CST401 ARTIFICIAL INTELLIGENCE

MODULE 5

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Explain the fundamental concepts of intelligent systems and their architecture. **(Cognitive Knowledge Level: Understanding)**
CO2	Illustrate uninformed and informed search techniques for problem solving in intelligent systems. **(Cognitive Knowledge Level: Understanding)**
CO3	Solve Constraint Satisfaction Problems using search techniques. **(Cognitive Knowledge Level: Apply)**
CO4	Represent AI domain knowledge using logic systems and use inference techniques for reasoning in intelligent systems. **(Cognitive Knowledge Level: Apply)**
CO5	Illustrate different types of learning techniques used in intelligent systems **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											
CO2	✓	✓										✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓			✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

****

SYLLABUS- MODULE 5 (Machine Learning)

Learning from Examples –Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and choosing the best hypothesis, Regression and classification with Linear models.

****

Learning From Example

> ***Agents that can improve their behavior through diligent study of their own experiences***

An agent is learning if it improves its performance on future tasks after making observations about the world. Machine Learning is defined as a technology that is used to train machines to perform various actions such as predictions, recommendations, estimations, etc., based on historical data or past experience.

Machine Learning enables computers to behave like human beings by training them with the help of past experience and predicted data.

There are three key aspects of Machine Learning, which are as follows:

1. Task: A task is defined as the main problem in which we are interested. This task/problem can be related to the predictions and recommendations and estimations, etc.
2. Experience: It is defined as learning from historical or past data and used to estimate and resolve future tasks.
3. Performance: It is defined as the capacity of any machine to resolve any machine learning task or problem and provide the best outcome for the same. However, performance is dependent on the type of machine learning problems.

Techniques in Machine Learning

Machine Learning techniques are divided mainly into the following 4 categories:

1. Supervised Learning

Supervised learning is applicable when a machine has sample data, i.e., input as well as output data with correct labels. Correct labels are used to check the correctness of the model using some labels and tags. Supervised learning technique helps us to predict future events with the help of past experience and labeled examples. Initially, it analyses the known training dataset, and later it introduces an inferred function that makes predictions about output values. Further, it also predicts errors during this entire learning process and also corrects those errors through algorithms.

Example: Let's assume we have a set of images tagged as "dog". A machine learning algorithm is trained with these dog images so it can easily distinguish whether an image is a dog or not.

2. Unsupervised Learning

In unsupervised learning, a machine is trained with some input samples or labels only, while output is not known. The training information is neither classified nor labeled; hence, a machine may not always provide correct output compared to supervised learning.

Although Unsupervised learning is less common in practical business settings, it helps in exploring the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Example: Let's assume a machine is trained with some set of documents having different categories (Type A, B, and C), and we have to organize them into appropriate groups. Because the machine is provided only with input samples or without output, so, it can organize these datasets into type A, type B, and type C categories, but it is not necessary whether it is organized correctly or not.

3. Reinforcement Learning

Reinforcement Learning is a feedback-based machine learning technique. In such type of learning, agents (computer programs) need to explore the environment, perform actions, and on the basis of their actions, they get rewards as feedback. For each good action, they get a positive reward, and for each bad action, they get a negative reward. The goal of a Reinforcement learning agent is to maximize the positive rewards. Since there is no labeled data, the agent is bound to learn by its experience only.

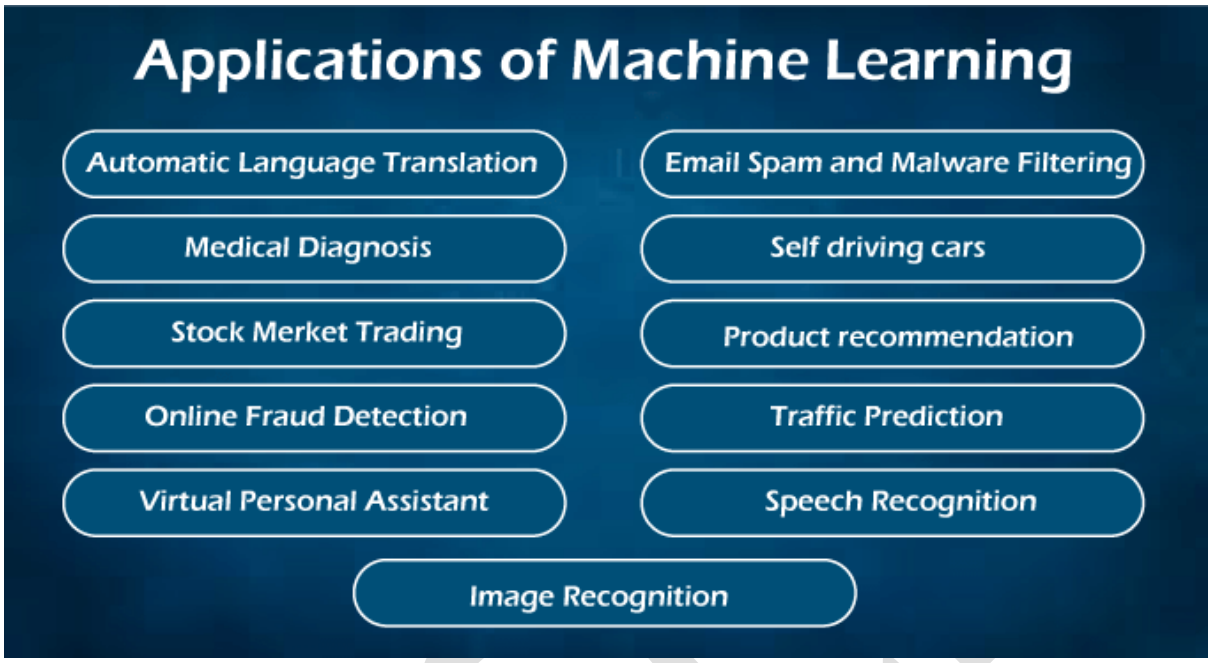
4. Semi-supervised Learning

Semi-supervised Learning is an intermediate technique of both supervised and unsupervised learning. It performs actions on datasets having few labels as well as unlabeled data. However, it generally contains unlabeled data. Hence, it also reduces the cost of the machine learning model as labels are costly, but for corporate purposes, it may have few labels. Further, it also increases the accuracy and performance of the machine learning model.

Sem-supervised learning helps data scientists to overcome the drawback of supervised and unsupervised learning. Speech analysis, web content classification, protein sequence classification, text documents classifiers., etc., are some important applications of Semi-supervised learning.

Applications of Machine Learning

Machine Learning is widely being used in approximately every sector, including healthcare, marketing, finance, infrastructure, automation, etc. There are some important real-world examples of machine learning, which are as follows:



Healthcare and Medical Diagnosis:

Machine Learning is used in healthcare industries that help in generating neural networks. These self-learning neural networks help specialists for providing quality treatment by analyzing external data on a patient's condition, X-rays, CT scans, various tests, and screenings. Other than treatment, machine learning is also helpful for cases like automatic billing, clinical decision supports, and development of clinical care guidelines, etc.

