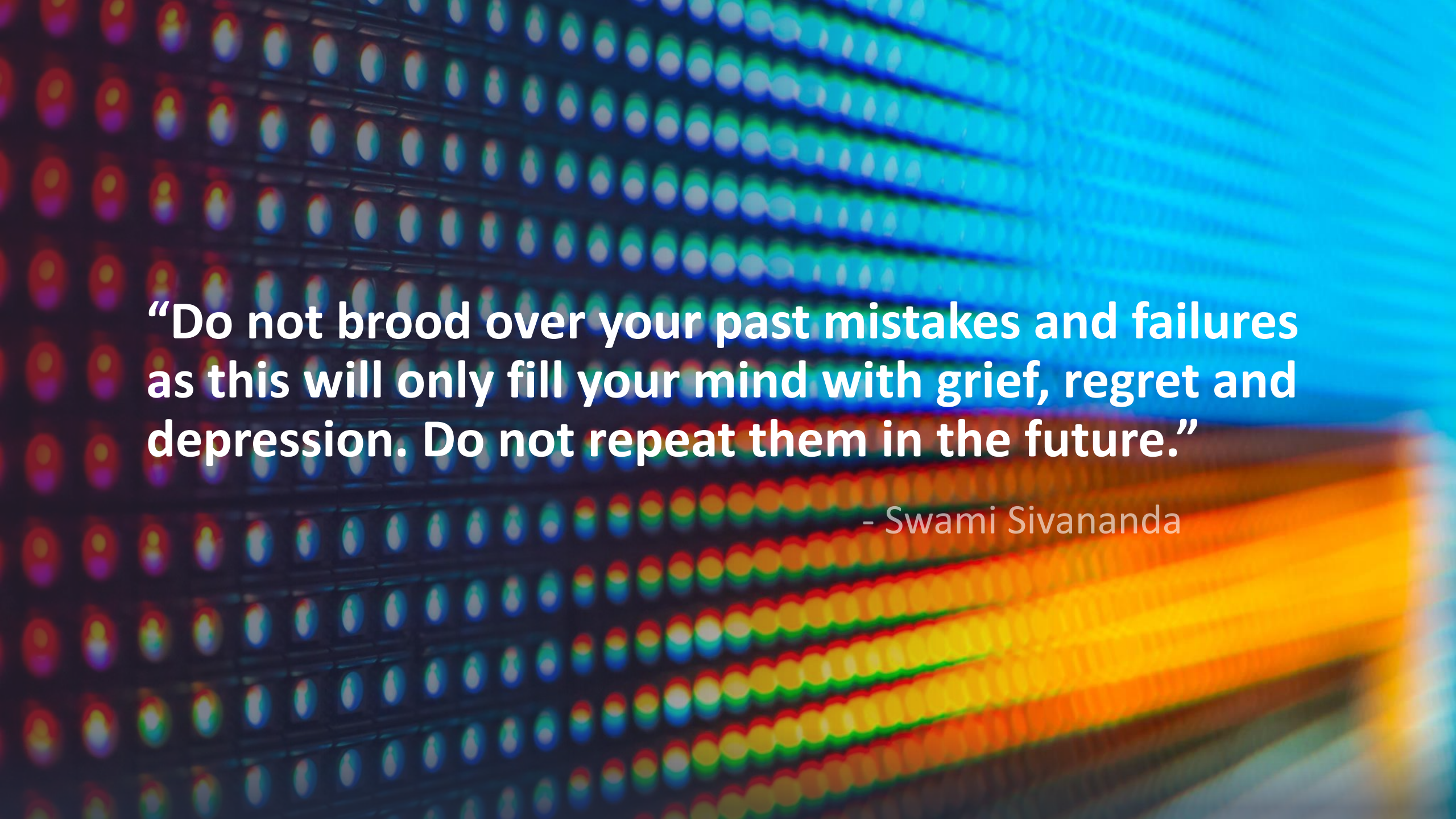

“Do not brood over your past mistakes and failures as this will only fill your mind with grief, regret and depression. Do not repeat them in the future.”
- Swami Sivananda

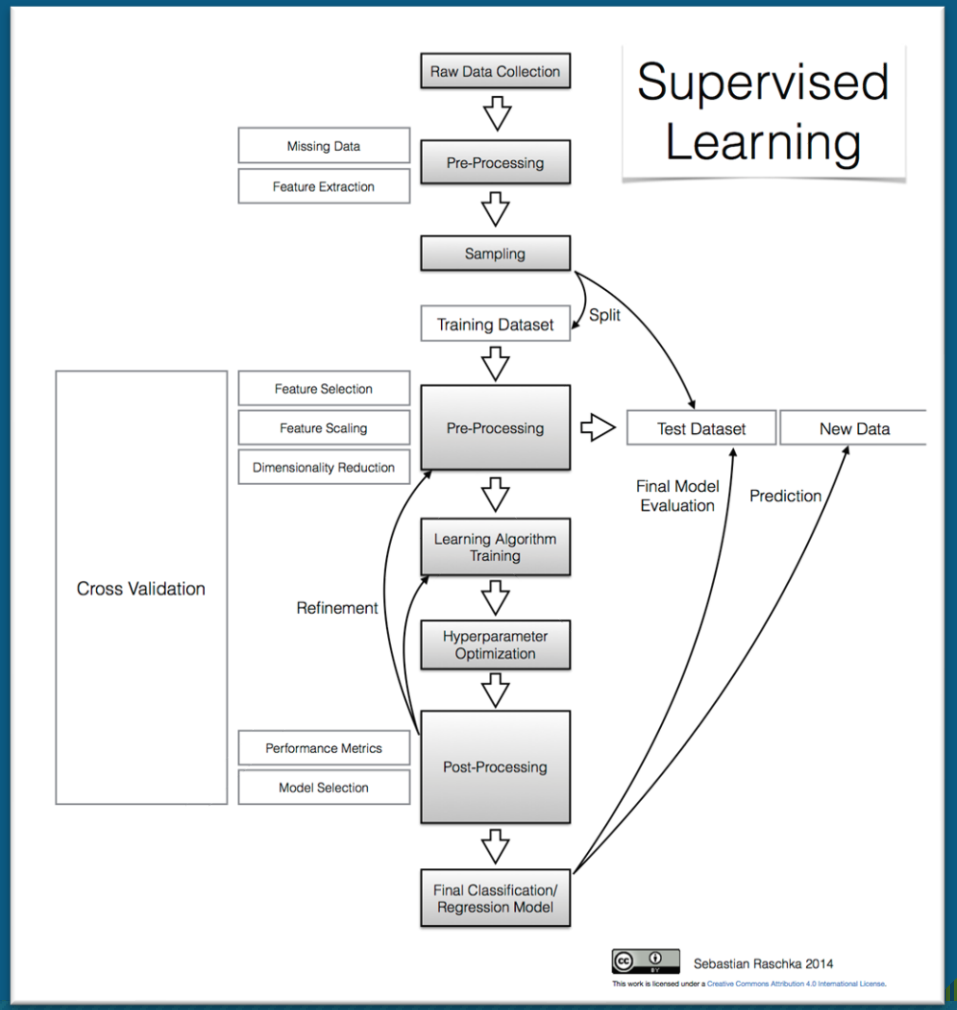
Supervised ML

- What is Supervised ML?
- Classification and Regression
- Supervised ML Algorithms

What is Supervised Learning?

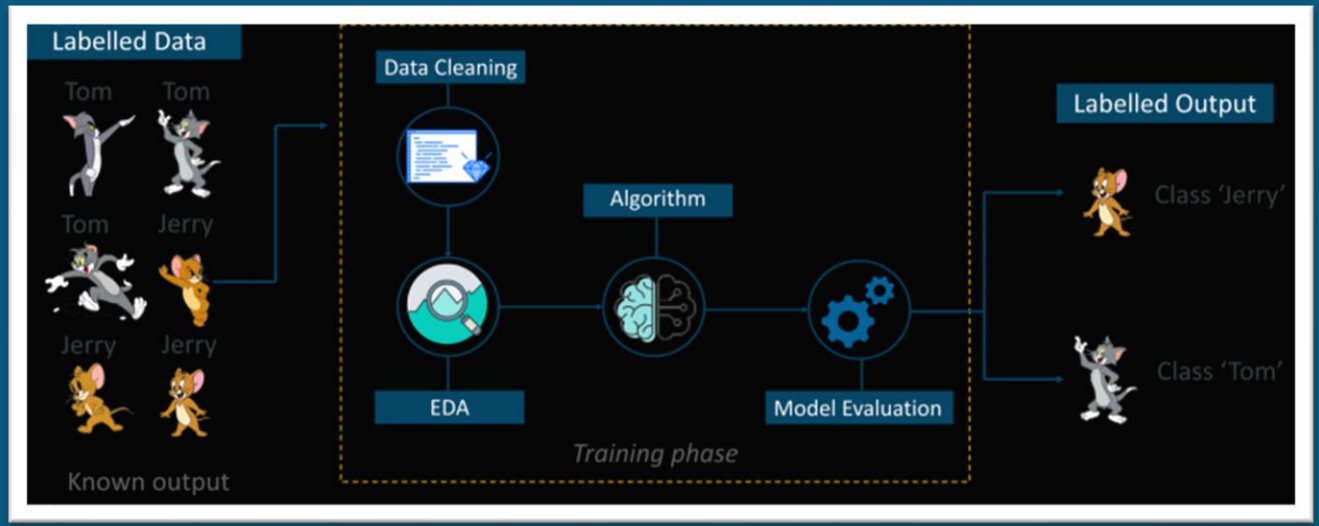
What is Supervised ML ?

