Unit I
INTRODUCTION TO MACHINE LEARNING

1.1 What is Machine Learning?

Machine learning (ML) is a discipline of artificial intelligence (AI) that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention.

Machine learning methods enable computers to operate autonomously without explicit programming. ML applications are fed with new data, and they can independently learn, grow, develop, and adapt. Machine learning derives insightful information from large volumes of data by leveraging algorithms to identify patterns and learn in an iterative process.

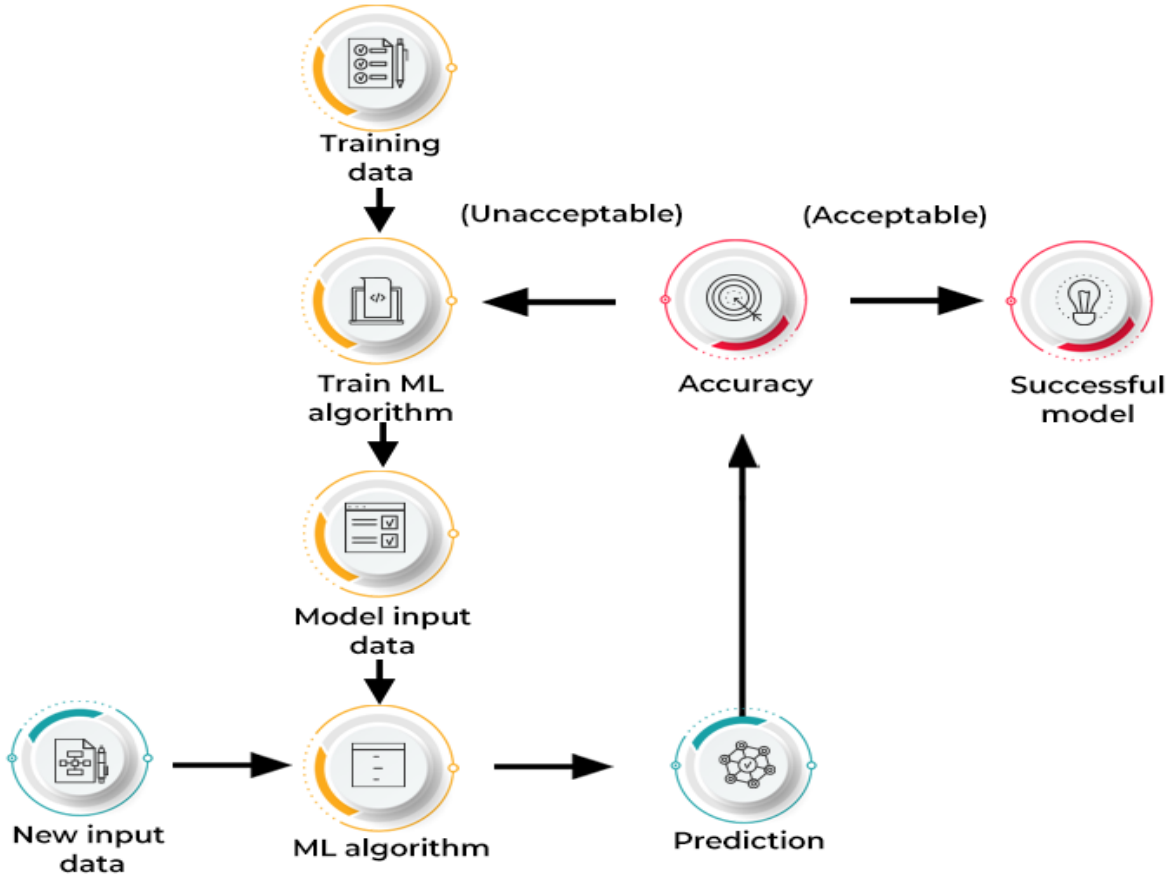
ML algorithms use computation methods to learn directly from data instead of relying on any predetermined equation that may serve as a model. The performance of ML algorithms adaptively improves with an increase in the number of available samples during the 'learning' processes.

While machine learning is not a new concept – dating back to World War II when the Enigma Machine was used – the ability to apply complex mathematical calculations automatically to growing volumes and varieties of available data is a relatively recent development. Today, with the rise of big data, IoT, and ubiquitous computing, machine learning has become essential for solving problems across numerous areas, such as

1. Computational finance (credit scoring, algorithmic trading)
2. Computer vision (facial recognition, motion tracking, object detection)
3. Computational biology (DNA sequencing, brain tumor detection, drug discovery)
4. Automotive, aerospace, and manufacturing (predictive maintenance)
5. Natural language processing (voice recognition)

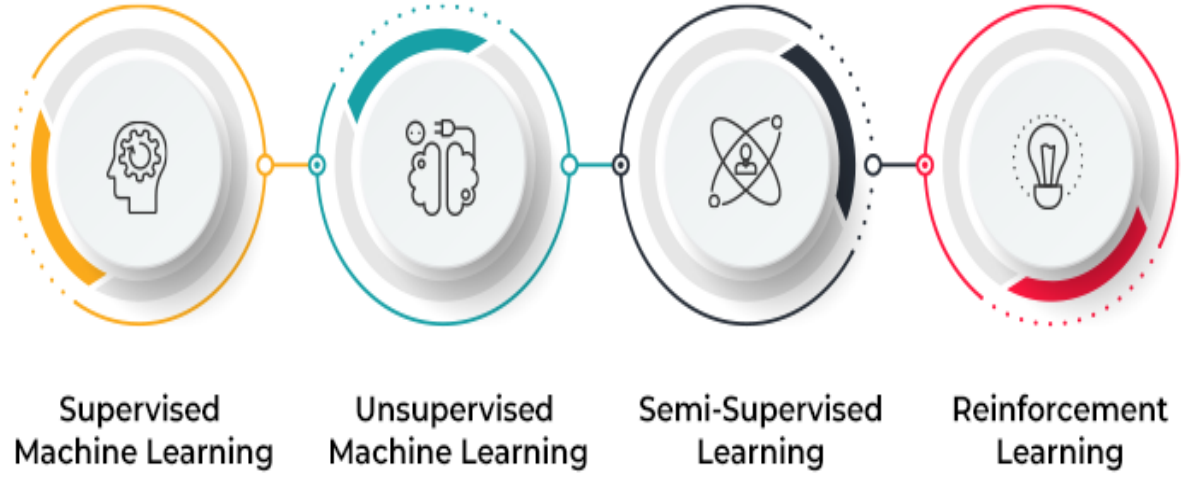
1.2 How does machine learning work?

