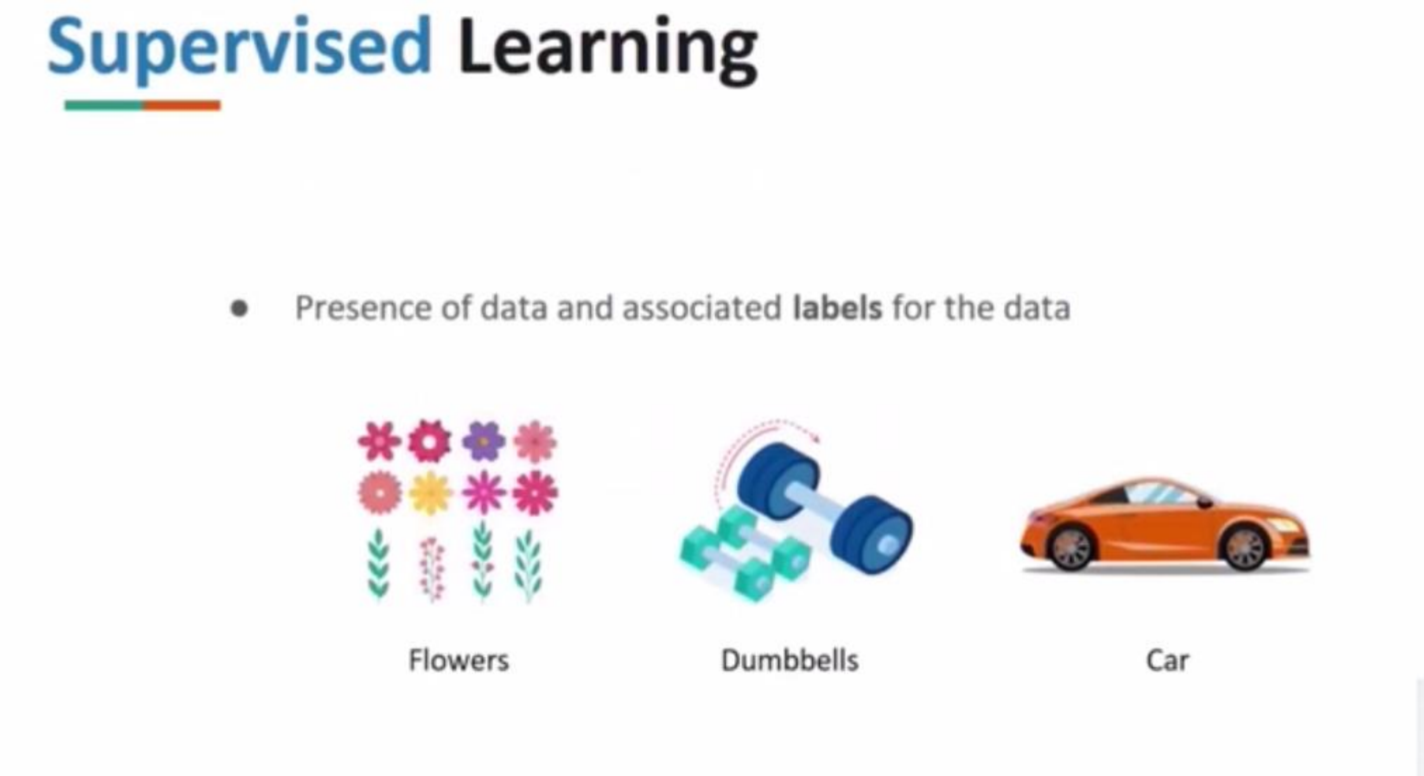
Lots of reasons!

- Helps reduce production cost
- Ability to easily process large amounts of data
- Deriving key insights about businesses
- Finding out hidden trends in data

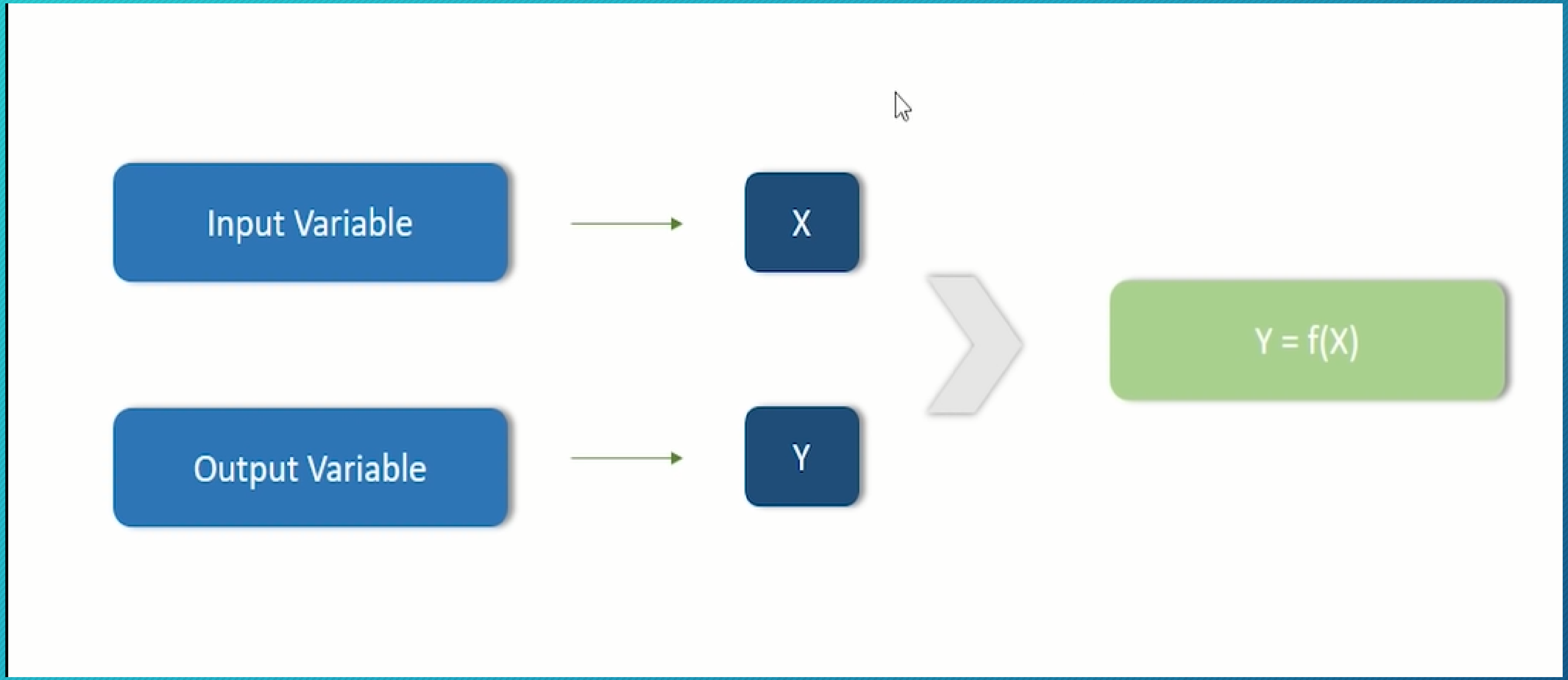
Categories Of Machine Learning



Supervised Learning



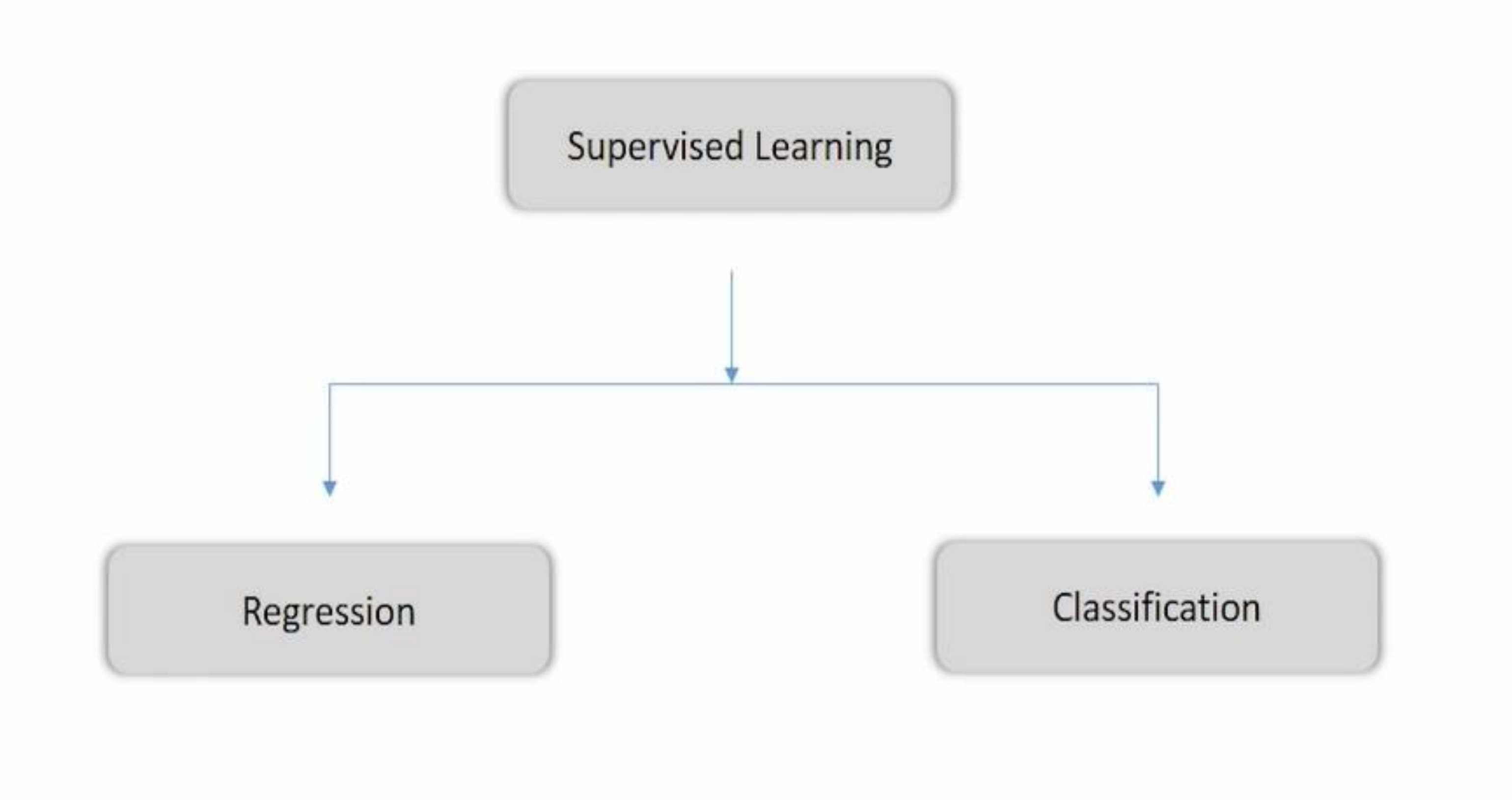
Supervised Machine Learning



Supervised Learning

- **y = f(x)** forms to be the foundation of supervised learning.
- The input variable is ‘x’, while the output variable is ‘y’.
- Mapping the output as a function of the input variable.