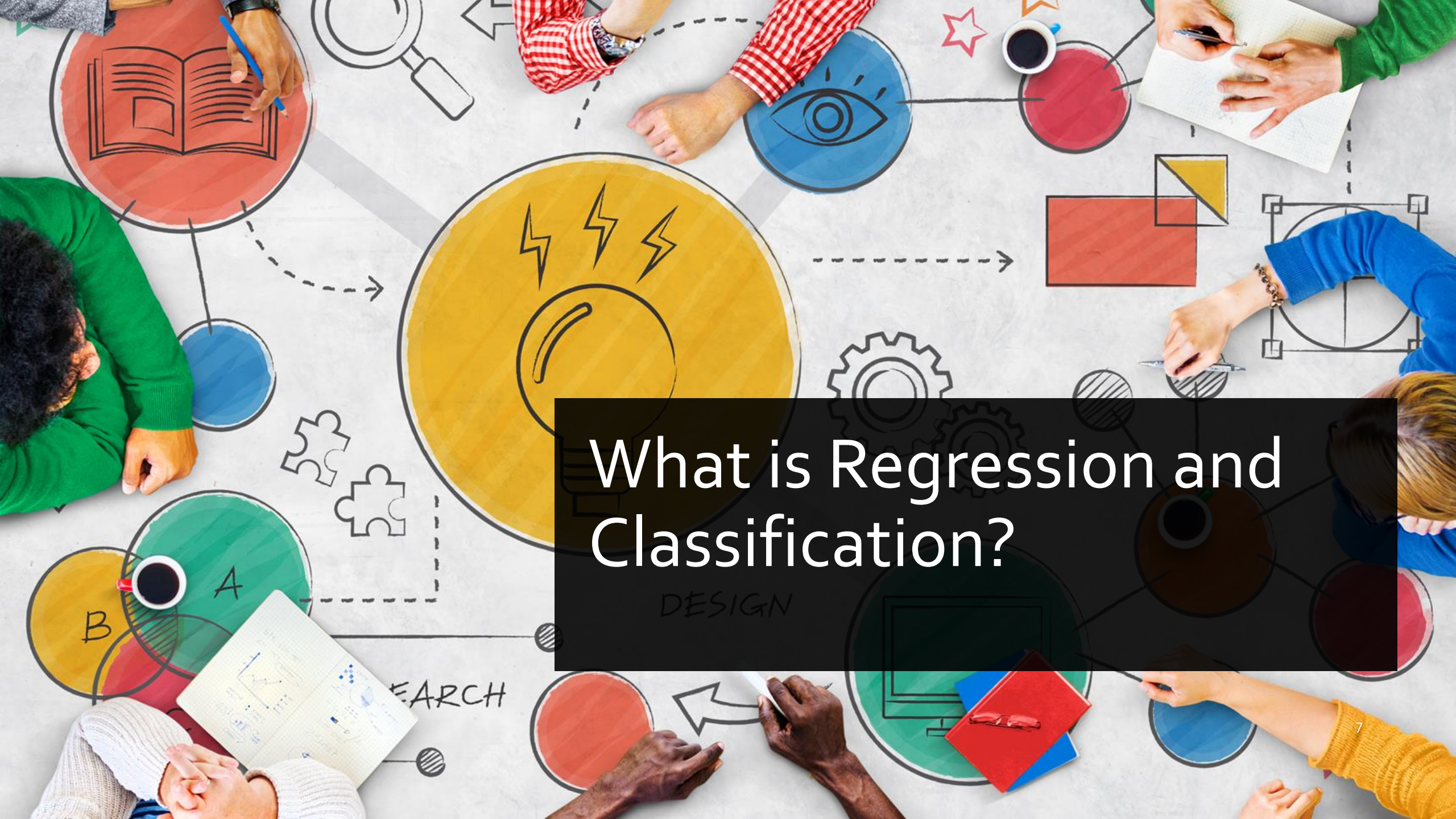
- In supervised learning, the algorithm is provided with a finite set of data which contains the right answers for each of the input values. The machine has the task to predict the right answers by analyzing the dataset correctly.



Practical Example

- As shown in the above example, we have initially taken some data and marked them as ‘Tom’ or ‘Jerry’. This labeled data is used by the training supervised model, this data is used to train the model.
- Once it is trained we can test our model by testing it with some test new data and checking of the model can predict the right output.





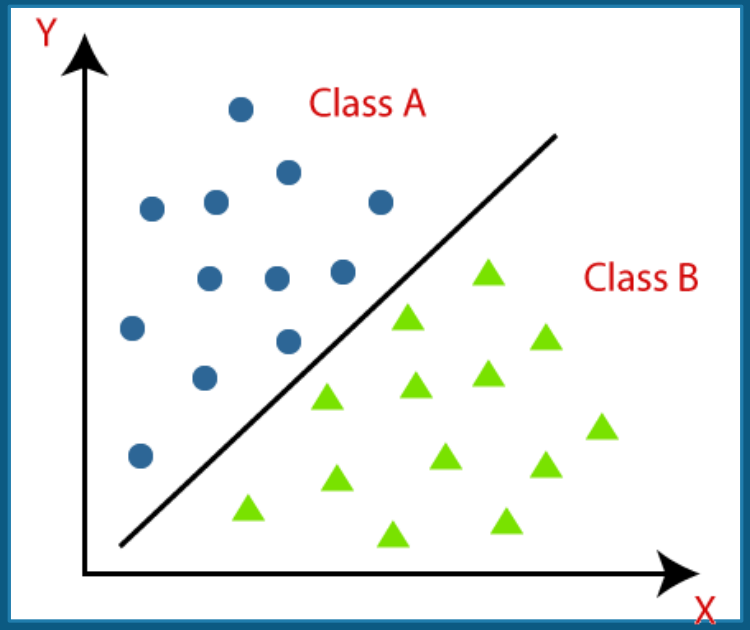
What is Regression and Classification?

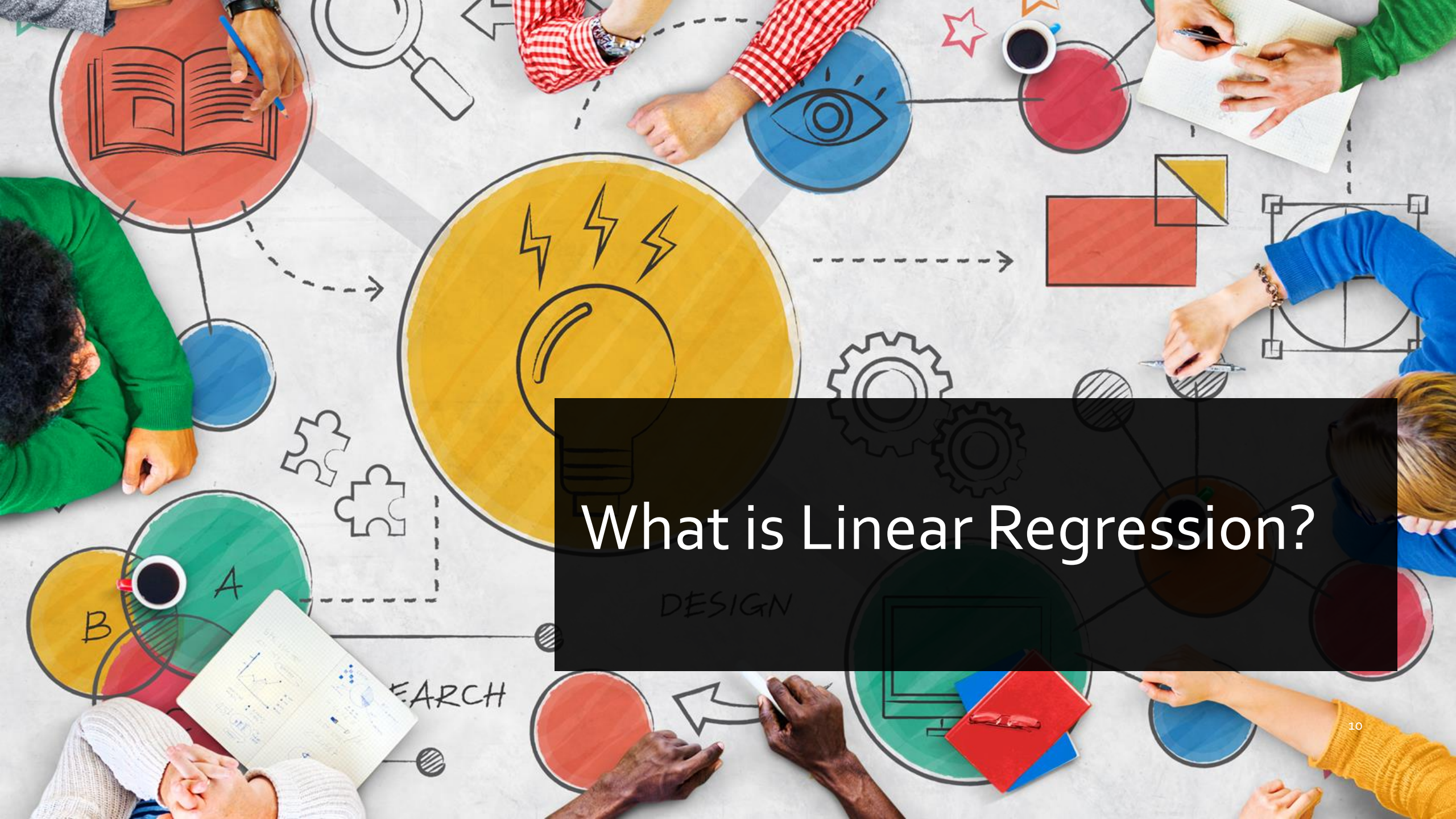
Regression

- It is a type of Supervised Machine Learning, where we have to find a relation between the one dependent variable [Y] and one or more independent variables [X1, X2, X3 ..... Xn] .
- The output here is a real value .
- It is basically a statistical method most commonly used in predictive analysis such as finance, investing and other disciplines.

Classification

- It is a type of supervised Machine Learning, where we must classify the data points into different classes based on the training data.
- The output here is a categorical value such as, 0 or 1, yes or no, true, or false, cat or dog.
- The algorithm implementing the classification on a dataset is known as classifier.





What is Linear Regression?

Linear Regression

- Linear Regression is a learning algorithm which is used to find linear relationship between the input features [X] and the single output [Y].
- In other words, we can say that 'Y' can be represented as a linear combination of features [X].

