Jaipur Engineering College & Research Centre, Jaipur
Department of Computer Science and Engineering

Lecture Notes

Artificial Intelligence [6CS4-05]

Unit 4

Vision of the Department:

To become renowned Centre of excellence in computer science and engineering and make competent engineers & professionals with high ethical values prepared for lifelong learning.

Mission of the Department:

M1: To impart outcome based education for emerging technologies in the field of computer science and engineering.
M2: To provide opportunities for interaction between academia and industry.
M3: To provide platform for lifelong learning by accepting the change in technologies.
M4: To develop aptitude of fulfilling social responsibilities.

Program Outcomes (PO):

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and Computer Science & Engineering specialization to the solution of complex Computer Science & Engineering problems.

2. **Problem analysis**: Identify, formulate, research literature, and analyze complex Computer Science and Engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex Computer Science and Engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety,and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of Computer Science and Engineering experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex Computer Science Engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Computer Science and Engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional Computer Science and Engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the Computer Science and Engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings in Computer Science and Engineering.

10. **Communication**: Communicate effectively on complex Computer Science and Engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the Computer Science and Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest contextof technological change in Computer Science and Engineering.

Program Educational Objectives (PEO):

PEO1: To provide students with the fundamentals of Engineering Sciences with more emphasis in Computer Science & Engineering by way of analyzing and exploiting engineering challenges.

PEO2: To train students with good scientific and engineering knowledge so as to comprehend, analyze, design, and create novel products and solutions for the real life problems inComputer Science and Engineering

PEO3: To inculcate professional and ethical attitude, effective communication skills, teamwork skills, multidisciplinary approach, entrepreneurial thinking and an ability to relate engineering issues with social issues for Computer Science & Engineering.

PEO4: To provide students with an academic environment aware of excellence, leadership, written ethical codes and guidelines, and the self-motivated life-long learning needed for a successful professional career in Computer Science & Engineering.

PEO5: To prepare students to excel in Industry and Higher education by Educating Students along with High moral values and Knowledge in Computer Science & Engineering.

Course Outcomes (COs):

CO1: Understand the concept of Artificial Intelligence and apply various Searching techniques.
CO2: Illustrate various Game Playing in Artificial Intelligence system.
CO3: Analyze different Knowledge Representation Techniques, Neural Network, Planning, UncertainKnowledge and Reasoning.
CO4: Apply basic concepts of Learning, Natural Language Processing, Robotics and Expert Systems in AI.

Syllabus:

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Syllabus

III Year-VI Semester: B.Tech. Computer Science and Engineering

6CS4-05: Artificial Intelligence

Credit: 2 **Max. Marks: 100(IA:20, ETE:80)**

2L+0T+0P **End Term Exam: 2 Hours**

SN	Contents	Hours
1	**Introduction:** Objective, scope and outcome of the course.	01
2	**Introduction to AI and Intelligent agent:** Different Approach of AI, Problem Solving : Solving Problems by Searching, Uninformed search, BFS, DFS, Iterative deepening, Bi directional search, Hill climbing, Informed search techniques: heuristic, Greedy search, A* search, AO* search, constraint satisfaction problems.	03
3	**Game Playing:** Minimax, alpha-beta pruning, jug problem, chess problem, tiles problem	06
4	**Knowledge and Reasoning:** Building a Knowledge Base: Propositional logic, first order logic, situation calculus. Theorem Proving in First Order Logic. Planning, partial order planning. Uncertain Knowledge and Reasoning, Probabilities, Bayesian Networks.	06
5	**Learning:** Overview of different forms of learning, Supervised base learning: Learning Decision Trees, SVM, Unsupervised based learning, Market Basket Analysis, Neural Networks.	07
6	**Introduction to Natural Language Processing:** Different issue involved in NLP, Expert System, Robotics.	05
	Total	**28**

Concepts of Learning

Learning is the process of converting experience into expertise or knowledge.

Learning can be broadly classified into three categories, as mentioned below, based on the nature of the learning data and interaction between the learner and the environment.

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning

Similarly, there are four categories of machine learning algorithms as shown below −

- Supervised learning algorithm
- Unsupervised learning algorithm
- Semi-supervised learning algorithm
- Reinforcement learning algorithm

However, the most commonly used ones are **supervised** and **unsupervised learning**.