Machine learning algorithms are moulded on a training dataset to create a model. As new input data is introduced to the trained ML algorithm, it uses the developed model to make a prediction.



1.3 Types of Machine Learning

Machine learning algorithms can be trained in many ways, with each method having its pros and cons. Based on these methods and ways of learning, machine learning is broadly categorized into four main types:



A. Supervised machine learning: This type of ML involves supervision, where machines are trained on labelled datasets and enabled to predict outputs based on the provided training. The labelled dataset specifies that some input and output parameters are already mapped. Hence, the machine is trained with the input and corresponding output. A device is made to predict the outcome using the test dataset in subsequent phases. For example, consider an input dataset of parrot and crow images. Initially, the machine is trained to understand the pictures, including the parrot and crow's colour, eyes, shape, and size. Post-training, an input picture of a parrot is provided, and the machine is expected to identify the object and predict the output. The trained machine checks for the various features of the object, such as colour, eyes, shape, etc., in the input picture, to make a final prediction. This is the process of object identification in supervised machine learning. The primary objective of the supervised learning technique is to map the input variable

(a) with the output variable

(b) Supervised machine learning is further classified into two broad categories:

1. Classification: These refer to algorithms that address classification problems where the output variable is categorical; for example, yes or no, true, or false, male, or female, etc. Real-world applications of this category are evident in spam detection and email filtering.

Some known classification algorithms include the Random Forest Algorithm, Decision Tree Algorithm, Logistic Regression Algorithm, and Support Vector Machine Algorithm.

2. Regression: Regression algorithms handle regression problems where input and output variables have a linear relationship. These are known to predict continuous output variables. Examples include weather prediction, market trend analysis, etc.

Popular regression algorithms include the Simple Linear Regression Algorithm, Multivariate Regression Algorithm, Decision Tree Algorithm, and Lasso Regression.

B. Unsupervised machine learning: Unsupervised learning refers to a learning technique that's devoid of supervision. Here, the machine is trained using an unlabelled dataset and is enabled to predict the output without any supervision. An unsupervised learning algorithm aims to group the unsorted dataset based on the input's similarities, differences, and patterns. For example, consider an input dataset of images of a fruit-filled container. Here, the images are not known to the machine learning model. When we input the dataset into the ML model, the task of the model is to identify the pattern of objects,

such as colour, shape, or differences seen in the input images and categorize them. Upon categorization, the machine then predicts the output as it gets tested with a test dataset. Unsupervised machine learning is further classified into two types:

1. Clustering: The clustering technique refers to grouping objects into clusters based on parameters such as similarities or differences between objects. For example, grouping customers by the products they purchase.

Some known clustering algorithms include the K-Means Clustering Algorithm, Mean-Shift Algorithm, DBSCAN Algorithm, Principal Component Analysis, and Independent Component Analysis.

2. Association: Association learning refers to identifying typical relations between the variables of a large dataset. It determines the dependency of various data items and maps associated variables. Typical applications include web usage task and market data analysis. Popular algorithms obeying association rules include the Apriori Algorithm, Eclat Algorithm, and FP-Growth Algorithm.

C. Semi-supervised learning: Semi-supervised learning comprises characteristics of both supervised and unsupervised machine learning. It uses the combination of labelled and unlabelled datasets to train its algorithms. Using both types of datasets, semi-supervised learning overcomes the drawbacks of the options mentioned above. Consider an example of a college student. A student learning a concept under a teacher's supervision in college is termed supervised learning. In unsupervised learning, a student self-learns the same concept at home without a teacher's guidance. Meanwhile, a student revising the concept after learning under the direction of a teacher in college is a semi-supervised form of learning.

D. Reinforcement learning: Reinforcement learning is a feedback-based process. Here, the AI component automatically takes stock of its surroundings by the hit & trial method, acts, learns from experiences, and improves performance. The component is rewarded for each good action and penalized for every wrong move. Thus, the reinforcement learning component aims to maximize the rewards by performing good actions. Unlike supervised learning, reinforcement learning lacks labelled data, and the agents learn via experiences only. Consider video games. Here, the game specifies the environment, and each move of the reinforcement agent defines its state. The agent is entitled to receive feedback via punishment and rewards, thereby affecting the overall game score. The goal of the agent is to achieve a high score. Reinforcement learning is applied across different

fields such as game theory, information theory, and multi-agent systems. Reinforcement learning is further divided into two types of methods or algorithms:

1. Positive reinforcement learning: This refers to adding a reinforcing stimulus after a specific behaviour of the agent, which makes it more likely that the behaviour may occur again in the future, e.g., adding a reward after a behaviour.

2. Negative reinforcement learning: Negative reinforcement learning refers to strengthening a specific behaviour that avoids a negative outcome.

1.4 Machine Learning Algorithms:

A. Linear regression: Linear regression gives a relationship between input (x) and an output variable (y), also referred to as independent and dependent variables. Let's understand the algorithm with an example where you are required to arrange a few plastic boxes of different sizes on separate shelves based on their corresponding weights. The task is to be completed without manually weighing the boxes. Instead, you need to guess the weight just by observing the boxes' height, dimensions, and sizes. In short, the entire task is driven based on visual analysis. Thus, you must use a combination of visible variables to make the final arrangement on the shelves. Linear regression in machine learning is of a similar kind, where the relationship between independent and dependent variables is established by fitting them to a regression line. This line has a mathematical representation given by the linear equation

y = mx + b, where y represents the dependent variable, m = slope, x = independent variable, and b = intercept.

The objective of linear regression is to find the best fit line that reveals the relationship between variables y and x.

