SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

SUBJECT: BIG DATA ANALYTICTS

BTech III -II Semester

R 18 regulation

UNIT 5 NOTES

UNIT V
Data Analytics with R Machine Learning: Introduction, Supervised Learning, Unsupervised Learning, Collaborative Filtering, Social Media Analytics, Mobile Analytics, Big Data Analytics with BigR.

Data Analytics with R Machine Learning:

What is machine learning

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information**. Currently, it is being used for various tasks such as **image recognition**, **speech recognition**, **email filtering**, **Facebook auto-tagging**, **recommender system**, and many more.

The machine learning techniques such as **Supervised**, **Unsupervised**, and **Reinforcement** learning. Regression and classification models, clustering methods, hidden Markov models, and various sequential models.

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

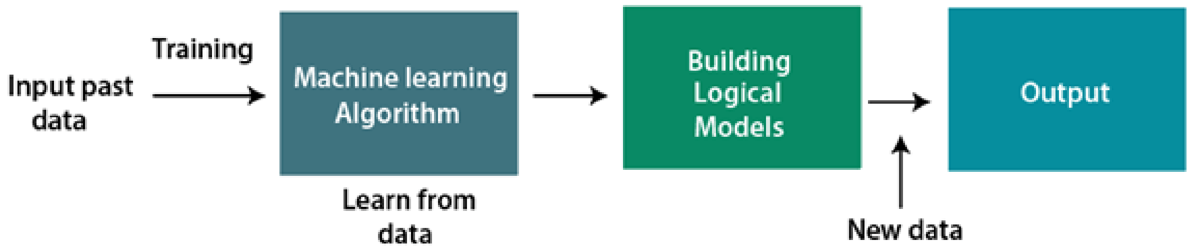
With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

A machine has the ability to learn if it can improve its performance by gaining more data.

How does Machine Learning work

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in **self-driving cars**, **cyber fraud detection**, **face recognition**, and **friend suggestion by Facebook**, etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

Following are some key points which show the importance of Machine Learning:

- Rapid increment in the production of data

- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

Data Analytics with R Machine Learning:

What is R Analytics?

R analytics is data analytics using R programming language, an open-source language used for statistical computing or graphics. This programming language is often used in statistical analysis and data mining. It can be used for analytics to identify patterns and build practical models. R not only can help analyze organizations' data, but also be used to help in the creation and development of software applications that perform statistical analysis.

With a graphical user interface for developing programs, R supports a variety of analytical modeling techniques such as classical statistical tests, clustering, time-series analysis, linear and nonlinear modeling, and more. The interface has four windows: the script window, console window, workspace and history window, and tabs of interest (help, packages, plots, and files). R allows for publication-ready plots and graphics and for storage of reusable analytics for future data.

R has become increasingly popular over many years and remains a top analytics language for many universities and colleges. It is well established today within academia as well as among corporations around the world for delivering robust, reliable, and accurate analytics. While R programming was originally seen as difficult for non-statisticians to learn, the user interface has become more user-friendly in recent years. It also now allows for extensions and other plugins like R Studio and R Excel, making the learning process easier and faster for new business analysts and other users. It has become the industry standard for statistical analysis and data mining projects and is due to grow in use as more graduates enter the workforce as R-trained analysts.

What are the Benefits of R Analytics?

Business analytics in R allows users to analyze business data more efficiently. The following are some of the main benefits realized by companies employing R in their analytics programs:

Democratizing Analytics Across the Organization: R can help democratize analytics by enabling business users with interactive data visualization and reporting tools. R can be used for data science by non data scientists so that business users and citizen data scientists can make better business decisions. R analytics can also reduce time spent on data preparation and data wrangling, allowing data scientists to focus on more complex data science initiatives.

Providing Deeper, More Accurate Insights: Today, most successful companies are data driven and therefore data analytics affects almost every area of business. And while there are a whole host of powerful data analytics tools, R can help create powerful models to analyze large amounts of data. With more precise data collection and storage through R analytics, companies can deliver more valuable insights to users. Analytics and statistical engines using R provide deeper, more accurate insights for the business. R can be used to develop very specific, in-depth analyses.

Leveraging Big Data: R can help with querying big data and is used by many industry leaders to leverage big data across the business. With R analytics, organizations can surface new insights in their large data sets and make sense of their data. R can handle these big datasets and is arguably as easy if not easier for most analysts to use as any of the other analytics tools available today.