Marketing:

Machine learning helps marketers to create various hypotheses, testing, evaluation, and analyze datasets. It helps us to quickly make predictions based on the concept of big data. It is also helpful for stock marketing as most of the trading is done through bots and based on calculations from machine learning algorithms. Various Deep Learning Neural network helps to build trading models such as Convolutional Neural Network, Recurrent Neural Network, Long-short term memory, etc.

Self-driving cars:

This is one of the most exciting applications of machine learning in today's world. It plays a vital role in developing self-driving cars. Various automobile companies like Tesla, Tata, etc., are continuously working for the development of self-driving cars. It also becomes possible by the machine learning method (supervised learning), in which a machine is trained to detect people and objects while driving.

Speech Recognition:

Speech Recognition is one of the most popular applications of machine learning. Nowadays, almost every mobile application comes with a voice search facility. This "Search By Voice" facility is also a part of speech recognition. In this method, voice instructions are converted into text, which is known as Speech to text" or "Computer speech recognition.

Google assistant, SIRI, Alexa, Cortana, etc., are some famous applications of speech recognition.

Traffic Prediction:

Machine Learning also helps us to find the shortest route to reach our destination by using Google Maps. It also helps us in predicting traffic conditions, whether it is cleared or congested, through the real-time location of the Google Maps app and sensor.

Image Recognition:

Image recognition is also an important application of machine learning for identifying objects, persons, places, etc. Face detection and auto friend tagging suggestion is the most famous application of image recognition used by Facebook, Instagram, etc. Whenever we upload photos with our Facebook friends, it automatically suggests their names through image recognition technology.

Product Recommendations:

Machine Learning is widely used in business industries for the marketing of various products. Almost all big and small companies like Amazon, Alibaba, Walmart, Netflix, etc., are using machine learning techniques for products recommendation to their users. Whenever we search for any products on their websites, we automatically get started with lots of advertisements for similar products. This is also possible by Machine Learning algorithms that learn users' interests and, based on past data, suggest products to the user.

Automatic Translation:

Automatic language translation is also one of the most significant applications of machine learning that is based on sequence algorithms by translating text of one language into other desirable languages. Google GNMT (Google Neural Machine Translation) provides this feature, which is Neural Machine Learning. Further, you can also translate the selected text on images as well as complete documents through Google Lens.

Virtual Assistant:

A virtual personal assistant is also one of the most popular applications of machine learning. First, it records out voice and sends to cloud-based server then decode it with the help of machine learning algorithms. All big companies like Amazon, Google, etc., are using these

features for playing music, calling someone, opening an app and searching data on the internet, etc.

Email Spam and Malware Filtering:

Machine Learning also helps us to filter various Emails received on our mailbox according to their category, such as important, normal, and spam. It is possible by ML algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier.

Supervised learning

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

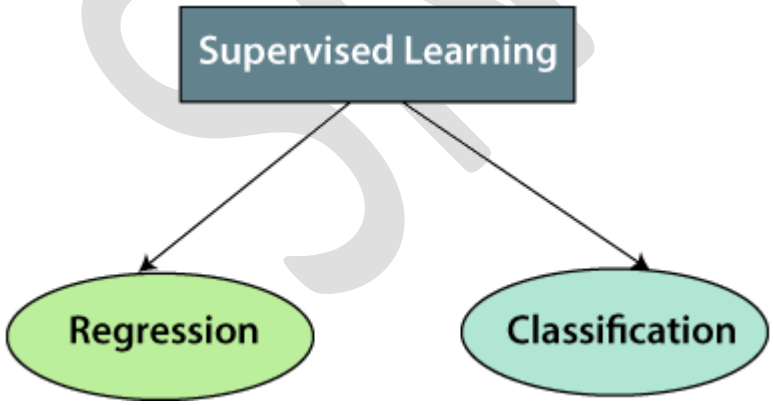
In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:



Supervised Machine learning

1. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc. Spam Filtering,

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Advantages of Supervised learning:

With the help of supervised learning, the model can predict the output on the basis of prior experiences. In supervised learning, we can have an exact idea about the classes of objects. Supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc.

Disadvantages of supervised learning:

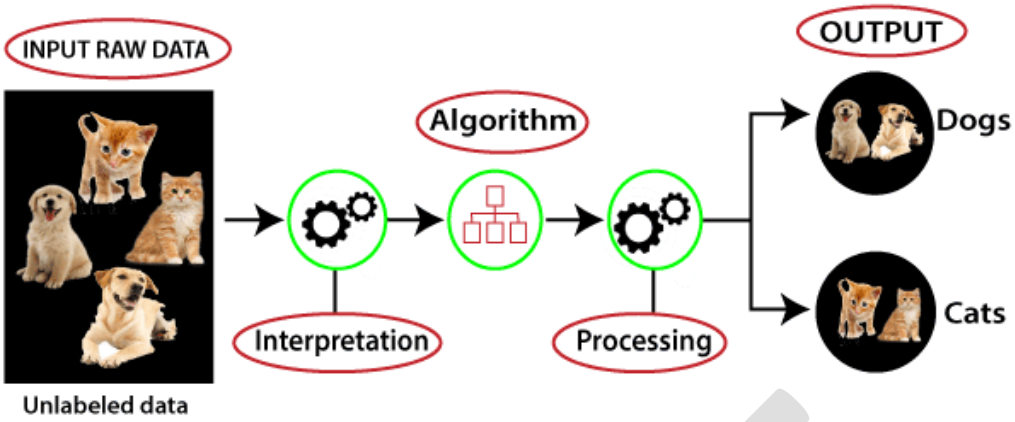
Supervised learning models are not suitable for handling the complex tasks. Supervised learning cannot predict the correct output if the test data is different from the training dataset. Training required lots of computation times. In supervised learning, we need enough knowledge about the classes of object.

Unsupervised Learning

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. It can be defined as:

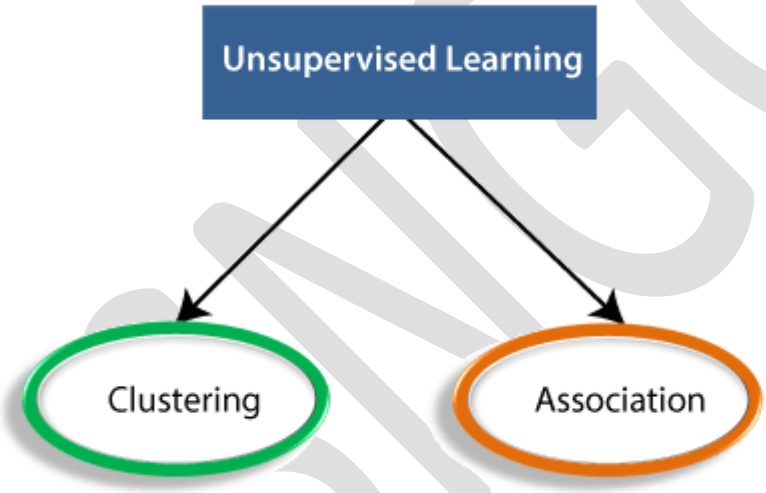
Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.



Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:



- Clustering: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.
- Association: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Unsupervised Learning algorithms:

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

Advantages of Unsupervised Learning

Unsupervised learning is used for more complex tasks as compared to supervised learning because, in unsupervised learning, we don't have labelled input data. Unsupervised learning is preferable as it is easy to get unlabelled data in comparison to labelled data.

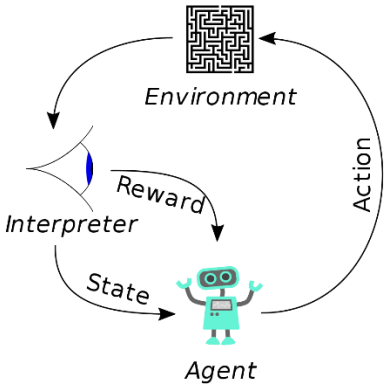
Disadvantages of Unsupervised Learning

Unsupervised learning is intrinsically more difficult than supervised learning as it does not have corresponding output. The result of the unsupervised learning algorithm might be less accurate as input data is not labelled, and algorithms do not know the exact output in advance.

Reinforcement learning

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Example: The problem is as follows: We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.



Semi-supervised learning

In semi-supervised learning we are given a few labeled examples and must make what we can of a large collection of unlabeled examples. Even the labels themselves may not be the oracular truths that we hope for. Imagine that you are trying to build a system to guess a person's age from a photo. You gather some labeled examples by snapping pictures of people and asking their age. That's supervised learning. But in reality some of the people lied about their age. It's not just that there is random noise in the data; rather the inaccuracies are systematic, and to uncover them is an unsupervised learning problem involving images, self-reported ages, and true (unknown) ages. Thus, both noise and lack of labels create a continuum between supervised and unsupervised learning.

CLASSIFICATION AND REGRESSION

When the output y is one of a finite set of values (such as sunny, cloudy or rainy), the learning problem is called classification, and is called Boolean or binary classification if there are only two values.

When y is a number (such as tomorrow's temperature), the REGRESSION learning problem is called regression

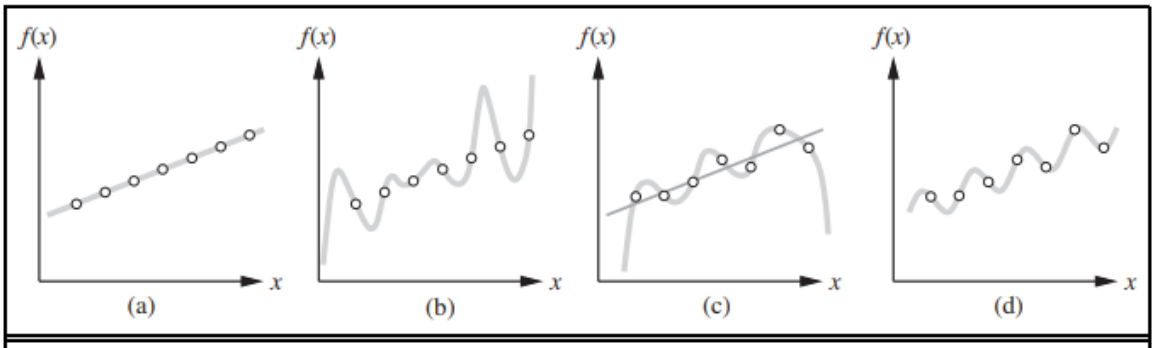


Figure 18.1 (a) Example $(x, f(x))$ pairs and a consistent, linear hypothesis. (b) A consistent, degree-7 polynomial hypothesis for the same data set. (c) A different data set, which admits an exact degree-6 polynomial fit or an approximate linear fit. (d) A simple, exact sinusoidal fit to the same data set.

Fitting a function of a single variable to some data points. The examples are points in the (x, y) plane, where y = f(x). We don't know what f is, but we will approximate it with a function h selected from a hypothesis space, H, which for this example we will take to be the set of polynomials, such as x^5+3x^2+2

Fig (a) shows some data with an exact fit by a straight line. The line is called a consistent hypothesis because it agrees with all the data.

Figure (b) shows a high degree polynomial that is also consistent with the same data. This illustrates a fundamental problem in inductive learning: how do we choose from among multiple consistent hypotheses? One answer is to prefer the simplest hypothesis consistent with the data.

This principle is called **Ockham's razor**, after the 14th-century English philosopher William of Ockham, who used it to argue sharply against all sorts of complications. Defining simplicity is not easy, but it seems clear that a degree-1 polynomial is simpler than a degree-7 polynomial, and thus (a) should be preferred to (b).

Figure (c) shows a second data set. There is no consistent straight line for this data set; in fact, it requires a degree-6 polynomial for an exact fit. There are just 7 data points, so a polynomial with 7 parameters does not seem to be finding any pattern in the data and we do not expect it to generalize well. A straight line that is not consistent with any of the data points, but might generalize fairly well for unseen values of x, is also shown in (c). In general, there is a tradeoff between complex hypotheses that fit the training data well and simpler hypotheses that may generalize better.

Figure (d) we expand the hypothesis space H to allow polynomials over both x and sin(x), and find that the data in (c) can be fitted exactly by a simple function of the form ax + b + c sin(x). This shows the importance of the choice of hypothesis space. We say that a learning problem is **realizable** if the hypothesis space contains the true function. Unfortunately, we cannot always tell whether a given learning problem is realizable, because the true function is not known.

Supervised learning can be done by choosing the hypothesis h^* that is most probable given the data:

$$h^* = \operatorname{argmax}_{h \in \mathcal{H}} P(h|data) \ .$$

By Bayes' rule this is equivalent to

$$h^* = \operatorname{argmax}_{h \in \mathcal{H}} P(data|h) \, P(h) \ .$$

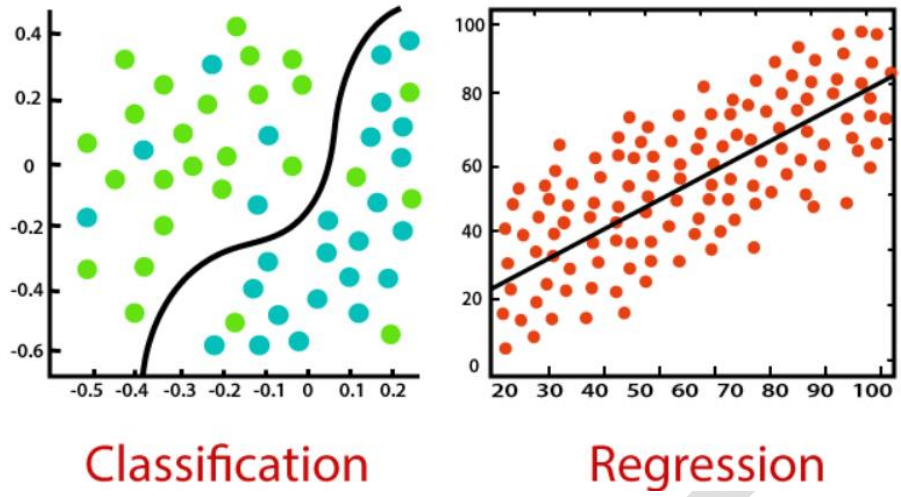
Then we can say that the prior probability P(h) is high for a degree-1 or -2 polynomial, lower for a degree-7 polynomial, and especially low for degree-7 polynomials with large, sharp spikes as in Figure 18.1(b). We allow unusual-looking functions when the data say we really need them, but we discourage them by giving them a low prior probability.