Supervised Learning

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types - **Regression** and **Classification**.

Regression trains on and predicts a continuous-valued response, for example predicting real estate prices.

Classification attempts to find the appropriate class label, such as analyzing positive/negative sentiment, male and female persons, benign and malignant tumors, secure and unsecure loans etc.

In supervised learning, learning data comes with description, labels, targets or desired outputs and the objective is to find a general rule that maps inputs to outputs. This kind of learning data is called **labeled data**. The learned rule is then used to label new data with unknown outputs.

Supervised learning involves building a machine learning model that is based on **labeled samples**. For example, if we build a system to estimate the price of a plot of land or a house based on various features, such as size, location, and so on, we first need to create a database and label it. We need to teach the algorithm what features correspond to what prices. Based on this data, the algorithm will learn how to calculate the price of real estate using the values of the input features.

Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyzes the training data and produces a derived function that can be used for mapping new examples. There are many **supervised learning algorithms** such as Logistic Regression, Neural networks, Support Vector Machines (SVMs), and Naive Bayes classifiers.

Common **examples** of supervised learning include classifying e-mails into spam and not-spam categories, labeling webpages based on their content, and voice recognition.

Unsupervised Learning

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. There is no labeled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called **unlabeled data**.

Suppose that we have a number of data points, and we want to classify them into several groups. We may not exactly know what the criteria of classification would be. So, an unsupervised learning algorithm tries to classify the given dataset into a certain number of groups in an optimum way.

Unsupervised learning algorithms are extremely powerful tools for analyzing data and for identifying patterns and trends. They are most commonly used for clustering similar input into logical groups. Unsupervised learning algorithms include Kmeans, Random Forests, Hierarchical clustering and so on.

Semi-supervised Learning

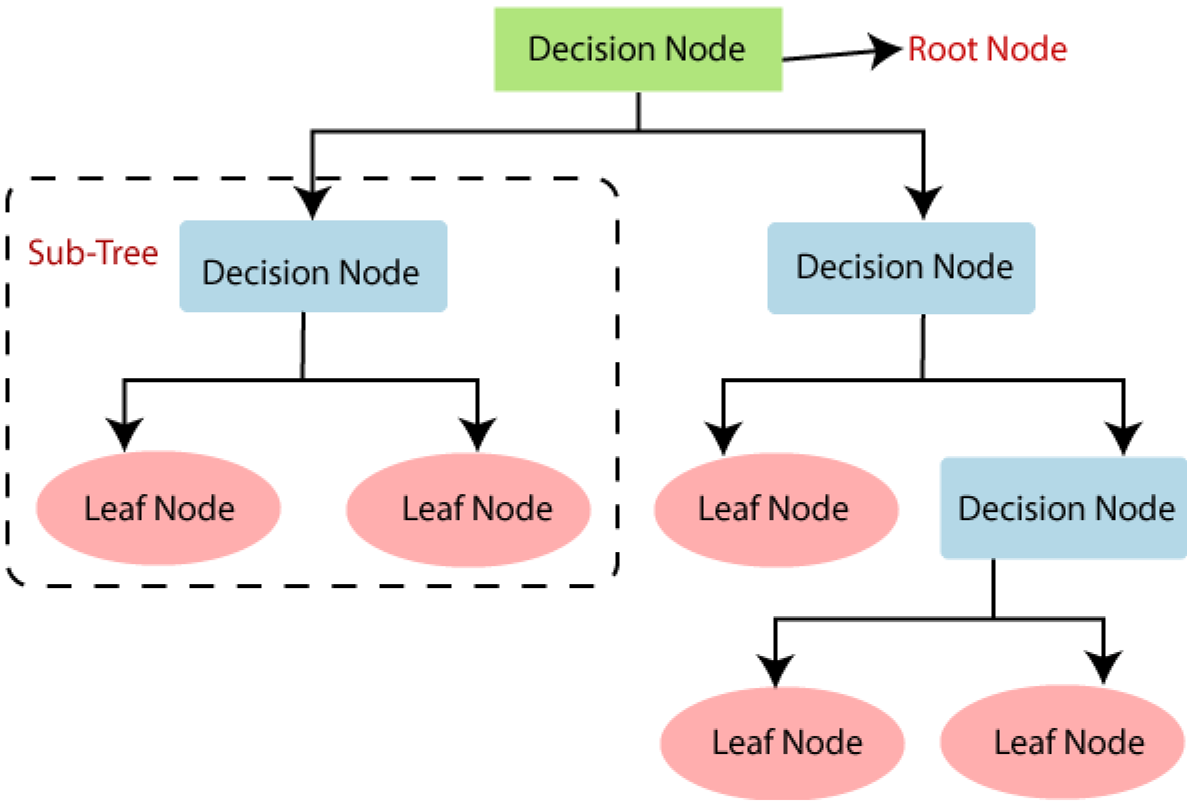
If some learning samples are labeled, but some other are not labeled, then it is semi-supervised learning. It makes use of a large amount of **unlabeled data for training** and a small amount of **labeled data for testing**. Semi-supervised learning is applied in cases where it is expensive to acquire a fully labeled dataset while more practical to label a small subset. For example, it often requires skilled experts to label certain remote sensing images, and lots of field experiments to locate oil at a particular location, while acquiring unlabeled data is relatively easy.