Creating Interactive Data Visualizations: R is also helpful for data visualization and >data exploration because it supports the creation of graphs and diagrams. It includes the ability to create interactive visualizations and 3D charts and graphs that are helpful for communicating with business users.

How Can R analytics Be Implemented?

While R programming was originally designed for statisticians, it can be implemented for a variety of uses including predictive analytics, data modeling, and data mining. Businesses can implement R to create custom models for data collection, clustering, and analytics. R analytics can provide a valuable way to quickly develop models targeted at understanding specific areas of the business and delivering tailored insights on day-to-day needs.

R analytics can be used for the following purposes:

- Statistical testing
- Prescriptive analytics
- Predictive analytics
- Time-series analysis
- What-if analysis
- Regression models
- Data exploration
- Forecasting
- Text mining
- Data mining
- Visual analytics
- Web analytics
- Social media analytics
- Sentiment analysis

R can be used to solve real-world business problems by turbocharging an organization's analytics program. It can be integrated into a business's analytics platform to help users get the most out of their data. With an extensive library of R functions and advanced statistical techniques, R can be used to apply statistical models to your analysis and better understand trends in the data. It can help predict potential business outcomes, identify opportunities and risks and create interactive dashboards to gain a comprehensive view of the data. This can lead to better business decisions and increased revenue.

Advantages to Implement Machine Learning Using R Language

- It provides good explanatory code. For example, if you are at the early stage of working with a machine learning project and you need to explain the work you do, it becomes easy to work with R language comparison to python language as it provides the proper statistical method to work with data with fewer lines of code.
- R language is perfect for data visualization. R language provides the best prototype to work with machine learning models.
- R language has the best tools and library packages to work with machine learning projects. Developers can use these packages to create the best pre-model, model, and post-model of the machine learning projects. Also, the packages for R are more advanced and extensive than python language which makes it the first choice to work with machine learning projects.

Popular R Language Packages Used to Implement Machine Learning

- **lattice:** The lattice package supports the creation of the graphs displaying the variable or relation between multiple variables with conditions.

- **DataExplorer:** This R package focus to automate the data visualization and data handling so that the user can pay attention to data insights of the project.
- **Dalex(Descriptive Machine Learning Explanations):** This package helps to provide various explanations for the relation between the input variable and its output. It helps to understand the complex models of machine learning
- **dplyr:** This R package is used to summarize the tabular data of machine learning with rows and columns. It applies the "split-apply-combine" approach.
- **Esquisse:** This R package is used to explore the data quickly to get the information it holds. It also allows to plot bar graph, histograms, curves, and scatter plots.
- **caret:** This R package attempts to streamline the process for creating predictive models.
- **janitor:** This R package has functions for examining and cleaning dirty data. It is basically built for the purpose of user-friendliness for beginners and intermediate users.
- **rpart:** This R package helps to create the classification and regression models using two-stage procedures. The resulting models are represented as binary trees.

Application Of R in Machine Learning

There are many top companies like Google, Facebook, Uber, etc using the R language for application of Machine Learning. The application are:

- Social Network Analytics
- To analyze trends and patterns
- Getting insights for behaviour of users
- To find the relationships between the users
- Developing analytical solutions
- Accessing charting components
- Embedding interactive visual graphics

Example of Machine Learning Problems

- **Web search like Siri, Alexa, Google, Cortona:** Recognize the user's voice and fulfill the request made
- **Social Media Service:** Help people to connect all over the world and also show the recommendations of the people we may know
- **Online Customer Support:** Provide high convenience of customer and efficiency of support agent

- **Intelligent Gaming:** Use high level responsive and adaptive non player characters similar to human like intelligence
- **Product Recommendation:** A software tool used to recommend the product that you might like to purchase or engage with
- **Virtual Personal Assistance:** It is the software which can perform the task according to the instructions provided
- **Traffic Alerts:** Help to switch the traffic alerts according to the situation provided
- **Online Fraud Detection:** Check the unusual functions performed by the user and detect the frauds
- **Healthcare:** Machine Learning can manage a large amount of data beyond the imagination of normal human being and help to identify the illness of the patient according to symptoms
- **Real world example:** When you search for some kind of cooking recipe on youTube, you will see the recommendations below with the title "You May Also Like This". This is a common use of Machine Learning.