- Formally,

  $Y = a + b1*X1 + b2*X2 + ... + bn*Xn$

  here,

  Y = Dependent variable

  X1, X2, X3 .... Xn = feature vectors [Column vectors]

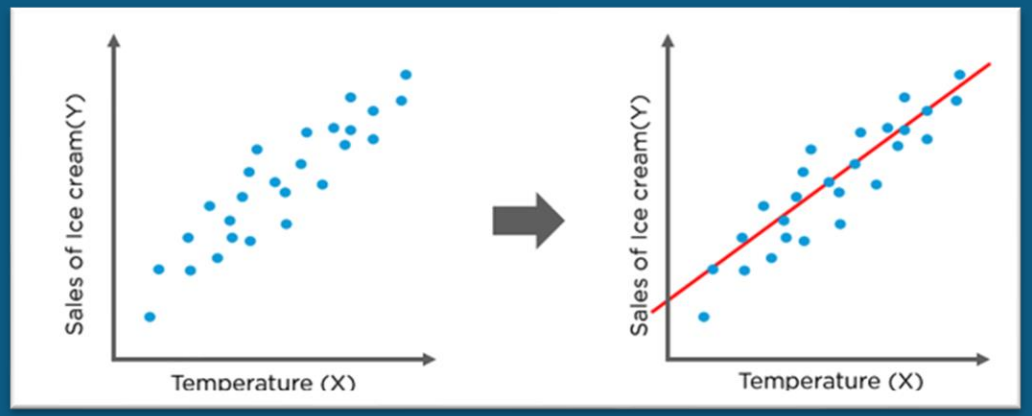
  a = intercept

  b1, b2 .... bn = slope of lines or regression coefficients

Linear Regression

What is Best Fit ?

- It can be of any shape depending on the number of independent variables.
- **Least Square Method:** The best fit is done by making sure that the sum of all the distances between the shape and the actual observations at each point is small as possible. The fit is "best" in the sense that no other position would produce less error given the choice of shape.
- The more randomly the data is spread on the scatter plot the less reliable line of best fit becomes.



Least Square Method

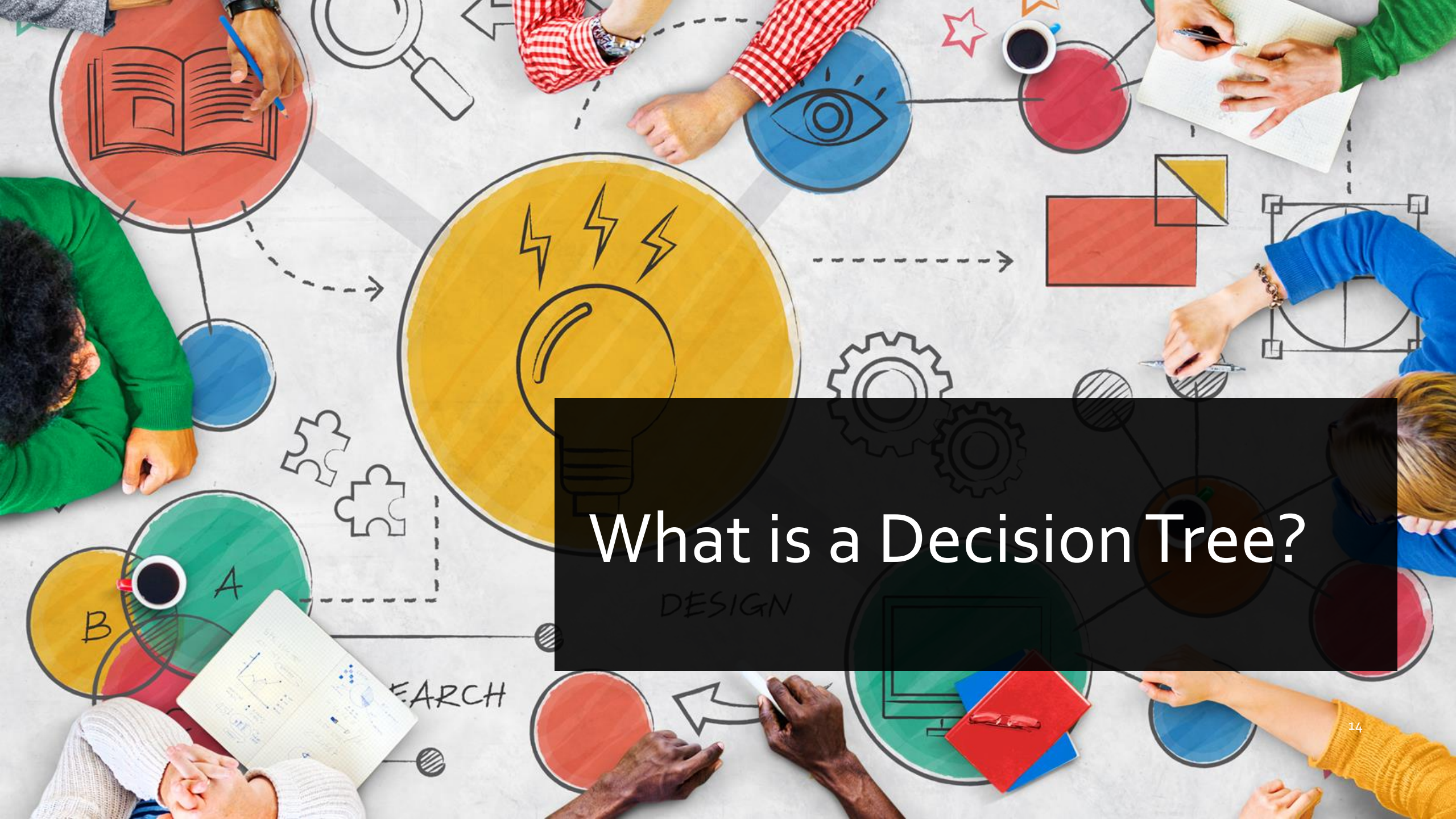
The least square regression line for the set of n data points is given by the equation of a line in slope intercept form:

$$y = a\,x + b$$

where a and b are given by

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

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

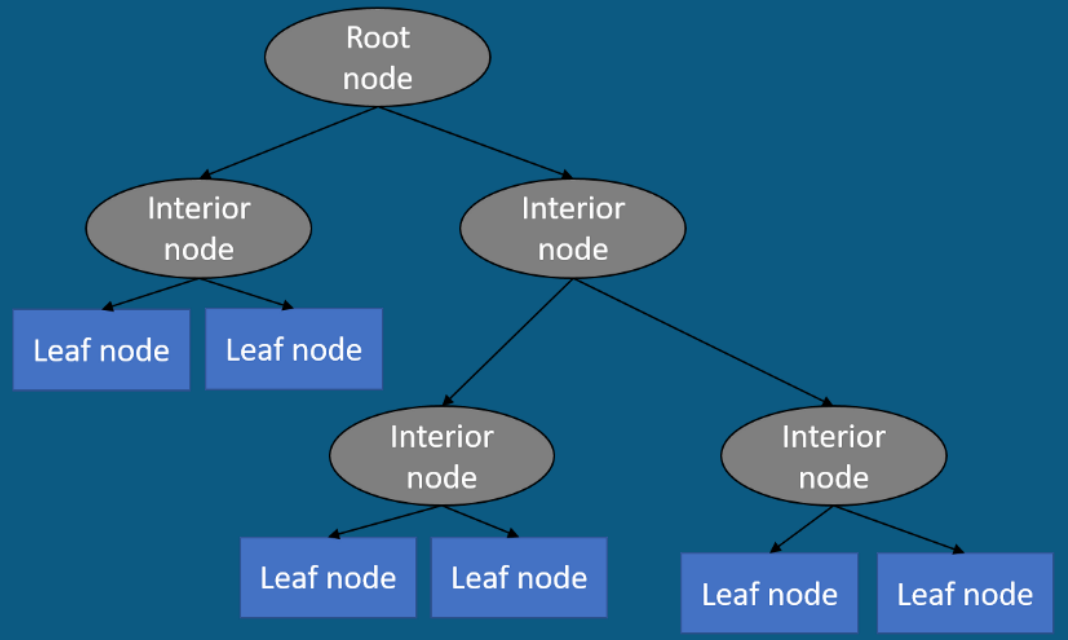


What is a Decision Tree?

Decision Trees

What is a Decision Tree ?

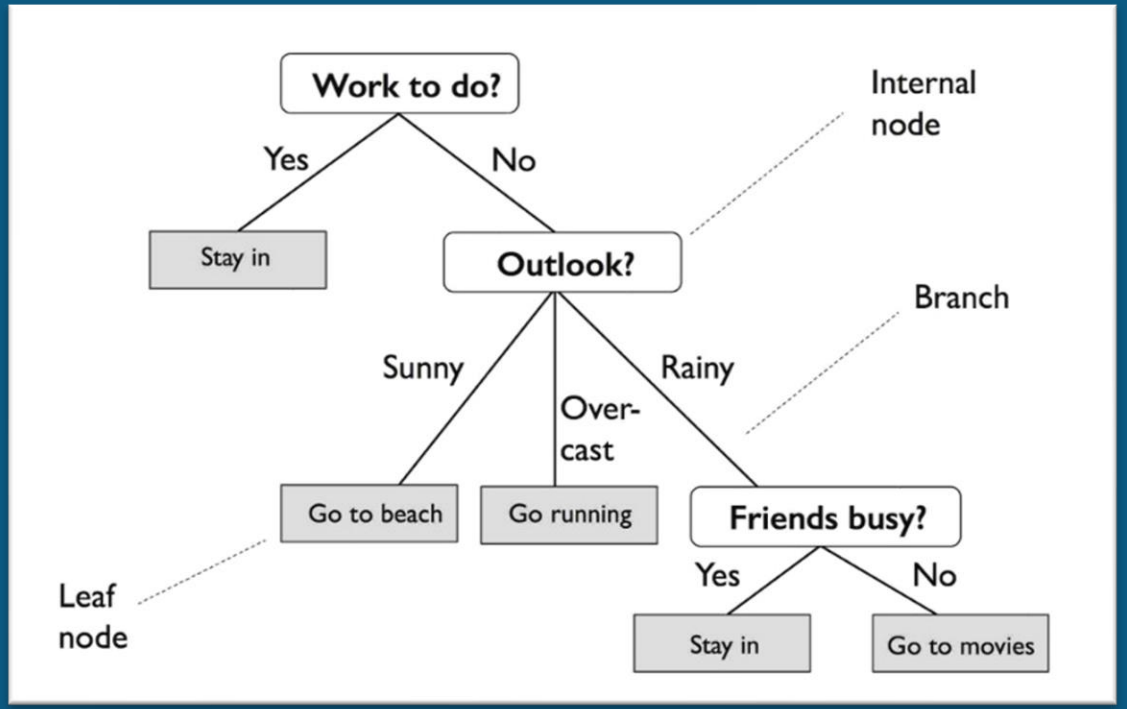
- A decision tree is tree is a tree in which consist of a root node, internal nodes and leaf nodes.
- The Root Node is the initial node which represents the entire sample and splits further into subset of samples.
- The Internal Nodes represents the features of data set.
- The Branches represent the decision rules.
- Leaf nodes represent the outcome.
- Represents all possible solutions to a given problem.
- Used for both classification as well as regression.