Categories of Supervised Machine Learning

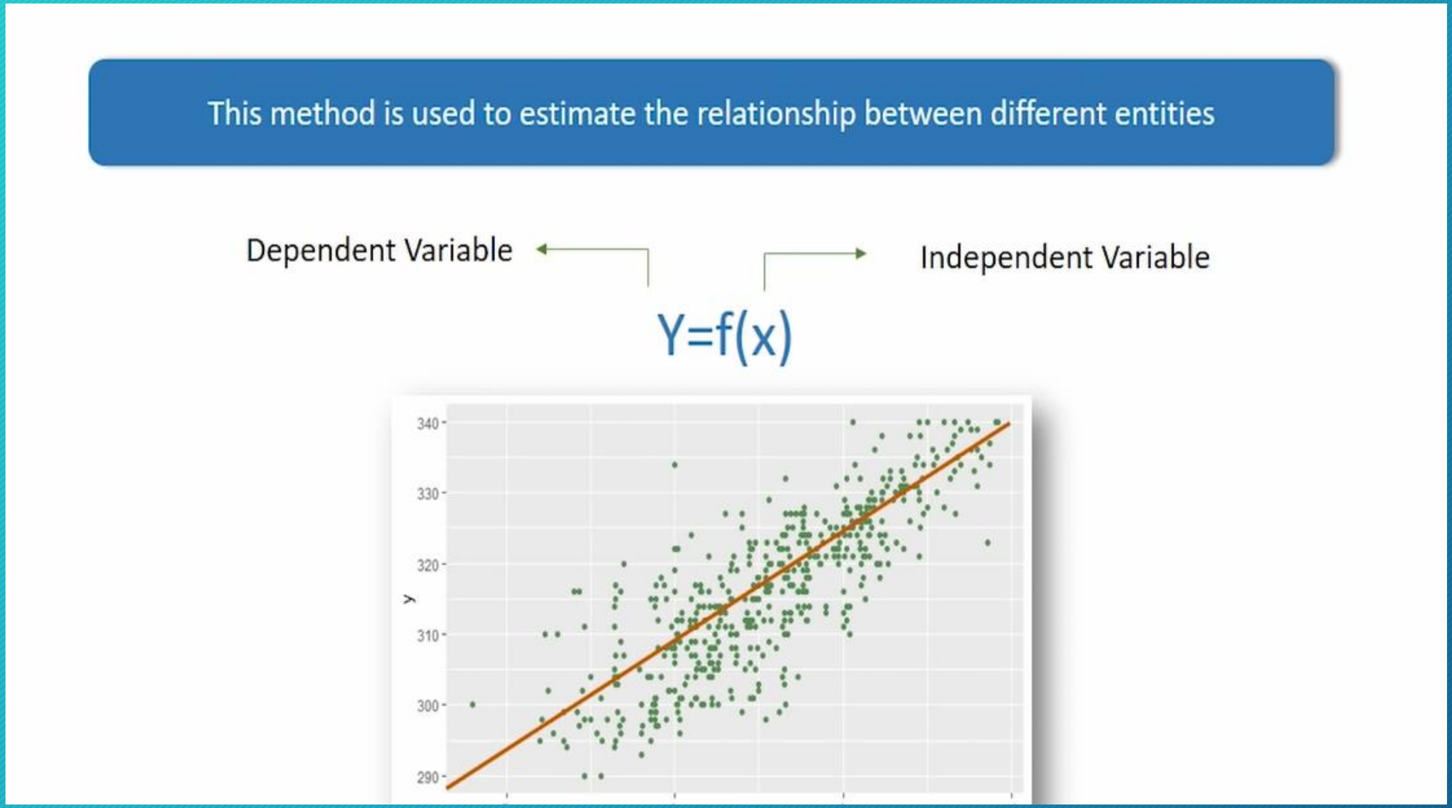


Supervised Learning

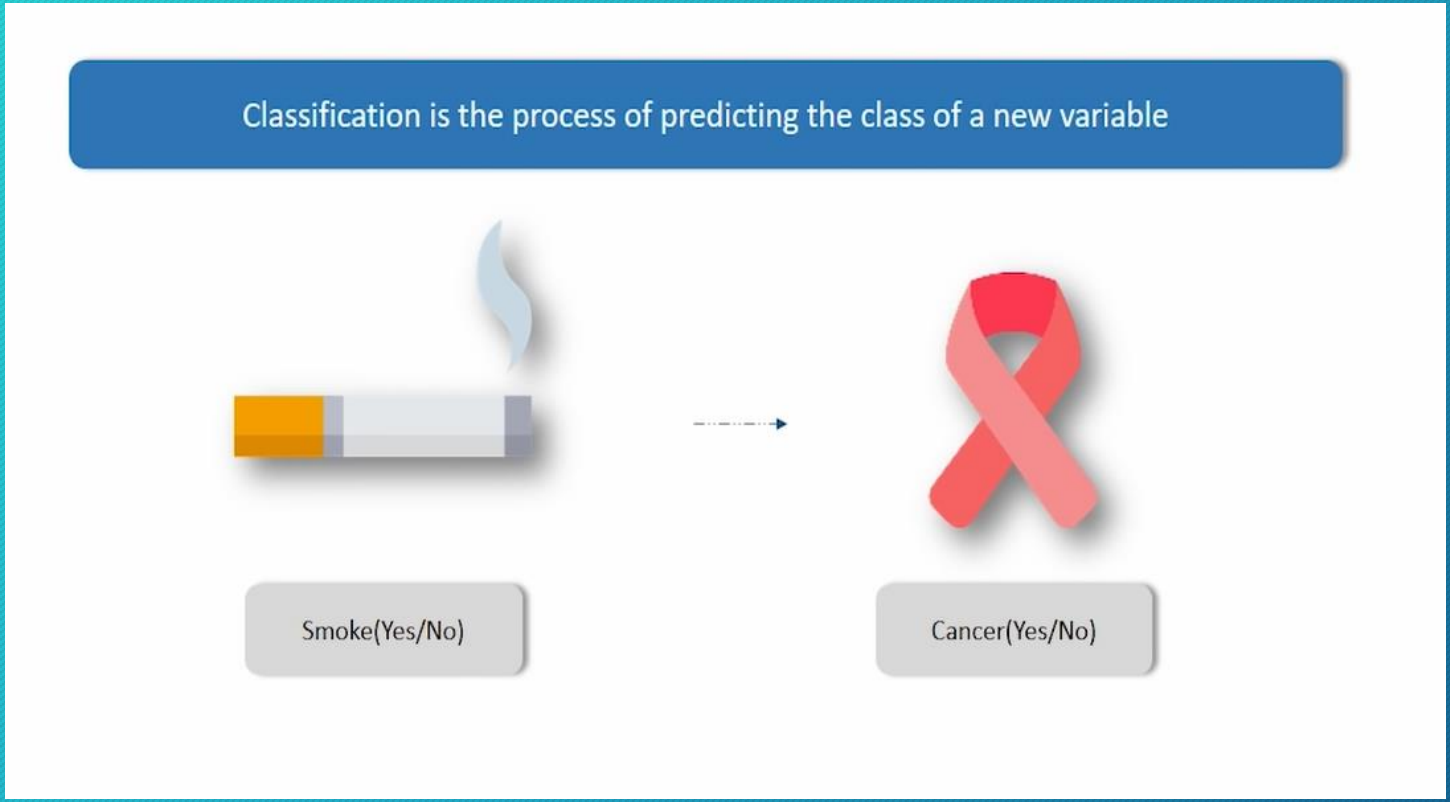
Grouping

- **Regression** – Prediction of future values from past data
- **Classification** – Categorization of items using data.

Regression


This method is used to estimate the relationship between different entities
Dependent Variable
Independent Variable
Y=f(x)
340
330
320
310
300
290
y

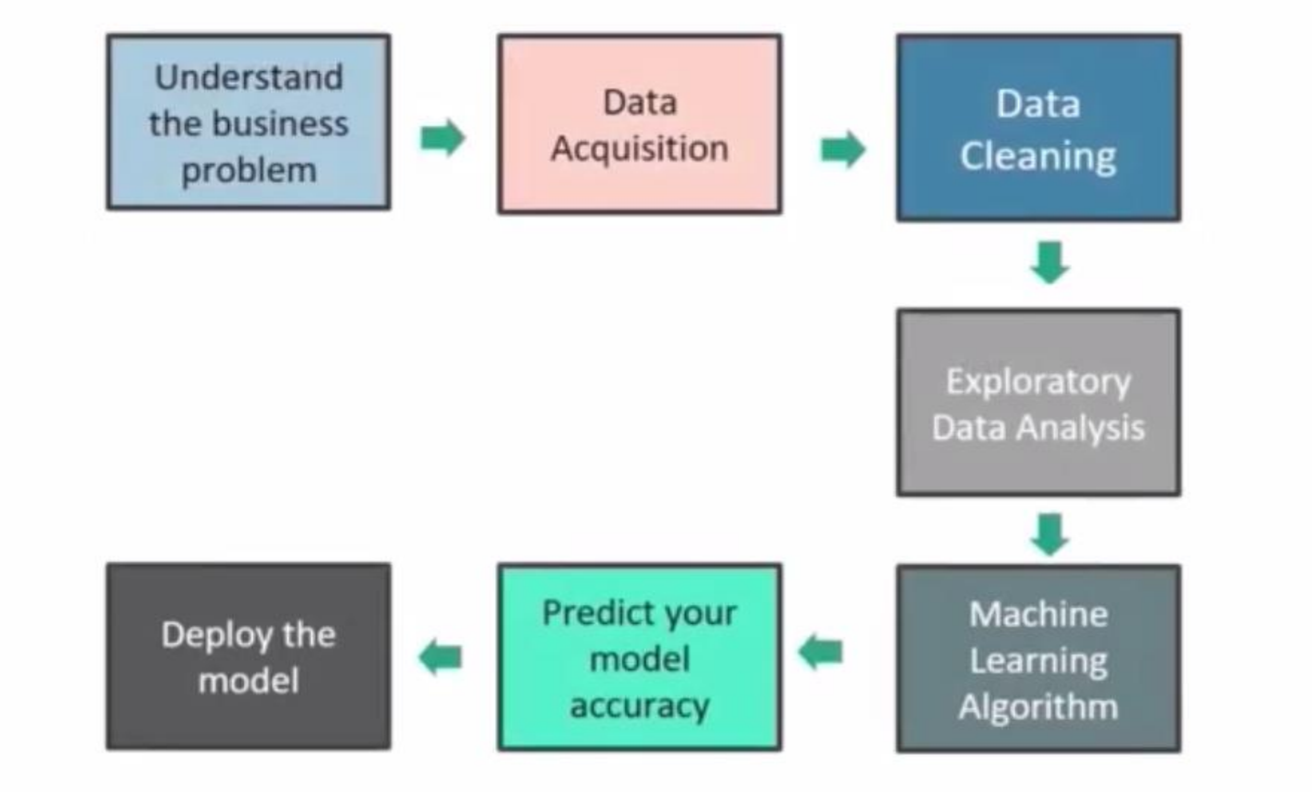
Classifiction :



How Machine Learning Model Learn:

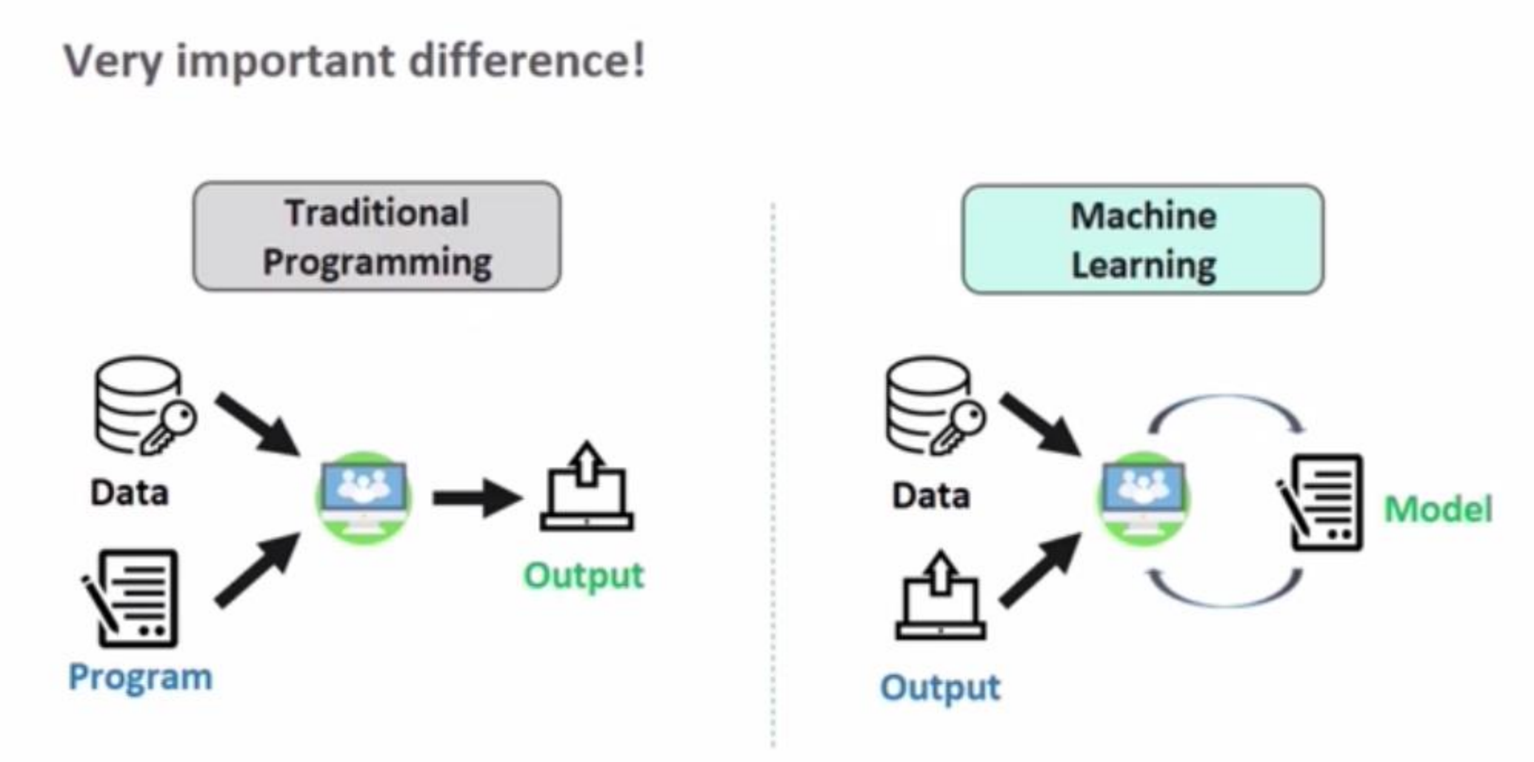
Data is split into two parts!