Regression vs. Classification in Machine Learning

Regression and Classification algorithms are Supervised Learning algorithms. Both the algorithms are used for prediction in Machine learning and work with the labeled datasets. But the difference between both is how they are used for different machine learning problems.

The main difference between Regression and Classification algorithms that Regression algorithms are used to predict the continuous values such as price, salary, age, etc. and Classification algorithms are used to predict/Classify the discrete values such as Male or Female, True or False, Spam or Not Spam, etc.

Consider the below diagram:



Classification:

Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters. In Classification, a computer program is trained on the training dataset and based on that training, it categorizes the data into different classes.

The task of the classification algorithm is to find the mapping function to map the input(x) to the discrete output(y).

Example: The best example to understand the Classification problem is Email Spam Detection. The model is trained on the basis of millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam or not. If the email is spam, then it is moved to the Spam folder.

Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the following types:

- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machines
- Kernel SVM
- Naïve Bayes
- Decision Tree Classification
- Random Forest Classification

Regression:

Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of Market Trends, prediction of House prices, etc. The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).

Example: Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

Types of Regression Algorithm:

Simple Linear Regression

Multiple Linear Regression

Polynomial Regression

Support Vector Regression

Decision Tree Regression

Random Forest Regression

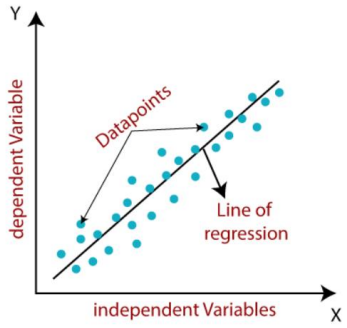
Difference between Regression and Classification

Regression Algorithm	Classification Algorithm
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete value.
The task of the regression algorithm is to map the input value (x) with the continuous output variable(y).	The task of the classification algorithm is to map the input value(x) with the discrete output variable(y).
Regression Algorithms are used with continuous data.	Classification Algorithms are used with discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, which can divide the dataset into different classes.
Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classification problems such as Identification of spam emails, Speech Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into Binary Classifier and Multi-class Classifier.

Linear Regression in Machine Learning

Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the

value of the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables.



Mathematically, we can represent a linear regression as:

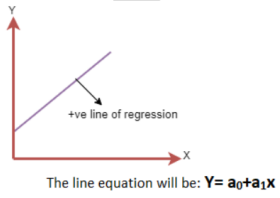
Here,

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

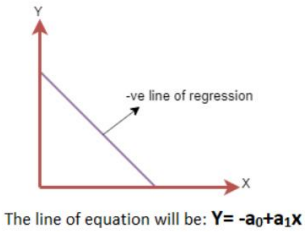
Linear Regression Line

A linear line showing the relationship between the dependent and independent variables is called a **regression line**. A regression line can show two types of relationship:

Positive Linear Relationship:
If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



Negative Linear Relationship:
If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized.

The best fit line will have the least error.

The different values for weights or the coefficient of lines (a0, a1) gives a different line of regression, so we need to calculate the best values for a0 and a1 to find the best fit line, so to calculate this we use cost function.

Cost function

The different values for weights or coefficient of lines (a0, a1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.

Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.

We can use the cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.

For Linear Regression, we use the Mean Squared Error (MSE) cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

$$MSE = 1\frac{1}{N}\sum_{i=1}^{n}(y_i - (a_1x_i + a_0))^2$$

Where,

N=Total number of observation

Yi = Actual value

(a1xi+a0)= Predicted value.

Gradient Descent

Gradient descent is used to minimize the MSE by calculating the gradient of the cost function. A regression model uses gradient descent to update the coefficients of the line by reducing the cost function. It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

Model Performance

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called optimization. It can be achieved by R-squared method. R-squared is a statistical method that determines the goodness of fit. It measures the strength of the relationship between the dependent and independent

variables on a scale of 0-100%. The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model. It is also called a coefficient of determination, or coefficient of multiple determination for multiple regression. It can be calculated from the below formula:

$$\text{R-squared} = \frac{\text{Explained variation}}{\text{Total Variation}}$$

Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- Simple Linear Regression: If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- Multiple Linear regression: If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Simple Linear Regression

Simple Linear Regression is a type of Regression algorithms that models the relationship between a dependent variable and a single independent variable. The relationship shown by a Simple Linear Regression model is linear or a sloped straight line, hence it is called Simple Linear Regression.

The key point in Simple Linear Regression is that the dependent variable must be a continuous/real value. However, the independent variable can be measured on continuous or categorical values.

Simple Linear regression algorithm has mainly two objectives:

- Model the relationship between the two variables. Such as the relationship between Income and expenditure, experience and Salary, etc.
- Forecasting new observations. Such as Weather forecasting according to temperature, Revenue of a company according to the investments in a year, etc.

Simple Linear Regression Model:

The Simple Linear Regression model can be represented using the below equation:

$y = a0 + a1x + \varepsilon$

Where,

a0= It is the intercept of the Regression line (can be obtained putting x=0)

a1= It is the slope of the regression line, which tells whether the line is increasing or decreasing.

ε = The error term. (For a good model it will be negligible)

Multiple Linear Regression

In Simple Linear Regression, where a single Independent/Predictor(X) variable is used to model the response variable (Y). But there may be various cases in which the response variable is affected by more than one predictor variable; for such cases, the Multiple Linear Regression algorithm is used.

Moreover, Multiple Linear Regression is an extension of Simple Linear regression as it takes more than one predictor variable to predict the response variable. We can define it as:

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.

Example:

Prediction of CO_2 emission based on engine size and number of cylinders in a car.

For MLR, the dependent or target variable(Y) must be the continuous/real, but the predictor or independent variable may be of continuous or categorical form.

Each feature variable must model the linear relationship with the dependent variable.

MLR tries to fit a regression line through a multidimensional space of data-points.

MLR equation:

In Multiple Linear Regression, the target variable(Y) is a linear combination of multiple predictor variables x1, x2, x3, ...,xn. Since it is an enhancement of Simple Linear Regression, so the same is applied for the multiple linear regression equation, the equation becomes:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations:

y_i = dependent variable

x_i = explanatory variables

β_0 = y-intercept (constant term)

β_p = slope coefficients for each explanatory variable

ϵ = the model's error term (also known as the residuals)

Multivariate Regression

Multivariate regression allows one to have a different view of the relationship between various variables from all the possible angles. It helps you to predict the behaviour of the response variables depending on how the predictor variables move. The multivariate regression method helps you find a relationship between multiple variables or features.