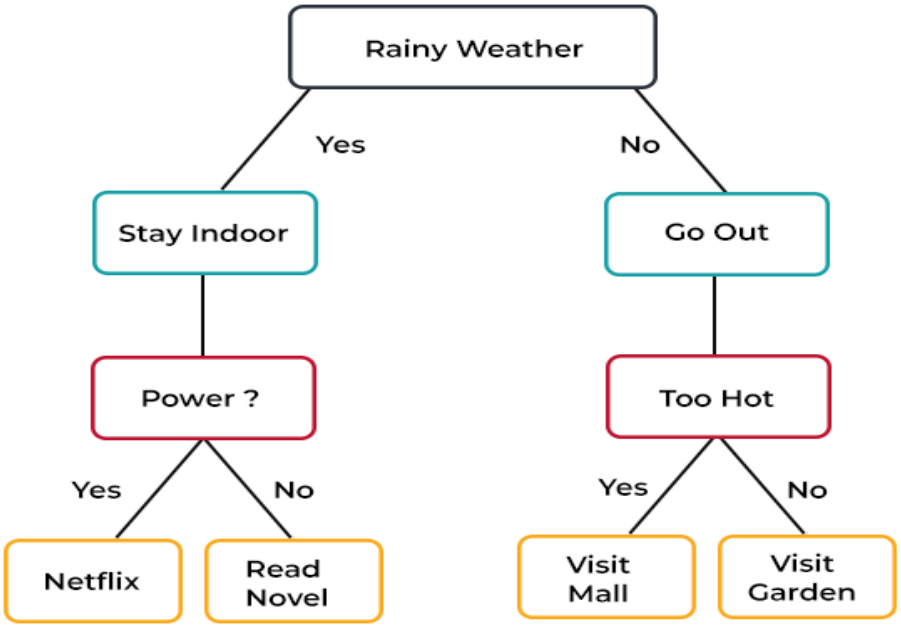
B. Logistic regression: The dependent variable is of binary type in logistic regression. This type of regression analysis describes data and explains the relationship between one dichotomous variable and one or more independent variables. Logistic regression is used in predictive analysis where pertinent data predict an event probability to a logit function. Thus, it is also called logit regression. Mathematically, logistic regression is represented by the equation:

y = e^ (b0 + b1*x) / (1 + e^ (b0 + b1*x))

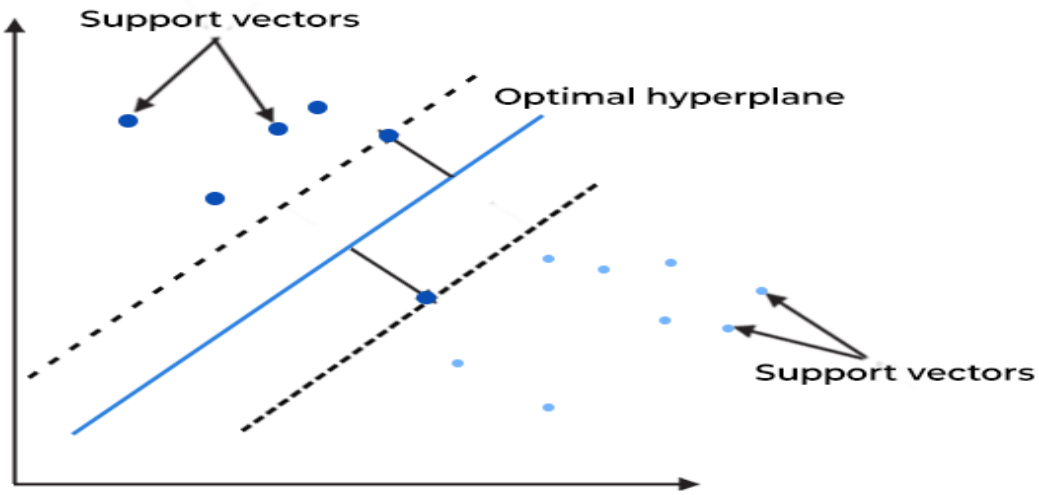
Here, x = input value, y = predicted output, b0 = bias or intercept term, b1 = coefficient for input (x).

Logistic regression could be used to predict whether a particular team will win (1) the FIFA World Cup 2022 or not (0), or whether a lockdown will be imposed (1) due to rising COVID-19 cases or not (0). Thus, the binary outcomes of logistic regression facilitate faster decision-making as you only need to pick one out of the two alternatives.

C. Decision trees: With a decision tree, you can visualize the map of potential results for a series of decisions. It enables companies to compare possible outcomes and then take a straightforward decision based on parameters such as advantages and probabilities that are beneficial to them. Decision tree algorithms can potentially anticipate the best option based on a mathematical construct and come in handy while brainstorming over a specific decision. The tree starts with a root node (decision node) and then branches into sub-nodes representing potential outcomes. Each outcome can further create child nodes that can open other possibilities. The algorithm generates a tree-like structure that is used for classification problems. For example, consider the decision tree below that helps finalize a weekend plan based on the weather forecast.



D. Support vector machines: (SVMs) Support vector machine algorithms are used to accomplish both classification and regression tasks. These are supervised machine learning algorithms that plot each piece of data in the n-dimensional space, with n referring to the number of features. Each feature value is associated with a coordinate value, making it easier to plot the features. Moreover, classification is further performed by distinctly deter task the hyper-plane that separates the two sets of support vectors or classes. A good separation ensures a good classification between the plotted data points.



In simple words, SVMs represent the coordinates for individual observations. These are popular machine learning classifiers used in applications such as data classification, facial expression classification, text classification, steganography detection in digital images, speech recognition, and others.

E. Naive Bayes algorithm: Naive Bayes refers to a probabilistic machine learning algorithm based on the Bayesian probability model and is used to address classification problems. The fundamental assumption of the algorithm is that features under consideration are independent of each other and a change in the value of one does not impact the value of the other.

For example, you can consider a ball, a cricket ball, if it is red, round, has a 7.1-7.26 cm diameter, and has a mass of 156-163 g. Although all these features could be interdependent, each one contributes to the probability that it is a cricket ball. This is the reason the algorithm is referred to as 'naïve'. Let's look at the mathematical representation of the algorithm.