- **Training** Data – Used to teach the algorithm
- **Testing** Data – Used to verify the learning capability



Understand the business problem
Data Acquisition
Data Cleaning
Exploratory Data Analysis
Machine Learning Algorithm
Predict your model accuracy
Deploy the model

Traditional Learning Vs Machine Learning

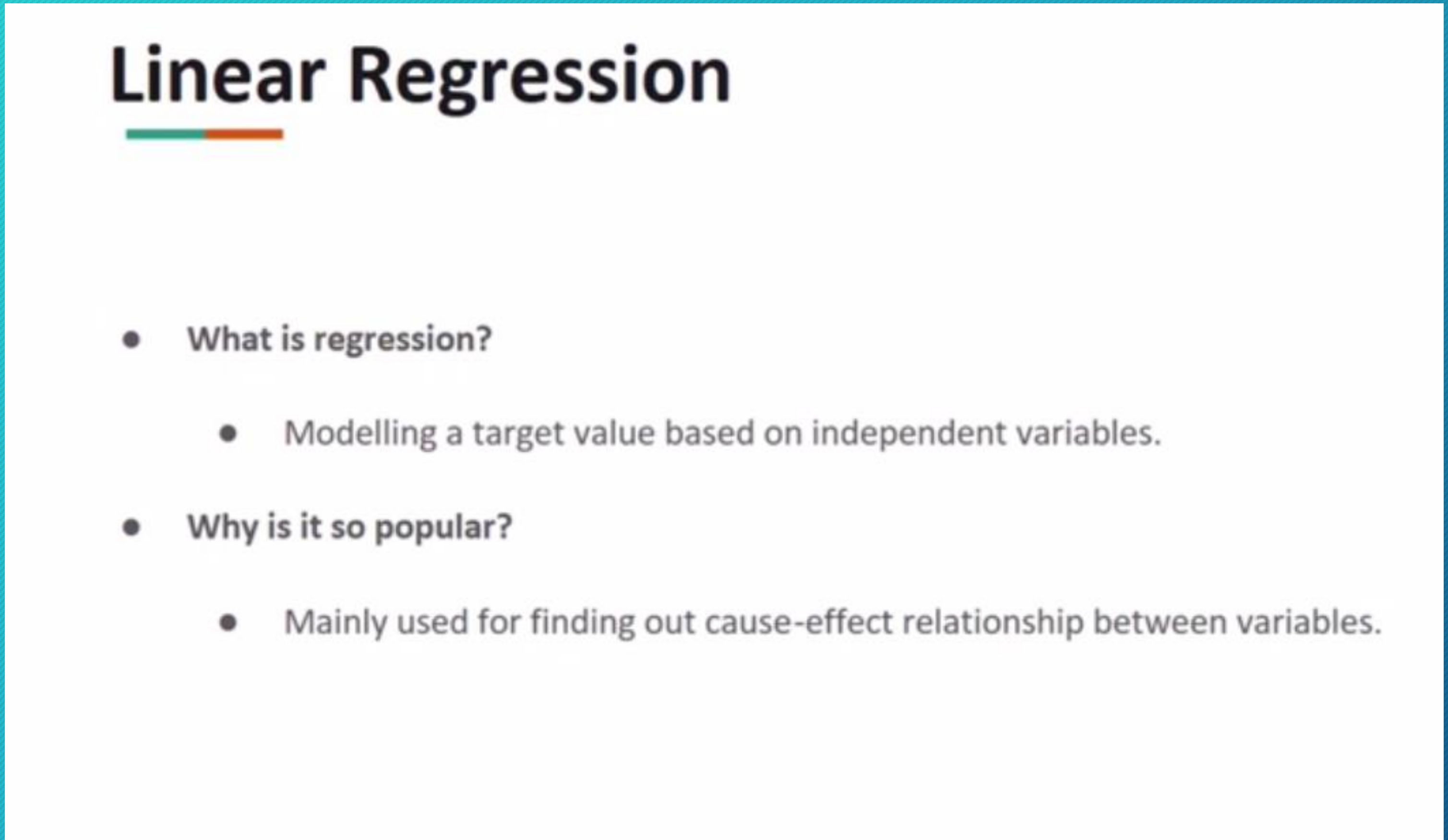


Machine Learning Algorithms

- Linear Regression
- Logistic Regression
- Naïve Bayes
- Support Vector Machine
- K-Nearest Neighbors
- Decision Tree
- Random Forest

Linear Regression

Linear Regression



Mean Squared Error

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

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

Mean Absolute Error

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

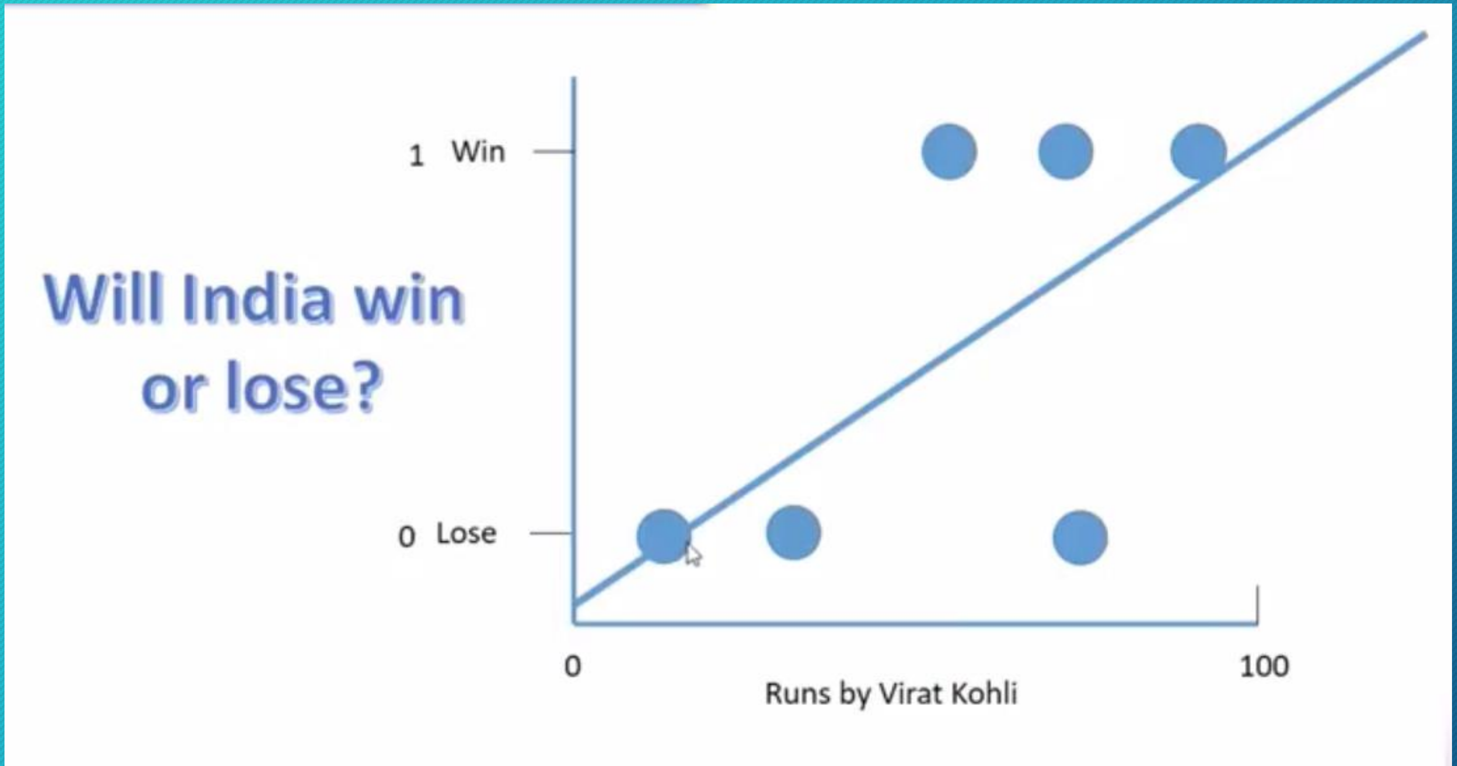
Root Mean Squared Error

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

$$\text{MAE} = \frac{|(y_i - y_p)|}{n}$$

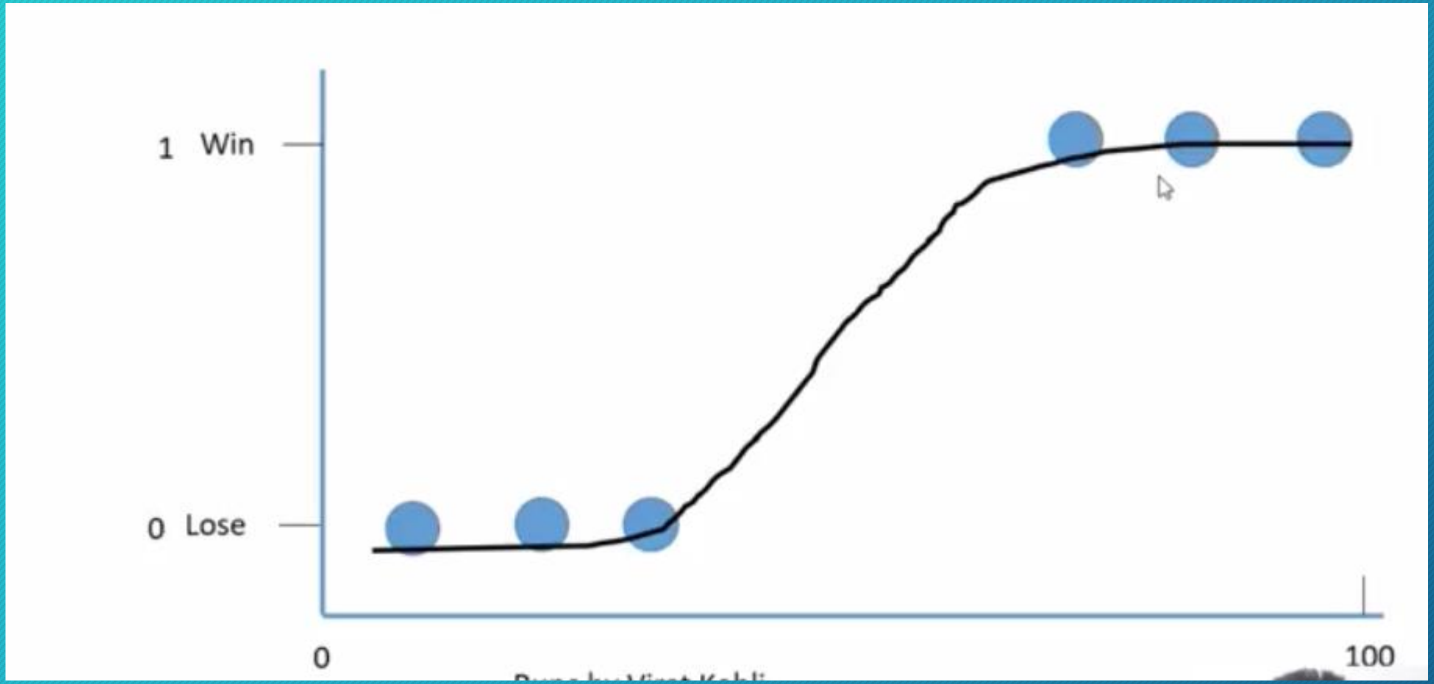
y_i = actual value
y_p = predicted value
n = number of observations/rows