Multivariate Regression is a supervised machine learning algorithm involving multiple data variables for analysis. Multivariate regression is an extension of multiple regression with one dependent variable and multiple independent variables. Based on the number of independent variables, we try to predict the output.

Multivariate Multiple Regression is the method of modeling multiple responses, or dependent variables, with a single set of predictor variables. For example, we might want to model both math and reading SAT scores as a function of gender, race, parent income, and so forth. This allows us to evaluate the relationship of, say, gender with each score. You may be thinking, "why not just run separate regressions for each dependent variable?" That's actually a good idea! And in fact that's pretty much what multivariate multiple regression does. It regresses each dependent variable separately on the predictors. However, because we have multiple responses, we have to modify our hypothesis tests for regression parameters and our confidence intervals for predictions.

Overfitting in Machine Learning

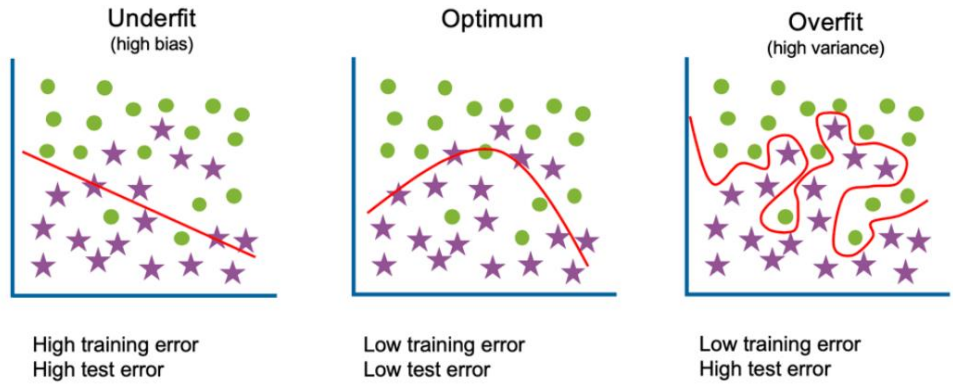
In the real world, the dataset present will never be clean and perfect. It means each dataset contains impurities, noisy data, outliers, missing data, or imbalanced data. Due to these impurities, different problems occur that affect the accuracy and the performance of the model. One of such problems is Overfitting in Machine Learning. Overfitting is a problem that a model can exhibit. A statistical model is said to be overfitted if it can't generalize well with unseen data.

Noise: Noise is meaningless or irrelevant data present in the dataset. It affects the performance of the model if it is not removed.

Bias: Bias is a prediction error that is introduced in the model due to oversimplifying the machine learning algorithms. Or it is the difference between the predicted values and the actual values.

Variance: If the machine learning model performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.

Generalization: It shows how well a model is trained to predict unseen data



Overfitting & underfitting are the two main errors/problems in the machine learning model, which cause poor performance in Machine Learning. Overfitting occurs when the model fits more data than required, and it tries to capture each and every datapoint fed to it. Hence it starts

capturing noise and inaccurate data from the dataset, which degrades the performance of the model. An overfitted model doesn't perform accurately with the test/unseen dataset and can't generalize well. An overfitted model is said to have low bias and high variance.

For example suppose there are three students, X, Y, and Z, and all three are preparing for an exam.

- X has studied only three sections of the book and left all other sections.
- Y has a good memory, hence memorized the whole book.
- And the third student, Z, has studied and practiced all the questions.

So, in the exam, X will only be able to solve the questions if the exam has questions related to section 3.

Student Y will only be able to solve questions if they appear exactly the same as given in the book.

Student Z will be able to solve all the exam questions in a proper way.

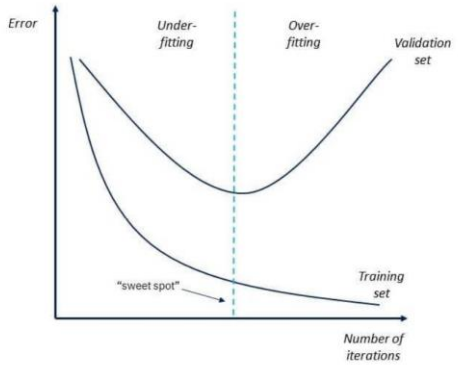
The same happens with machine learning;

- if the algorithm learns from a small part of the data, it is unable to capture the required data points and hence under fitted.
- Suppose the model learns the training dataset, like the Y student. They perform very well on the seen dataset but perform badly on unseen data or unknown instances. In such cases, the model is said to be Overfitting.
- And if the model performs well with the training dataset and also with the test/unseen dataset, similar to student Z, it is said to be a good fit.

How to detect Overfitting?

Overfitting in the model can only be detected once you test the data. To detect the issue, we can perform Train/test split. In the train-test split of the dataset, we can divide our dataset into random test and training datasets. We train the model with a training dataset which is about 80% of the total dataset. After training the model, we test it with the test dataset, which is 20 % of the total dataset.

Now, if the model performs well with the training dataset but not with the test dataset, then it is likely to have an overfitting issue. For example, if the model shows 85% accuracy with training data and 50% accuracy with the test dataset, it means the model is not performing well.



Ways to prevent the Overfitting

Although overfitting is an error in Machine learning which reduces the performance of the model, however, we can prevent it in several ways.

With the use of the linear model, we can avoid overfitting; however, many real-world problems are non-linear ones. Below are several ways that can be used to prevent overfitting:

1. Early Stopping:

- the training is paused before the model starts learning the noise within the model
- training the model iteratively, measure the performance of the model after each iteration.
- Continue up to a certain number of iterations until a new iteration improves the performance of the model.
- the model begins to overfit the training data; hence we need to stop the process before the learner passes that point.

2. Train with more data

Increasing the training set by including more data can enhance the accuracy of the model, as it provides more chances to discover the relationship between input and output variables

3. Feature Selection

we identify the most important features within training data, and other features are removed.

this process helps to simplify the model and reduces noise from the data.

4. Cross-Validation

divided the dataset into k-equal-sized subsets of data; these subsets are known as folds.

5. Data Augmentation

adding more data to prevent overfitting,

slightly modified copies of already existing data are added to the dataset.

Ensemble Methods

prediction from different machine learning models is combined to identify the most popular result.

The most commonly used ensemble methods are Bagging and Boosting.

In bagging, individual data points can be selected more than once. After the collection of several sample datasets, these models are trained independently, and depending on the type of

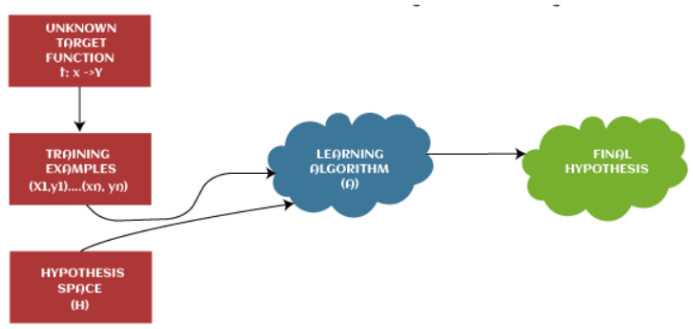
task-i.e., regression or classification-the average of those predictions is used to predict a more accurate result. Moreover, bagging reduces the chances of overfitting in complex models.