Decision Trees

How does Decision Tree algorithm work ?

- The algorithm starts from the root node, containing the complete dataset.
- It tries to find the best attribute from the given dataset using Attribute Selection Measure (ASM)
- Then, the algorithm divides the root node into subtrees representing subsets to the given problem.
- This process continues recursively until the algorithm reaches a point where it cannot further classify the nodes.



Decision Trees

Attribute Selection Measures [ASM]

- **Information Gain**
  - Calculates how much information a feature provides us about a class.
  - According to its value, the tree is split into further nodes.
  - The algorithm always tries to maximize the information gain in order to get better splits resulting into accurate outcome.

    IG = Entropy [S] – [(Weighted Avg) * Entropy{Xi}]
  - Entropy is the measure of randomness in the data. In other words, It tells us about the impurity in a given attribute.

    Entropy [S] = -P(yes) * log2[P(Yes)] – P(no)*log2[P(no)]

    here,

    S = total no. of samples

    P(yes) = probability of yes

    P(no) = probability of no

Decision Trees

Attribute Selection Measures [ASM]

- **Gini Index**
  - It is the measure of purity or impurity while creating a decision tree.
  - An attribute of low Gini index is preferred over high Gini index.
  - Only used for binary splits.

$$GI = 1 - \sum_j (Pj)^2$$

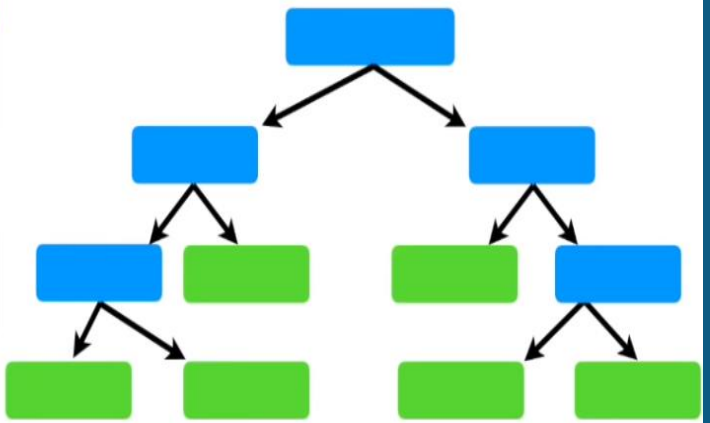
Pruning

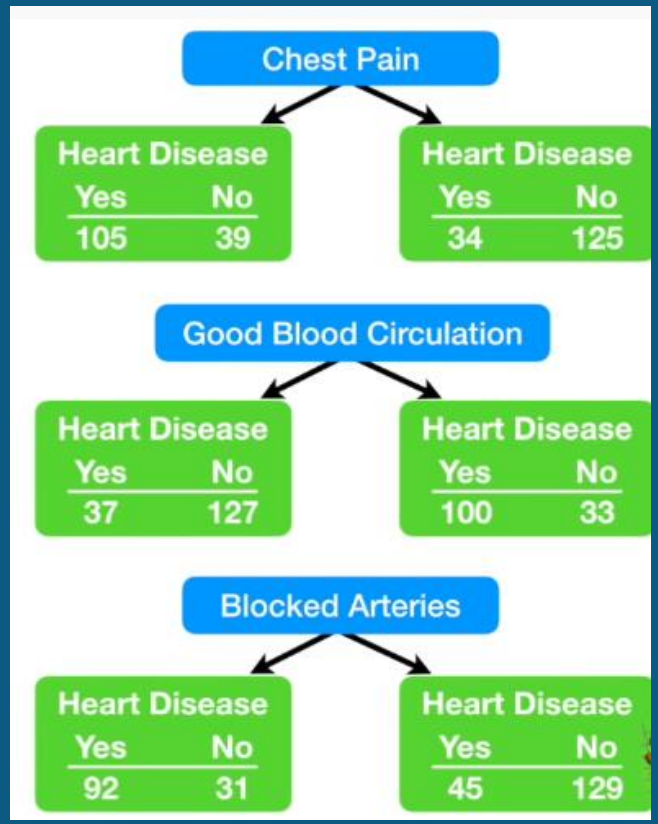
- While creating a decision tree, it may happen that the tree may become too large, increasing the risk of overfitting.
- Thus, Pruning is used to decrease the size of tree without decreasing its accuracy by removing the unnecessary nodes from the tree.

Practical Example

Given the data for chest pain, blood circulation and blocked arteries we need to accurately classify, if a new sample is heart disease or not.

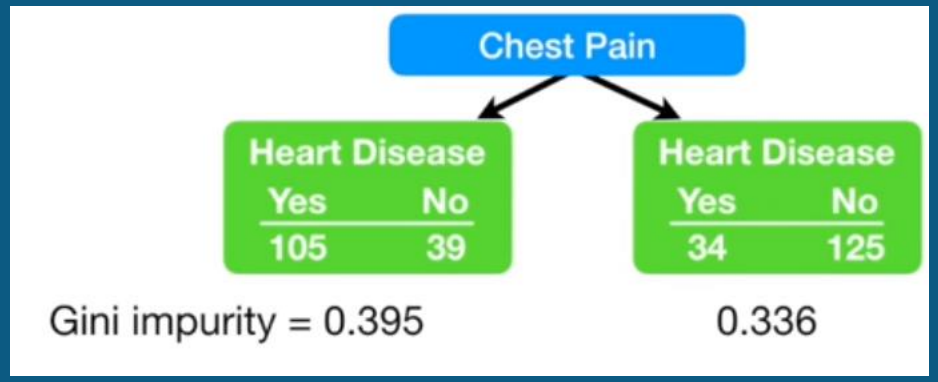
Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc...	etc...	etc...	etc...





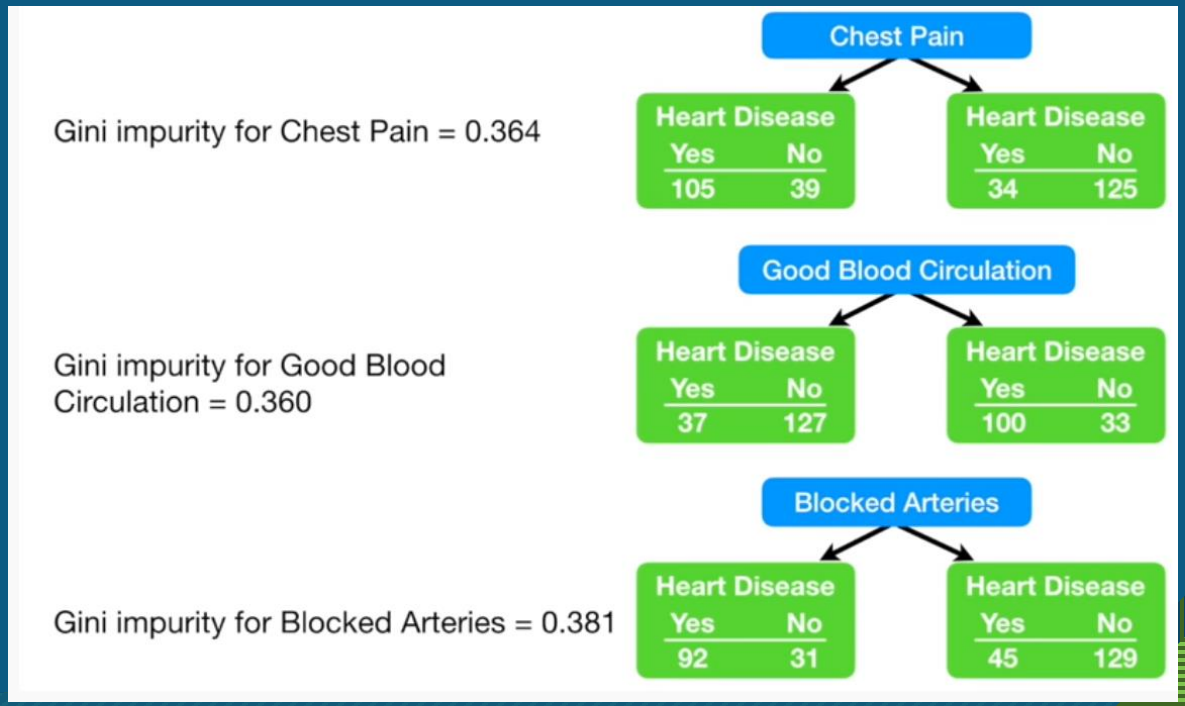
Practical Example

As, we have previously seen every feature has some impurity. Thus, we will use Gini Index and Information Gain to define which feature will be our root node.



$$= \left(\frac{144}{144 + 159}\right) 0.395 + \left(\frac{159}{144 + 159}\right) 0.336$$

