Unit-IV

4.1Learning

Learning is one of the fundamental building blocks of artificial intelligence (AI) solutions. From a conceptual standpoint, learning is a process that improves the knowledge of an AI program by making observations about its environment. From a technical/mathematical standpoint, AI learning processes focused on processing a collection of input-output pairs for a specific function and predicts the outputs for new inputs. Most of the artificial intelligence(AI) basic literature identifies two main groups of learning models: supervised and unsupervised. However, that classification is an oversimplification of real world AI learning models and techniques.

To understand the different types of AI learning models, we can use two of the main elements of human learning processes: knowledge and feedback. From the knowledge perspective, learning models can be classified based on the representation of input and output data points. In terms of the feedback, AI learning models can be classified based on the interactions with the outside environment, users and other external factors.

4.2 AI Learning Models: Knowledge-Based Classification

Factoring its representation of knowledge, AI learning models can be classified in two main types: inductive and deductive.

— Inductive Learning: This type of AI learning model is based on inferring a general rule from datasets of input-output pairs.. Algorithms such as knowledge based inductive learning(KBIL) are a great example of this type of AI learning technique. KBIL focused on finding inductive hypotheses on a dataset with the help of background information.

— Deductive Learning: This type of AI learning technique starts with te series of rules nad infers new rules that are more efficient in the context of a specific AI algorithm. Explanation-Based Learning(EBL) and Relevance-0Based Learning(RBL) are examples examples o f deductive techniques. EBL extracts general rules from examples by "generalizing" the explanation. RBL focuses on identifying attributes and deductive generalizations from simple example.

AI Learning Models: Feedback-Based Classification

Based on the feedback characteristics, AI learning models can be classified as supervised, unsupervised, semi-supervised or reinforced.

— Unsupervised Learning: Unsupervised models focus on learning a pattern in the input data without any external feedback. Clustering is a classic example of unsupervised learning models.

— Supervised Learning: Supervised learning models use external feedback to learning functions that map inputs to output observations. In those models the external environment acts as a "teacher" of the AI algorithms.

— Semi-supervised Learning: Semi-Supervised learning uses a set of curated, labeled data and tries to infer new labels/attributes on new data data sets. Semi-Supervised learning models are a solid middle ground between supervised and unsupervised models.

—Reinforcement Learning: Reinforcement learning models use opposite dynamics such as rewards and punishment to "reinforce" different types of knowledge. This type of learning technique is becoming really popular in modern AI solutions.

4.2 What is Supervised Machine Learning?

In Supervised learning, you train the machine using data which is well "**labeled.**" It means some data is already tagged with the correct answer. It can be compared to learning which takes place in the presence of a supervisor or a teacher.

A supervised learning algorithm learns from labeled training data, helps you to predict outcomes for unforeseen data.

Successfully building, scaling, and deploying accurate supervised machine learning models takes time and technical expertise from a team of highly skilled data scientists. Moreover, Data scientist must rebuild models to make sure the insights given remains true until its data changes.

In this tutorial, you will learn:

- What is Supervised Machine Learning?
- How Supervised Learning Works
- Types of Supervised Machine Learning Algorithms
- Supervised vs. Unsupervised Machine learning techniques
- Challenges in Supervised machine learning
- Advantages of Supervised Learning:
- Disadvantages of Supervised Learning
- Best practices for Supervised Learning

4.2.1 How Supervised Learning Works

For example, you want to train a machine to help you predict how long it will take you to drive home from your workplace. Here, you start by creating a set of labeled data. This data includes

- Weather conditions
- Time of the day
- Holidays

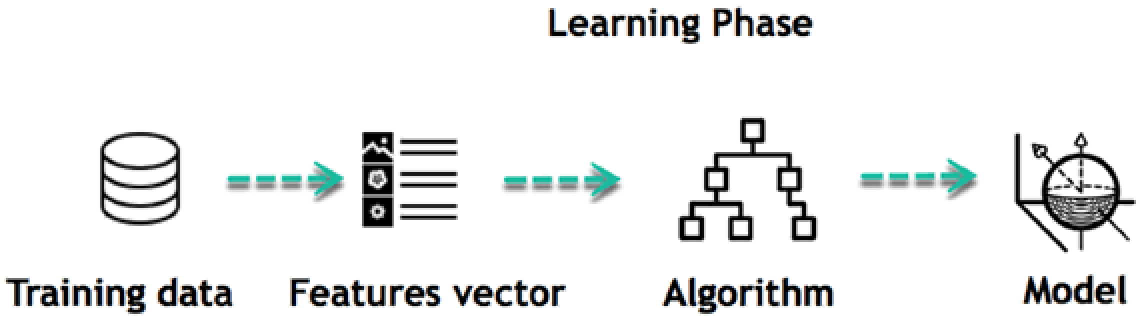
All these details are your inputs. The output is the amount of time it took to drive back home on that specific day.

You instinctively know that if it's raining outside, then it will take you longer to drive home. But the machine needs data and statistics.

Let's see now how you can develop a supervised learning model of this example which help the user to determine the commute time. The first thing you requires to create is a training set. This training set will contain the total commute time and corresponding factors like weather, time, etc. Based on this training set, your machine might see there's a direct relationship between the amount of rain and time you will take to get home.

So, it ascertains that the more it rains, the longer you will be driving to get back to your home. It might also see the connection between the time you leave work and the time you'll be on the road.

The closer you're to 6 p.m. the longer it takes for you to get home. Your machine may find some of the relationships with your labeled data.



This is the start of your Data Model. It begins to impact how rain impacts the way people drive. It also starts to see that more people travel during a particular time of day.

4.2.2 Types of Supervised Machine Learning Algorithms

4.2.2.1Regression:

Regression technique predicts a single output value using training data.

Example: You can use regression to predict the house price from training data. The input variables will be locality, size of a house, etc.

Strengths: Outputs always have a probabilistic interpretation, and the algorithm can be regularized to avoid overfitting.

Weaknesses: Logistic regression may underperform when there are multiple or non-linear decision boundaries. This method is not flexible, so it does not capture more complex relationships.

4.2.2.2 Logistic Regression:

Logistic regression method used to estimate discrete values based on given a set of independent variables. It helps you to predicts the probability of occurrence of an event by fitting data to a logit function. Therefore, it is also known as logistic regression. As it predicts the probability, its output value lies between 0 and 1.

Here are a few types of Regression Algorithms

4.2.2.3 Classification:

Classification means to group the output inside a class. If the algorithm tries to label input into two distinct classes, it is called binary classification. Selecting between more than two classes is referred to as multiclass classification.

Example: Determining whether or not someone will be a defaulter of the loan.

Strengths: Classification tree perform very well in practice

Weaknesses: Unconstrained, individual trees are prone to overfitting.

Here are a few types of Classification Algorithms

Naïve Bayes Classifiers

Naïve Bayesian model (NBN) is easy to build and very useful for large datasets. This method is composed of direct acyclic graphs with one parent and several children. It assumes independence among child nodes separated from their parent.

Decision Trees

Decisions trees classify instance by sorting them based on the feature value. In this method, each mode is the feature of an instance. It should be classified, and every branch represents a value which the node can assume. It is a widely used technique for classification. In this method, classification is a tree which is known as a decision tree.

It helps you to estimate real values (cost of purchasing a car, number of calls, total monthly sales, etc.).

Support Vector Machine

Support vector machine (SVM) is a type of learning algorithm developed in 1990. This method is based on results from statistical learning theory introduced by Vap Nik.

SVM machines are also closely connected to kernel functions which is a central concept for most of the learning tasks. The kernel framework and SVM are used in a variety of fields. It includes multimedia information retrieval, bioinformatics, and pattern recognition.

4.2.3 Supervised vs. Unsupervised Machine learning techniques

Based On	**Supervised machine learning technique**	**Unsupervised machine learning technique**
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labelled
Computational Complexity	Supervised learning is a simpler method.	Unsupervised learning is computationally complex
Accuracy	Highly accurate and trustworthy method.	Less accurate and trustworthy method.

4.2.4Challenges in Supervised machine learning

Here, are challenges faced in supervised machine learning:

- Irrelevant input feature present training data could give inaccurate results
- Data preparation and pre-processing is always a challenge.
- Accuracy suffers when impossible, unlikely, and incomplete values have been inputted as training data
- If the concerned expert is not available, then the other approach is "brute-force." It means you need to think that the right features (input variables) to train the machine on. It could be inaccurate.