Reinforcement Learning:

Here learning data gives feedback so that the system adjusts to dynamic conditions in order to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly. The best known instances include self-driving cars and chess master algorithm AlphaGo.

Decision Trees

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:



There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

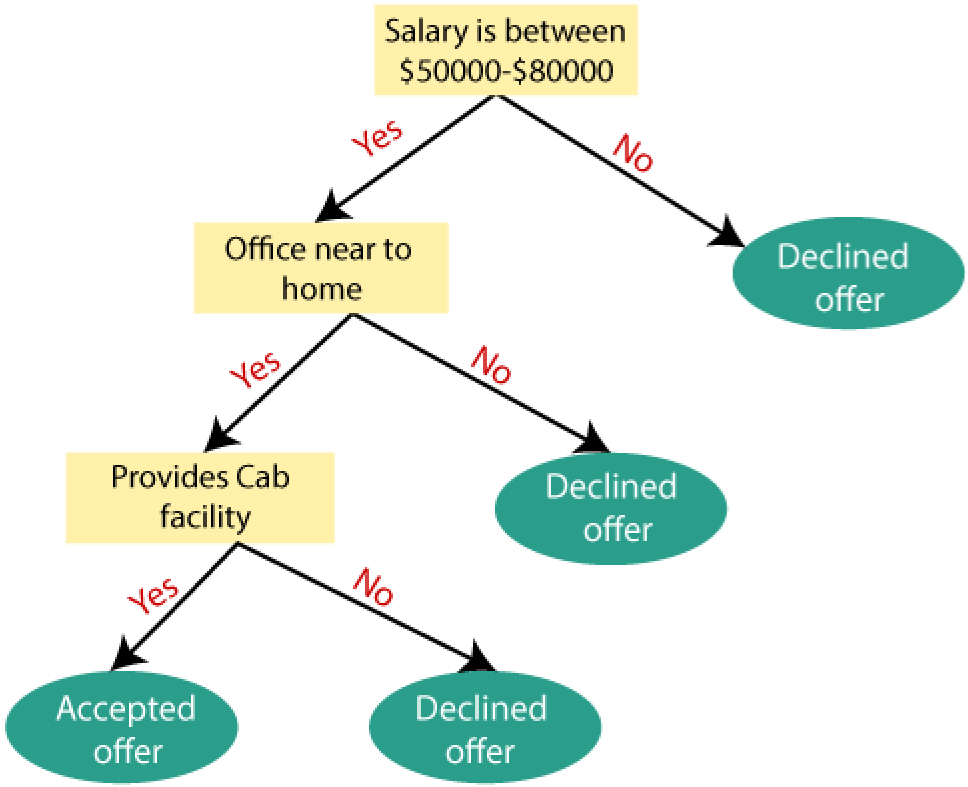
Working

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



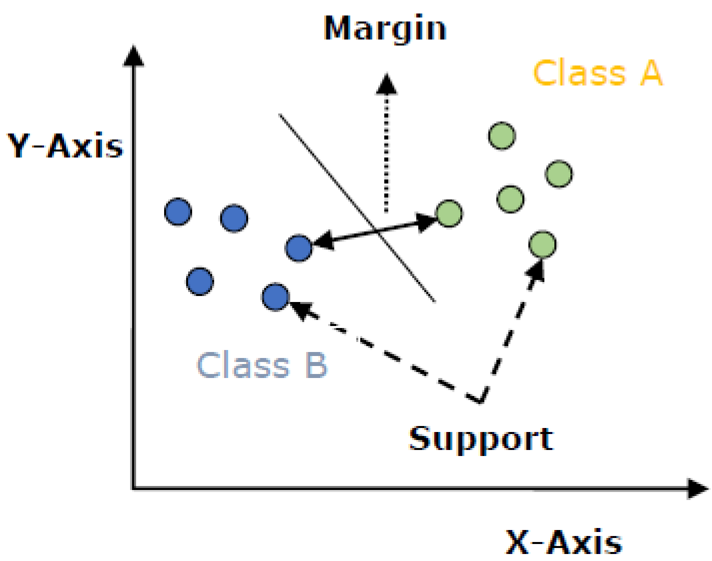
SVM (Support Vector Machine)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

Working of SVM

An SVM model is basically a representation of different classes in a hyper plane in multidimensional space. The hyper plane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyper plane (MMH).



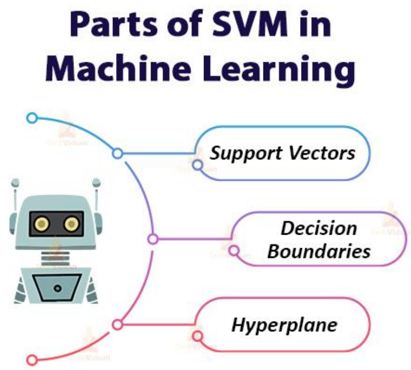
The followings are important concepts in SVM −

- **Support Vectors** − Data points that are closest to the hyper plane is called support vectors. Separating line will be defined with the help of these data points.
- **Hyper plane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
- **Margin** − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyper plane (MMH) and it can be done in the following two steps −

- First, SVM will generate hyper planes iteratively that segregates the classes in best way.
- Then, it will choose the hyper plane that separates the classes correctly.