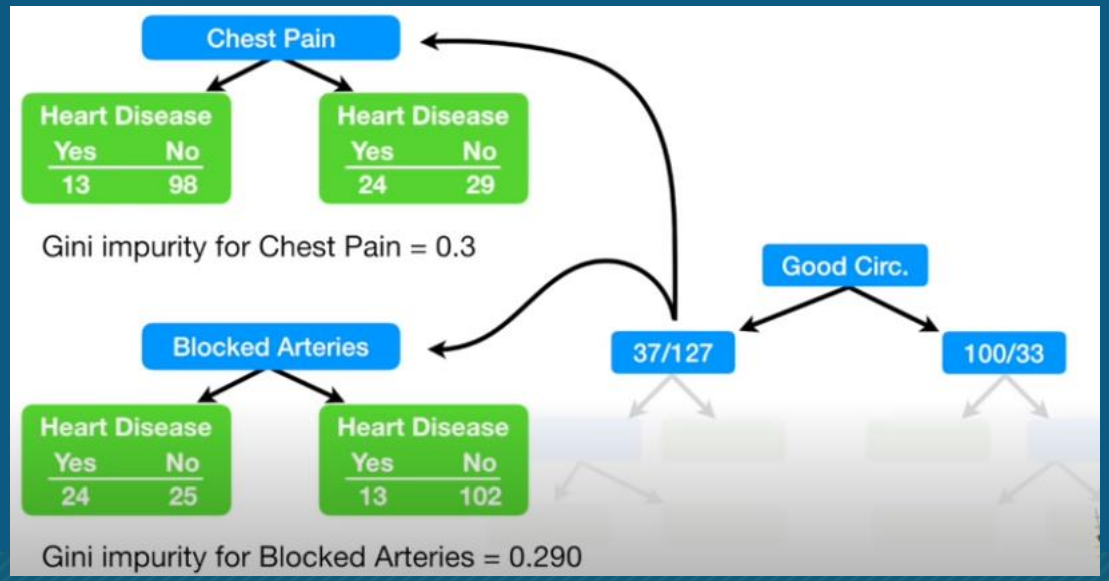
$$= 0.364$$



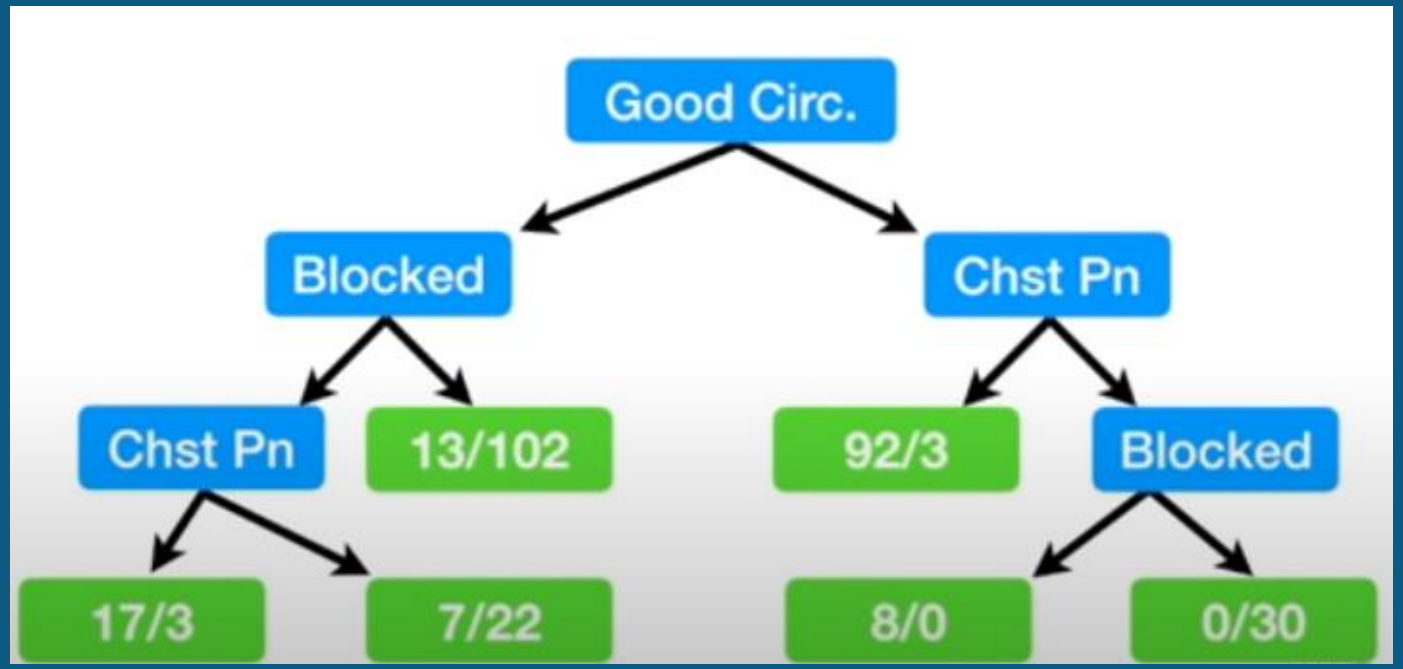
Practical Example

Criteria for splitting:

- Calculate all of the Gini impurity scores.
- If the node itself has the lowest score, than there is no point in separating the patients any more and it becomes a leaf node.
- If separating the data results in an improvement, than pick the separation with the lowest impurity value.



Practical Example

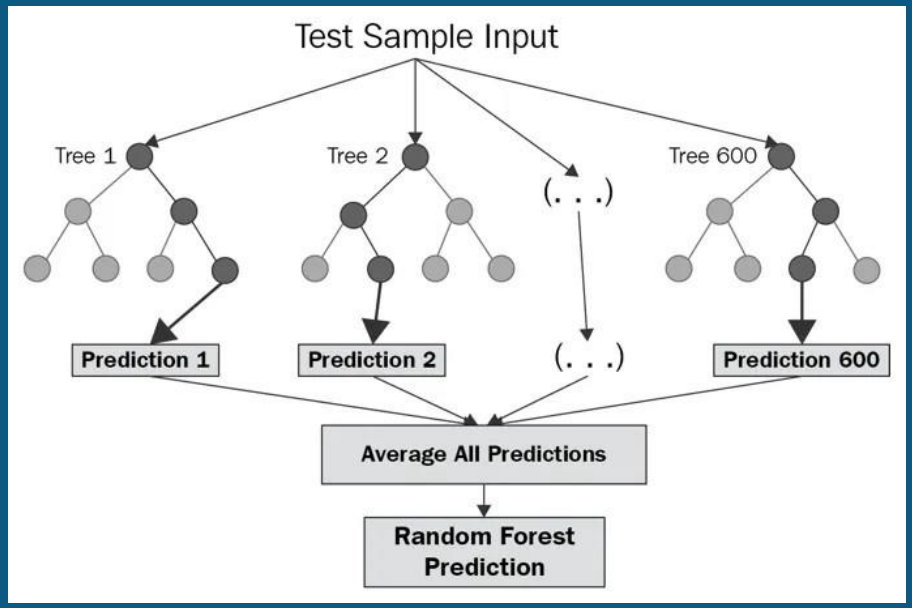


What is a Random Forest?

Random Forest

What is a Random Forest ?

- Random Forest is a Supervised Machine Learning algorithm which builds several decision trees and merges them together to get a more accurate and stable prediction.
- It builds an ensemble of decision trees, usually trained with the “bagging” method.
- The general idea behind the bagging method is that a combination of learning models increases the overall result.
- It can be used for both classification as well as regression.
- It has the same hyper parameters as a decision tree or bagging classifier.
- Instead of searching for the most important feature while splitting a node, Random Forest searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.



Random Forest

Difference between Random Forest and Decision Tree

- If we input a training dataset with features and labels into a decision tree, it will formulate some set of rules, which will be used to make the predictions. In comparison, the random forest algorithm randomly selects observations and features to build several decision trees and then averages the results.
- Another difference is “deep” decision trees might suffer from overfitting. Most of the time, random forest prevents this by creating random subsets of the features and building smaller trees using those subsets. Afterwards, it combines the subtrees.

Random Forest

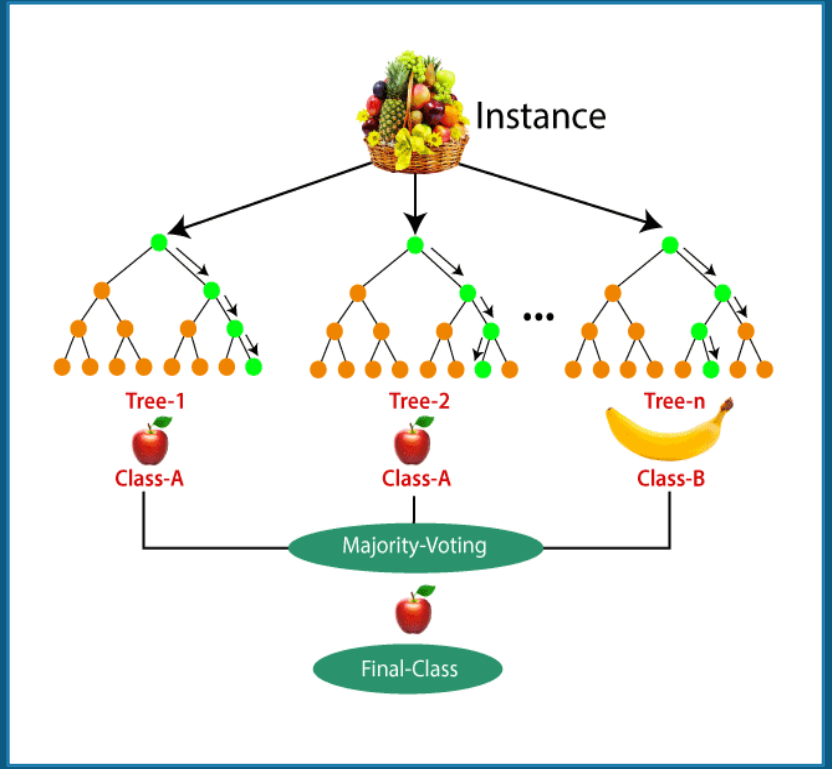
How does Random Forest algorithm work?