4.2.5 Advantages of Supervised Learning:

- Supervised learning allows you to collect data or produce a data output from the previous experience
- Helps you to optimize performance criteria using experience
- Supervised machine learning helps you to solve various types of real-world computation problems.

4.2.6 Disadvantages of Supervised Learning

- Decision boundary might be overtrained if your training set which doesn't have examples that you want to have in a class
- You need to select lots of good examples from each class while you are training the classifier.
- Classifying big data can be a real challenge.
- Training for supervised learning needs a lot of computation time.

4.2.7 Best practices for Supervised Learning

- Before doing anything else, you need to decide what kind of data is to be used as a training set
- You need to decide the structure of the learned function and learning algorithm.
- Gather corresponding outputs either from human experts or from measurements

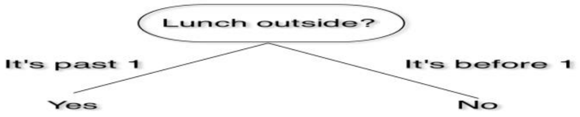
Summary

- In Supervised learning, you train the machine using data which is well "labelled."
- You want to train a machine which helps you predict how long it will take you to drive home from your workplace is an example of supervised learning
- Regression and Classification are two types of supervised machine learning techniques.
- Supervised learning is a simpler method while Unsupervised learning is a complex method.
- The biggest challenge in supervised learning is that Irrelevant input feature present training data could give inaccurate results.
- The main advantage of supervised learning is that it allows you to collect data or produce a data output from the previous experience.
- The drawback of this model is that decision boundary might be overstrained if your training set doesn't have examples that you want to have in a class.
- As a best practice of supervise learning, you first need to decide what kind of data should be used as a training set.

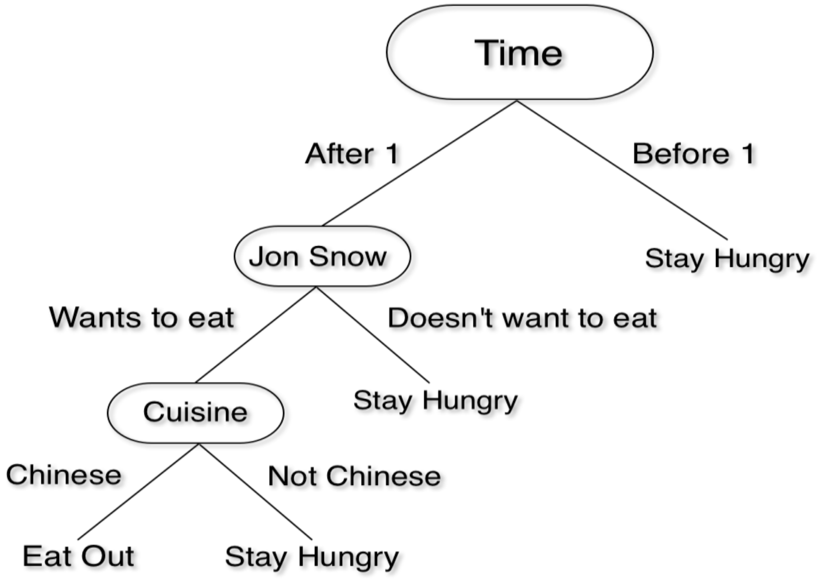
4.3 Learning Decision Tree

4.3.1 What is a decision tree?

Let's skip the formal definition and think conceptually about decision trees. Imagine you're sitting in your office and feeling hungry. You want to go out and eat, but lunch starts at 1 PM. What do you do? Of course, you look at the time and then decide if you can go out. You can think of your logic like this:



We just made a decision tree! This is a simple one, but we can build a complicated one by including more factors like weather, cost, etc. If you want to go to lunch with your friend, Jon Snow, to a place that serves Chinese food, the logic can be summarized in this tree:



This is also a decision tree. You start at the top, follow the paths that describe the current condition, and keep doing that until you reach a decision.

Some notation

Let’s shift to the world of computers. Each box we just drew is called a node. The topmost node is called the root and all the nodes at the bottom layer are leaf nodes. Think of it as a real-world tree, but inverted.

Each node tests some property (attribute) of our world (dataset) and each branch going out from the node corresponds to a value of that attribute. Given a tree, the process of deciding will be:

1. Start at the root
2. Observe value of the attribute at the root
3. Follow the path that corresponds to the observed value
4. Repeat until we reach a leaf node, which will give us our decision

4.3.2 How to construct a decision tree?

You won’t ever need to construct a decision tree from scratch (unless you’re a student like me). Nonetheless, it’s a good learning experience and you’ll learn some interesting concepts along the way.

The most popular algorithm for constructing decision trees is ID3 and it’s quite simple. Here’s the algorithm pseudocode:

```
ID3 (Examples, Target_Attribute, Attributes)
  Create a root node for the tree
  If all examples are positive, Return the single-node tree Root, with label = +.
  If all examples are negative, Return the single-node tree Root, with label = -.
  If Atributes list is empty, then Return the single node tree Root,
  with label = most common value of the target attribute in the examples.
  Otherwise Begin
    A ← The Attribute that best classifies examples.
    Decision Tree attribute for Root = A.
    For each possible value, vi, of A,
      Add a new tree branch below Root, corresponding to the test A = vi.
      Let Examples(vi) be the subset of examples that have the value vi for A
      If Examples(vi) is empty
```

```
            Then below this new branch add a leaf node with label = most common target value in the examples
          Else below this new branch add the subtree ID3 (Examples(vi), Target_Attribute, Attributes – {A})
      End
      Return Root
```

One detail you'll notice is that just after the beginning of the loop, the algorithm has to pick the attribute that best classifies the examples. How will it do that? To understand that, we'll have to dive into a little bit of math. Don't worry, it's not too hard, and if you get stuck, I can answer any questions in the comments.

4.3.3 Information Gain and Entropy

One of the commonly used and beginner friendly ways to figure out the best attribute is information gain. It's calculated using another property called entropy.

Entropy is a concept used in physics and mathematics that refers to the randomness or the impurity of a system. In information theory, it refers to the impurity of a group of examples.

Let's see an example to make it clear: You have 2 bags of full of chocolates. The chocolates can be either red or blue. You decide to measure the entropy of bags by counting the number of chocolates. So you sit down and start counting. After 2 minutes, you discover the first bag has 50 chocolates. 25 of them are red and 25 are blue. Second bag also has 50 chocolates, all of them blue.

In this case, the first bag has entropy 1 as the chocolates are equally distributed. The second bag has entropy zero because there is no randomness.

If you want to calculate the entropy of a system, we use this formula:

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

Here, c is the total number of classes or attributes and pi is number of examples belonging to the ith class. Confused? Let's try an example to clarify.

We will go back to our chocolate boxes. We have two classes, red(R) and blue(B). For the first box, we have 25 red chocolates. The total number of chocolates is 50. So pi becomes 25 divided by 50. Same goes for blue class. Plug those values into entropy equation and we get this:

$$Entropy(C) = -\frac{25}{50} log_2 \frac{25}{50} - \frac{25}{50} log_2 \frac{25}{50}$$

Solve the equation and here are the results:

$$
\begin{aligned}
\frac{25}{50} &= 0.5 \\
log_2\frac{25}{50} &= log_2\frac{1}{2} \\
&= log_2 1 - log_2 2 \\
&= 0 - 1 \\
&= -1 \\
Entropy(C) &= -(0.5 \times -1) - (0.5 \times -1) \\
&= 0.5 + 0.5 \\
&= 1
\end{aligned}
$$

If you'd like to verify the result or play with more examples, check Wolfram Alpha.

Go ahead and calculate entropy for the second box, which has 50 red chocolates and 0 blue ones. You will get 0 entropy.

If you understand the concept, excellent! We'll move to information gain now. If you have any doubts, just leave a comment, and I'll be happy to answer any questions.

Information Gain

Information gain is simply the expected reduction in entropy caused by partitioning all our examples according to a given attribute. Mathematically, it's defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

This may seem like a lot, so let's break it down. S refers to the entire set of examples that we have. A is the attribute we want to partition or split. |S| is the number of examples and |Sv| is the number of examples for the current value of attribute A.

Still very complicated, right? Let's try the measure on an example and see how it works.

