Deep Learning and Neural Networks

Rishabh Arya

Roll number: 180040080 **Mentor: Mohd Safwan**

May 2019

1. Introduction

Deep learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

Deep learning is an artificial intelligence function that initiates the workings of the human brain in processing data and creating patterns for use in decision making.

It has networks capable of learning unsupervised from data that is unstructured or unlabelled.

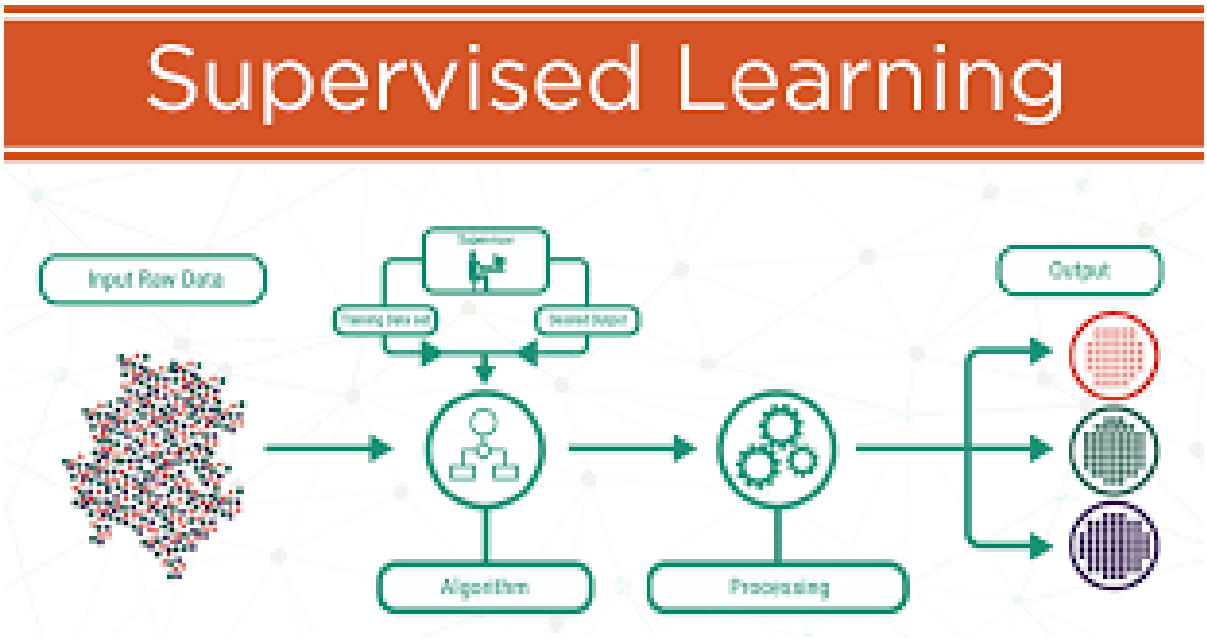
2. Machine learning algorithms:

- Reinforcement learning
- Supervised learning
- Unsupervised learning
- Semi-Supervised learning