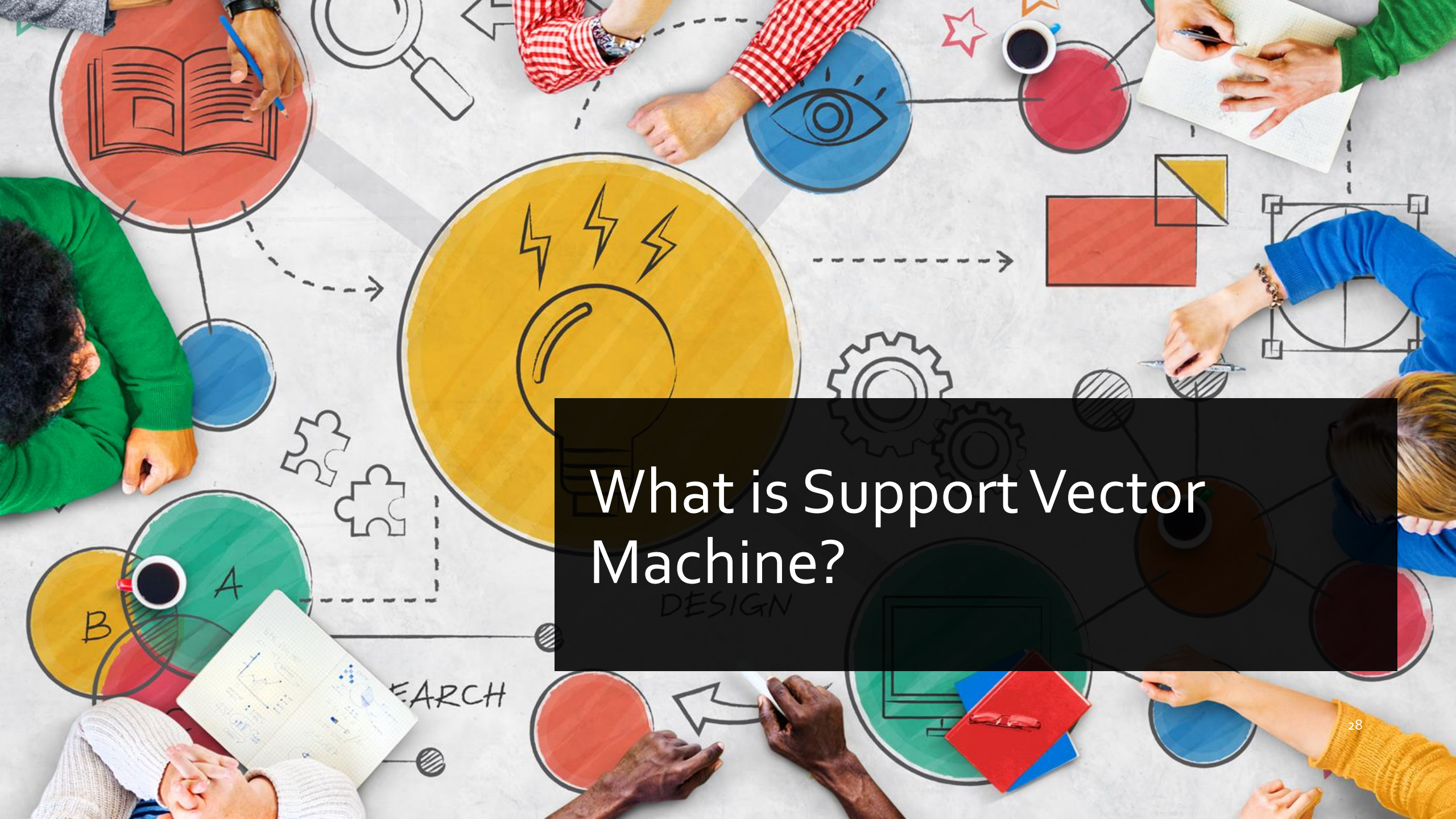
- Select random K data points from the training set.
- Build the decision trees associated with the selected data points (Subsets).
- Choose the number N for decision trees that we want to build.
- Repeat points 1 & 2.
- For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Random Forest

How does Random Forest algorithm work?

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on most results, the Random Forest classifier predicts the final decision.



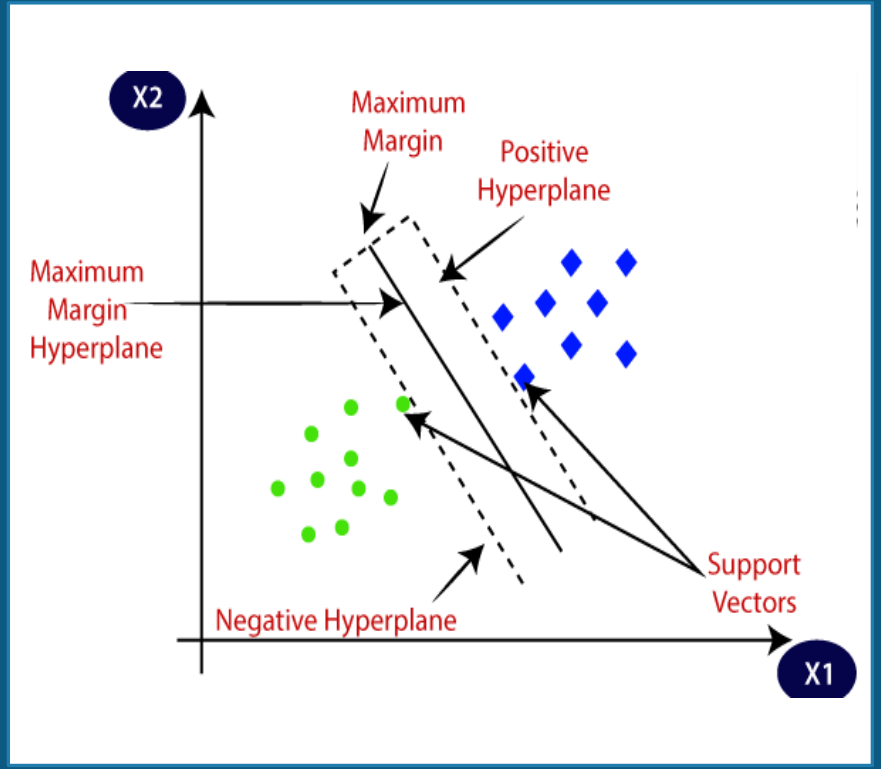


What is Support Vector Machine?

Support Vector Machine

What is a Support Vector Machine?

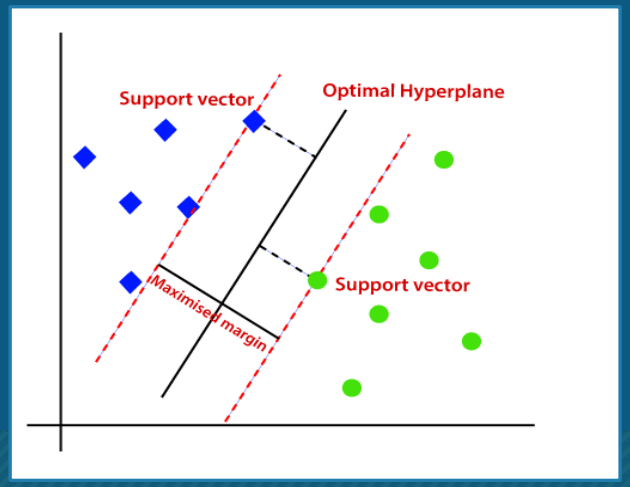
- It is a supervised machine learning algorithm which is used for both Classification as well as Regression. However, primarily it is used for Classification problems in Machine Learning.
- SVM creates the best line or hyperplane boundary that can segregate n-dimensional space into classes so that new data points can be easily put in the correct category.
- SVM choses extreme vectors to create the hyperplane. These extreme vectors are called support vectors.

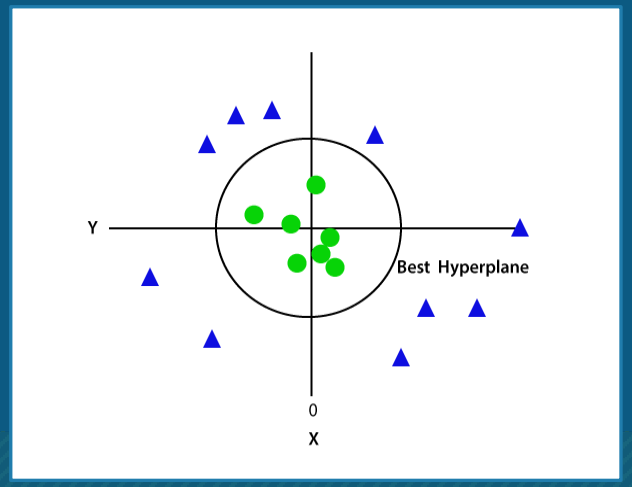


Support Vector Machine

Types of SVM

- **Linear SVM:** Linear SVM is used for linearly separable data. It means that the data can be classified into two classes using a straight line.
- **Non – Linear SVM:** It is used for non-linearly separable data. It means that the data cannot be classified into two classes using a straight line.