If X, Y = probabilistic events, P (X) = probability of X being true, P(X|Y) = conditional probability of X being true in case Y is true. Then, Bayes' theorem is given by the equation: P (X|Y) = (P (Y|X) x P (X)) /P (Y)

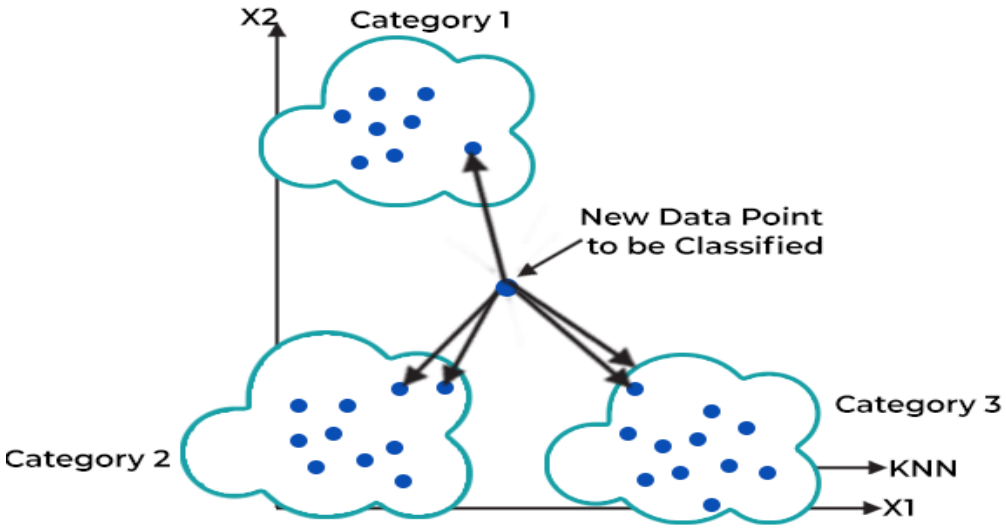
A naive Bayesian approach is easy to develop and implement. It is capable of handling massive datasets and is useful for making real-time predictions. Its applications include spam filtering, sentiment analysis and prediction, document classification, and others.

F. KNN classification algorithm: The K Nearest Neighbours (KNN) algorithm is used for both classification and regression problems. It stores all the known use cases and classifies new use cases (or data points) by segregating them into different classes. This

classification is accomplished based on the similarity score of the recent use cases to the available ones. KNN is a supervised machine learning algorithm, wherein 'K' refers to the number of neighbouring points we consider while classifying and segregating the known n groups. The algorithm learns at each step and iteration, thereby eliminating the need for any specific learning phase. The classification is based on the neighbour's majority vote.

The algorithm uses these steps to perform the classification:

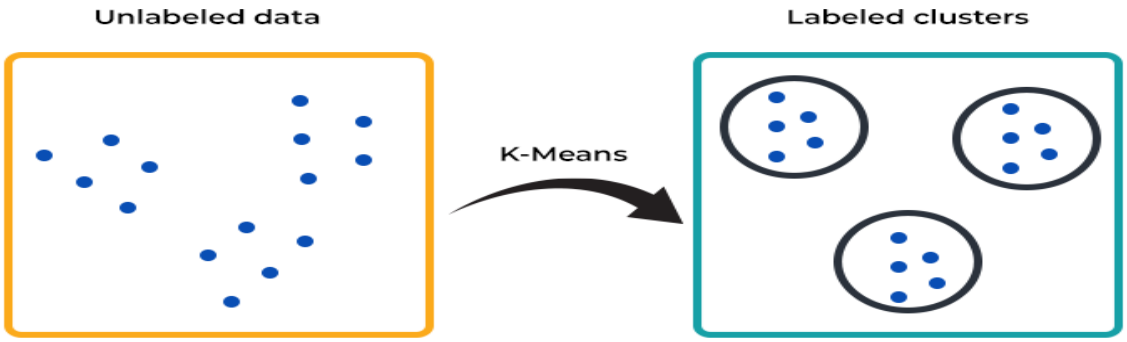
- For a training dataset, calculate the distance between the data points that are to be classified and the rest of the data points.
- Choose the closest 'K' elements based on the distance or function used.
- Consider a 'majority vote' between the K points–the class or label dominating all data points reveals the final ranking.



G. K-Means: K-Means is a distance-based unsupervised machine learning algorithm that accomplishes clustering tasks. In this algorithm, you classify datasets into clusters (K clusters) where the data points within one set remain homogenous, and the data points from two different clusters remain heterogeneous. The clusters under K-Means are formed using these steps:

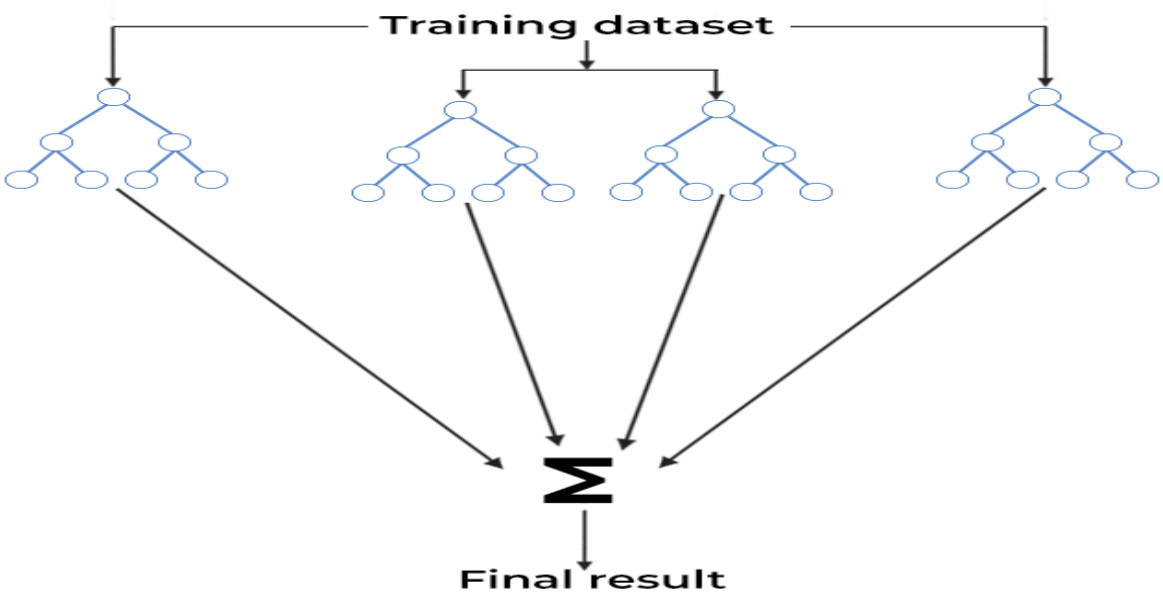
- Initialization: The K-means algorithm selects centroids for each cluster ('K' number of points).
- Assign objects to centroid: Clusters are formed with the closest centroids (K clusters) at each data point.