Building the Decision Tree

First, let's take our chocolate example and add a few extra details. We already know that the box 1 has 25 red chocolates and 25 blue ones. Now, we will also consider the brand of chocolates. Among red ones, 15 are Snickers and 10 are Kit Kats. In blue ones, 20 are Kit Kats and 5 are Snickers. Let's assume we only want to eat red Snickers. Here, red Snickers (15) become positive examples and everything else like blue Snickers and red Kit Kats are negative examples.

Now, the entropy of the dataset with respect to our classes (eat/not eat) is:

$$\begin{aligned} Entropy &= -\frac{15}{50}log_2\frac{15}{50} - \frac{35}{50}log_2\frac{35}{50} \\ &= 0.5210 + 0.3602 \\ &= 0.8812 \end{aligned}$$

Let's take a look back now — we have 50 chocolates. If we look at the attribute color, we have 25 red and 25 blue ones. If we look at the attribute brand, we have 20 Snickers and 30 Kit Kats.

To build the tree, we need to pick one of these attributes for the root node. And we want to pick the one with the highest information gain. Let's calculate information gain for attributes to see the algorithm in action.

Information gain with respect to color would be:

$$\begin{aligned} Information\ Gain(Chocolates, Colors) = Entropy(Chocolates) \\ -(\frac{|red\ chocolates|}{|total\ chocolates|} \times Entropy(red\ chocolates)) \\ -(\frac{|blue\ chocolates|}{|total\ chocolates|} \times Entropy(blue\ chocolates)) \end{aligned}$$

We just calculated the entropy of chocolates with respect to class, which is 0.8812. For entropy of red chocolates, we want to eat 15 Snickers but not 10 Kit Kats. The entropy for red chocolates is:

$$\begin{aligned} Entropy(red\ chocolates) &= -\frac{15}{25}log_2\frac{15}{25} - \frac{10}{25}log_2\frac{10}{25} \\ &= 0.9709 \end{aligned}$$

For blue chocolates, we don't want to eat them at all. So entropy is 0.

Our information gain calculation now becomes:

$$\begin{aligned} Information\ Gain(Chocolates, Colors) &= 0.8812 - (\frac{25}{50} \times 0.9709) - (\frac{25}{50} \times 0) \\ &= 0.3958 \end{aligned}$$

If we split on color, information gain is 0.3958.

Let's look at the brand now. We want to eat 15 out of 20 Snickers. We don't want to eat any Kit Kats. The entropy for Snickers is:

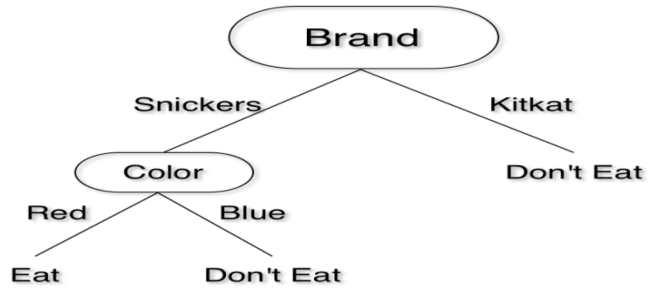
$$\begin{aligned} Entropy(Snickers) &= -\frac{15}{20}log_2\frac{15}{20} - \frac{5}{20}log_2\frac{5}{20} \\ &= 0.8112 \end{aligned}$$

We don't want to eat Kit Kats at all, so Entropy is 0. Information gain:

$$Information\ Gain(Chocolates, Brand) = 0.8812 - (\frac{20}{50} \times 0.8112) - (\frac{30}{50} \times 0)$$
$$= 0.5567$$

Information gain for the split on brand is 0.5567.

Since information gain for brand is larger, we will split based on brand. For the next level, we only have color left. We can easily split based on color without having to do any calculations. Our decision tree will look like this:



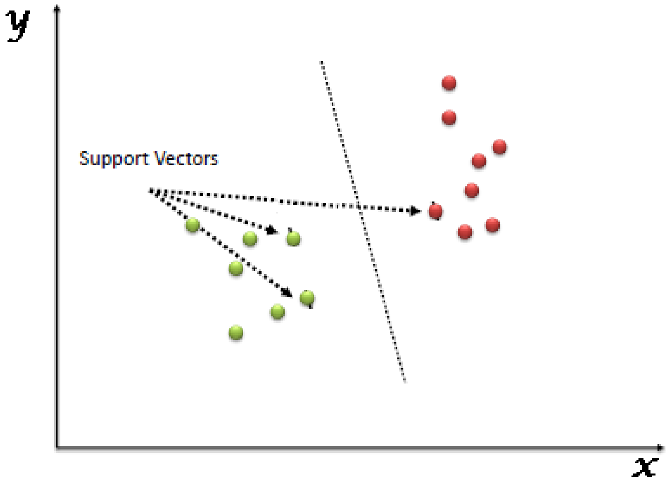
Who thought eating chocolates would be this hard?

You should have a solid intuition about how decision trees work now.

4.4 SVM(Support Vector Machine

4.4.1 What is 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).



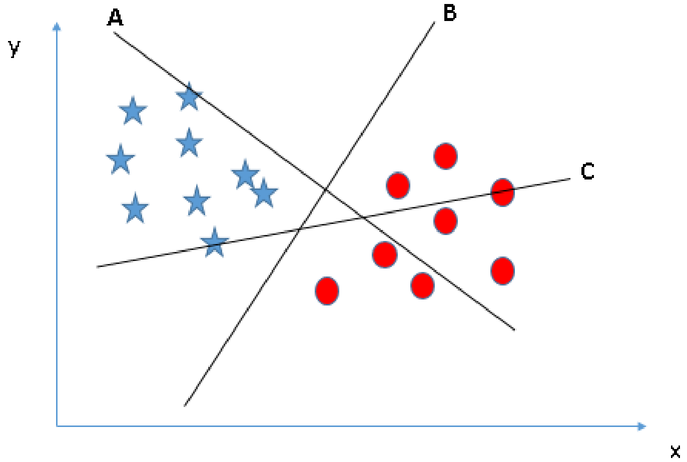
Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

4.4.2 How does it work?

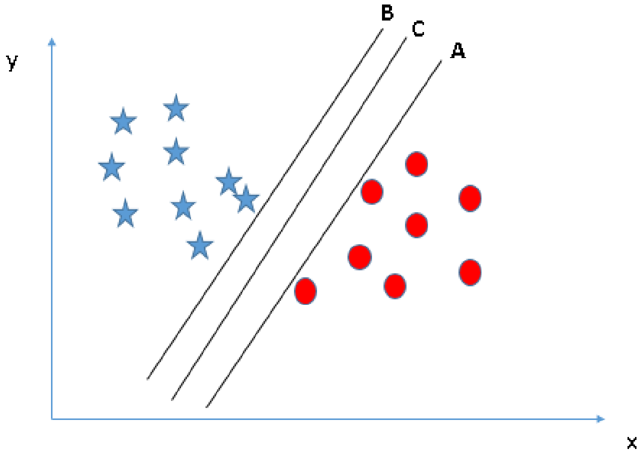
Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is “How can we identify the right hyper-plane?”. Don’t worry, it’s not as hard as you think!

Let’s understand:

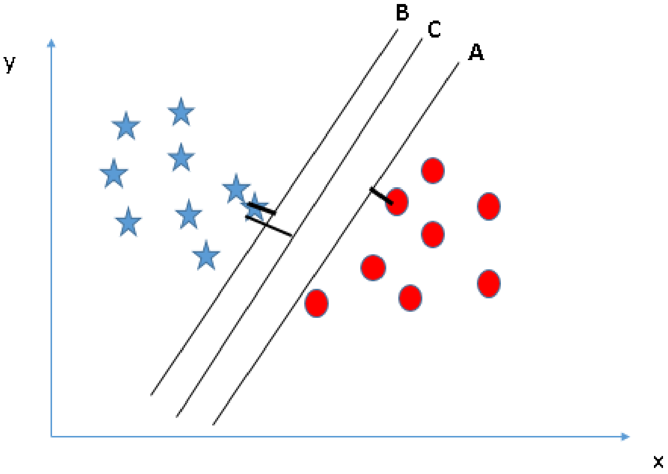
- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.



  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?