In boosting, a large number of weak learners arranged in a sequence are trained in such a way that each learner in the sequence learns from the mistakes of the learner before it. It combines all the weak learners to come out with one strong learner. In addition, it improves the predictive flexibility of simple models.

Hypothesis in Machine Learning

The hypothesis is defined as the supposition or proposed explanation based on insufficient evidence or assumptions. It is just a guess based on some known facts but has not yet been proven. A good hypothesis is testable, which results in either true or false.

Example: Let's understand the hypothesis with a common example. Some scientist claims that ultraviolet (UV) light can damage the eyes then it may also cause blindness. In this example, a scientist just claims that UV rays are harmful to the eyes, but we assume they may cause blindness. However, it may or may not be possible. Hence, these types of assumptions are called a hypothesis.



In supervised learning techniques, the main aim is to determine the possible hypothesis out of hypothesis space that best maps input to the corresponding or correct outputs.

There are some common methods given to find out the possible hypothesis from the Hypothesis space, where hypothesis space is represented by uppercase-h (H) and hypothesis by lowercase-h (h).

Hypothesis space (H):

Hypothesis space is defined as a set of all possible legal hypotheses; hence it is also known as a hypothesis set. It is used by supervised machine learning algorithms to determine the best possible hypothesis to describe the target function or best maps input to output. It is often constrained by choice of the framing of the problem, the choice of model, and the choice of model configuration

Hypothesis (h)

It is defined as the approximate function that best describes the target in supervised machine learning algorithms. It is primarily based on data as well as bias and restrictions applied to data.

Hence hypothesis (h) can be concluded as a single hypothesis that maps input to proper output and can be evaluated as well as used to make predictions.

The hypothesis (h) can be formulated in machine learning as follows: y= mx + b

Where,

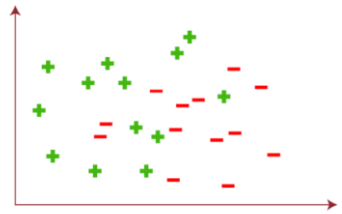
Y: Range

m: Slope of the line which divided test data or changes in y divided by change in x.

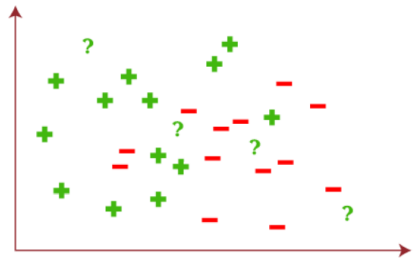
x: domain

c: intercept (constant)

Example



Now, assume we have some test data by which ML algorithms predict the outputs for input as follows

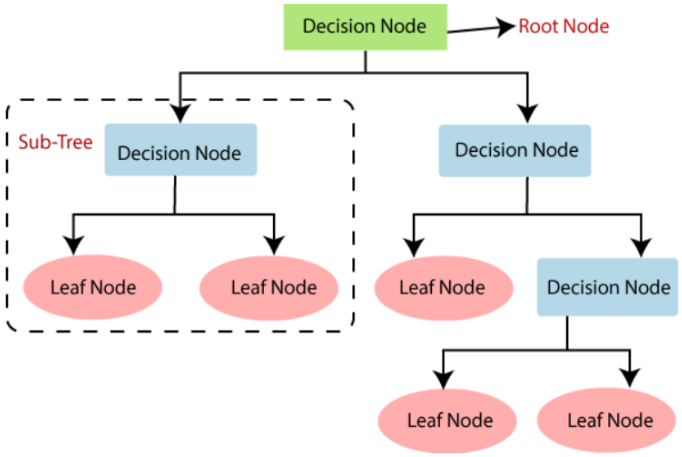


Decision Tree Classification Algorithm

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. o 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.



Why use Decision Trees?

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 treelike 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.

How does the Decision Tree algorithm Work?

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.

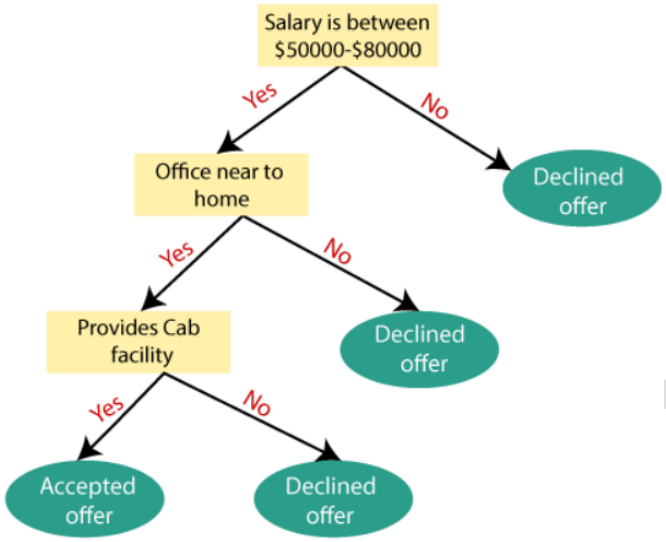
Step-4: Generate the decision tree node, which contains the best attribute.

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).



Selection Measures: Attribute selection measure or ASM

1. Information Gain
2. Gini Index

Information Gain

Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute. It calculates how much information a feature provides us about a class. According to the value of information gain, we split the node and build the decision tree. A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy

Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

$$\text{Entropy}(s) = -P(\text{yes})\log_2 P(\text{yes}) - P(\text{no})\log_2 P(\text{no})$$

Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

Gini Index

Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm. An attribute with the low Gini index should be preferred as compared to the high Gini index. It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.Gini index can be calculated using the below formula:

Gini Index= $1-\sum_j P_j^2$

Consider the following data set comprised of two binary input attributes (A1 and A2) and one binary output. **(8)**

Example	A_1	A_2	Output y
x_1	1	1	1
x_2	1	1	1
x_3	1	0	0
x_4	0	0	1
x_5	0	1	0
x_6	0	1	0

Use the DECISION-TREE-LEARNING algorithm to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

First find Entropy(S) whole dataset

Entropy(s)= -P(yes)$\log_2$ P(yes)- P(no) $\log_2$ P(no)

$$=-\frac{3}{6}log_2\frac{3}{6}-\frac{3}{6}log_2\frac{3}{6}=1$$

Gain(S, A1)= ES $-\{\frac{|S_{A1=1}|}{|S|}Entropy(S_{A=1})+\frac{|S_{A1=0}|}{|S|}Entropy(S_{A1=0})\}$

$S_{A1=1}$

X	A1	Y
X1	1	1
X2	1	1
X3	1	0

$S_{A1=0}$

X	A1	Y
X4	0	1
X5	0	0
X6	0	0

$|S_{A1=1}|=3$

$Entropy(S_{A1=1})=-P_1\log_2 P_1 - P_0 \log_2 P_0$

$$=-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3}=0.9183$$

$|S_{A1=0}|=3$

$$Entropy(S_{A1=0})=-\frac{1}{3}log_2\frac{1}{3}-\frac{2}{3}log_2\frac{2}{3}=0.9183$$