- Centroid update: Create new centroids based on existing clusters and determine the closest distance for each data point based on new centroids. Here, the position of the centroid also gets updated whenever required.
- Repeat: Repeat the process till the centroids do not change.



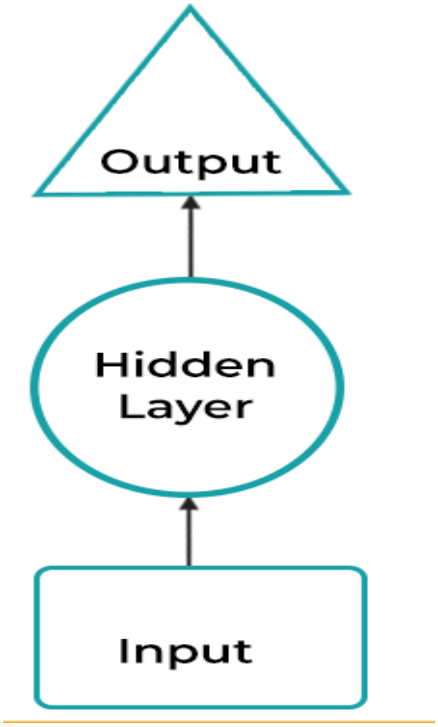
H. Random Forest algorithm: Random Forest algorithms use multiple decision trees to handle classification and regression problems. It is a supervised machine learning algorithm where different decision trees are built on different samples during training. These algorithms help estimate missing data and tend to keep the accuracy intact in situations when a large chunk of data is missing in the dataset. Random forest algorithms follow these steps:

- Select random data samples from a given data set.
- Build a decision tree for each data sample and provide the prediction result for each decision tree.
- Carry out voting for each expected result.
- Select the final prediction result based on the highest voted prediction result.



I. **Artificial neural networks (ANNs):** Artificial neural networks are machine learning algorithms that mimic the human brain (neuronal behaviour and connections) to solve complex problems. ANN has three or more interconnected layers in its computational model that process the input data. The first layer is the input layer or neurons that send input data to deeper layers. The second layer is called the hidden layer. The components of this layer change or tweak the information received through various previous layers by performing a series of data transformations. These are also called neural layers. The third layer is the output layer that sends the final output data for the problem. ANN algorithms find applications in smart home and home automation devices such as door locks, thermostats, smart speakers, lights, and appliances. They are also used in the field of computational vision, specifically in detection systems and autonomous vehicles.

J. Recurrent neural networks (RNNs): Recurrent neural networks refer to a specific type of ANN that processes sequential data. Here, the result of the previous step acts as the input to the current step. This is facilitated via the hidden state that remembers information about a sequence. It acts as a memory that maintains the information on what was previously calculated. The memory of RNN reduces the overall complexity of the neural network.



RNN analyses time series data and possesses the ability to store, learn, and maintain contexts of any length. RNN is used in cases where time sequence is of paramount importance, such as speech recognition, language translation, video frame processing, text generation, and image captioning. Even Siri, Google Assistant, and Google Translate use the RNN architecture.

1.5 Real life applications:

1. Healthcare industry: Machine learning is being increasingly adopted in the healthcare industry, credit to wearable devices and sensors such as wearable fitness trackers, smart health watches, etc. All such devices monitor users' health data to assess their health in real-time. Moreover, the technology is helping medical practitioners in analysing trends or flagging events that may help in improved patient diagnoses and treatment. ML algorithms even allow medical experts to predict the lifespan of a patient suffering from a fatal disease with increasing accuracy. Additionally, machine learning is contributing significantly to two areas:

- **Drug discovery:** Manufacturing or discovering a new drug is expensive and involves a lengthy process. Machine learning helps speed up the steps involved in such a multi-step process. For example, Pfizer uses IBM's Watson to analyse massive volumes of disparate data for drug discovery.
- **Personalized treatment:** Drug manufacturers face the stiff challenge of validating the effectiveness of a specific drug on a large mass of the population. This is because the drug works only on a small group in clinical trials and possibly causes side effects on some subjects.

To address these issues, companies like Genentech have collaborated with GNS Healthcare to leverage machine learning and simulation AI platforms, innovating biomedical treatments to address these issues. ML technology looks for patients' response markers by analysing individual genes, which provides targeted therapies to patients.

2. Finance sector: Today, several financial organizations and banks use machine learning technology to tackle fraudulent activities and draw essential insights from vast volumes of data. ML-derived insights aid in identifying investment opportunities that allow investors to decide when to trade. Moreover, data task methods help cyber-surveillance systems zero in on warning signs of fraudulent activities, subsequently neutralizing them. Several financial institutes have already partnered with tech companies to leverage the benefits of machine learning. For example,

- Citibank has partnered with fraud detection company Feedzai to handle online and in-person banking frauds.
- PayPal uses several machine learning tools to differentiate between legitimate and fraudulent transactions between buyers and sellers.

3. Retail sector: Retail websites extensively use machine learning to recommend items based on users' purchase history. Retailers use ML techniques to capture data, analyse it, and deliver personalized shopping experiences to their customers. They also implement ML for marketing campaigns, customer insights, customer merchandise planning, and price optimization. According to a September 2021 report by Grand View Research, Inc., the global recommendation engine market is expected to reach a valuation of $17.30 billion by 2028. Common day-to-day examples of recommendation systems include:

- When you browse items on Amazon, the product recommendations that you see on the homepage result from machine learning algorithms. Amazon uses artificial neural networks (ANN) to offer intelligent, personalized recommendations relevant to customers based on their recent purchase history, comments, bookmarks, and other online activities.
- Netflix and YouTube rely heavily on recommendation systems to suggest shows and videos to their users based on their viewing history.

Moreover, retail sites are also powered with virtual assistants or conversational chatbots that leverage ML, natural language processing (NLP), and natural language understanding (NLU) to automate customer shopping experiences.

4. Travel industry: Machine learning is playing a pivotal role in expanding the scope of the travel industry. Rides offered by Uber, Ola, and even self-driving cars have a robust machine learning backend. Consider Uber's machine learning algorithm that handles the dynamic pricing of their rides. Uber uses a machine learning model called 'Geosurge' to manage dynamic pricing parameters. It uses real-time predictive modelling on traffic patterns, supply, and demand. If you are getting late for a meeting and need to book an Uber in a crowded area, the dynamic pricing model kicks in, and you can get an Uber ride immediately but would need to pay twice the regular fare. Moreover, the travel industry uses machine learning to analyse user reviews. User comments are classified through sentiment analysis based on positive or negative scores. This is used for campaign monitoring, brand monitoring, compliance monitoring, etc., by companies in the travel industry.