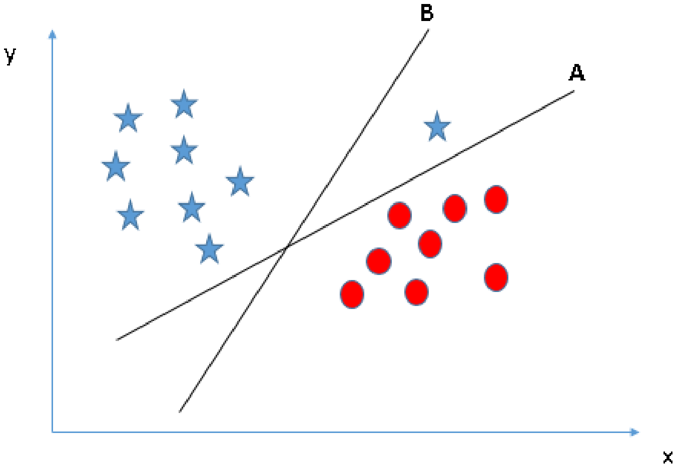


  Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let’s look at the below snapshot:



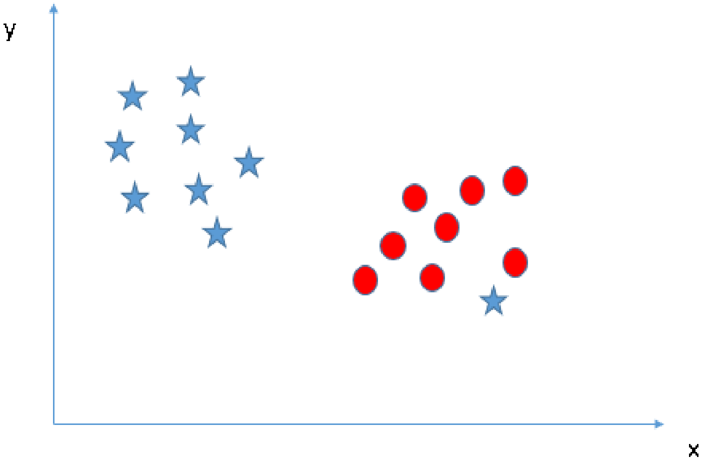
Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

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

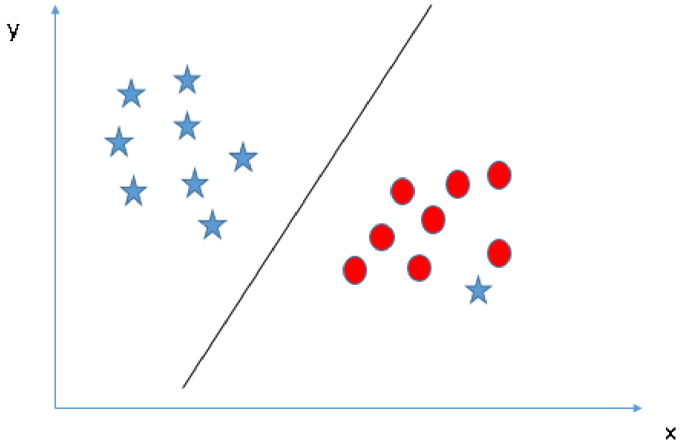


Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore,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

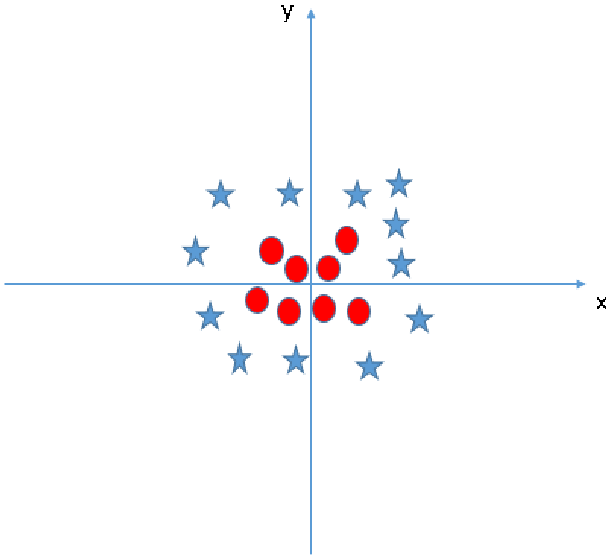
- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier.



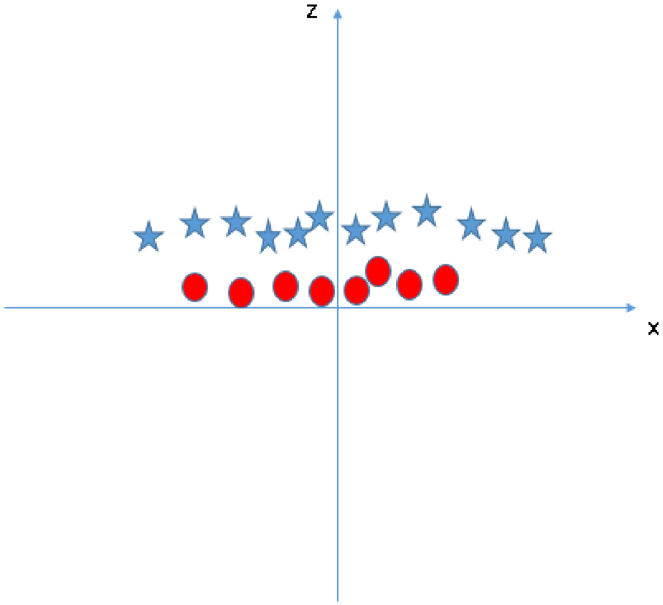
As I have already mentioned, one star at other end is like an outlier for star class. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to outliers.



- **Find the hyper-plane to segregate to classes (Scenario-5):** In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.



SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:

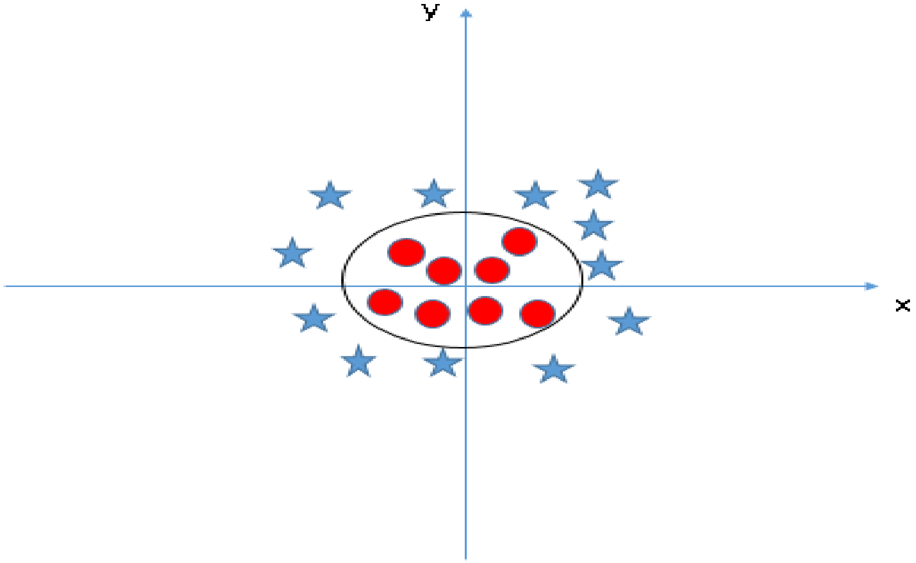


In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

In SVM, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, SVM has a technique called the **kernel trick**. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined.

When we look at the hyper-plane in original input space it looks like a circle:



Now, let's look at the methods to apply SVM algorithm in a data science challenge.

4.4.3 Pros and Cons associated with SVM

- **Pros:**
  - It works really well with clear margin of separation
  - It is effective in high dimensional spaces.
  - It is effective in cases where number of dimensions is greater than the number of samples.
  - It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- **Cons:**
  - It doesn't perform well, when we have large data set because the required training time is higher
  - It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
  - SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

4.5 Unsupervised Learning

4.5.1 What is Unsupervised Learning?

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabelled data.

Unsupervised learning algorithms allows you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods.