- It is a strong data classifier. The support vector machine uses two or more labelled classes of data. It separates two different classes of data by a hyper plane. The data points based on their position according to the hyper plane will be put in separate classes. In addition, an important thing to note is that SVM in Machine Learning always uses graphs to plot the data. Therefore, we will be seeing some graphs in the article. Now, let’s learn some more stuff.

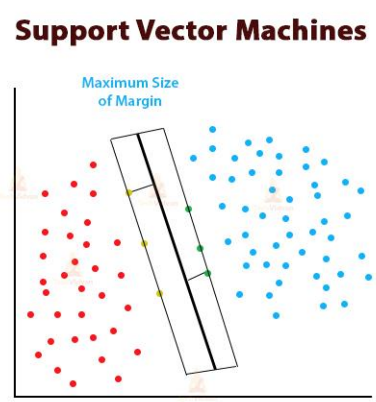
- Parts of SVM in Machine Learning



To understand SVM mathematically, we have to keep in mind a few important terms. These terms will always come whenever you use the SVM algorithm. So let's start looking at them one by one.

1. Support Vectors

Support vectors are special data points in the dataset. They are responsible for the construction of the hyperplane and are the closest points to the hyperplane. If these points were removed, the position of the hyperplane would be altered. The hyperplane has decision boundaries around it. And, the support vectors help in decreasing and increasing the size of the boundaries. They are the main components in making an SVM. We can see the picture for this.



The yellow and green points here are the support vectors. Red and blue dots are separate classes. The middle dark line is the hyperplane in 2-D and the two lines alongside the hyperplane are the decision boundaries. They collectively form the decision surface.

2. Decision Boundaries

Decision boundaries in SVM are the two lines that we see alongside the hyperplane. The distance between the two light-toned lines is called the margin. An optimal or best hyperplane form when the margin size is maximum. The SVM algorithm adjusts the hyperplane and its margins according to the support vectors.