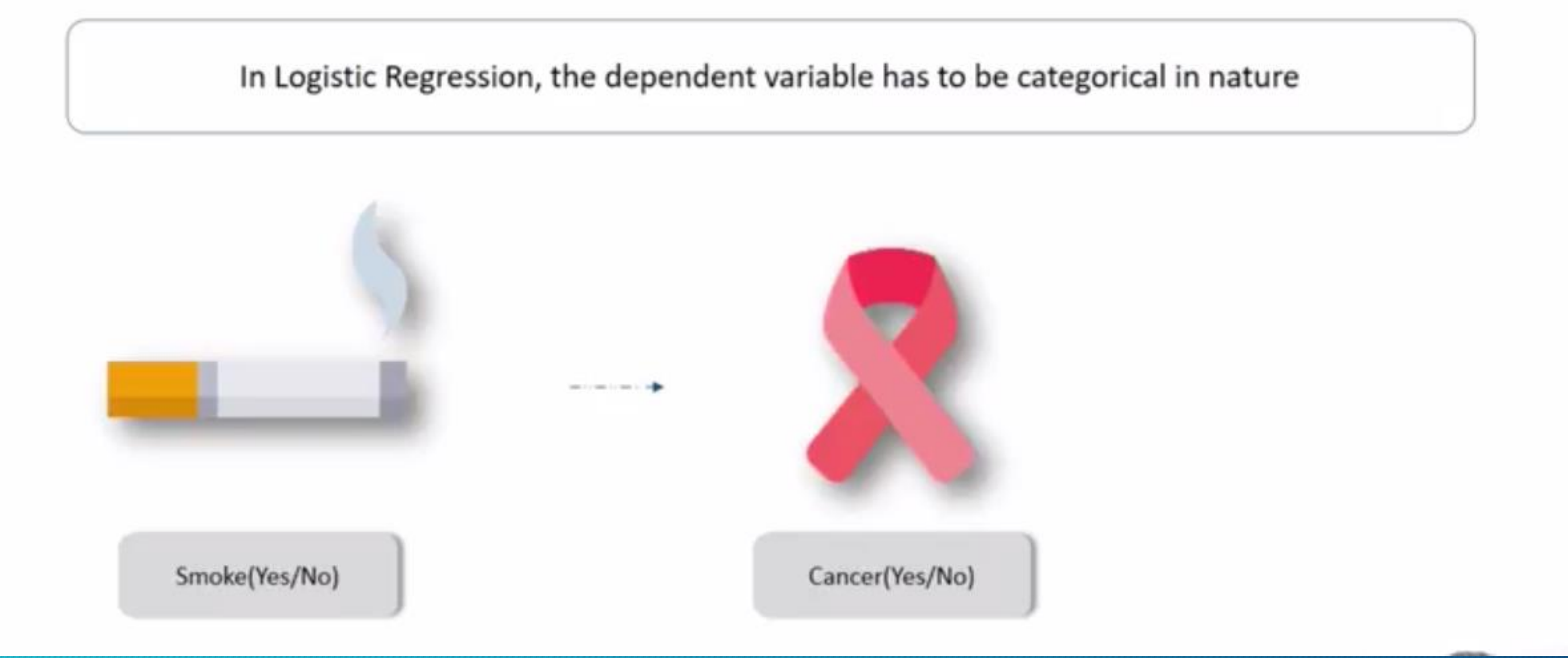
Problem With Linear Regression



Logistic Regression

Logistic Regression




In Logistic Regression, the dependent variable has to be categorical in nature
Smoke(Yes/No)
Cancer(Yes/No)

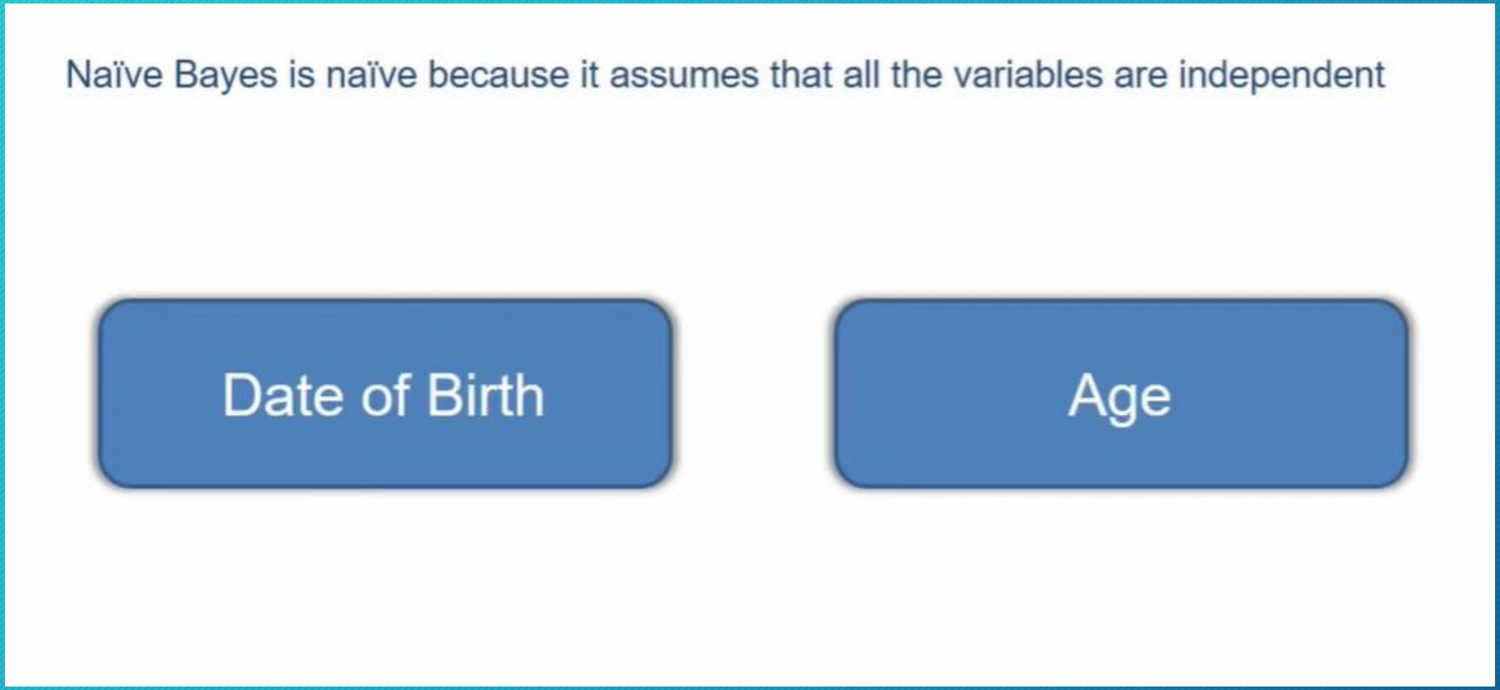
Sigmoid Function

Below formula gives us a sigmoid curve

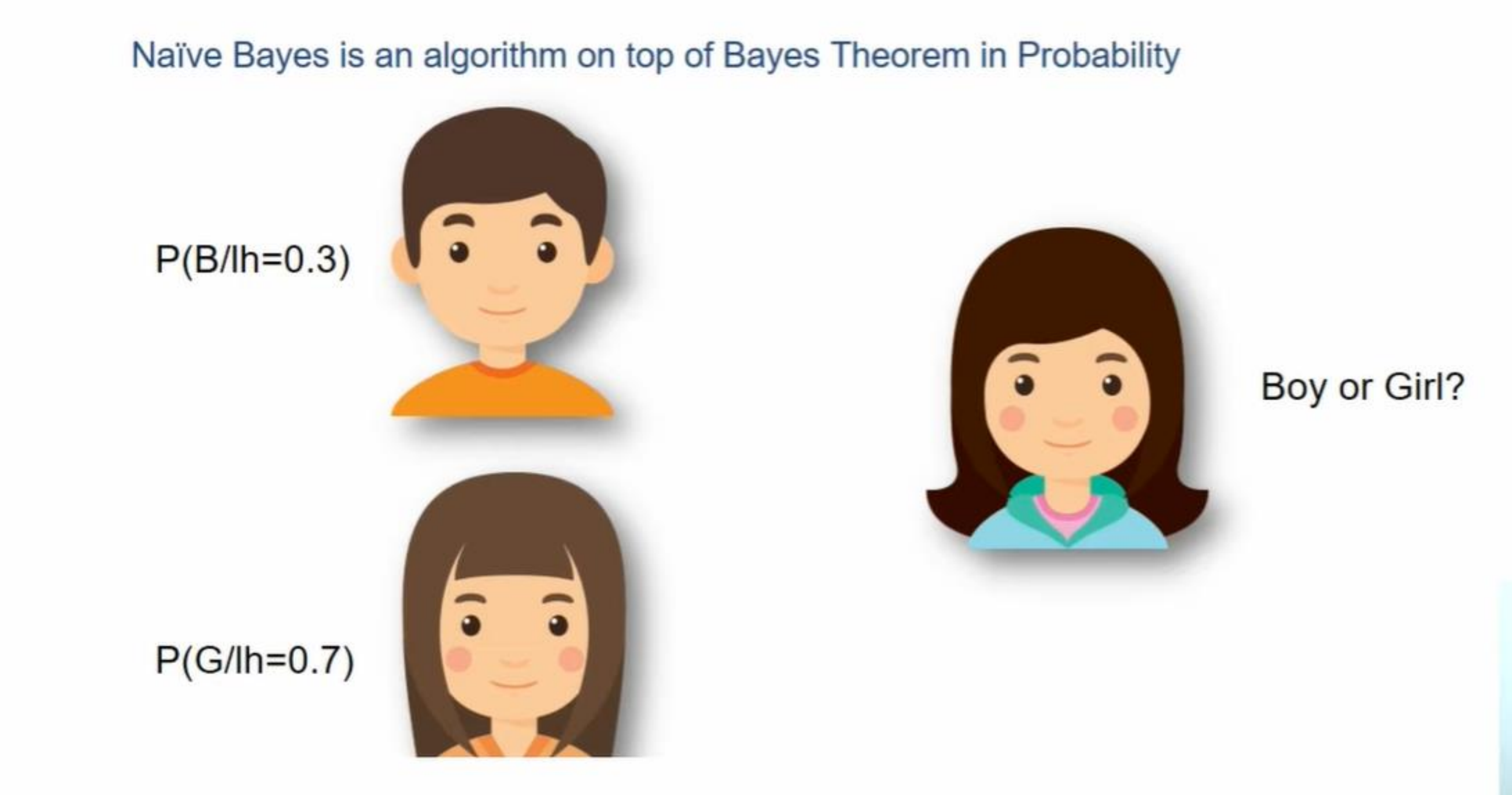
$$f(x) = \frac{e^x}{1 + e^x}$$

Naïve Bayes

What is Naïve ..?