Types of Machine Learning Problems

- **Regression:** The regression technique helps the machine learning approach to predict continuous values. For example, the price of a house.
- **Classification:** The input is divided into one or more classes or categories for the learner to produce a model to assign unseen modules. For example, in the case of email fraud, we can divide the emails into two classes i.e "spam" and "not spam".
- **Clustering:** This technique follows the summarization, finding a group of similar entities. For example, we can gather and take readings of the patients in the hospital.
- **Association:** This technique finds co-occurring events or items. For example, market-basket.
- **Anomaly Detection:** This technique works by discovering abnormal cases or behavior. For example, credit card fraud detection.
- **Sequence Mining:** This technique predicts the next stream event. For example, click-stream event.
- **Recommendation:** This technique recommends the item. For example, songs or movies according to the celebrity in it.

Packages

We will be using, directly or indirectly, the following packages through the chapters:

- caret
- ggplot2
- mlbench
- class

- caTools
- randomForest
- impute
- ranger
- kernlab
- class
- glmnet
- naivebayes
- rpart
- rpart.plot

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.

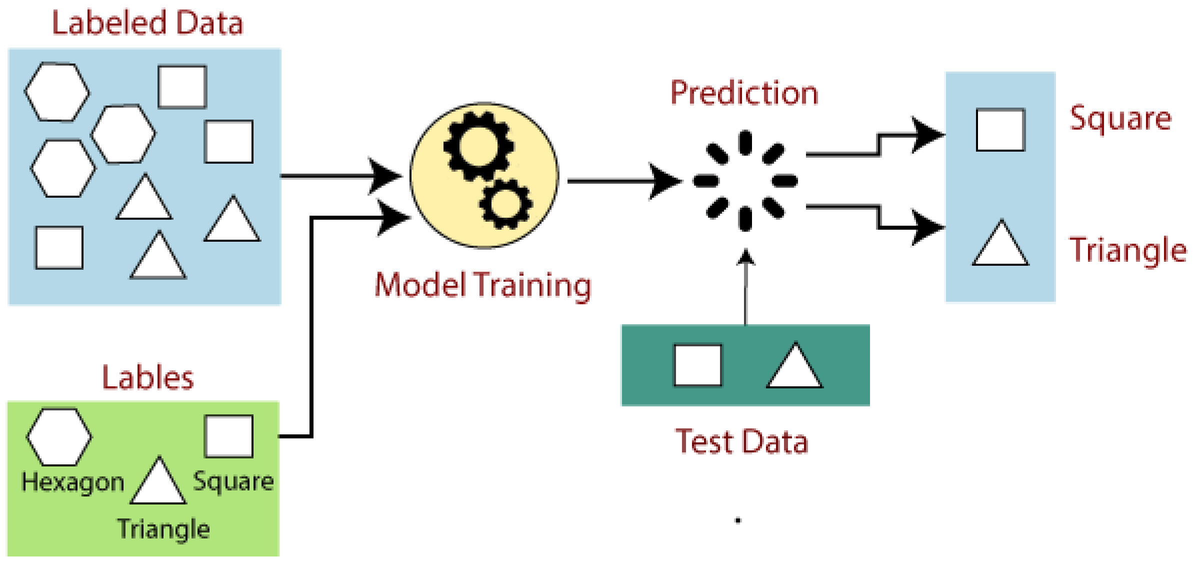
Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

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

How Supervised Learning Works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

The working of Supervised learning can be easily understood by the below example and diagram:



Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

- If the given shape has four sides, and all the sides are equal, then it will be labelled as a **Square**.
- If the given shape has three sides, then it will be labelled as a **triangle**.
- If the given shape has six equal sides then it will be labelled as **hexagon**.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training **dataset, test dataset, and validation dataset**.
- Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.

- Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

Supervised learning is classified into two categories of algorithms:

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

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

Types:-

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

Regression:

- A regression is a statistical technique that relates a dependent variable to one or more independent (explanatory) variables.
- A regression model is able to show whether changes observed in the dependent variable are associated with changes in one or more of the explanatory variables.
- In regression, we normally have one dependent variable and one or more independent variables.
- Here we try to “regress” the value of the dependent variable “Y” with the help of the independent variables.
- In other words, we are trying to understand, how the value of ‘Y’ changes w.r.t change in ‘X’.

For the regression analysis is be a successful method, we understand the following terms:

- **Dependent Variable:** This is the variable that we are trying to understand or forecast.
- **Independent Variable:** These are factors that influence the analysis or target variable and provide us with information regarding the relationship of the variables with the target variable.