Another classification

- Regression
- Classification
- Clusterization

3 Supervised Learning

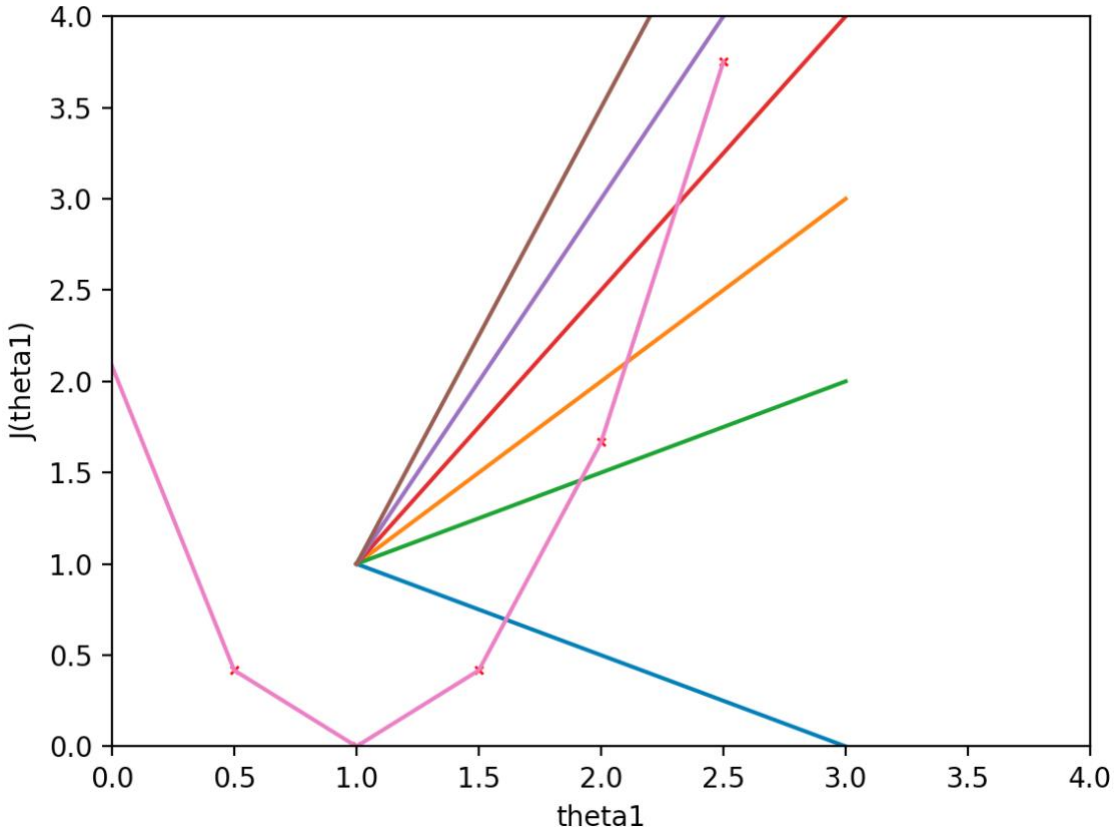


Supervised learning is the Data mining task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the **supervisory signal**).

A **supervised learning algorithm** analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for **unseen instances**. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

4. Cost Function

In ML, cost function are used to estimate how badly models are performing. Put simply, a cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and Y. This is typically expressed as a difference or distance between the predicted value and the actual value.



The objective of a ML model, therefore, is to find parameters, weights or a structure that minuses the cost function

5. Gradient Descent:

It is a first-order iterartive optimization algorithm for finding the minimum of a function.

If instead, one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

Cost Function

$$J\left(\Theta_0, \Theta_1\right) = \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y_i]^2$$

$h_\Theta(x_i)$: Predicted Value; y_i: True Value

Gradient Descent

$$\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J\left(\Theta_0, \Theta_1\right)$$

α: Learning Rate

Now,

$$\begin{aligned}
\frac{\partial}{\partial \Theta} J_\Theta &= \frac{\partial}{\partial \Theta} \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y]^2 \\
&= \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x_i) - y) \frac{\partial}{\partial \Theta_j} (\Theta x_i - y) \\
&= \frac{1}{m} (h_\Theta(x_i) - y) x_i
\end{aligned}$$

Therefore,

$$\Theta_j := \Theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} [(h_\Theta(x_i) - y) x_i]$$

6. BackPropagation:

Backpropagation algorithms are a family of methods used to efficiency train artificial neural artificial neural networks(ANNs) following a gradient-based optimization algorithm that exploits exploits the chain rule. The main feature of backpropagation iterative, recursive and efficient method for calculating the weight updates to improve the network until it is able to perform the task for which it is being trained.[1] It is closely related to the Gauss-Newton algorithm.

Backpropagation requires the derivatives of activation functions to be known at network design time. Automatic differentiation is a technique that can automatically and analytically provide the derivatives to the training algorithm. In the context of learning, backpropagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function; backpropagation computes the gradient(s), whereas (stochastic) gradient descent uses the gradients for training the model (via optimization).

The Equations of Backpropagation

$$\frac{\partial J_{\text{net}}}{\partial \mathbf{W}^{[l]}} = \frac{1}{m}\boldsymbol{\Delta}^{[l]}\left(\boldsymbol{A}^{[l-1]}\right)^{\mathrm{T}} + \frac{\lambda}{m}\mathbf{W}^{[l]}$$

$$\frac{\partial J}{\partial \boldsymbol{b}^{[l]}} = \frac{1}{m}\sum_{i=1}^{m}\Delta_{ji}^{[l]}$$

$$\boldsymbol{\Delta}^{[l]} = \left\{\nabla_{\mathbf{A}^{[l]}}J\right\} \odot g^{[l]'}\left(\mathbf{Z}^{[l]}\right)$$

$$\nabla_{\mathbf{A}^{[l]}}J = \mathbf{W}^{[l+1]^{T}}\boldsymbol{\Delta}^{[l+1]}\ , \quad l \neq L$$

7. Probability Theory

7.1 Conditional Probability

Conditional probability is a measure of the probability of an event occurring given that another event has occurred.