Naïve Bayes


Naïve Bayes is an algorithm on top of Bayes Theorem in Probability
P(B/lh=0.3)
Boy or Girl?
P(G/lh=0.7)

Naïve Bayes Classifier -

a. Naive Bayes classifiers are linear classifiers based on Bayes' theorem. The model generated is probabilistic

b. It is called naive due to the assumption that the features in the dataset are mutually independent

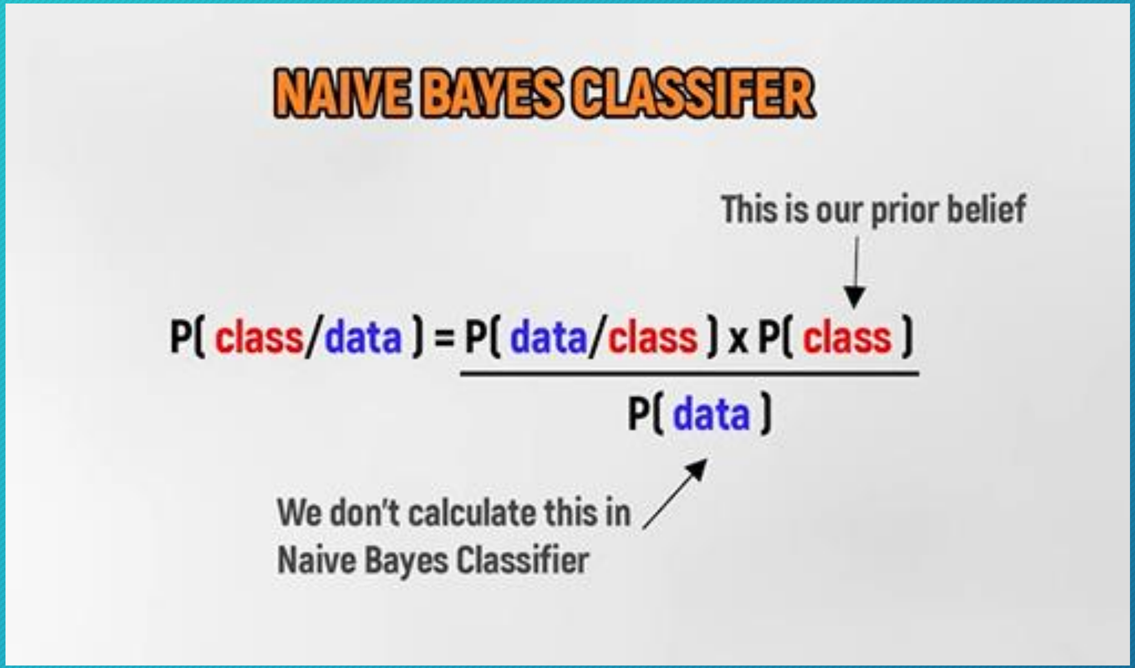
c. In real world, the independence assumption is often violated, but naive Bayes classifiers still tend to perform very well

d. Idea is to factor all available evidence in form of predictors into the naïve Bayes rule to obtain more accurate probability for class prediction

e. It estimates conditional probability which is the probability that something will happen, *given that something else* has already occurred. For e.g. the given mail is likely a spam given appearance of words such as “prize”

f. Being relatively robust, easy to implement, fast, and accurate, naive Bayes classifiers are used in many different fields

Naïve Bayes Classifier

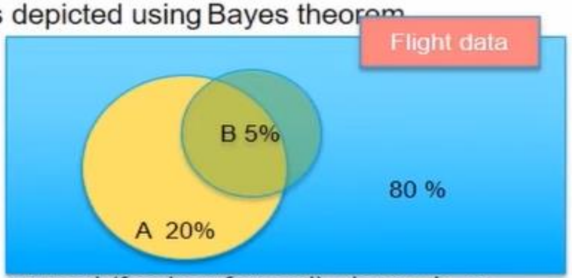


Joint/Conditional Probability

Naïve Bayes Classifier -

Joint Probabilities (Contd…) -

a. The relationship between dependent events is depicted using Bayes theorem



Likelihood Prior prob

Posterior

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

Evidence

b. Probability of event A given that event B has occurred (fog has formed) depends on
   I. Apriori probability of fog occurring whenever there was flight delay – P (B/A)
   II. Apriori probability of flight delay P(A) which is 20% in the example
   III. Apriori probability of flight facing fog P(B) which is 5% in the example

c. When it is a matter of deciding the class of an output such as whether flight will get delayed or not, we calculate P(A/B) and P(!A/B), compare which is higher. Since in both the denominator is P(B), it is ignored as it has no influence on which class will it be

d. However, to calculate the updated probability of a class, denominator P(B) is required

Naïve Bayes Classifier

Naïve Bayes Classifier -

a. The following two tables reflect the apriori probabilities of the events A and B. Probabilities based on past data of 100 points

T1	FOG		
Frequency	**Yes**	**No**	**Total**
Flight delayed	4	16	20
Not Delayed	1	79	80
Total	5	95	100

T2	FOG		
Likelihood	**Yes**	**No**	**Total**
Flight delayed	4 / 20	16 / 20	20
Not Delayed	1 / 80	79 /80	80
Total	5 / 100	95 / 100	100

b. In the likelihood table (T2) reveals that P(fog = Yes / flight delayed) = 4/20 = .20 indicating that the probability is 20 percent that a flight will be delayed given fog

c. $P(A \cap B)$ => P(flight delay | fog) = P(fog / flight delay) * P(flight delay)

d. P(flight delay | fog) = ((4/20) * (20 / 100)) = .04 (maximal probability) (no need to divide by P(B), probability of fog, as it is a constant. **This is Naïve Bayes probability.**

e. **Joint probability -** $P(A \cap B)$ = ((20 / 100) * (5/100)) = .01

K-Nearest Neighbors

K-NN Algorithm

- Input data is indexed to find the closest neighbors.
- Data is compared in the inferencing phase to save time.
- Belongs to the category of lazy learners!

K-NN Classifier

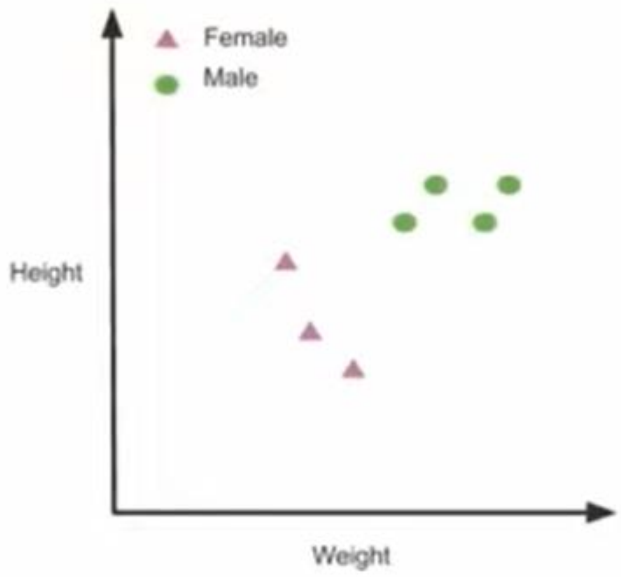
Let us understand a simple K-NN Classifier:

- Predict if person is male or female using K-NN.
- Prediction is based on height and weight of the person.

DATASET EXAMPLE

Height	Weight	Gender
187	80	Male
165	50	Female
199	99	Male
145	70	Female
180	87	Male
178	65	Female
187	60	Male

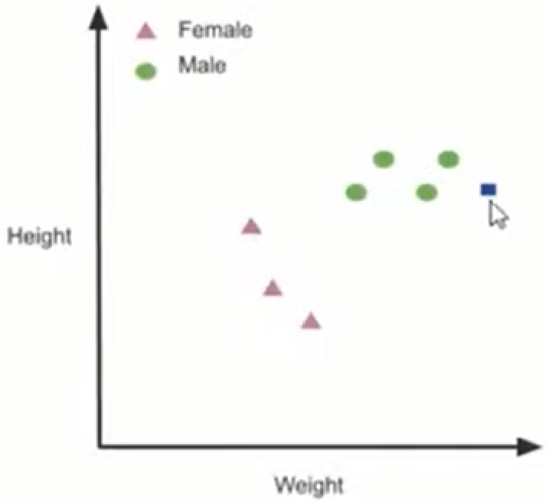
Adding New Dataset:



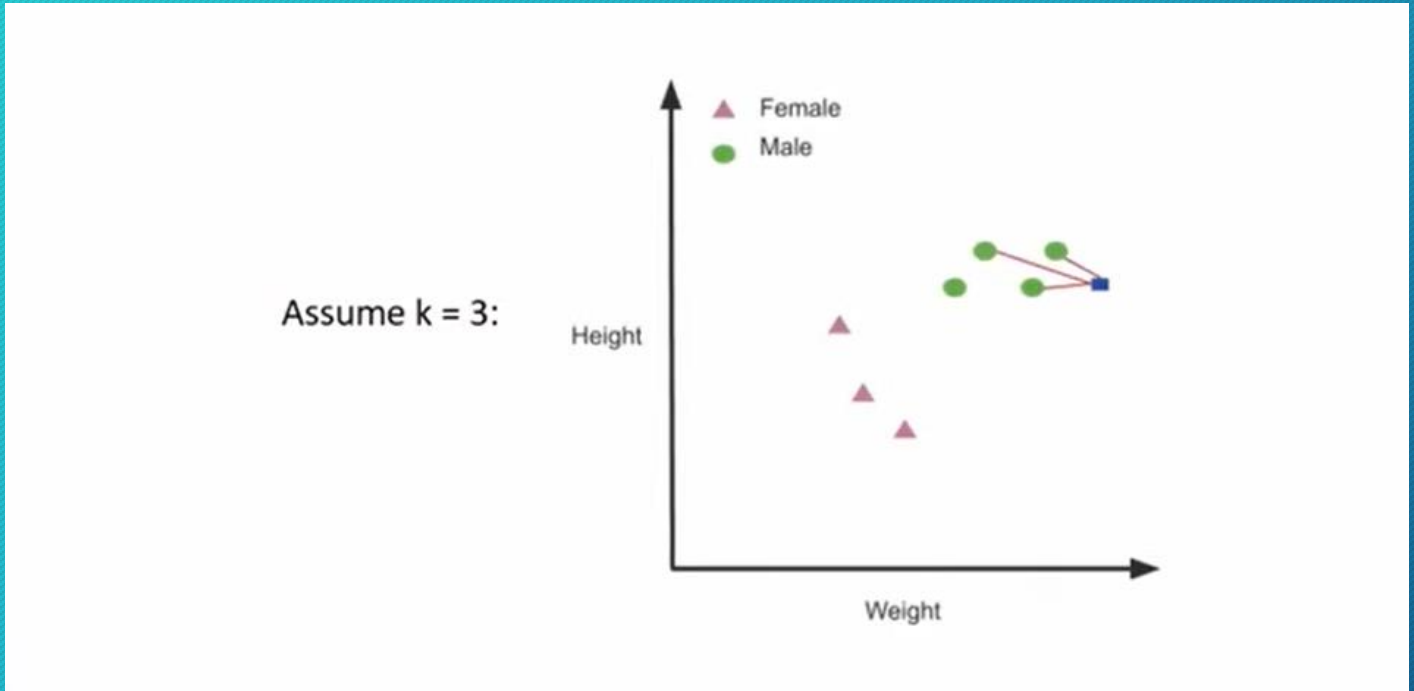
Now, need to add a new entry of a person with height 190cm and weight 100kg.

Add New Plot

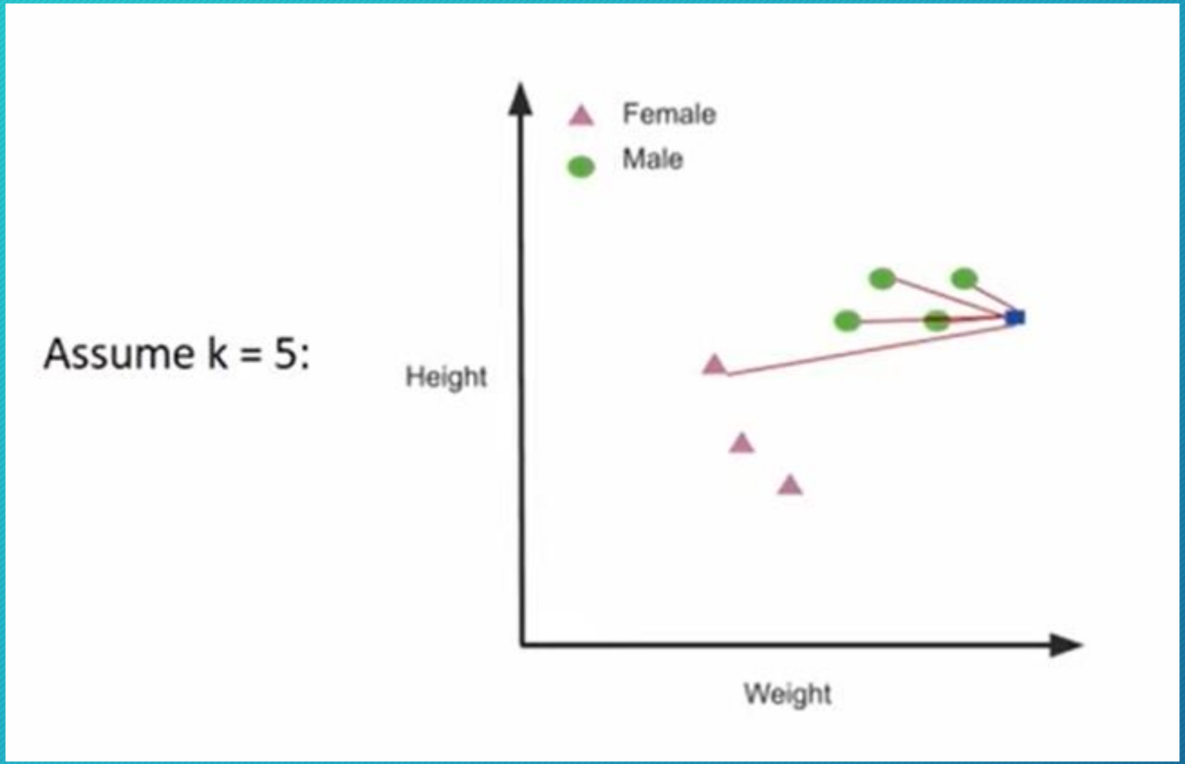
Added a new entry and plotted the relevant graph for it.