Regression analysis is used for prediction and forecasting. This statistical method is used across different industries such as,

- Financial Industry- Understand the trend in the stock prices, forecast the prices, and evaluate risks in the insurance domain
- Marketing- Understand the effectiveness of market campaigns, and forecast pricing and sales of the product.
- Manufacturing- Evaluate the relationship of variables that determine to define a better engine to provide better performance
- Medicine- Forecast the different combinations of medicines to prepare generic medicines for diseases.

Logistic Regression

- Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of data belonging to a given class or not.
- It is a kind of statistical algorithm, which analyze the relationship between a set of independent variables and the dependent binary variables.
- It is a powerful tool for decision-making. For example email spam or not.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
- Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

Classification

In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

What is the Classification Algorithm?

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

Naïve Bayes classifer

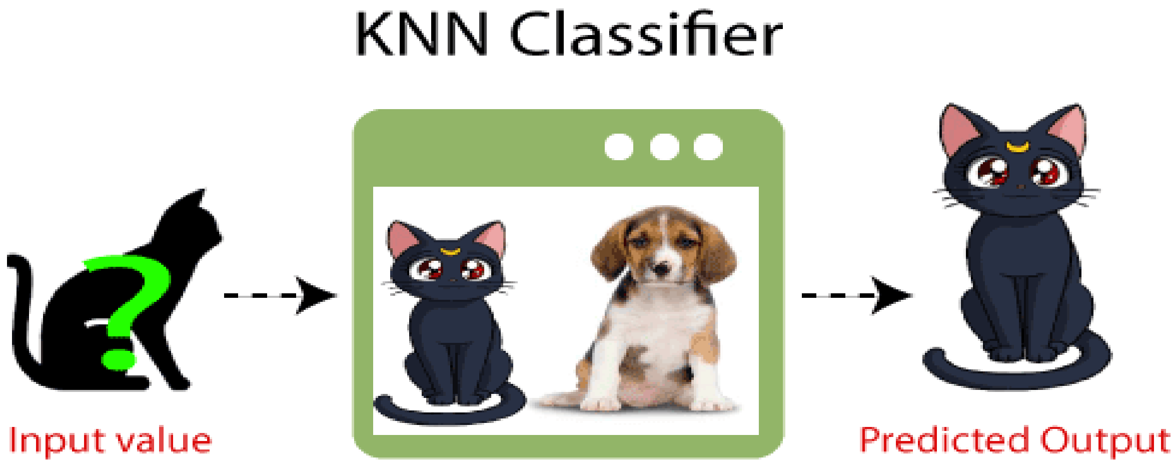
- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

K-NN (k nearest neighbors)

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity.
- This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

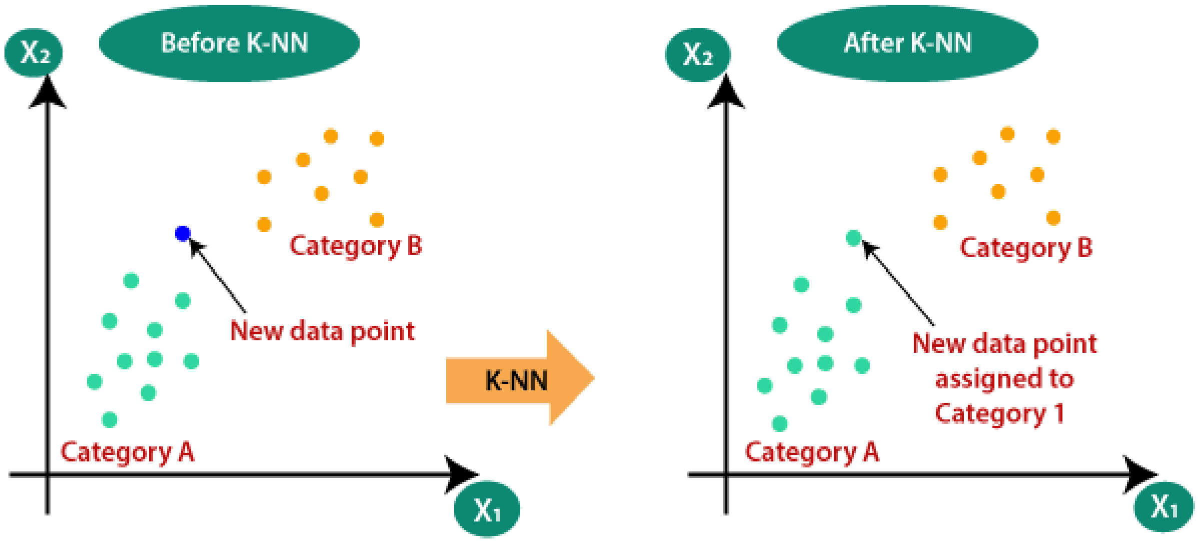
Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