5. social media: With machine learning, billions of users can efficiently engage on social media networks. Machine learning is pivotal in driving social media platforms from personalizing news feeds to delivering user-specific ads. For example, Facebook's auto-tagging feature employs image recognition to identify your friend's face and tag them

automatically. The social network uses ANN to recognize familiar faces in users' contact lists and facilitates automated tagging. Similarly, LinkedIn knows when you should apply for your next role, whom you need to connect with, and how your skills rank compared to peers. All these features are enabled by machine learning.

1.6 Learning Tasks

1. Descriptive Data Task:

This term is basically used to produce correlation, cross-tabulation, frequency etc. These technologies are used to determine the similarities in the data and to find existing patterns. One more application of descriptive analysis is to develop the captivating subgroups in a major part of the data available.

This analytics emphasis on the summarization and transformation of the data into meaningful information for reporting and monitoring.

2. Predictive Data Task:

The main goal of this task is to say something about future results not of current behaviour. It uses the supervised learning functions which are used to predict the target value. The methods come under this type of task category are called classification, time-series analysis, and regression. Modelling of data is the necessity of the predictive analysis, and it works by utilizing a few variables of the present to predict the future not known data values for other variables.

S. No	Comparison	Descriptive Data Task	Predictive Data Task
1	Basic	It determines, what happened in the past by analysing stored data.	It determines, what can happen in the future with the help past data analysis.
2	Preciseness	It provides accurate data.	It produces results does not ensure accuracy.
3	Practical analysis methods	Standard reporting, query/drill down and ad-hoc reporting.	Predictive modelling, forecasting, simulation, and alerts.
4	Require	It requires data aggregation and data mining	It requires statistics and forecasting methods
5	Type of approach	Reactive approach	Proactive approach
6	Describe	Describes the characteristics of the data in a target data set.	Carry out the induction over the current and past data so that predictions can be made.

1.7 Learning Paradigms:

1. Supervised Learning: Supervised learning is a form of machine learning which effectively works as concept mapping. You have an input, and you get an output. As you feed in data and assess it, you get a function which tries to abstract the system to have rules that probably make no sense to an actual person.

An image contains a car, and your expected output is a yes, this image doesn't contain a car, so you get a no. As the system learns, it can rule out non-cars with increasing accuracy. When you teach certain concepts, you can't directly explain them and expect your audience to understand. You describe what makes that concept work but also give examples. There is a level of inference that if you have this input, you get this specific output, and they draw their own conclusions about what connects them. If you're teaching a kid what a dog is, you don't explain the legs, the eyes, the ears, etc., you point and say: "That's a dog," or: "That's not a dog," until they get it. You're providing a task to be done and examples of the right and wrong answer.

For our car example, this would mean we've codified images as containing a car or not, and the system abstracts the pattern. The main limitation is that it relies entirely on the training data. If all your pictures are red cars in the woods for the car set, a tomato with green around it might be determined to be a car because there is a red shape and green around it, or a blue car may register as a false negative. It found a correlation that fits a pattern without fitting the pattern we wanted.

2. Unsupervised Learning: Unsupervised learning is the opposite of supervised learning in the sense that instead of assessing the accuracy of the training, you're assessing the results of the process. You determine a set and supervise the process in getting the results with supervised learning, but you feed a set and let the system classify it to determine if a novel input is a member of a derived set or not. This type of self-organization can lead to some interesting results. You let the system sort things out in a way that makes the clusters make sense based on the requirements.

For instance, if someone handed you a bunch of vegetables and asked you to sort them, and you couldn't tell what or the purpose of the sorting was, you might go off of colour. Potatoes, carrots, etc. all have different colours, but are the same thing, though your sorting method might disagree. You're providing data and letting the system make sense of it based on some vague initial premise implied in the data. This approach is useful in something like a medical application.

If you pass a bunch of similar cell slides to an algorithm, it might find something which then correlates to cancer which can be explored further. While with supervised learning we gave the system our desired categories for output, here we give the system our inputs and let it make sense of it. You might find something that you didn't even know to search for as an output with this sort of system.

3. Reinforcement Learning: Reinforcement learning is where the process of learning itself is fed back into. As your algorithm explores its environment, it learns more and tries to accomplish some specific goal(s) it is rewarded for. The better it does at approaching the goal, the more it is rewarded. Your algorithm reacts to the environment it is in to grow and adapt to fit the rules of the system.

A real-life example of this is teaching a dog to walk on a leash. You don't want the dog to pull or fall behind, but the dog really doesn't understand the rules. When the dog does what it's supposed to, you give them a reward and walk. When they don't, they don't get to continue walking or similar which shows them they've violated the rules (ideally you don't unreasonably punish the dog since they don't understand, but it's fine to do to an algorithm).

The dog learns the rules by trial and error and eventually knows instinctively what they can and cannot do when you have their leash. Your algorithm is adapting to the changes as they come in rather than based on a static input set. This makes less sense for anything which doesn't involve a continual process (a robot trying to walk and fighting gravity) or a dynamic environment (a video game algorithm where there aren't necessarily fixed states). This learning system is based on trial and error rather than a fixed set of data.

1.8 :

1.9 Dimensionality reduction techniques:

In Machine Learning and Statistic, Dimensionality Reduction the process of reducing the number of random variables under consideration via obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

two main algorithms in Dimensionality Reduction

1. Principle Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)

1. Principle Component Analysis (PCA): In projection methods, we are interested in finding a mapping from the inputs in the original d-dimensional space to a new (k < d)-

dimensional space, with minimum loss of information. The projection of x on the direction of w is $z = w^T x$. Principal component analysis (PCA) is an unsupervised method in that it does not use the output information; the criterion to be maximized is the variance. The principal component is w_1 such that the sample, after projection on to W_1, is most spread out so that the difference between the sample points becomes most apparent. For a unique solution and to make the direction the important factor, we require $||w_1|| = 1$.

If $z_1 = w_1^T x$ with $Cov(x) = \Sigma$, then $Var(z_1) = w_1^T \Sigma w_1$.