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

Probability of event A occured and event B occured

Probability of event A given B has occured

Probability of event B

7.2 Bayes' Theorem

Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes' theorem, a person's age can be used to more accurately assess the probability that they have cancer than can be done without knowledge of the person's age.

One of the many applications of Bayes' theorem is Bayesian inference, a particular approach to statistical inference, a particular approach to statistical inference. When applied, the probabilities involved in Bayes' theorem may have different probability interpretations.With ths Bayesian probability interpretation the theorem expresses how a degree of belief, expressed as a probability, should rationally change to account for availaibility of related evidence. Bayesian inference is fundamental to Bayesian statistics.

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

$$= \frac{P(B|A)P(A)}{\sum_{i=1}^{n}[P(B|A_i)P(A_i)}$$

8.Unsupervised Learning

Unsupervised learning is the training of machine using information that is nei- ther classified nor labeled and allowing the algorithm to act on that information without guidance.
The task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self.

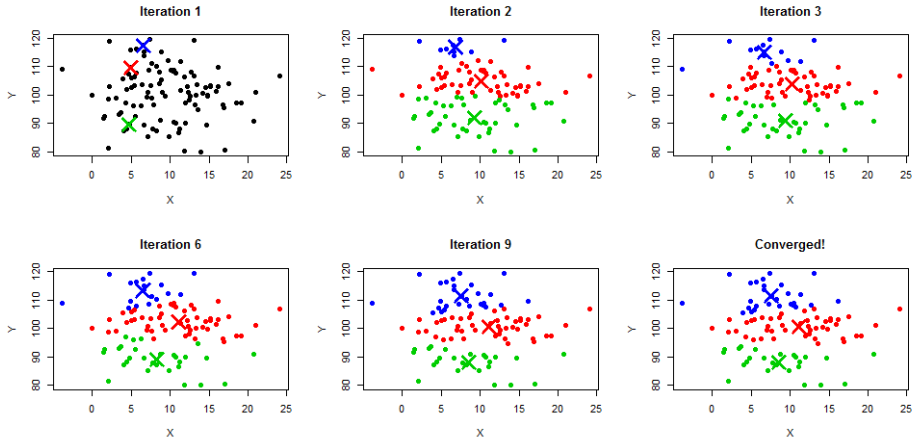
Unsupervised learning classified into two categories of algorithms:

1. Clustering

2. Association

9.1. Clustering

We are given a data set of items, with certain features, and values for these features. The task is to categorize those items into groups(It will help if you think of items as points in an n-dimensional space). To achieve this, we will use the kMeans algorithm



The algorithm works as follows:

1. First we initialize k points, called means, randomly.
2. We categorize each item to its closest mean and we update the means coordinates, which are the averages of the items categorized in that mean so far.
3. We repeat the process for a given number of iterations and at the end, we have our clusters.
4. We repeat the process for a given number of iterations and at the end, we have our clusters.

9.2 Association Rule

Association rule learning is a rule-based machine learning method for discover- ing interesting relations between variables in large databases.

Some basic definitions:

1. Support Count: Frequency of occurrence of a itemset.
2. Frequent Itemset: An itemset whose support is greater than or equal to minsup threshold.
3. Association Rule: An implication expression of the form $X \rightarrow Y$, where X and Y are any 2 itemsets.