So $Gain(S,A_1)$ = ES $-\{\frac{|S_{A1=1}|}{|S|}\,Entropy(S_{A1=1})+\frac{|S_{A1=0}|}{|S|}\,Entropy(S_{A1=0})\}$

$$=1-\left\{\frac{3}{6}X0.9183+\frac{3}{6}X\,0.9183\right\}=0.0817\text{ -----(1)}$$

$S_{A2=1}$

X	A2	Y
X1	1	1
X2	1	1
X5	1	0
X6	1	0

$S_{A2=0}$

X	A2	Y
X3	0	0
X4	0	1

$|S_{A2=1}|=4$

$Entropy(S_{A2=1})=-P_1\log_2 P_1 - P_0 \log_2 P_0$

$$=-\frac{2}{4}log_2\frac{2}{4}-\frac{2}{4}log_2\frac{2}{4}=1$$

$|S_{A2=0}|=2$

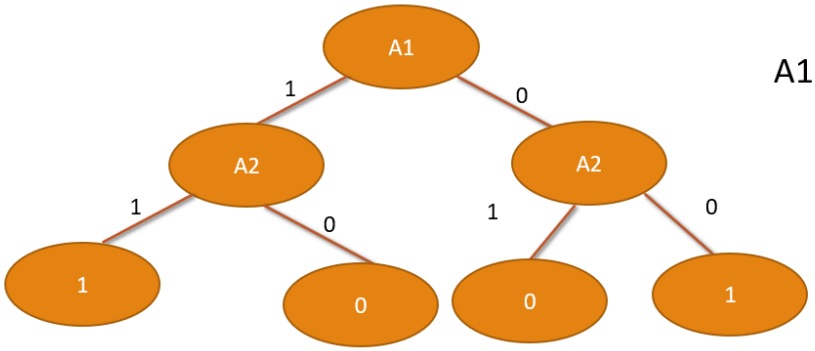
$$Entropy(S_{A1=0})=-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2}=1$$

So $Gain(S, A_2)$ = ES $-\{\frac{|S_{A2=1}|}{|S|} Entropy(S_{A2=1}) + \frac{|S_{A2=0}|}{|S|} Entropy(S_{A2=0})\}$

$= 1 - \left\{\frac{4}{6} X\ 1 + \frac{2}{6} X\ 1\right\} = 0$ ----------(2)

From (1) and (2) We found that A1 give more information gain.

So we make A1 as root node



A1=1

X	A2	Y
X1	1	1
X2	1	1
X3	0	0

A1=0

X	A2	Y
X4	0	1
X5	1	0
X6	1	0

Consider the following data set comprised of three binary input attributes (A_1, A_2 and A_3) and one binary output

Example	A_1	A_2	A_3	Output Y
x_1	1	0	0	0
x_2	1	0	1	0
x_3	0	1	0	0
x_4	1	1	1	1
x_5	1	1	0	1

Use the DECISION-TREE-LEARNING algorithm to learn a decision tree for these data. Show the computation made to determine the attribute to split each node.

$$\text{Entropy}(S) = E(S) = -P(1)\log_2 P(1) - P(0)\log_2 P(0)$$

$$= -\frac{2}{5}\log_2 2/5 - 3/5 \log_2 3/5$$

$$= -2/5 \times -1.322 - 3/5 \times -0.737$$

$$= 0.971$$

S_{A1}

$S_{A1}=1$

	A_1	y
x_1	1	0
x_2	1	0
x_4	1	1
x_5	1	1

$|S_{A1}=1| = 4$

$\text{Entropy}(S_{A1}=1) = -2/4 \log_2 2/4 - 2/4 \log_2 2/4$

$= 1$

$S_{A1}=0$

$|S_{A1}=0| = 1$

	A_1	y
x_3	0	0

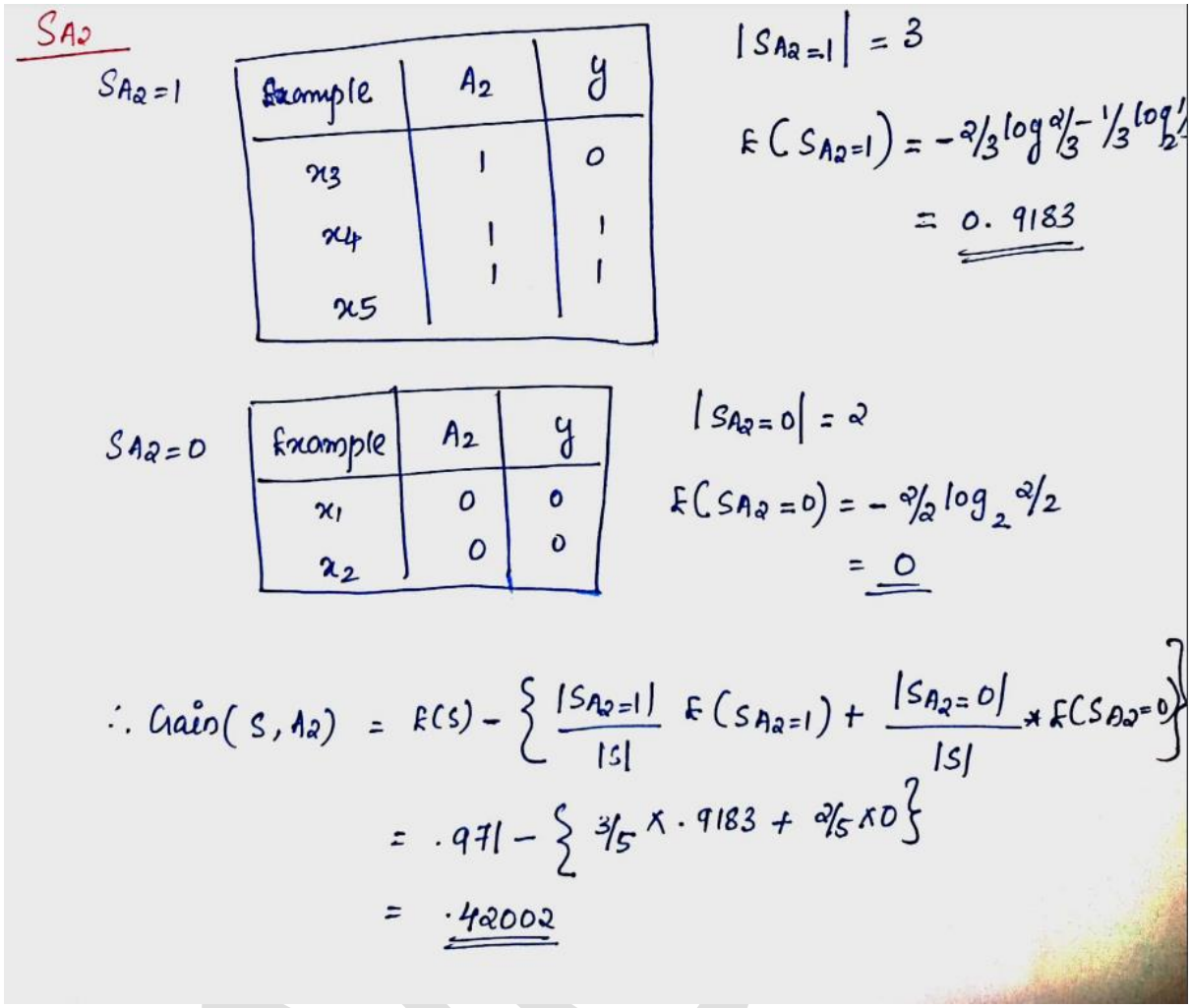
$\text{Entropy}(S_{A1}=0) = -\frac{1}{1}\log_2 1/1$