What is K ?



NOTE : Most common practice to set k as odd when there is comparison.

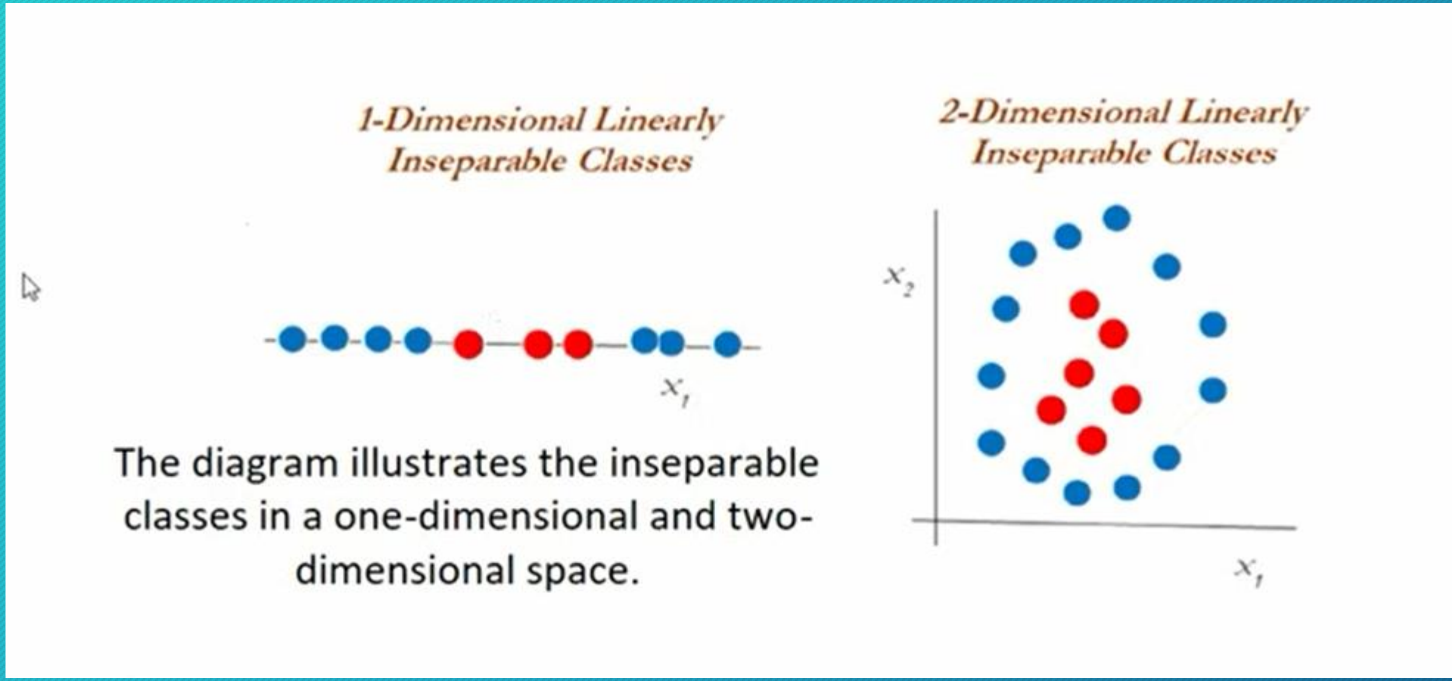


Support Vector Machine

Support Vector Machine:

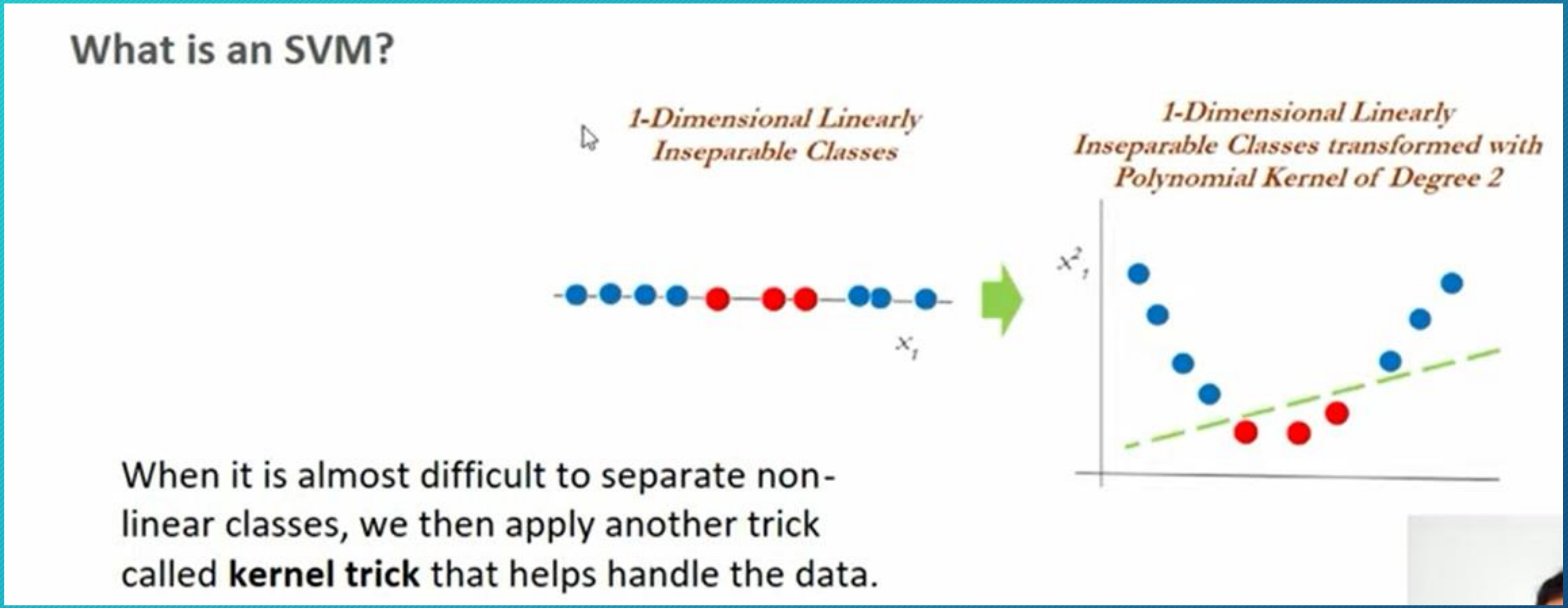
- SVM is a supervised learning algorithm in Machine Learning that can be used for both regression and classification applications.
- The support vector machine approach is considered during a non-linear decision.
- And, the data is not separable by a support vector classifier irrespective of the cost function.

Inseprable Data:

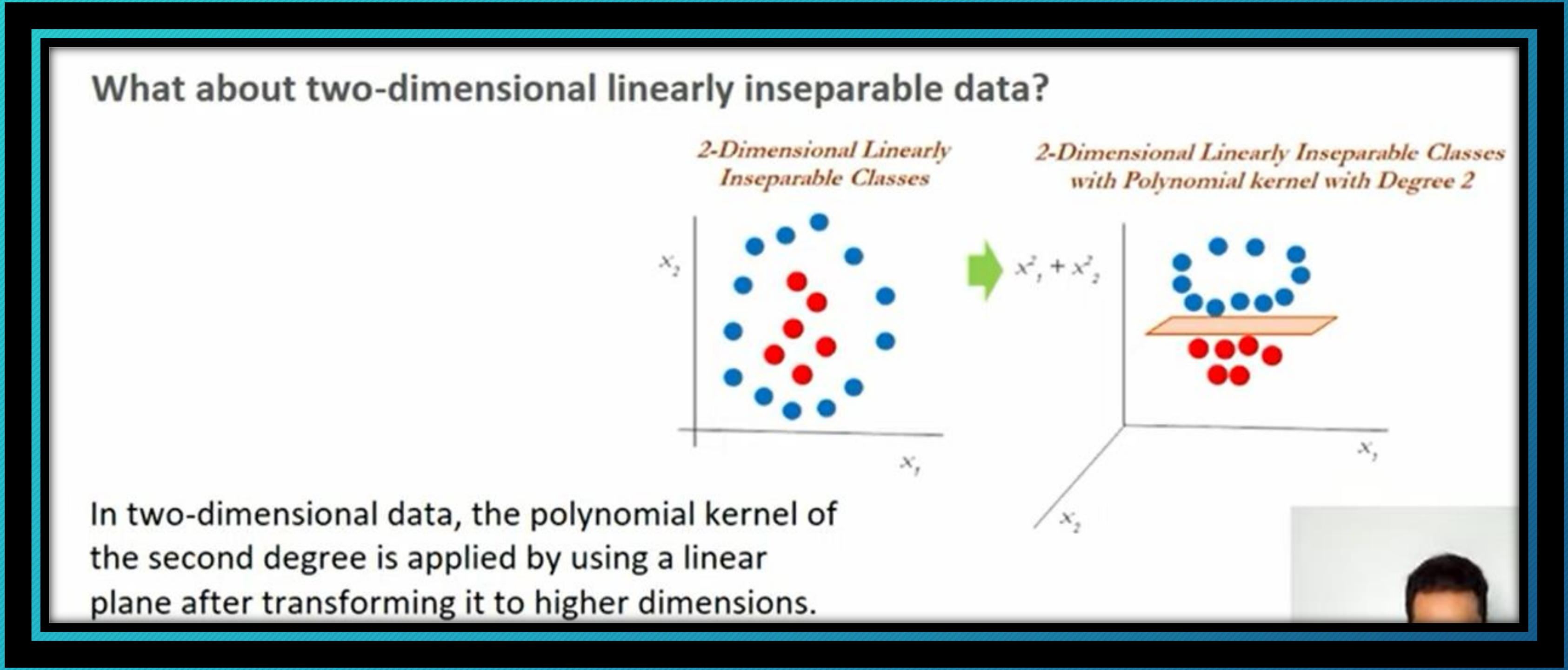


The diagram illustrates the inseparable classes in a one-dimensional and two-dimensional space.