We seek w_1 such that $Var(z_1)$ is maximized subject to the constraint that $w_1^T w_1 = 1$. Writing this as a Lagrange problem, we have

$$\max_{\boldsymbol{w}_1} \boldsymbol{w}_1^T \boldsymbol{\Sigma} \boldsymbol{w}_1 - \alpha(\boldsymbol{w}_1^T \boldsymbol{w}_1 - 1)$$

Taking the derivative with respect to w1 and setting it equal to 0, we have

$2\Sigma w_1 - 2\alpha w_1 = 0$, and therefore $\Sigma w_1 = \alpha w_1$

which holds if w_1 is an eigenvector of Σ and α the corresponding eigenvalue. Because we want to maximize

$w_1^T \Sigma w_1 = \alpha w_1^T w_1 = \alpha$

we choose the eigenvector with the largest eigenvalue for the variance to be maximum. Therefore, the principal component is the eigenvector of the covariance matrix of the input sample with the largest eigenvalue, $\lambda_1 = \alpha$.

The second principal component, w_2, should also maximize variance, be of unit length, and be orthogonal to w_1. This latter requirement is so that after projection $z_2 = w_2^T x$ is uncorrelated with z_1. For the second principal component, we have

$$\max_{\boldsymbol{w}_2} \boldsymbol{w}_2^T \boldsymbol{\Sigma} \boldsymbol{w}_2 - \alpha(\boldsymbol{w}_2^T \boldsymbol{w}_2 - 1) - \beta(\boldsymbol{w}_2^T \boldsymbol{w}_1 - 0)$$

Taking the derivative with respect to w_2 and setting it equal to 0, we have

$2\Sigma w_2 - 2\alpha w_2 - \beta w_1 = 0$

Premultiply by w_1^T and we get

$2 w_1^T \Sigma w_2 - 2\alpha w_1^T w_2 - \beta w_1^T w_1 = 0$

Note that $w_1^T w_2 = 0$. $w_1^T \Sigma w_2$ is a scalar, equal to its transpose $w_2^T \Sigma w_1$ where, because w1 is the leading eigenvector of Σ, $\Sigma w_1 = \lambda_1 w_1$. Therefore

$w_1^T \Sigma w_2 = w_2^T \Sigma w1 = \lambda_1 w_2^T w_1 = 0$

Then $\beta = 0$ and $\Sigma w_2 = \alpha w_2$

which implies that w_2 should be the eigenvector of Σ with the second largest eigenvalue, $\lambda_2 = \alpha$. Similarly, we can show that the other dimensions are given by the eigenvectors with decreasing eigenvalues.

Because Σ is symmetric, for two different eigenvalues, the eigenvectors are orthogonal. If Σ is positive definite ($x^T \Sigma x > 0$, for all non-null x), then all its eigenvalues are positive. If Σ is singular, then its rank, the effective dimensionality, is k with $k < d$ and λ_i, $i = k + 1, \ldots, d$ are 0 (λ_i are sorted in descending order). The k eigenvectors with nonzero eigenvalues are the dimensions of the reduced space. The first eigenvector (the one with the largest eigenvalue), w_1, namely, the principal component, explains the largest part of the variance; the second explains the second largest; and so on.

We define $z = w^T (x - m)$, where the k columns of w are the k leading eigenvectors of S, the estimator to Σ. We subtract the sample mean m from x before projection to center the data on the origin. After this linear transformation, we get to a k-dimensional space whose dimensions are the eigenvectors, and the variances over these new dimensions are equal to the eigenvalues. To normalize variances, we can divide by the square roots of the eigenvalues.

Let us see another derivation: We want to find a matrix W such that when we have $z=W^Tx$ (assume without loss of generality that x is already centered), we will get $Cov(z) = D$ where D is any diagonal matrix; that is, we would like to get uncorrelated z_i. If we form a $(d \times d)$ matrix C whose i^{th} column is the normalized eigenvector c_i of S, then $C^TC = I$ and

$$
\begin{aligned}
S &= SCC^T \\
&= S(c_1, c_2, \ldots, c_d)\, C^T \\
&= (Sc_1, Sc_2, \ldots, Sc_d)\, C^T \\
&= (\lambda_1 c_1, \lambda_2 c_2, \ldots, \lambda_d c_d) C^T \\
&= \lambda_1 c_1 c_1^T + \cdots + \lambda_d c_d c^T \\
&= CDC^T
\end{aligned}
$$

where D is a diagonal matrix whose diagonal elements are the eigenvalues, $\lambda_1, \ldots, \lambda_d$. This is called the spectral decomposition of S. Since C is orthogonal and $CC^T = C^TC = I$, we can multiply on the left by C^T and on the right by C to obtain $C^T SC = D$.

We know that if $z = W^T x$, then $Cov(z) = W^T SW$, which we would like to be equal to a diagonal matrix. We can set $W = C$.

Let us see an example to get some intuition (Rencher 1995): Assume we are given a class of students with grades on five courses, and we want to order these students. That is, we

want to project the data onto one dimension, such that the difference between the data points become most apparent. We can use PCA. The eigenvector with the highest eigenvalue is the direction that has the highest variance, that is, the direction on which the students are most spread out. This works better than taking the average because we consider correlations and differences in variances.

In practice even if all eigenvalues are greater than 0, if |S| is small, remembering that $|S|=\pi_{i=1}^{d} =\lambda_i$, we understand that some eigenvalues have little contribution to variance and may be discarded. Then, we consider the leading k components that explain more than, for example, 90 percent, of the variance. When λ_i are sorted in descending order, the proportion of variance explained by the k principal components is

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

2. Linear Discriminant Analysis (LDA):

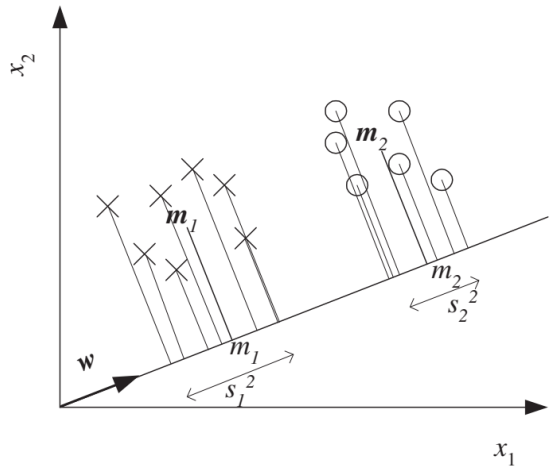
Linear discriminant analysis (LDA) is a supervised method for dimensionality reduction for classification problems. We start with the case where there are two classes, then generalize to K > 2 classes. Given samples from two classes C1 and C2, we want to find the direction, as defined by a vector w, such that when the data are projected onto w, the examples from the two classes are as well separated as possible. As we saw before,