3. Hyperplane

The hyperplane is the central line in the diagram above. In this case, the hyperplane is a line because the dimension is 2-D. If we had a 3-D plane, the hyperplane would have been a 2-D plane itself. There is a lot of mathematics involved in studying the hyperplane. We will be looking at that. But, to understand a hyperplane we need to imagine it first. Imagine there is a **feature space (a blank piece of paper)**. Now, imagine a line cutting through it from the center. That is the hyperplane. The math equation for the hyperplane is a linear equation.

$\mathbf{a_0 + a_1x_1 + a_2x_2 + \ldots\ldots. + a_nx_n}$

This is the equation. Here a0 is the intercept of the hyperplane. Also, a1 and a2 define the first and second axes respectively. X1 and X2 are for two dimensions. Let us assume that the equation is equal to E. So if the data points lie beneath the hyperplane then E<0. If they are above it, the E>=0. This is how we classify data using a hyperplane.

In any ML method, we would have the training and testing data. So here we have n*p matrix which has n observations and p dimensions. We have a variable Y, which decides in which class the points would lie. So, we have two values 1 and -1. Y can only be these two values in any case. If Y is 1 then data is in class 1. If Y is -1 then data is in class -1.

Working of SVM

For this, let us compare the working of SVM and other classifiers for better understanding. Let us talk about SVMs and perceptrons. Perceptrons and other classifiers focus on all the points that are present in the data. For these classifiers, the focus is more on just separating the complex points and adjusting the dividing line.

Perceptrons are made by taking one point at a time and fixing the dividing line accordingly. When all the complex points are separated, the perceptron algorithm stops. This is the end of the process for these classifiers, as they do not improve the position of the dividing line. So, finding the optimal dividing line is not what the perceptron does.

Whereas, the case with SVM is a little different. In this, the algorithm only focuses on the points that are complex to separate and it ignores the rest of the points. The algorithm finds the points that are closest to each other. It then draws a line between them. The dividing line would be optimal if it is perpendicular to the line connecting the points.

The best part is that two different classes are formed on either side of the line. Whatever new point enters the dataset, it won't be affecting the hyperplane. The only points that would affect the hyperplane are the support vectors. The hyperplane won't allow the data from both classes to mix in most cases. Also, the hyperplane can adjust itself by maximizing the size of its margin. The margin is the space between the hyperplane and the decision boundaries. This is how the SVM in Machine Learning works.

Pros and Cons of SVM in Machine Learning

Now, let's discuss the advantages and disadvantages of SVM in Machine Learning.

Pros of SVM in Machine Learning

- SVMs have better results in production than ANNs do.
- They can efficiently handle higher dimensional and linearly inseparable data. They are quite memory efficient.
- Complex problems can be solved using kernel functions in the SVM. This comes under the kernel trick which is a big asset for SVM.
- SVM works well with all three types of data (structured, semi-structured and unstructured).
- Over-fitting is a problem avoided by SVM. This is because SVM has regularisation parameters and generalization in its models.
- There are various types of kernel functions for various decision functions.
- We can add different kernel functions together to achieve more complex hyperplanes.

Cons of SVM in Machine Learning

- Choosing a kernel function is not an easy task (especially a good one).
- The tuning of SVM parameters is not easy. Their effect on the model is hard to see.
- SVM takes a lot of time for training with large datasets.
- It is hard to predict the final model as there can be a lot of minute changes. So, recalibrating the model each time is not a solution.
- If there are more features than samples in the data, the model will give a poor performance.

Supervised and Unsupervised Learning

Supervised learning

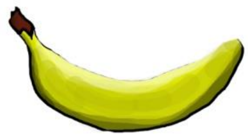
Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one like this:

- If shape of object is rounded and depression at top having color Red then it will be labeled as –**Apple**.

- If shape of object is long curving cylinder having color Green-Yellow then it will be labeled as –**Banana**.

Now suppose after training the data, you have given a new separate fruit say Banana from basket and asked to identify it.



Since the machine has already learned the things from previous data and this time have to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in Banana category. Thus the machine learns the things from training data(basket containing fruits) and then apply the knowledge to test data(new fruit).