$= 0$

$$\therefore \text{Gain}(S, A_1) = E(S) - \left\{ \frac{|S_{A1}=1|}{|S|} \text{Entropy}(S_{A_1}=1) + \frac{|S_{A1}=0|}{|S|} * \text{Entropy}(S_{A1}=0) \right\}$$

$$= 0.971 - \{ 4/5 \times 1 + 1 \times 0 \}$$

$$= .171$$


S_{A_2}
$S_{A_2=1}$
Example | A_2 | y
x_3 | 1 | 0
x_4 | 1 | 1
x_5 | 1 | 1
$|S_{A_2=1}| = 3$
$E(S_{A_2=1}) = -2/3 \log 2/3 - 1/3 \log 1/$
$= 0.9183$
$S_{A_2=0}$
Example | A_2 | y
x_1 | 0 | 0
x_2 | 0 | 0
$|S_{A_2=0}| = 2$
$E(S_{A_2=0}) = -2/2 \log_2 2/2$
$= 0$
$\therefore Gain(S, A_2) = E(S) - \{ \frac{|S_{A_2=1}|}{|S|} E(S_{A_2=1}) + \frac{|S_{A_2=0}|}{|S|} * E(S_{A_2=0}) \}$
$= .971 - \{ 3/5 \times .9183 + 2/5 \times 0 \}$
$= .42002$

S_{A3}

$S_{A3} = 1$

Example	A_3	y
x_2	1	0
x_4	1	1

$|S_{A3}=1| = 2$

$E(S_{A3}=1) = -\frac{1}{2}\log_2 \frac{1}{2} - \frac{1}{2}\log_2 \frac{1}{2}$

$= 1$

$S_{A3} = 0$

Example	A_3	y
x_1	0	0
x_3	0	0
x_5	0	1

$|S_{A3}=0| = 3$

$E(S_{A3}=0) = -\frac{2}{3}\log_2 \frac{2}{3} - \frac{1}{3}\log_2 \frac{1}{3}$

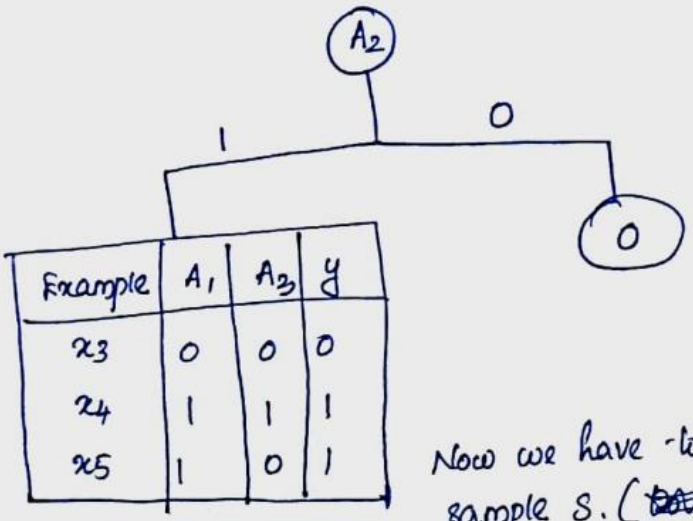
$= .9183$

$\therefore Gain(S, A_3) = E(S) - \left\{ \frac{|S_{A3}=1|}{|S|} E(S_{A3}=1) + \frac{|S_{A3}=0|}{|S|} \times E(S_{A3}=0) \right\}$

$= .971 - \left\{ 2/5 \times 1 + 3/5 \times .9183 \right\}$

$= .0.02002$

Attribute with maximum gain is A_2 so A_2 is the root node



Example	A_1	A_2	y
x3	0	0	0
x4	1	1	1
x5	1	0	1

Now we have to take this as new sample S. (~~bottom up~~ above Follow the procedure)

Entropy $(S_{new}) = -2/3 \log_2 2/3 - 1/3 \log_2 1/3$

$= 0.9183$

S_{new} A_1

$S_{new} A_1 = 1$

Example	A_1	y
x4	1	1
x5	1	1

$|S_{new} A_1 = 1| = 2$

Entropy $(S_{new} A_1 = 1) = -2/2 \log_2 2/2$

$= 0$

S_{new} $A_1 = 0$

Example	A_1	y
x3	0	0

$|S_{new} A_1 = 0| = 1$

Entropy $(S_{new} A_1 = 0) = 1/1 \log_2 1/1$

$= 0$

$\therefore$ Gain $(S_{new}, S_{new} A_1) = f(S_{new}) - 0$

$= .9183$

S_{new} A_3

S_{new} $A_3 = 1$

Example	A_3	Y
x_4	1	1

$E(S_{new} A_3 = 1) = -1/1 \log_2 1/1 = 0$

S_{new} $A_3 = 0$

Example	A_3	Y
x_3	0	0
x_5	0	1

$E(S_{new} A_3 = 0)$

$= -1/2 \log_2 1/2 - 1/2 \log_2 1/2$

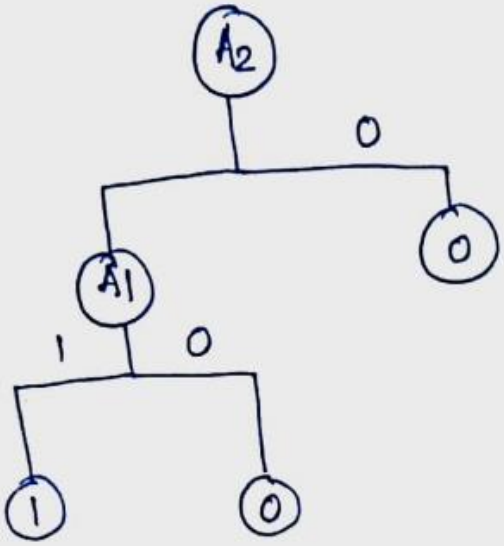
$= 1$

Gain $(S_{new}, A_3) = .9183 - \{ 1/3 \times 0 + 2/3 \times 1 \}$

$= .2516$

So in S_{new} A_1 is the attribute with more gain

Decision tree



Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree. A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning.

There are mainly two types of tree pruning technology used:

- Cost Complexity Pruning
- Reduced Error Pruning.