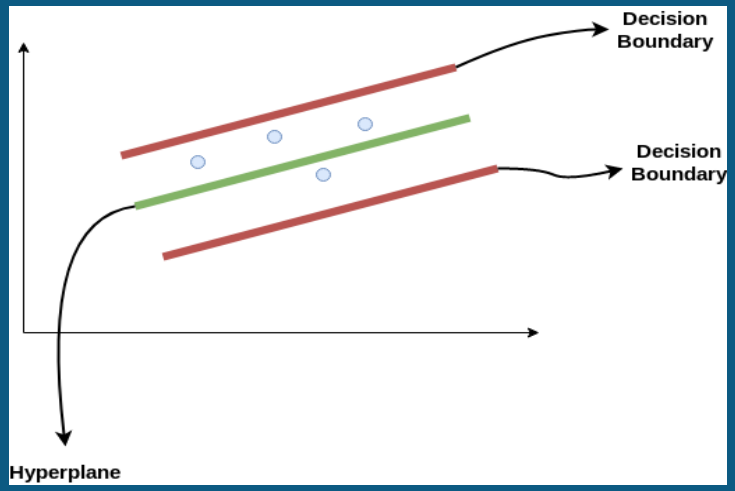
Support Vector Machine

Hyperplane and Support Vectors in SVM

- **Hyperplane:** There can be multiple decision boundaries which will be able to segregate the classes in n-dimensional space. The best decision boundary that helps to classify the data points is called as Hyperplane.
- **Support Vectors:** The data points that are closest to the hyperplane are called as Support Vectors. These vectors affect the position of the hyperplane and support the hyperplane.

How does SVM works as a Regressor?

Consider these two red lines as the decision boundary and the green line as the hyperplane. **Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line.** Our best fit line is the hyperplane that has a maximum number of points.



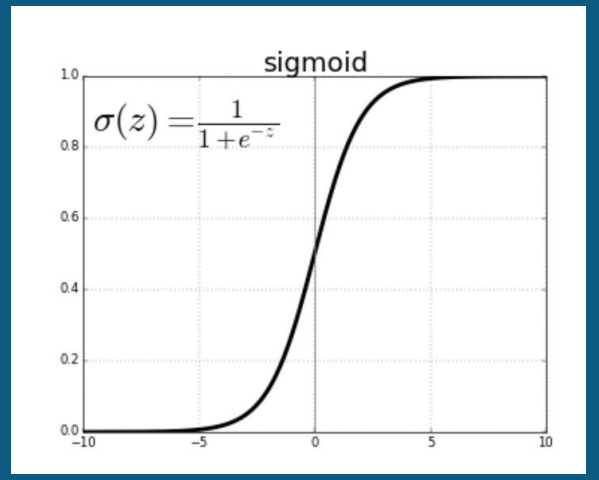
What is Logistic Regression?

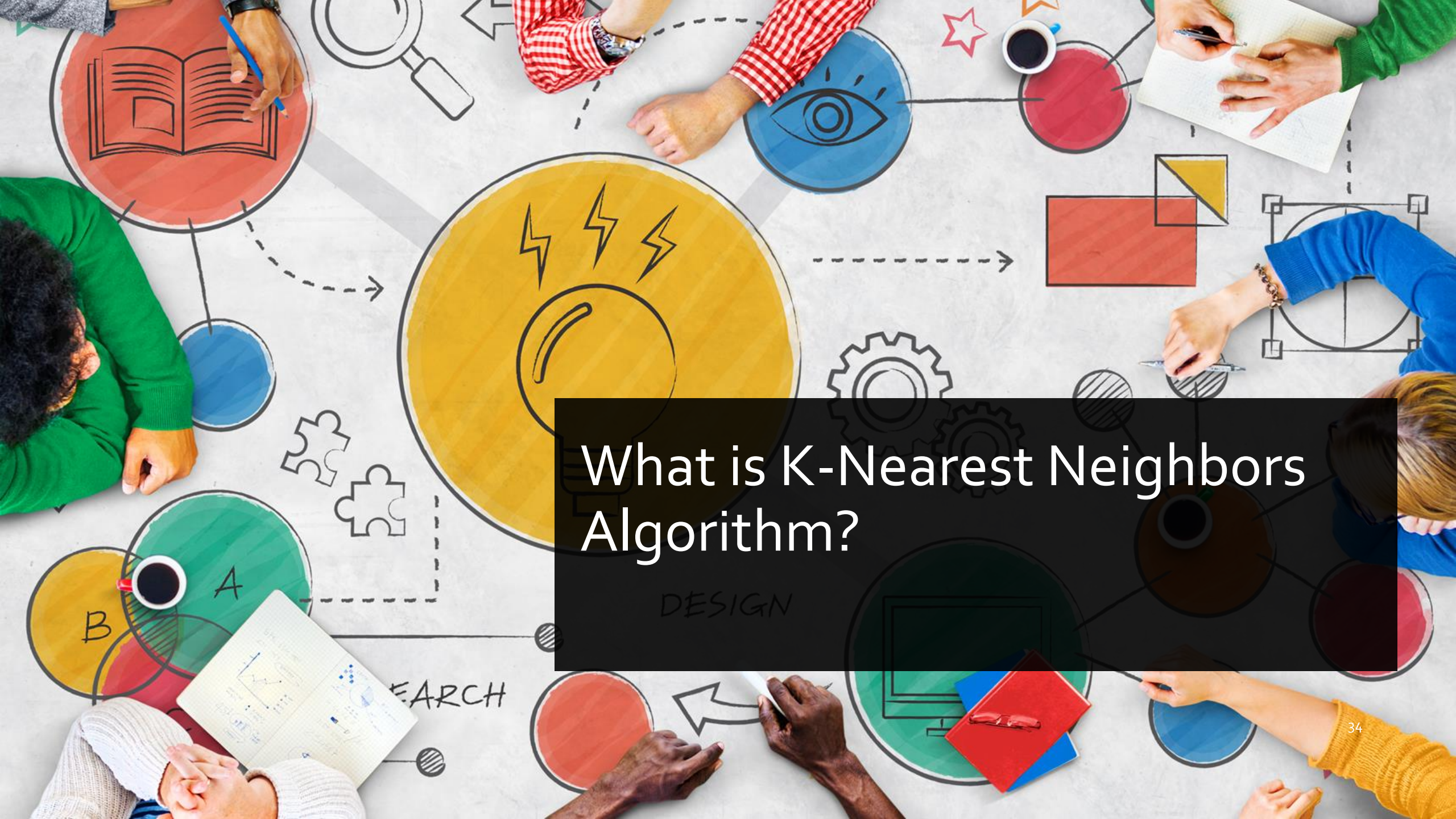
Logistic Regression

What is Logistic Regression?

- Logistic regression is used for predicting the categorical dependent variable using a given set of independent variables.
- The outcome here is a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- It is used for solving classification problems.
- The logistic function used is called as Sigmoid Function
- The logistic Regression function is given by : -

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$



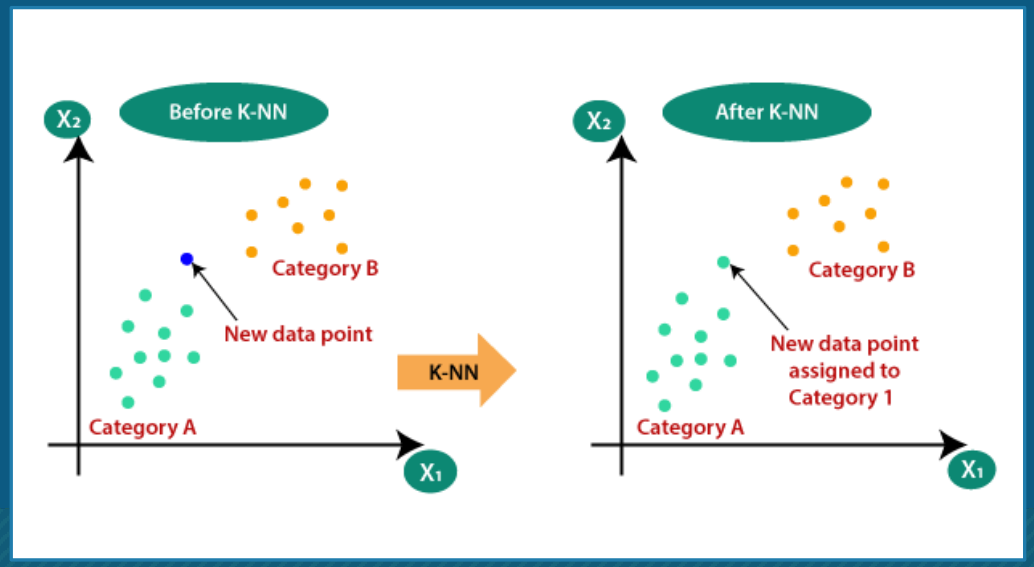


What is K-Nearest Neighbors Algorithm?

K-Nearest Neighbors

What is K-Nearest Neighbour Algorithm?

- KNN algorithm is used to classify the data points based on previously stored data points. It puts the new data point into the category that is most like the available categories.
- It is a non-parametric algorithm, which means it does not make any assumption about the data.
- It is also called lazy learner algorithm as it does not learn from the training set immediately instead stores the dataset and at the time of classification, it performs an action the dataset.



K-Nearest Neighbors

How does KNN work ?

- Select the number K of the neighbours.
- Calculate the distance of **K number of neighbours.**
- Take the K nearest neighbours as per the calculated distance.
- Among these k neighbours, count the number of the data points in each category.
- Assign the new data points to that category for which the number of the neighbour is maximum.

How to choose the value of 'K' ?

There is no method described to find the value of 'K'. The optimal choice of the value is highly data-dependent: in general, a larger suppresses the effects of noise, but makes the classification boundaries less distinct. A lower value can be noisy and lead to the effect of outliers in the model.

K-Nearest Neighbors

Decision Metrices in KNN

- **Makowski Distance:**
  - It is a generalized distance metrics and is used for real-valued vector spaces.
  - The p value in the formula can be manipulated to give us different distances like:
  - p = 1, when p is set to 1, we get Manhattan distance
  - p = 2, when p is set to 2, we get Euclidean distance

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

- **Manhattan Distance:** This distance is calculated between two points as the sum of the absolute difference of their cartesian coordinates.