In this topic, you will learn:

- What is Unsupervised Learning?
- Example of Unsupervised Machine Learning
- Why Unsupervised Learning?
- Types of Unsupervised Learning
- Clustering

- Clustering Types
- Association
- Supervised vs. Unsupervised Machine Learning
- Applications of unsupervised machine learning
- Disadvantages of Unsupervised Learning

4.5.2 Example of Unsupervised Machine Learning

Let's, take the case of a baby and her family dog.

She knows and identifies this dog. Few weeks later a family friend brings along a dog and tries to play with the baby.

Baby has not seen this dog earlier. But it recognizes many features (2 ears, eyes, walking on 4 legs) are like her pet dog. She identifies the new animal as a dog. This is unsupervised learning, where you are not taught but you learn from the data (in this case data about a dog.) Had this been supervised learning, the family friend would have told the baby that it's a dog.

4.5.3 Why Unsupervised Learning?

Here, are prime reasons for using Unsupervised Learning:

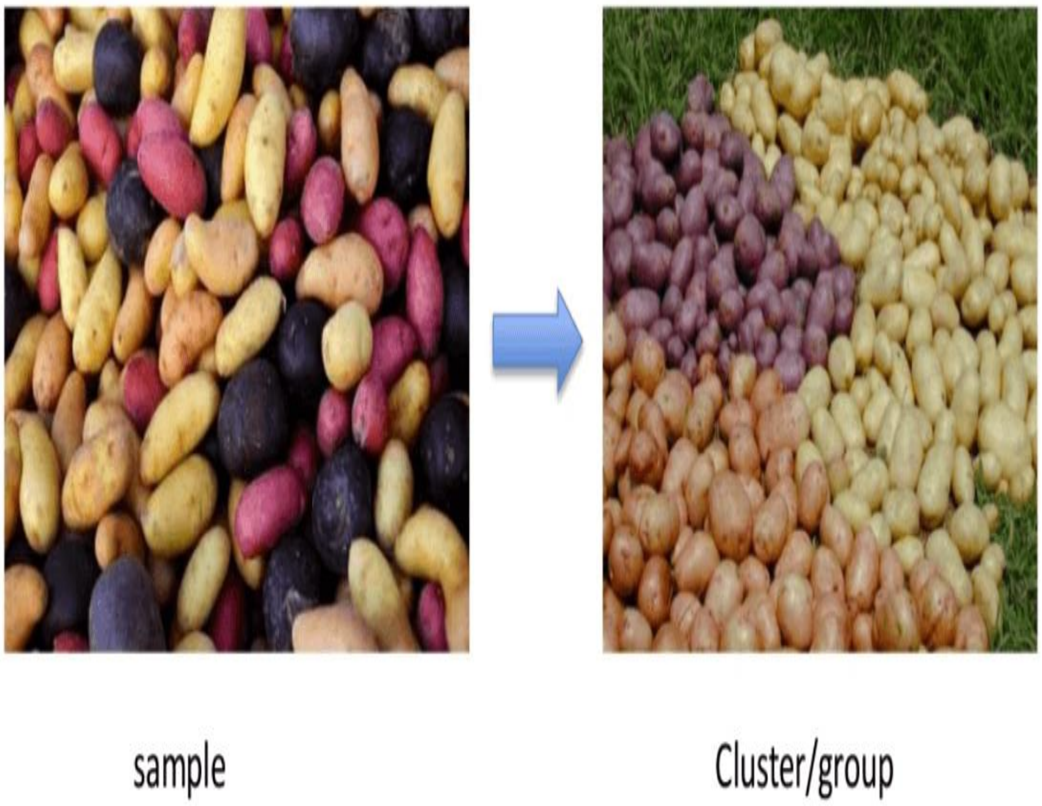
- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.

- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

4.5.4 Types of Unsupervised Learning

Unsupervised learning problems further grouped into clustering and association problems.

4.5.5 Clustering



Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data. You can also modify how many clusters your algorithms should identify. It allows you to adjust the granularity of these groups.

There are different types of clustering you can utilize:

Exclusive (partitioning)

In this clustering method, Data are grouped in such a way that one data can belong to one cluster only.

Example: K-means

Agglomerative

In this clustering technique, every data is a cluster. The iterative unions between the two nearest clusters reduce the number of clusters.

Example: Hierarchical clustering

Overlapping

In this technique, fuzzy sets is used to cluster data. Each point may belong to two or more clusters with separate degrees of membership.

Here, data will be associated with an appropriate membership value. Example: Fuzzy C-Means

Probabilistic

This technique uses probability distribution to create the clusters

Example: Following keywords

- "man's shoe."
- "women's shoe."
- "women's glove."
- "man's glove."

can be clustered into two categories "shoe" and "glove" or "man" and "women."

4.5.6 Clustering Types

- Hierarchical clustering
- K-means clustering
- K-NN (k nearest neighbors)
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

Hierarchical Clustering:

Hierarchical clustering is an algorithm which builds a hierarchy of clusters. It begins with all the data which is assigned to a cluster of their own. Here, two close cluster are going to be in the same cluster. This algorithm ends when there is only one cluster left.

K-means Clustering

K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration. Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.

The output of the algorithm is a group of "labels." It assigns data point to one of the k groups. In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

K-mean clustering further defines two subgroups:

- Agglomerative clustering
- Dendrogram

Agglomerative clustering:

This type of K-means clustering starts with a fixed number of clusters. It allocates all data into the exact number of clusters. This clustering method does not require the number of clusters K as an input. Agglomeration process starts by forming each data as a single cluster.

This method uses some distance measure, reduces the number of clusters (one in each iteration) by merging process. Lastly, we have one big cluster that contains all the objects.

Dendrogram:

In the Dendrogram clustering method, each level will represent a possible cluster. The height of dendrogram shows the level of similarity between two join clusters. The closer to the bottom of the process they are more similar cluster which is finding of the group from dendrogram which is not natural and mostly subjective.

K- Nearest neighbors

K- nearest neighbour is the simplest of all machine learning classifiers. It differs from other machine learning techniques, in that it doesn't produce a model. It is a simple algorithm which stores all available cases and classifies new instances based on a similarity measure.

It works very well when there is a distance between examples. The learning speed is slow when the training set is large, and the distance calculation is nontrivial.

Principal Components Analysis:

In case you want a higher-dimensional space. You need to select a basis for that space and only the 200 most important scores of that basis. This base is known as a principal component. The subset you select constitute is a new space which is small in size compared to original space. It maintains as much of the complexity of data as possible.

4.5.7Association

Association rules allow you to establish associations amongst data objects inside large databases. This unsupervised technique is about discovering interesting relationships between variables in large databases. For example, people that buy a new home most likely to buy new furniture.

Other Examples:

- A subgroup of cancer patients grouped by their gene expression measurements
- Groups of shopper based on their browsing and purchasing histories
- Movie group by the rating given by movies viewers

4.5.8 Supervised vs. Unsupervised Machine Learning

Parameters	**Supervised machine learning technique**	**Unsupervised machine learning technique**
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labelled
Computational Complexity	Supervised learning is a simpler method.	Unsupervised learning is computationally complex
Accuracy	Highly accurate and trustworthy method.	Less accurate and trustworthy method.

4.5.9 Applications of unsupervised machine learning

Some applications of unsupervised machine learning techniques are:

- Clustering automatically split the dataset into groups base on their similarities
- Anomaly detection can discover unusual data points in your dataset. It is useful for finding fraudulent transactions
- Association mining identifies sets of items which often occur together in your dataset
- Latent variable models are widely used for data preprocessing. Like reducing the number of features in a dataset or decomposing the dataset into multiple components

4.5.10Disadvantages of Unsupervised Learning

- You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known
- Less accuracy of the results is because the input data is not known and not labeled by people in advance. This means that the machine requires to do this itself.
- The spectral classes do not always correspond to informational classes.
- The user needs to spend time interpreting and label the classes which follow that classification.
- Spectral properties of classes can also change over time so you can't have the same class information while moving from one image to another.

Summary

- Unsupervised learning is a machine learning technique, where you do not need to supervise the model.
- Unsupervised machine learning helps you to finds all kind of unknown patterns in data.
- Clustering and Association are two types of Unsupervised learning.
- Four types of clustering methods are 1) Exclusive 2) Agglomerative 3) Overlapping 4) Probabilistic.
- Important clustering types are: 1)Hierarchical clustering 2) K-means clustering 3) K-NN 4) Principal Component Analysis 5) Singular Value Decomposition 6) Independent Component Analysis.
- Association rules allow you to establish associations amongst data objects inside large databases.
- In Supervised learning, Algorithms are trained using labelled data while in Unsupervised learning Algorithms are used against data which is not labelled.
- Anomaly detection can discover important data points in your dataset which is useful for finding fraudulent transactions.
- The biggest drawback of Unsupervised learning is that you cannot get precise information regarding data sorting.