1-d Seprable Using Kernal Function:



2-d Seprated Data Using Kernal Function:



Important Points

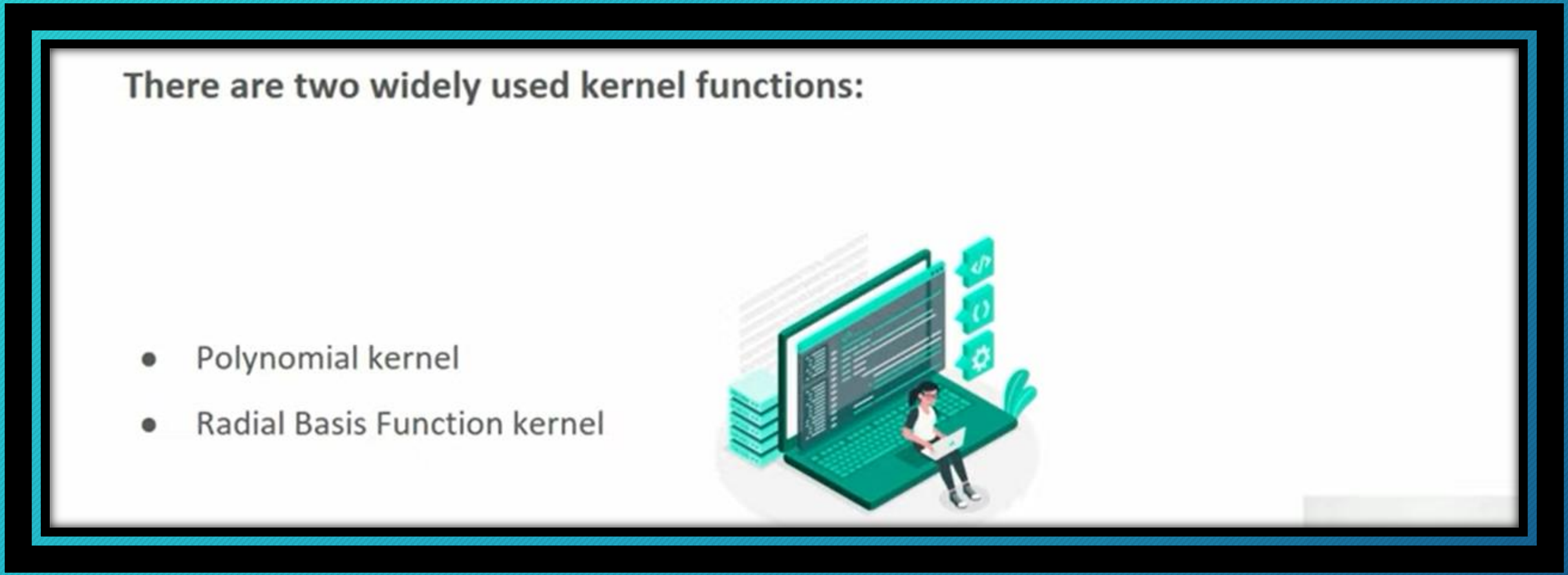
- Very flexible working with a variety of data (unstructured, structured and semi-structured)
- Overfitting is very less compared to other models.
- But training time is more if the dataset is large.
- Very popular in healthcare and banking sectors.

Kernal Functions

Kernel functions are tunable parameters in an SVM model!

- They are responsible for removing the computational requirement to achieve the higher dimensional vector space.
- Along with that they help in dealing with the non-linear separable data as we saw.

Types of Kernal Function



Polynomial Function

- A polynomial function is used with a degree 2 to separate the non-linear data by transforming them into higher dimensions.
- Take a look at the following equation:

$$K(x, x') = (1 + x * x')^k$$

RBF Kernal:

- This kernel function is also known as the Gaussian kernel function.
- It is capable of producing an infinite number of dimensions to separate the non-linear data.
- It depends on a hyperparameter ‘γ'(gamma) that needs to be scaled while normalizing the data.

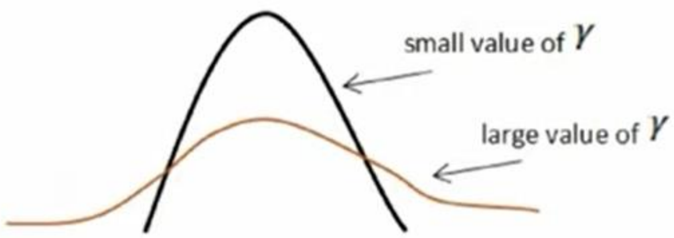
Gamma Function:

- The smaller the value of the hyperparameter, the smaller the bias and higher the variance it gives.
- While a higher value of hyperparameter gives a higher bias and lower variance solutions.
- It is explained with the help of the following equation:

$$K(x, x') = e(-\gamma \,||x - x'\,||^2\ )\,;\ \ \gamma\ = hyperparameter$$

Example Of Gamma Function:

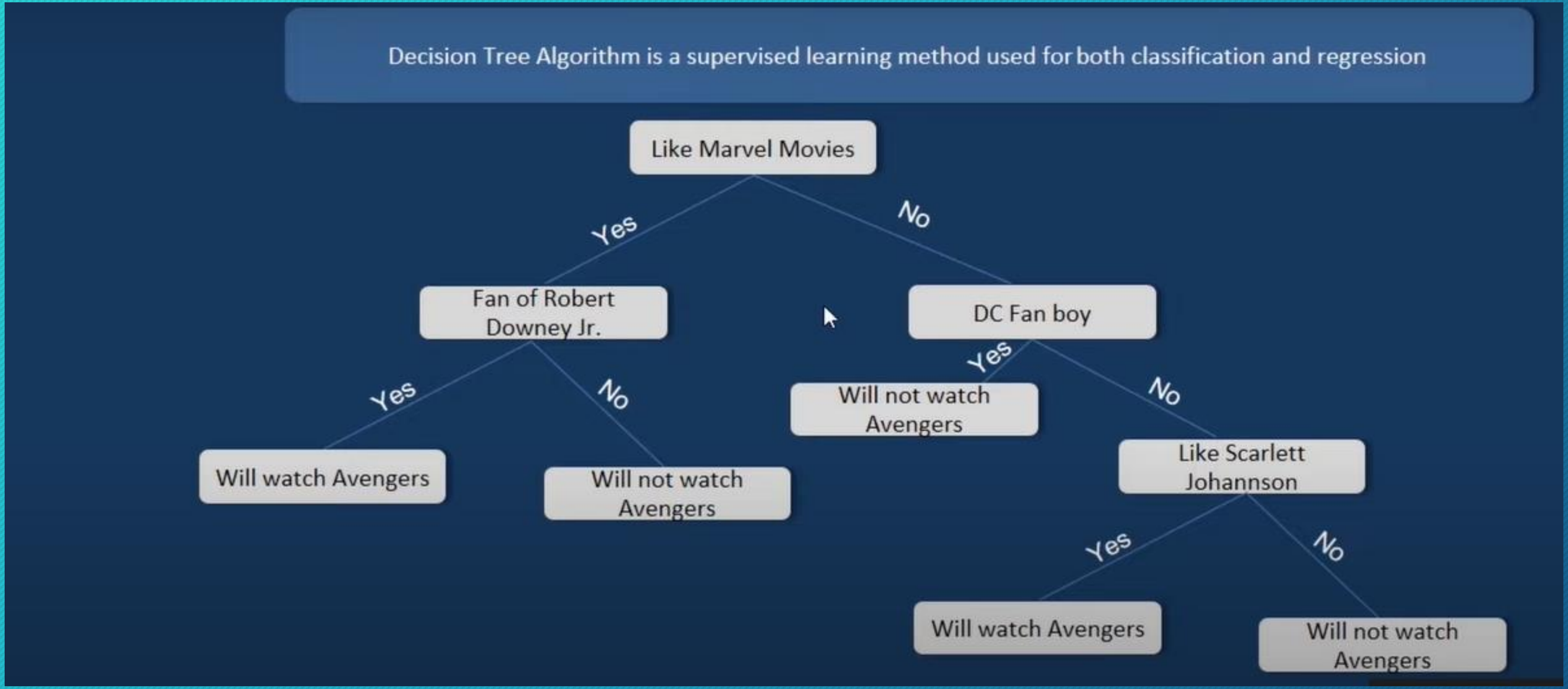
- Let us understand the impact of gamma with an example:



The large value of gamma gives us a softer and broader bump compared to the small value that gives us a pointed bump in higher dimensions.

Decision Tree

Example



Decision Tree Algorithm is a supervised learning method used for both classification and regression
Like Marvel Movies
Yes
No
Fan of Robert Downey Jr.
DC Fan boy
Yes
No
Yes
No
Will watch Avengers
Will not watch Avengers
Will not watch Avengers
Like Scarlett Johannson
Yes
No
Will watch Avengers
Will not watch Avengers

X-Y relation in it (Decision Tree - CART)

X1	X2	Y
0.2	0.3	Good
0.4	0.3	Bad
0.2	0.1	Good
0.6	0.5	Bad
0.5	0.5	Good

→ Categorical in Nature

- Spam/Not Spam
- Tumor/No Tumor
- Lend Money/Deny

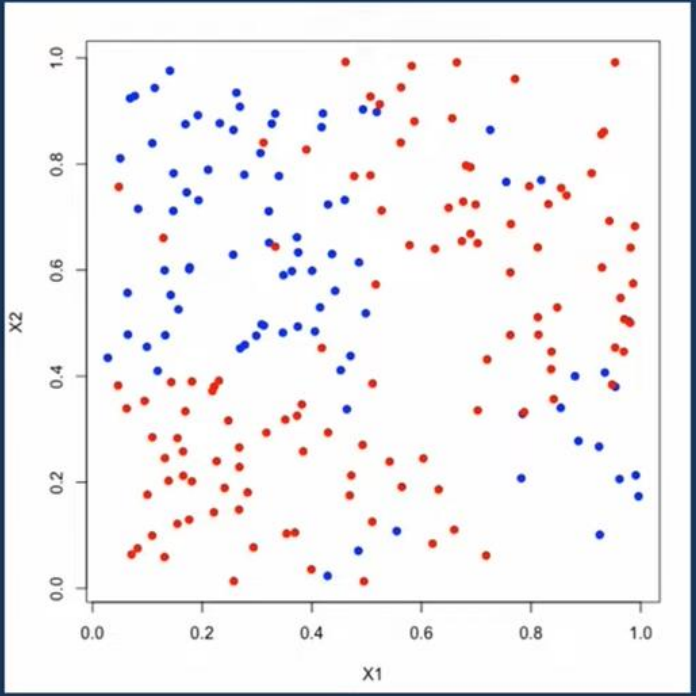
X1	X2	Y
0.2	0.3	56
0.4	0.3	34
0.2	0.1	76
0.6	0.5	12
0.5	0.5	45