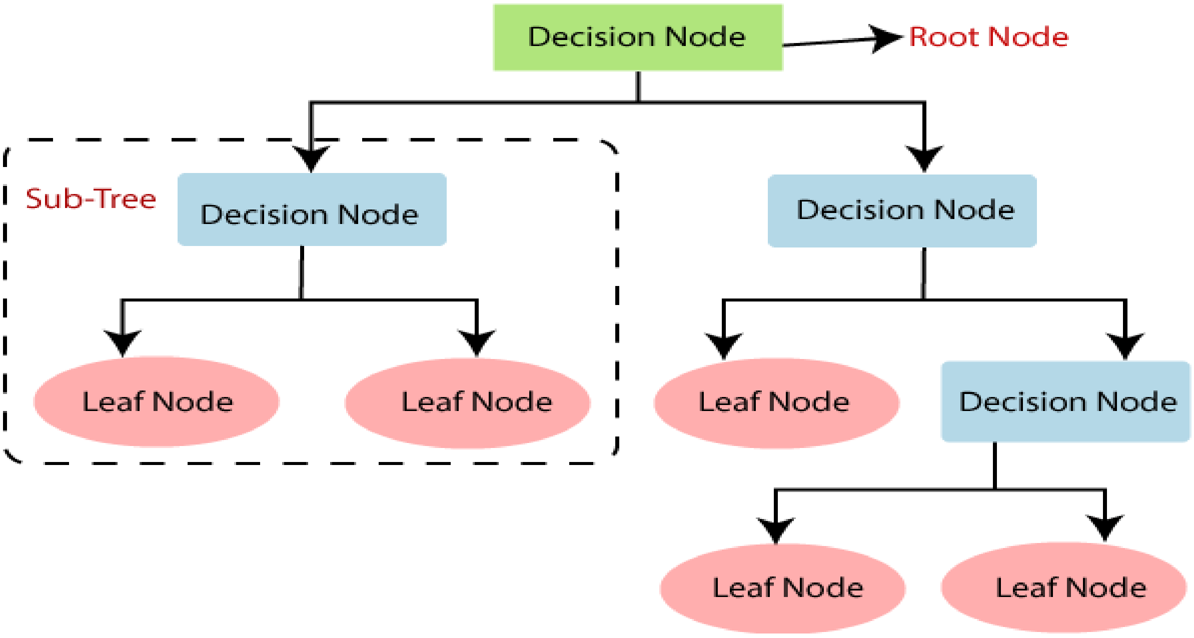


How does K-NN work?

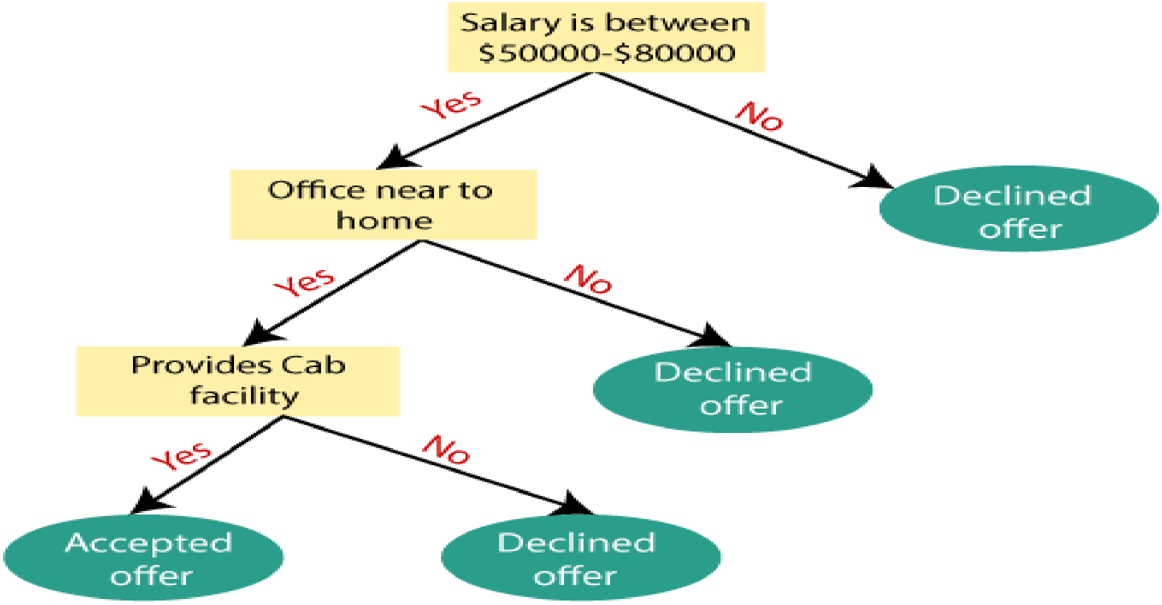
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.