4.6 Market basket analysis

Market basket analysis is a data mining technique, generally used in the retail industry in an effort to understand purchasing behaviour. It looks for combinations of items that frequently occur in the same transaction. In other words, it gives insights into items that may have some association or affinity.

For example, customers purchasing flour and sugar are also likely to buy eggs. The outcome of the analysis is to derive a set of rules that can be understood as "if this, then that". Retailers can use these insights to do product placements or offer discounts.

In fact, market basket analysis is being applied outside retail. Therefore, it's more generally called **Affinity Analysis**.

Discussion

- Could you give examples of market basket analysis?

  Market basket analysis uncovers associations between products by looking for combinations of products that frequently co-occur in transactions. Thus, supermarkets can identify relationships between products that people buy.

  For example, customers who buy a pencil and paper are likely to buy an eraser or ruler. A customer in an English pub buying a pint of beer without a bar meal is more likely to buy crisps/chips than somebody who didn't buy beer. Someone who buys shampoo is likely to buy conditioner. Retailers can use this information to modify the store layout or offer discount on shampoo but not on conditioner.

  Online retails such as Amazon make product purchase recommendations. If you add any item to your cart, Amazon will recommended other items that other customers often bought together with your selected item.

- What are the applications of market basket analysis?

  Within retail, market basket analysis helps determine purchasing behaviour, build recommendation engines, customize loyalty programs, cross-sell, up-sell, place products in stores in the right places, offer right combinations of discounts, and so on.

Beyond retail, affinity analysis has been used on medical data. Does high BMI and smoking lead to greater chance of high blood pressure? Affinity analysis answers such questions.

In web browsing, it enables click stream analysis. For example, given the last two clicks, how likely is the user to click a specific link? It can be used to detect intrusions.

In banking, credit card purchases are analysed to detect frauds and cross-sell. In insurance, user profiles formed via market basket analysis can be used to flag fraudulent claims. In telecom, market basket analysis can suggest the right bundle of services to retain customers or analyse calling patterns.

- What's the typical data pipeline for market basket analysis?

  Data for market basket analysis typically comes from point-of-sale (POS) transaction data or invoices. These usually include list of products purchased, unit price and quantity of each item. To be statistically significant, the dataset must be large. One analyst reported a dataset of 32 million records of 50K unique items.

  The next step is to look for combinations of items that occur most often together within a transaction. The selection of the right algorithm is important here. Otherwise, we will end up with too many combinations of items (perhaps in millions) that may be computationally difficult to analyse.

  Once frequently occurring item combinations are identified, we next look for associations. This step is often called **Association Rule Mining**.

  The last step is to pick out strong association rules, seek explanations as to why such associations exists and drive business decisions. The usefulness or "interestingness" of a rule is application dependent.

- Could you give more details on Association Rule Mining?

  Graphical view of rules: coffee and toast show a high lift. Source: García 2018.

  Association Rule Mining counts the frequency of items that occur together across a large collection of items or actions. The goal is to find associations that take place together far more often than you would find in a random sampling of possibilities.

  Output of above will be a set of association rules in the following form: IF {item 1, item 2} THEN {item 3}. This states that when items 1 and 2 are purchased, then item 3 is likely to be purchased with a certain probability. The first part of the rule is called **antecedent**. The second part is called **consequent**.

  For example, a customer who buys pencil and paper (antecedent) is likely to buy an eraser (consequent). But how likely is such a customer to buy an eraser? To quantify this, we have a few measures: support, confidence, and lift. These tell us how important or reliable is an association rule.

- Could you explain the terms Support, Confidence and Lift?

  Measures to evaluate association rules. Source: Li 2017.

  These are common quantitative measures to identify most important association rules:

  - **Support**: Given all transactions, support is the percentage of transactions that contain a given item combination. Often combination that fall below a support threshold are ignored in further analysis. When dataset has thousands of items and millions of transactions, a threshold of 0.01% is reasonable.
  - **Confidence**: Given item A is purchased, what's the chance that customer will buy item C? This question is answered by the confidence measure. Thus, rather than looking at just probability of purchasing item C (which support does), confidence looks at conditional probability.
  - **Lift**: Suppose data shows that items A and C are occurring together in many transactions. Do A and C have an association or are they occurring together purely by chance? This question is answered by the lift measure.

Association rules must satisfy both minimum support and minimum confidence values. To filter the results further to a smaller list, lift is a popular measure.

- Could you give example calculations of Support, Confidence and Lift?

  For example, given a million transactions, 24K transactions contain {flour,sugar}; 30K transactions contain {eggs}; 20K transactions contain both {flour,sugar} and {eggs}. Thus,

  - Support({flour,sugar}) = 24K / 1M = 0.024
  - Support(eggs) = 30K / 1M = 0.03
  - Support({flour,sugar}, eggs) = 20K / 1M = 0.02
  - Confidence({flour,sugar}->eggs) = Support({flour,sugar}, eggs) / Support({flour,sugar}) = 0.02 / 0.024 = 0.83
  - Confidence(eggs->{flour,sugar}) = Support({flour,sugar}, eggs) / Support(eggs) = 0.02 / 0.03 = 0.66
  - Lift({flour,sugar}, eggs) = Support({flour,sugar}, eggs) / (Support({flour,sugar})*Support(eggs)) = 0.02 / (0.024*0.03) = 27.8

  Note that Confidence is directional but Lift is not. In the above example, a purchase of {flour,sugar} drives purchase of eggs more strongly than eggs driving purchase of {flour,eggs}.

  A lift value of 1 implies there's no association. A value more than 1 implies a positive association. In our example, the denominator value (0.024*0.03) = 0.00072 = 0.072% is how often both items would occur together if they had no relationship. Lift is giving us a measure of association relative to being random.

- What are the tools or packages available to do market basket analysis?

  R language has the **arules** package for association rule mining. This includes C implementations of Apriori and Eclat algorithms. The **arulesViz** package has useful visualizations that can help in exploratory analysis. It includes visualizations of support, confidence and lift.

  KNIME offers a tool for market basket analysis. It provides a graphical block-diagram-based interface that can be ideal for non-programmers. It offers the Apriori algorithm in traditional as well as the more optimized Borgelt implementation.

  In Python, we can use the **MLxtend** package. Christian Borgelt has also released a C implementation that can be compiled for the Python environment. He calls this **PyFIM**, where FIM stands for Frequency Itemset Mining.

Sample Code

- Python

- *# Example code-snippet showing important methods in python*
- *# Source: https://pbpython.com/market-basket-analysis.html*
- *# Accessed: 2019-03-16*
-
- **from** mlxtend.frequent_patterns **import** apriori
- **from** mlxtend.frequent_patterns **import** association_rules
-
- *#basket_sets are the items bought together derived from invoices after approriate transformation and data cleaning.*
- *#minimum support taken here as 0.07*
- frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
-
- *# recommneded rules/items based on association learning*
- rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
-
- rules.head()

4.6. Neural Network

What is a Neural Network?

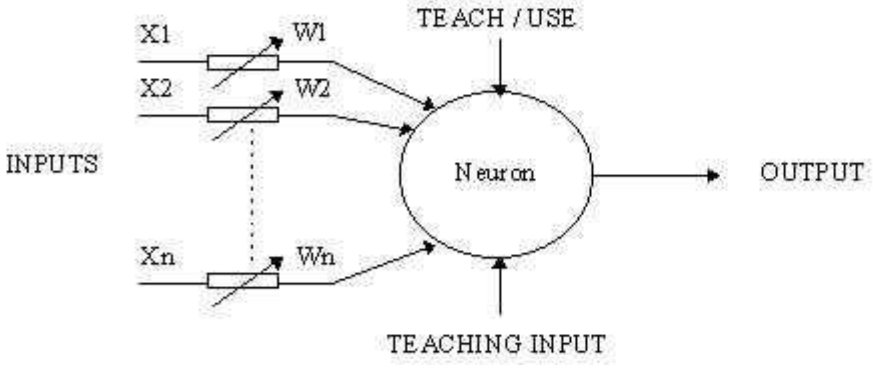
A neural network is either a system software or hardware that works similar to the tasks performed by neurons of human brain. Neural networks include various technologies like deep learning, and machine learning as a part of Artificial Intelligence (AI).