Supervised learning classified into two categories of algorithms:

- **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.
- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Supervised learning deals with or learns with “labeled” data.Which implies that some data is already tagged with the correct answer.

Types:-

- Regression
- Logistic Regression
- Classification
- Naive Bayes Classifiers
- K-NN (k nearest neighbors)
- Decision Trees
- Support Vector Machine

Advantages:-

- Supervised learning allows collecting data and produce data output from the previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problems.

Disadvantages:-

- Classifying big data can be challenging.
- Training for supervised learning needs a lot of computation time.So,it requires a lot of time.

<u>Unsupervised learning</u>

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here 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.
For instance, suppose it is given an image having both dogs and cats which have not seen ever.

Thus the machine has no idea about the features of dogs and cat so we can't categorize it in dogs and cats. But it can categorize them according to their similarities, patterns, and differences i.e., we can easily categorize the above picture into two parts. First first may contain all pics having **dogs** in it and second part may contain all pics having **cats** in it. Here you didn't learn anything before, means no training data or examples.
It allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with unlabelled data.

Unsupervised learning classified into two categories of algorithms:

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Types of Unsupervised Learning:-

Clustering
1. Exclusive (partitioning)
2. Agglomerative
3. Overlapping
4. Probabilistic

Clustering Types:-
1. Hierarchical clustering
2. K-means clustering
3. Principal Component Analysis
4. Singular Value Decomposition
5. Independent Component Analysis

Market Basket Analysis

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together.

The adoption of market basket analysis was aided by the advent of electronic point-of-sale (POS) systems. Compared to handwritten records kept by store owners, the digital records generated by POS systems made it easier for applications to process and analyze large volumes of purchase data.

Implementation of market basket analysis requires a background in statistics and data science, as well as some algorithmic computer programming skills. For those without the needed technical skills, commercial, off-the-shelf tools exist.

One example is the Shopping Basket Analysis tool in Microsoft Excel, which analyzes transaction data contained in a spreadsheet and performs market basket analysis. The items to be analyzed must be related by a transaction ID. The Shopping Basket Analysis tool then creates two worksheets: the Shopping Basket Item Groups worksheet, which lists items that are frequently purchased together, and the Shopping Basket Rules worksheet, which shows how items are related (For example, purchasers of Product A are likely to buy Product B).

Types of market basket analysis

There are two types of market basket analysis:

1. Predictive market basket analysis: This type considers items purchased in sequence to determine cross-sell
2. Differential market basket analysis: This type considers data across different stores, as well as purchases from different customer groups during different times of the day, month or year. If a rule holds in one dimension (like store, time period or customer group), but does

not hold in the others, analysts can determine the factors responsible for the exception. These insights can lead to new product offers that drive higher sales.

Algorithms associated with market basket analysis

In market basket analysis, association rules are used to predict the likelihood of products being purchased together. Association rules count the frequency of items that occur together, seeking to find associations that occur far more often than expected.

Algorithms that use association rules include AIS, SETM and Apriori. The Apriori algorithm is commonly cited by data scientists in research articles about market basket analysis and is used to identify frequent items in the database, then evaluate their frequency as the datasets are expanded to larger sizes.

The arules package for R is an open source toolkit for association mining using the R programming language. This package supports the Apriori algorithm, along with other mining algorithms, including arulesNBMiner, opusminer, RKEEL and RSarules.

Examples of market basket analysis

The Amazon website employs a well-known example of market basket analysis. On a product page, Amazon presents users with related products, under the headings of "Frequently bought together" and "Customers who bought this item also bought."

Market basket analysis also applies to bricks-and-mortar stores. If analysis showed that magazine purchases often include the purchase of a bookmark (which could be considered an unexpected combination, since the consumer did not purchase a book), then the book store might place a selection of bookmarks near the magazine rack.

Benefits of market basket analysis

Market basket analysis can increase sales and customer satisfaction. Using data to determine that products are often purchased together, retailers can optimize product placement, offer special deals and create new product bundles to encourage further sales of these combinations.

These improvements can generate additional sales for the retailer, while making the shopping experience more productive and valuable for customers. By using market basket analysis, customers may feel a stronger sentiment or brand loyalty toward the company.