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

K-Nearest Neighbors

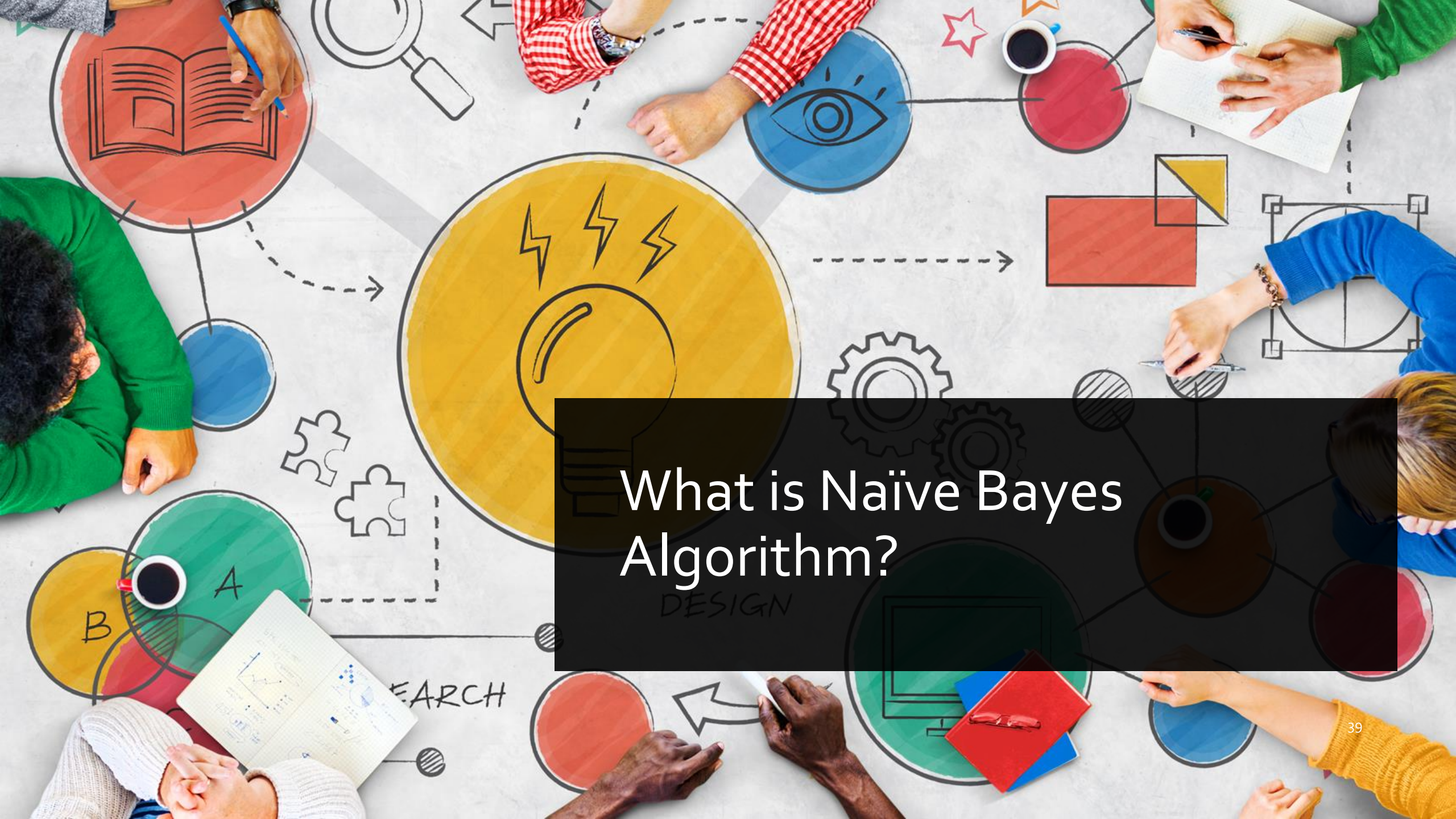
Decision Metrices in KNN

- **Euclidean Distance:** It is the measure of straight-line distance between the two points in Euclidean space. It is calculated as the sum of squares of the absolute difference of cartesian coordinates of two points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- **Cosine Distance:** This distance metric is used mainly to calculate similarity between two vectors. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in the same direction.

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$



What is Naïve Bayes Algorithm?

Naïve Bayes

What is Naïve Bayes Algorithm ?

- Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is a probabilistic classifier, which means it predicts based on the probability of an object.
- Naive Bayes is made up of two words, **“Naive”** meaning the classifier assumes that the occurrence of a certain feature is independent of the occurrence of other features and **“Bayes”** as it is based on Bayes’ Theorem.

Bayes' Theorem:

- Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

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

Where,

- P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
- P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
- P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
- P(B) is Marginal Probability: Probability of Evidence.

Practical Example

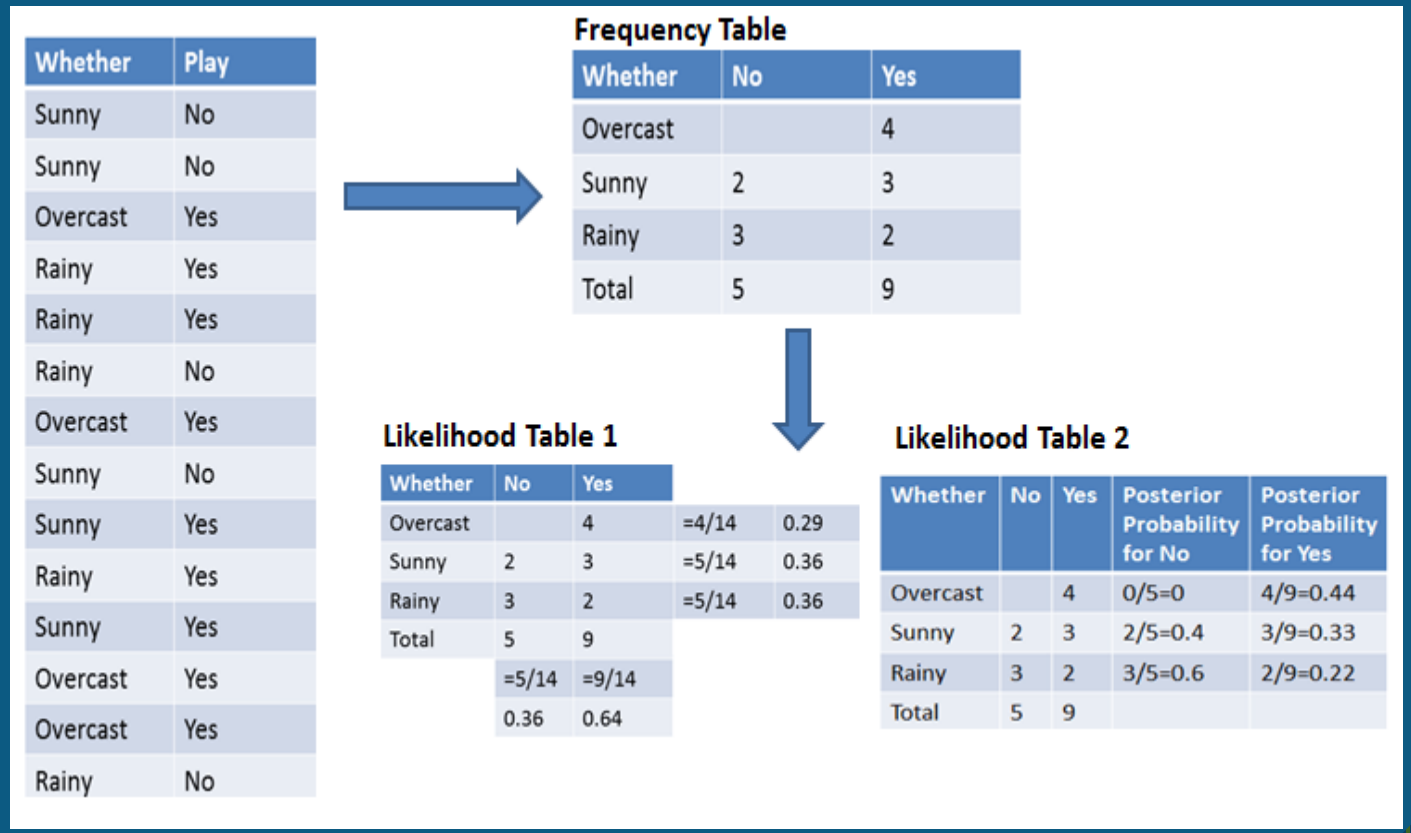
Given an example of weather conditions and playing sports. We need to calculate the probability of playing sports. Now, we need to classify whether players will play or not, based on the weather condition.

Probability of playing:

- P(Yes | Overcast) = P(Overcast | Yes) P(Yes) / P (Overcast)
- P(Overcast) = 4/14 = 0.29
- P(Yes)= 9/14 = 0.64
- P(Overcast |Yes) = 4/9 = 0.44

Then,

- P (Yes | Overcast) = 0.44 * 0.64 / 0.29 = 0.98



Whether	Play
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Likelihood Table 1

Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

Practical Example

Probability of not playing:

- P(No | Overcast) = P(Overcast | No) P(No) / P (Overcast)
- P(Overcast) = 4/14 = 0.29
- P(No)= 5/14 = 0.36
- P(Overcast |No) = 0/9 = 0

Then,

- P (No | Overcast) = 0 * 0.36 / 0.29 = 0

The probability of a 'Yes' class is higher. So you can determine here if the weather is overcast than players will play the sport.

Whether	Play
Sunny	No
Sunny	No
Overcast	Yes
Rainy	Yes
Rainy	Yes
Rainy	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Whether	No	Yes
Overcast		4
Sunny	2	3
Rainy	3	2
Total	5	9

Likelihood Table 1

Whether	No	Yes		
Overcast		4	=4/14	0.29
Sunny	2	3	=5/14	0.36
Rainy	3	2	=5/14	0.36
Total	5	9		
	=5/14	=9/14		
	0.36	0.64		

Likelihood Table 2

Whether	No	Yes	Posterior Probability for No	Posterior Probability for Yes
Overcast		4	0/5=0	4/9=0.44
Sunny	2	3	2/5=0.4	3/9=0.33
Rainy	3	2	3/5=0.6	2/9=0.22
Total	5	9		

Thank You