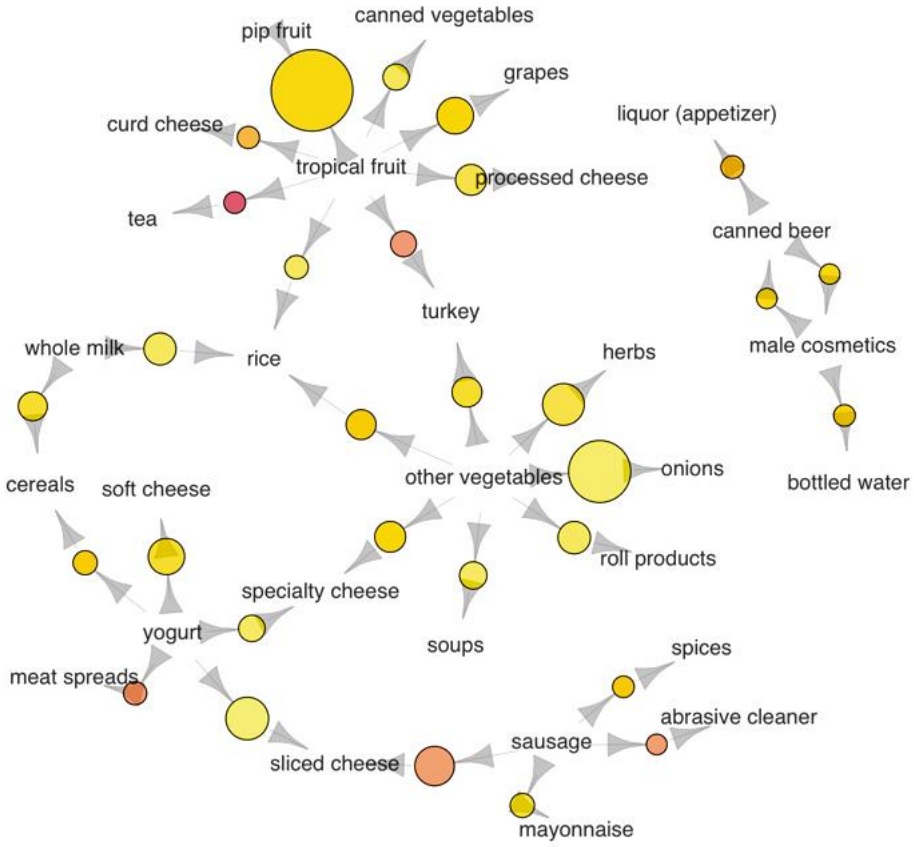
Rules for Evalution Metrics:

1. Support(s) - The number of transactions that include items in the X and Y parts of the rule as a percentage of the total number of transaction.

It is a measure of how frequently the collection of items occur together as a percentage of all transactions.

2. Confidence(c) - It is the ratio of the number of transactions that includes all items in B as well as the no of transactions that includes all items in A to the no of transactions that includes all items in A.
   Conf(X→Y) - Supp(X∪Y) ÷ Supp(X)
3. Lift(l) - The lift of the rule X→Y is the confidence of the rule divided by the expected confidence, assuming that the itemsets X and Y are inde- pendent of each other.The expected confidence is the confidence divided by the frequency of Y.

   Lift(X→Y) = Conf(X→Y) ÷ Supp(Y)
   Lift value near 1 indicates X and Y almost often appear together as ex- pected, greater than 1 means they appear together more than expected and less than I means they appear less than expected. Greater lift values indicate stronger association.



Process:

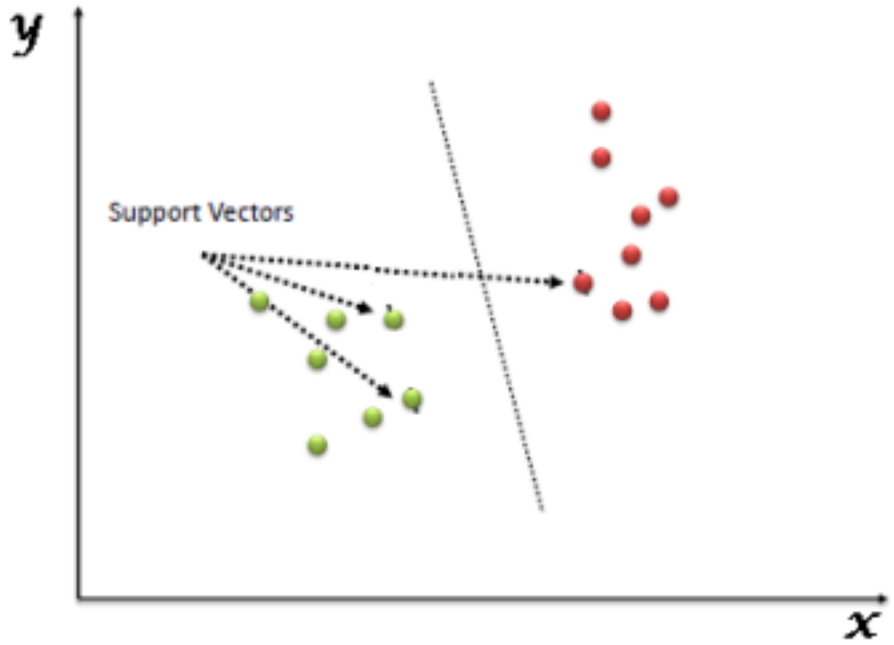
Association rules are usually required to satisfy a user-specified minimum sup- port and a user-specified minimum confidence at the same time.
Association rule generation is usually split up into two separate steps:

1. A minimum support threshold is applied to find all frequent itemsets in a database.

2. A minimum confidence constraint is applied to these frequent itemsets in order to form rules.

10. Support vector Machine

Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).

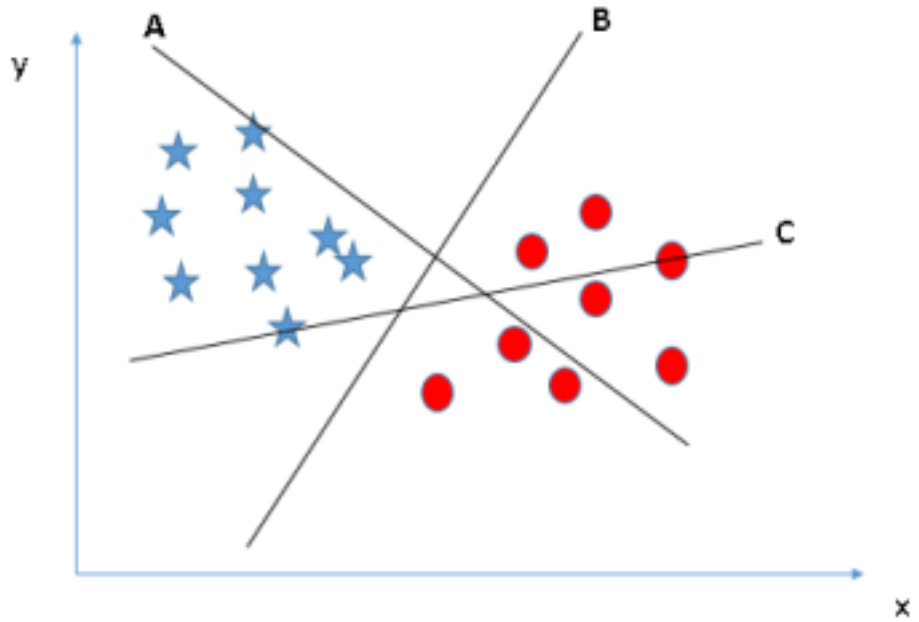


How does it work?

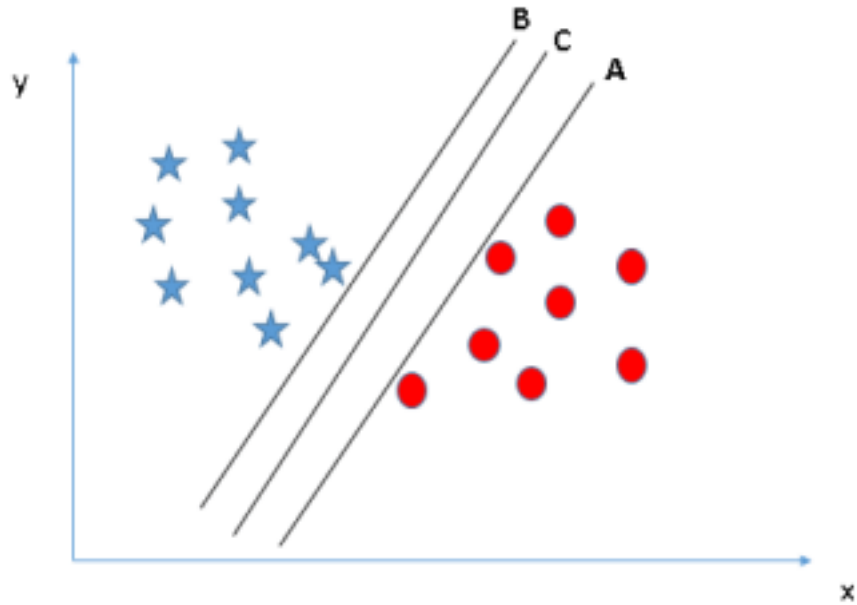
Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is "How can we identify the right hyper-plane?". Don't worry, it's not as hard as you think!