Artificial neural networks (ANN) is the key tool of machine learning. These are systems developed by the inspiration of neuron functionality in the brain, which will replicate the way we humans learn. Neural networks (NN) constitute both input & output layer, as well as a hidden layer containing units that change input into output so that output layer can utilise the value. These are the tools for finding patterns which are numerous & complex for programmers to retrieve and train the machine to recognize the patterns.

Most of the business applications and commercial companies make use of these technologies. Their main aim is to solve complex problems like pattern recognition or facial recognition, and several other applications include -- speech-to-text transcription, data analysis, handwriting recognition for check processing, weather prediction, and signal processing.

4.6.1 Structure of ANNs

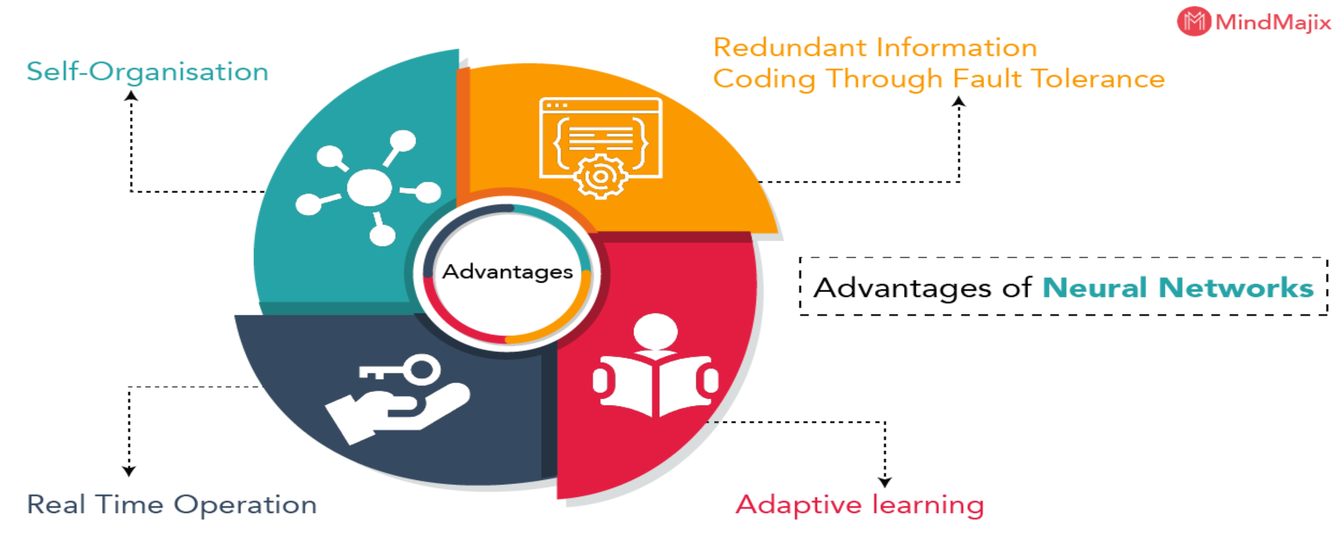
ANN works quite similar to human-brain. By making necessary connections, we can duplicate the working of brain using silicon and wires which act similar to dendrites and neurons. As stimuli from external environment are accepted by dendrites in the same way, the input creates electric impulses that travel through the neural network. ANN consists of several nodes which behave as neurons. The nodes are connected by links (wires) for communication with one another. Nodes take input data to perform small operations on trained data and results of these operations are passed to other nodes (neurons). The output at the node is called its node value. Following is the image representing the basic structure of neuron.



4.6.2 Need for Neural Networks

Neural networks have a remarkable ability to retrieve meaningful data from imprecise data, that is used in detecting trends and extract patterns which are difficult to understand either by computer or humans. A trained NN can be made an "expert" in information that has been given to analyse and can be used for provide projections.

4.6.3Advantages of Neural Networks

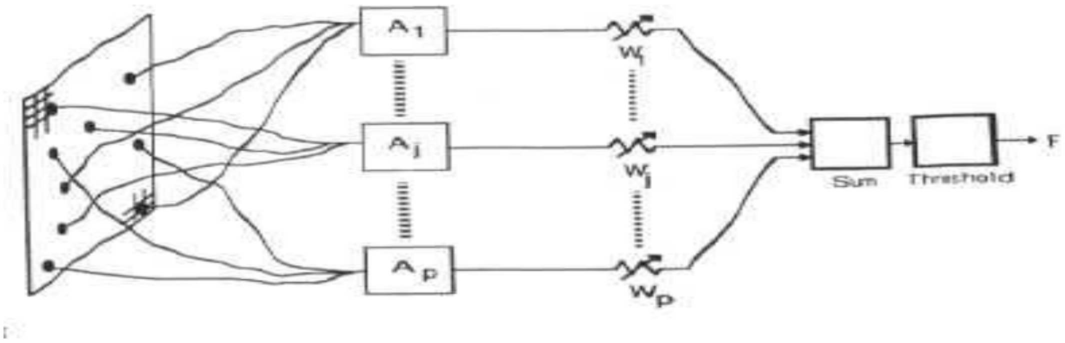


Some of the advantages of neural networks are listed below

- **Self-Organisation:** An ANN can generate its own representation of the information that it receives at the time of learning.
- **Real Time Operation:** ANN calculations may be done simultaneously, and some special (hardware) devices are manufactured which take advantage of this capability.
- **Adaptive learning:** Capability to learn how to solve tasks is based on the data given for training set.
- **Redundant Information Coding Through Fault Tolerance**: Semi destruction of a network leads to degradation of corresponding performance. Moreover, some network will have the ability to retain data even when a major network damage occurs.

4.6.4 Working of Artificial Neural Networks

An ANN includes a huge number of processors working parallelly, which are arranged in layers. The first layer receives the raw data as input, similar to optic nerves of human eye visual processing. Every successive layer receives the raw input data as output from the previous layer, similar to neurons of optic nerve receiving signals from those close to it. The final layer generates the output. Below image shows several layers.

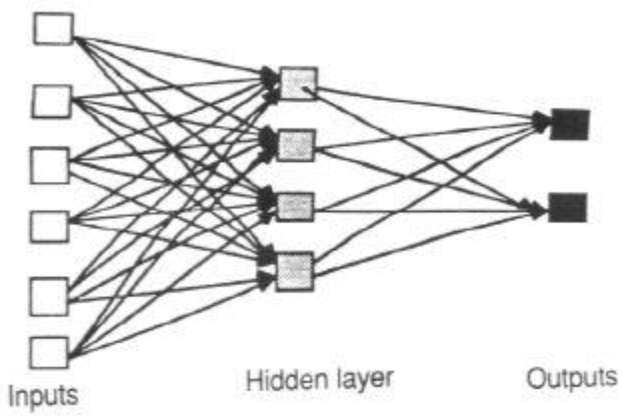


Neural networks are adaptable i.e. they can modify themselves according to the training and run parallely to provide more information about the world. If the network generates a “desired” output, then there is no need to change the trained input data, and vise-versa. If the network generates an “undesired” output resulting errors, then the system modifies the trained input data to improve the results.

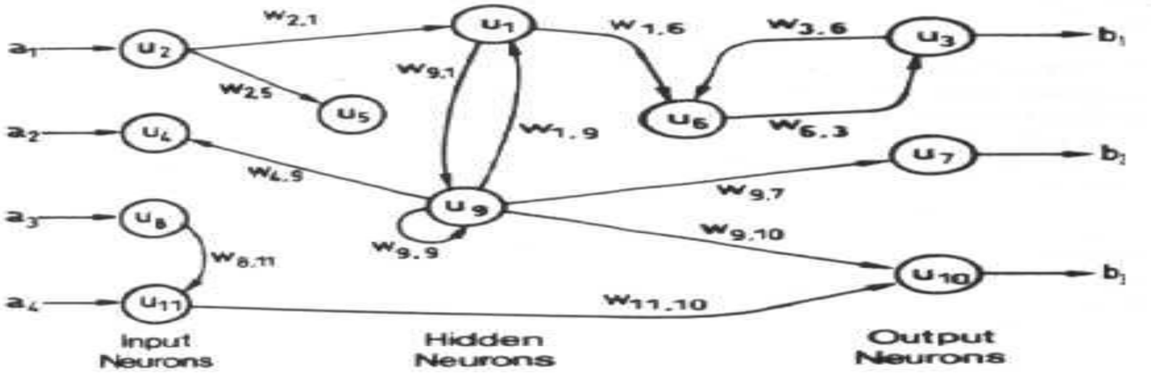
4.6.5 Types of Neural Networks

Neural Networks are of many types and each of these come with a particular use case.