$z = w^T x$

is the projection of x onto w and thus is a dimensionality reduction from d to 1. **m1** and m1 are the means of samples from C_1 before and after projection, respectively. Note that $m1 \in R^d$ and $m1 \in R$. We are given a sample $X = \{x^t, r^t\}$ such that $r^t = 1$ if $x^t \in C_1$ and $r^t = 0$ if $x^t \in C_2$

$$m_1 = \frac{\sum_t \boldsymbol{w}^T \boldsymbol{x}^t r^t}{\sum_t r^t} = \boldsymbol{w}^T \boldsymbol{m}_1$$

$$m_2 = \frac{\sum_t \boldsymbol{w}^T \boldsymbol{x}^t (1 - r^t)}{\sum_t (1 - r^t)} = \boldsymbol{w}^T \boldsymbol{m}_2$$



The scatter of samples from C_1 and C_2 after projection are

$$s_1^2 = \sum_t (\boldsymbol{w}^T \boldsymbol{x}^t - m_1)^2 r^t$$
$$s_2^2 = \sum_t (\boldsymbol{w}^T \boldsymbol{x}^t - m_2)^2 (1 - r^t)$$

After projection, for the two classes to be well separated, we would like the means to be as far apart as possible and the examples of classes be scattered in as small a region as possible. So, $|m_1 - m_2|$ to be large and $s_1^2 + s_2^2$ to be small. Fisher's linear discriminant is $\boldsymbol{w}$ that maximizes

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

Rewriting the numerator, we get

$$\begin{aligned}(m_1 - m_2)^2 &= (\boldsymbol{w}^T \boldsymbol{m}_1 - \boldsymbol{w}^T \boldsymbol{m}_2)^2 \\ &= \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} \\ &= \boldsymbol{w}^T \mathbf{S}_B \boldsymbol{w}\end{aligned}$$

where $S_B = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T$ is the between-class scatter matrix. The denominator is the sum of scatter of examples of classes around their means after projection and can be rewritten as

$$\begin{aligned}s_1^2 &= \sum_t (\boldsymbol{w}^T \boldsymbol{x}^t - m_1)^2 r^t \\ &= \sum_t \boldsymbol{w}^T (\boldsymbol{x}^t - \boldsymbol{m}_1)(\boldsymbol{x}^t - \boldsymbol{m}_1)^T \boldsymbol{w} r^t \\ &= \boldsymbol{w}^T \mathbf{S}_1 \boldsymbol{w}\end{aligned}$$

where

$$\mathbf{S}_1 = \sum_t r^t (\boldsymbol{x}^t - \boldsymbol{m}_1)(\boldsymbol{x}^t - \boldsymbol{m}_1)^T$$

is the within-class scatter matrix for C1.

$\mathbf{S}_1/\sum_t r^t$ is the estimator of Σ_1. Similarly, $s_2^2 = w^T S_2 w$ with $S_2 = \sum_t (1 - r^t)(x^t - m_2)(x^t - m_2)^T$, and we get

$$s_1^2 + s_2^2 = w^T S_W w$$

where $S_W = S_1 + S_2$ is the total within-class scatter. Note that $S_1^2 + S_2^2$ divided by the total number of samples is the variance of the pooled data. The above equation of J(w)can be rewritten as

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \mathbf{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}} = \frac{|\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)|^2}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}}$$

Taking the derivative of J with respect to $\boldsymbol{w}$ and setting it equal to 0, we get

$$\frac{\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}} 2 \left((\boldsymbol{m}_1 - \boldsymbol{m}_2) - \frac{\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)}{\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}} \mathbf{S}_W \boldsymbol{w} \right) = 0$$

Given that $\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)/\boldsymbol{w}^T \mathbf{S}_W \boldsymbol{w}$ is a constant, we have

$$\boldsymbol{w} = c\mathbf{S}_W^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)$$

where c is some constant. Because it is the direction that is important and not the magnitude, we can just take c = 1 and find **w**.

Remember that when $p(x|C_i) \sim N(\mu_i, \Sigma)$, we have a linear discriminant where $w=\Sigma^{-1}(\mu 1-\mu 2)$, and we see that Fisher's linear discriminant is optimal if the classes are normally distributed. Under the same assumption, a threshold, w0, can also be calculated to separate the two classes. But Fisher's linear discriminant can be used even when the classes are not normal. We have projected the samples from d dimensions to 1and any classification method can be used afterward. We see two-dimensional synthetic data with two classes. As we see, and as expected, because it uses the class information, LDA direction is superior to the PCA direction in terms of the ease of discrimination afterwards. In the case of K > 2 classes, we want to find the matrix W such that $z = W^T x$ where z is k-dimensional and W is d × k.

The within-class scatter matrix for C_i is

$$\mathbf{S}_i = \sum_t r_i^t (\boldsymbol{x}^t - \boldsymbol{m}_i)(\boldsymbol{x}^t - \boldsymbol{m}_i)^T$$

where $r_i^t = 1$ if $\boldsymbol{x}^t \in C_i$ and 0 otherwise. The total within-class scatter is

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i$$

When there are $K > 2$ classes, the scatter of the means is calculated as how much they are scattered around the overall mean

$$\boldsymbol{m} = \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{m}_i$$

and the between-class scatter matrix is

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

Thus, we are interested in the matrix W that maximizes

$$J(\mathbf{W}) = \frac{|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_W \mathbf{W}|}$$

The largest eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_B$ are the solution. S_B is the sum of K matrices of rank 1, namely, $(m_i - m)(m_i - m)^T$, and only K − 1 of the mare independent. Therefore, S_B has a maximum rank of K − 1 and we take k = K − 1. Thus, we define a new lower, (K − 1) dimensional space where the discriminant is then to be constructed. Though LDA uses class separability as its goodness criterion, any classification method can be used in this new space for estimating the discriminants.

We see that to be able to apply LDA, SW should be invertible. If this is not the case, we can first use PCA to get rid of singularity and then apply LDA to its result; however, we should make sure that PCA does not reduce dimensionality so much that LDA does not have anything left to work on.