- **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.

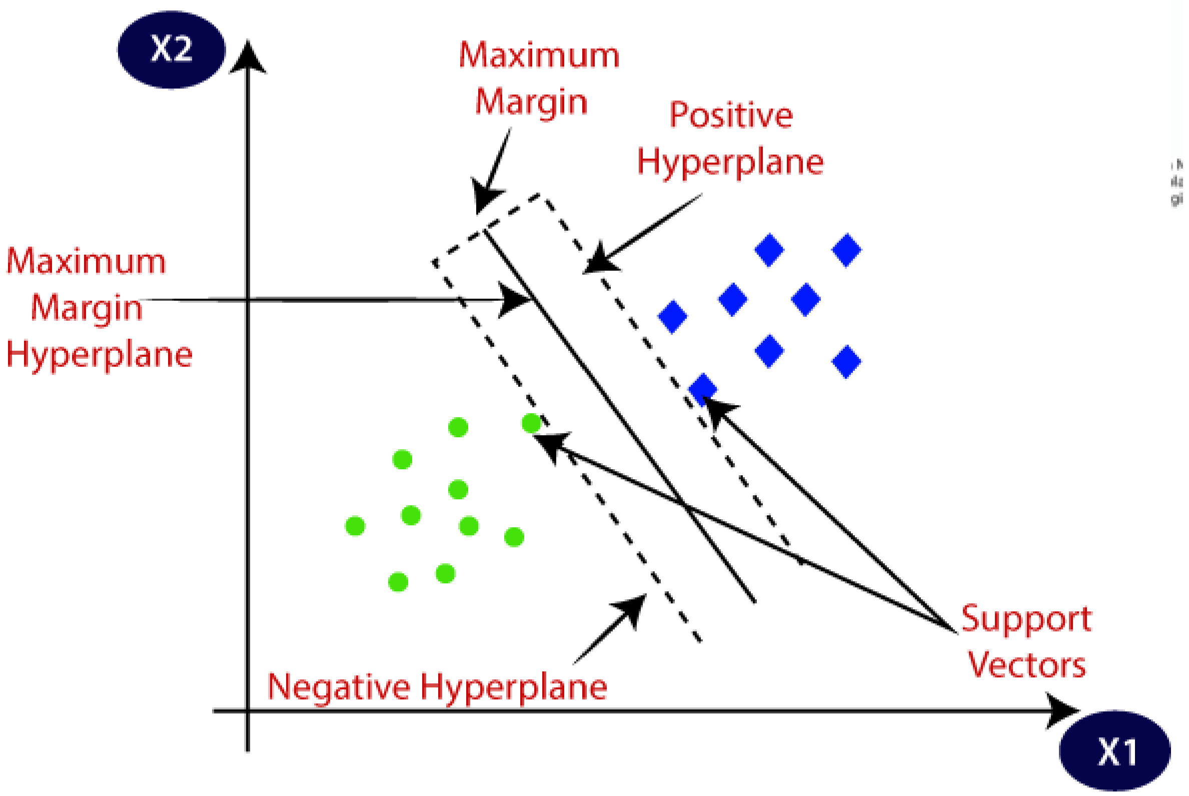


SUPPORT VECTOR MACHINES

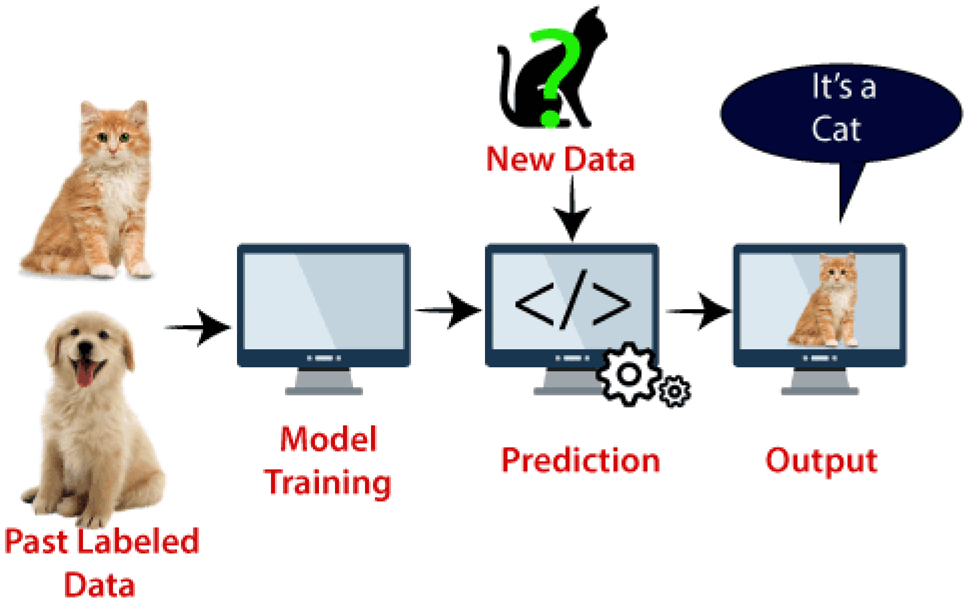
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

Types of SVM

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

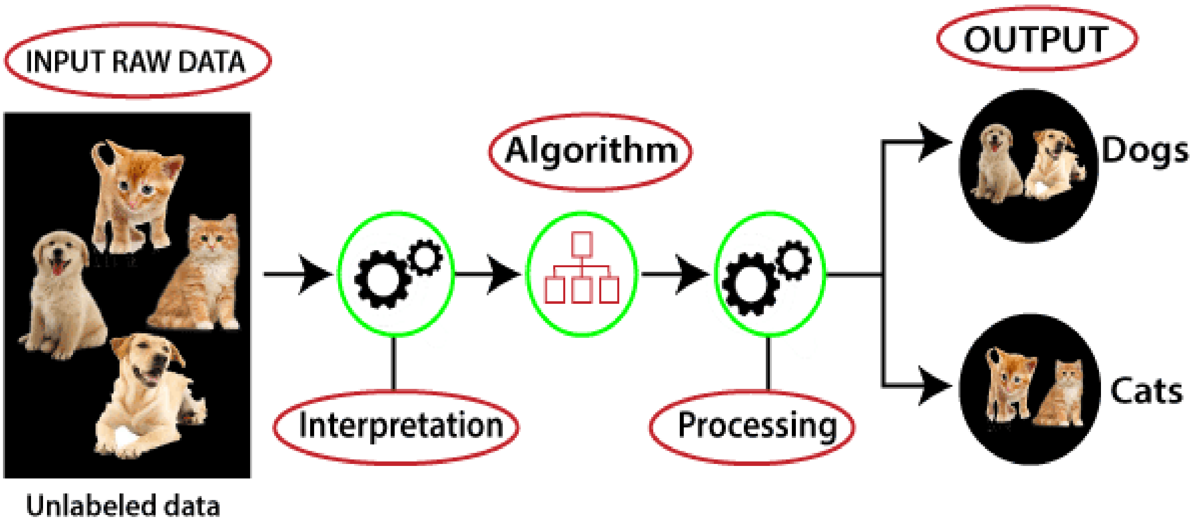
Unsupervised learning

It uses machine learning algorithms to analyze and cluster unlabeled datasets.

These algorithms discover hidden patterns or data groupings without the need for human intervention.

- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

Working of Unsupervised Learning



Clustering Types of Unsupervised Learning Algorithms

Below are the clustering types of Unsupervised Machine Learning algorithms:

Unsupervised learning problems further grouped into clustering and association problems.

Clustering

Clustering is an important concept when it comes to unsupervised learning.

It mainly deals with finding a structure or pattern in a collection of uncategorized data. Unsupervised Learning Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data.

There are different types of clustering you can utilize:

Exclusive (partitioning)

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

Example: K-means

Agglomerative

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

Example: Hierarchical clustering

Overlapping

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

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

Probabilistic

This technique uses probability distribution to create the clusters.