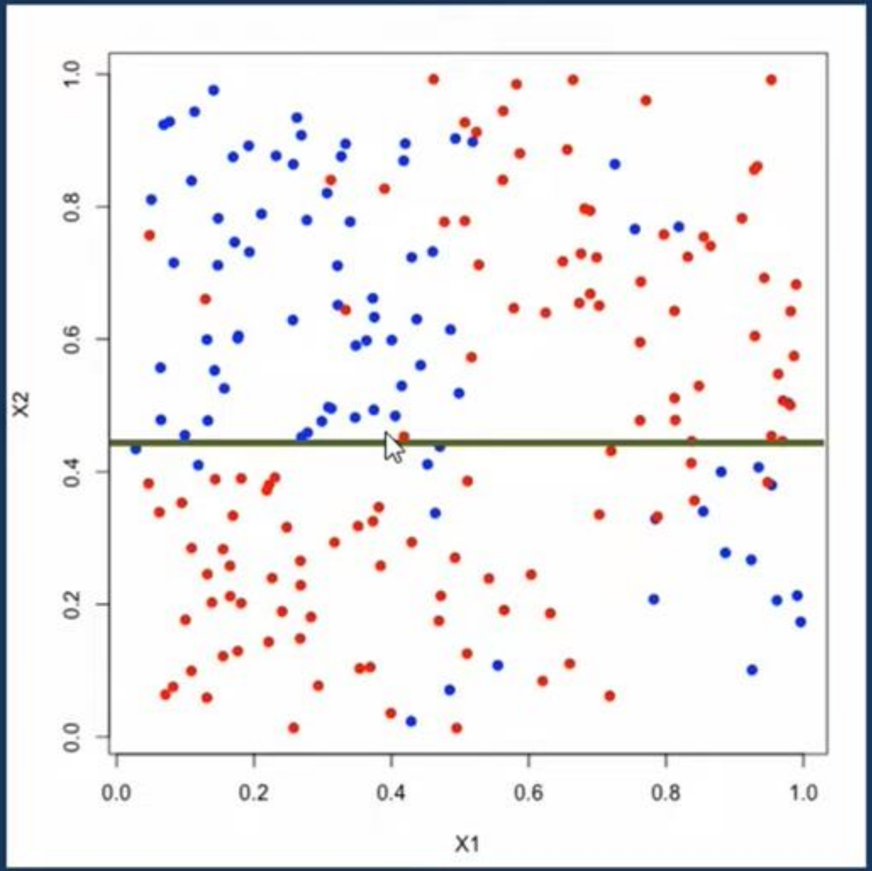
→ Numerical in Nature

- Predict Stock Returns
- Predicting Sports Scores
- Pricing a house

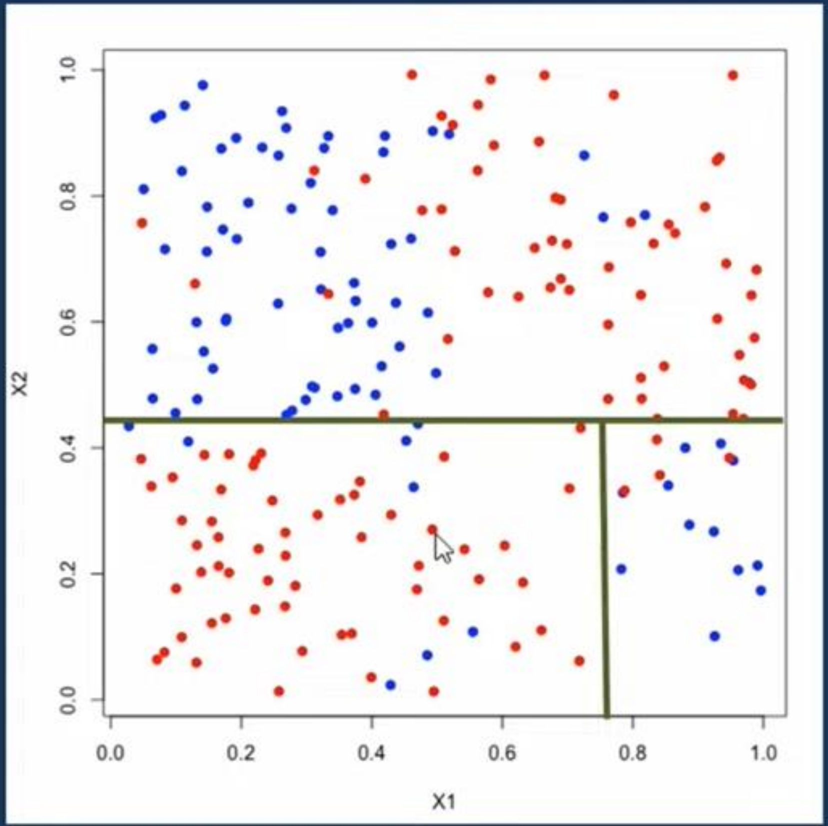
How Decision Tree Built ?

The general idea is that we will segment the space into a number of simple regions

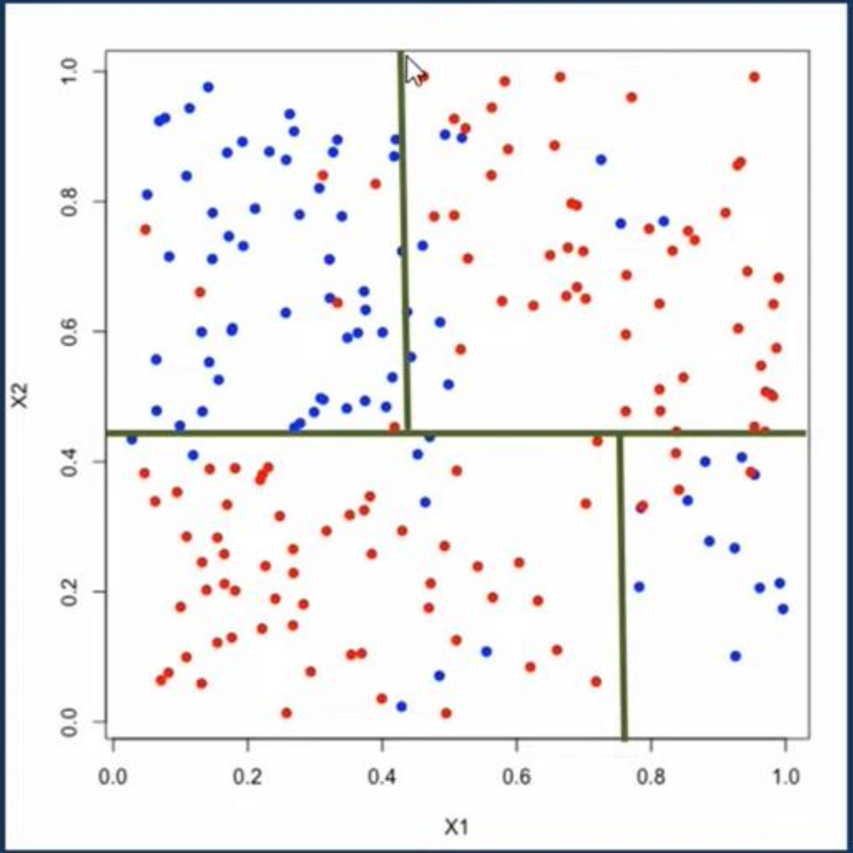




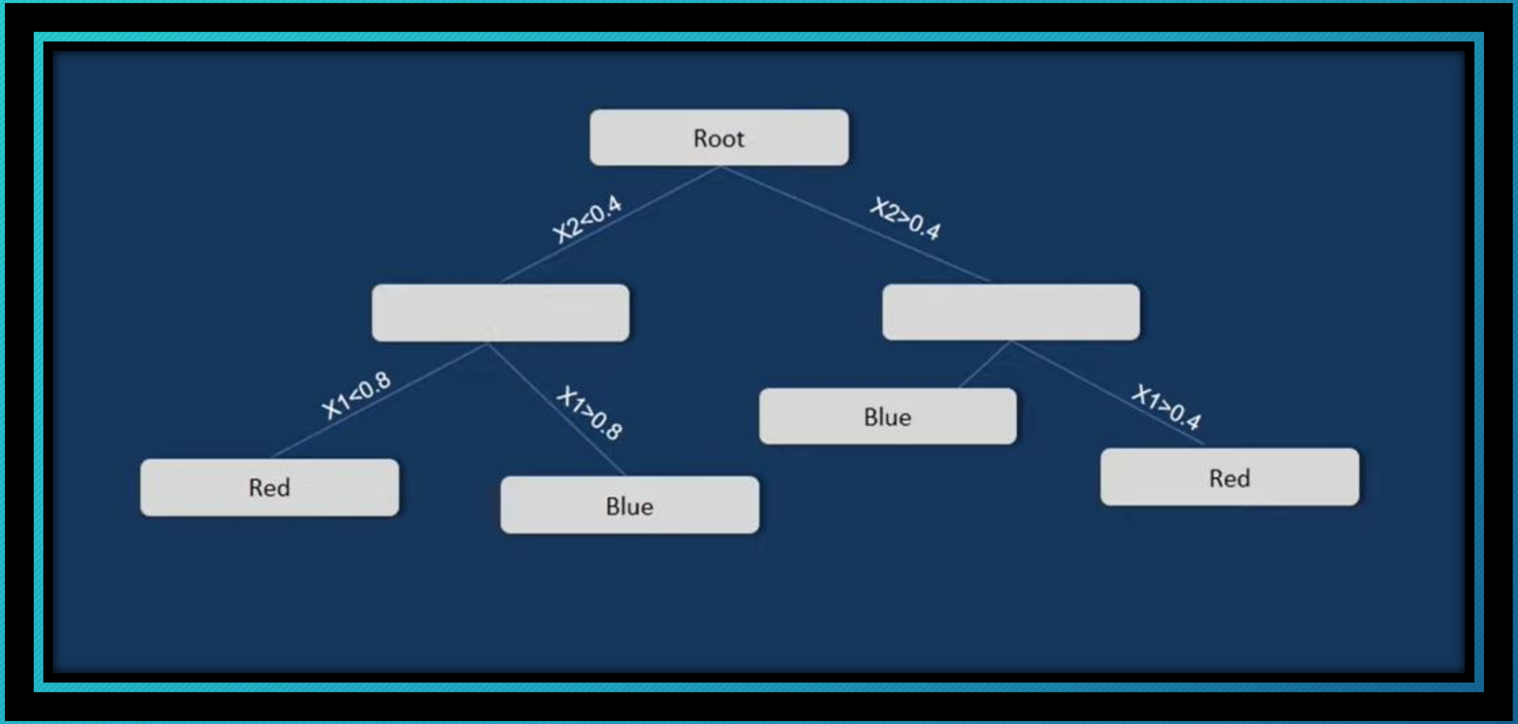
X2
X1
0.0
0.2
0.4
0.6
0.8
1.0



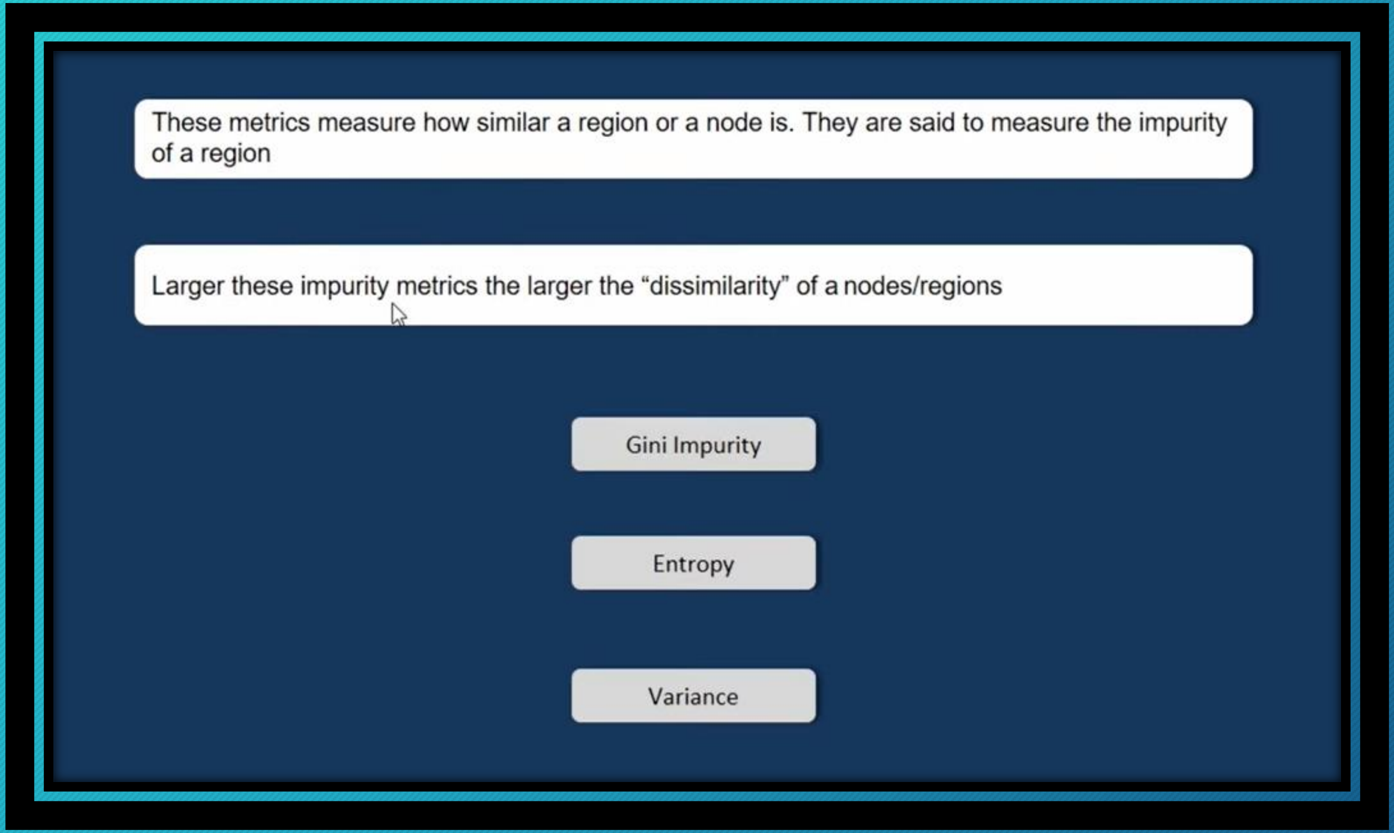
X2
X1
0.0
0.2
0.4
0.6
0.8
1.0



0.0
0.2
0.4
0.6
0.8
1.0
X1
X2


Root
X2<0.4
X2>0.4
X1<0.8
X1>0.8
Blue
X1>0.4
Red
Blue
Red

Impurity:


These metrics measure how similar a region or a node is. They are said to measure the impurity of a region
Larger these impurity metrics the larger the “dissimilarity” of a nodes/regions
Gini Impurity
Entropy
Variance

Random Forest

Tree to a Forest:

Decision trees are very sensitive to even small changes in the data - usually called unstable

Can we get a whole bunch of decision trees to work together to yield a better and more robust prediction?

Then for prediction we could use the mean for regression trees and mode for classification trees

Bagging And Random Forest