Let's understand:

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

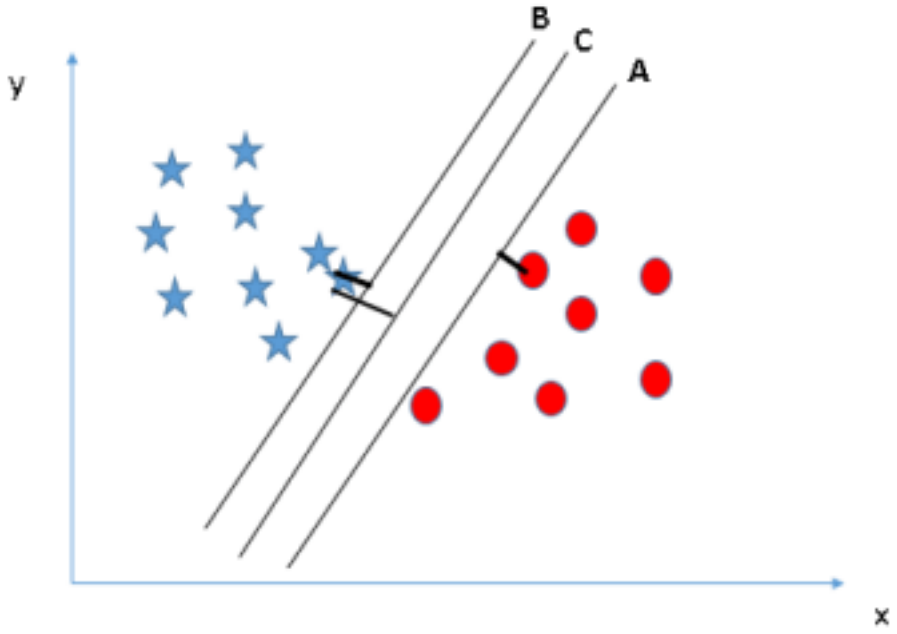


- You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.
- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

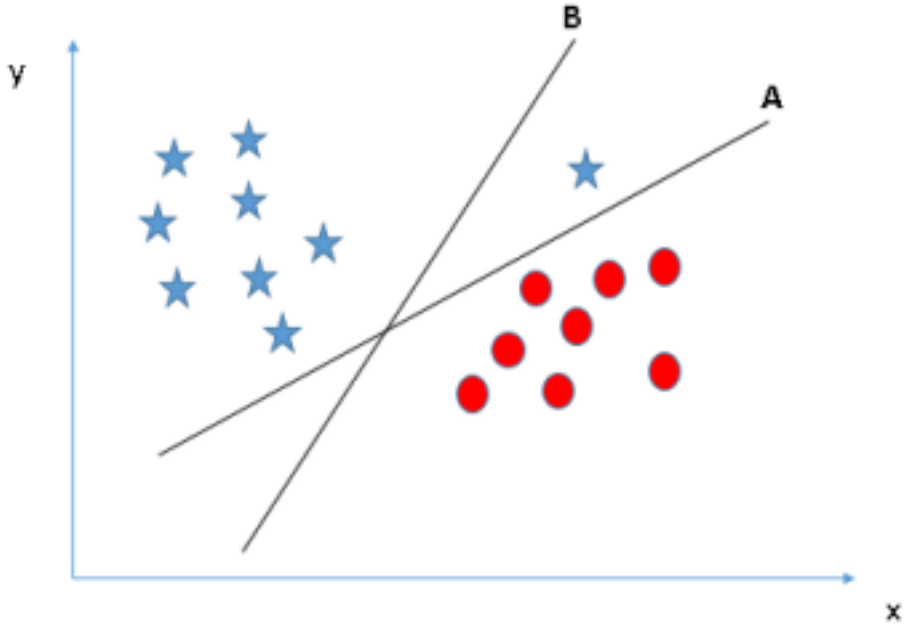


Here, maximizing the distances between nearest data point (either class) and hyper-

plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let's look at the below snapshot:



- **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane



Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.

- As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.
-