Feedforward Neural Network: This is the most common type of neural network. where information travels in uni direction, that is from input to output.



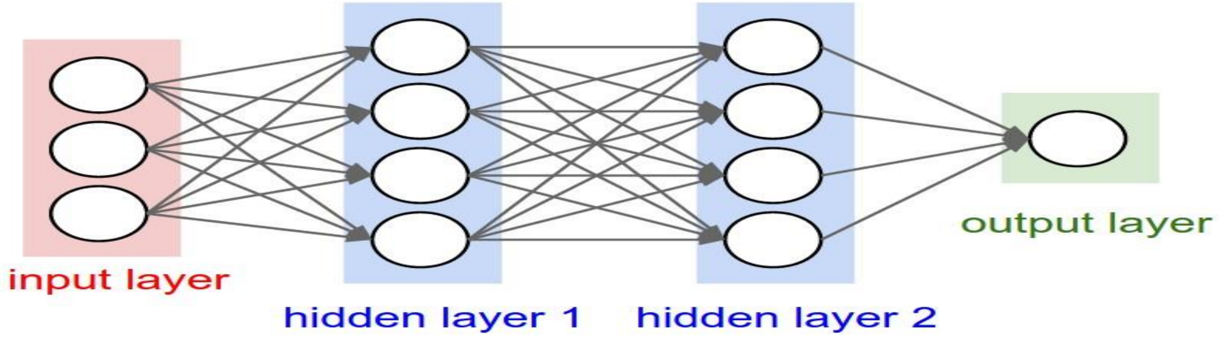
Recurrent Neural Network: A more frequently used network type in which data can be transferred in various directions. These possess greater learning capabilities and are often used for complex tasks like learning handwriting or language recognition.

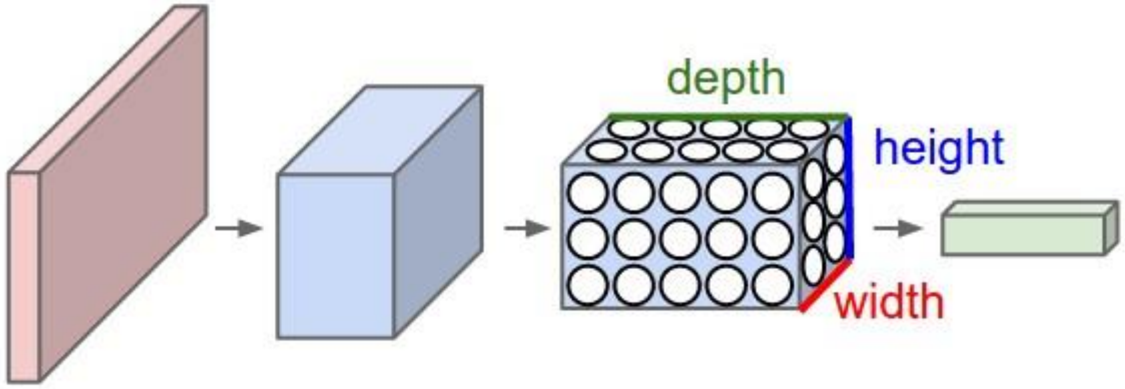


There are also other types which are rarely used, and some of them are, Boltzmann machine networks, Hopfield networks and convolutional neural networks.

Convolutional Neural Networks:

Convolutional Neural Networks are similar to ordinary Neural Networks but, with two hidden layers and they are made up of neurons that have ability to learn. Every neuron receives some inputs, performs a dot product, and sometimes follow non-linearity. The complete network will show a single differentiable score function i.e., from class scores on one end to the raw image pixels on the other end. And, have a loss function (e.g. SVM/Softmax) on the fully-connected layer. The tricks developed by developers to learning neural networks still apply. Choosing the proper network depends on your choice and raw data that is to be given as input must be trained by you according to your preference. At times, we can use multiple approaches, in complex cases like voice recognition.





Artificial Intelligence Tutorials

4.6.6 How do Neural Networks learn from trained Data

In the initial stages, neural networks (NN) are fed with huge amounts of data. Training is generally given by providing input and educating the network what should be the output. For example, facial recognition is the latest technology implemented by many smartphone companies. Each input is gathered by the identification of matching data, like image of the person’s face, iris, various facial expressions, and all these inputs have to be trained. Providing proper answers will allow it to accommodate its internal data to learn how better it can perform.

Rules must be defined in such a way that, each node decides what to be sent to next layer considering its own inputs from the previous layer. This is done by considering many principles like, genetic algorithms, fuzzy logic, gradient-based training Bayesian method. ANNs are given basic rules related to object relationships. Right decision must be taken in building the rules.

Strategies of Machine Learning in ANN

Artificial neural networks have the ability to learn but they should be trained. There are many learning strategies namely:

- **Supervised Learning :** It involves a scholar. For example, the scholar gives examples while preaching for better understanding of the moral. In the same way, ANN implements pattern recognition where it starts guessing while recognizing. Then, the trained data patterns provide the ANN with the answers.
- **Unsupervised Learning :** It comes to action when there is no sample data set with known answers. Searching hidden pattern is one such example. The concept of clustering involves dividing the elements into sets of groups, is based unknown pattern that are carried out using existing data sets.
- **Reinforcement Learning:** It is a strategy built based on observation. The ANN takes decision by considering its environment. If the observations are supposed to be negative, the network adjusts its data to make a different decision for next time.

Applications

Recognition of Image was the first area where neural networks were successfully applied, but the technology expanded to many areas such as

- Natural language processing, translation and language generation.
- Drug discovery and development.
- Stock market prediction.

- Delivery driver route planning and optimization.
- Chatbots.

Neural Networks currently in Practice

In what real time applications neural networks are best suited for ?

A NN has a broad applicability to real time business problems. Now-a-days, they are successfully implemented by many industries including the telecom sectors. Ever since neural networks evolved as a new trend, identifying patterns in data has become much easier as they are well suited for forecasting needs and prediction. For example, industrial process control, sales forecasting, data target marketing, validation, risk management, and customer research are some of its real-time applications.

To be more specific, ANNs are also used in recognition of speakers in communication (speech recognition), undersea mine detection, the google assistant (SIRI), recovery of telecommunications from faulty software, diagnosis of hepatitis, words texture analysis three-dimensional object recognition, handwritten word recognition, and facial recognition.

The above ones are specific areas where neural networks are being applied today. Primary uses involve processes that operate according to strict patterns and have huge amount of data. If data involved is too heavy for a human brain to understand in a reasonable amount of time, the process of automation is easier through artificial neural networks.

4.6.7 Limitations of Neural Networks

From technical point of view, one of the most biggest challenges is the amount of time it takes to train networks, which often require acceptable amount of computing power for even complex tasks. Second most issue to be considered is that neural networks are **black boxes**, in which the user groups the trained data and receives answers. They are allowed to tune the answers, and drawback is that they have no access to the exact process of decision making. This is the reason why researchers are working actively, but artificial neural networks play a very big role in changing day-to-day lives.

4.6.8 Future scope:

Being a highly competitive world, we have a lot to gain from neural networks. Their capability to learn through better example makes them powerful and flexible. Moreover, we need not devise any algorithm to perform a particular task. We don’t require internal mechanisms of that task. These are well suited for real time systems as they respond fast with best computational times because of their parallel architecture.

Neural networks are also contributing to other areas of research like psychology and neurology. In neurology, it is used to investigate the internal mechanisms of the brain and model parts of living organisms. The most exciting aspect of neural networks is that there is a possibility that one-day '**conscious**' networks might arise. Some scientists are arguing that consciousness is a “mechanical property” and conscious neural networks are realistic and are possible. Neural networks have a huge potential and we can get the best from them by collaborating with fuzzy logic, computing, AI and ML.