Clustering Types

Following are the clustering types of Machine Learning:

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

Hierarchical Clustering

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm.

The hierarchical clustering technique has two approaches:

1. **Agglomerative:** Agglomerative is a **bottom-up** approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
2. **Divisive:** Divisive algorithm is the reverse of the agglomerative algorithm as it is a **top-down approach.**

Agglomerative Hierarchical clustering

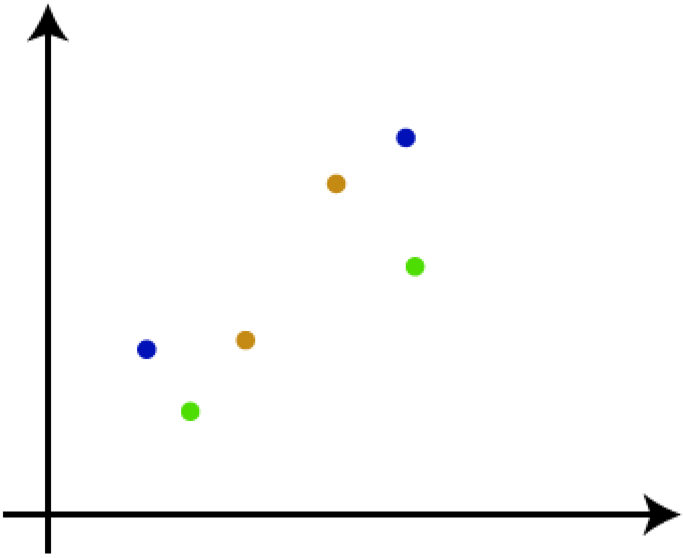
The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the **bottom-up approach**. It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets.

This hierarchy of clusters is represented in the form of the dendrogram.

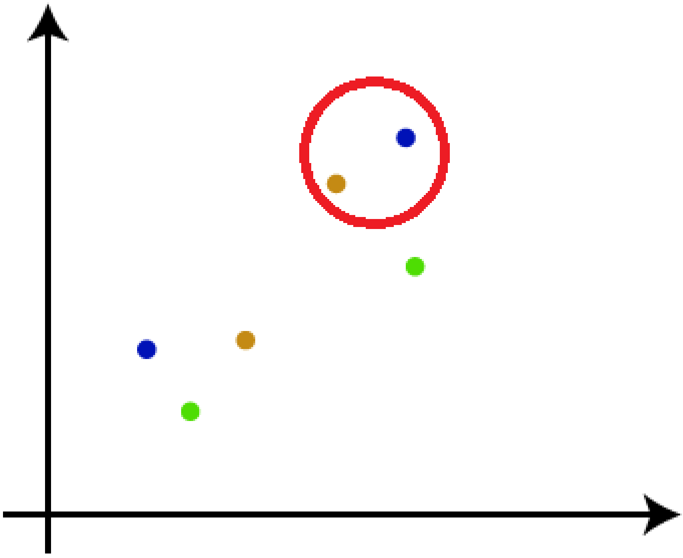
How the Agglomerative Hierarchical clustering Work?

The working of the AHC algorithm can be explained using the below steps:

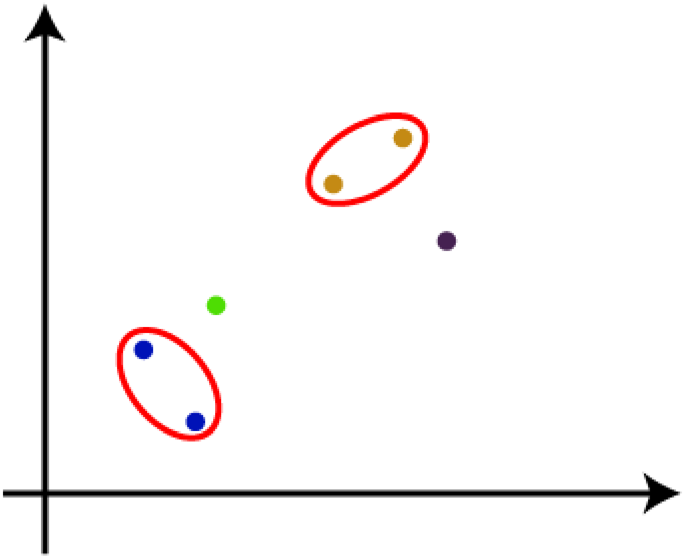
- **Step-1:** Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



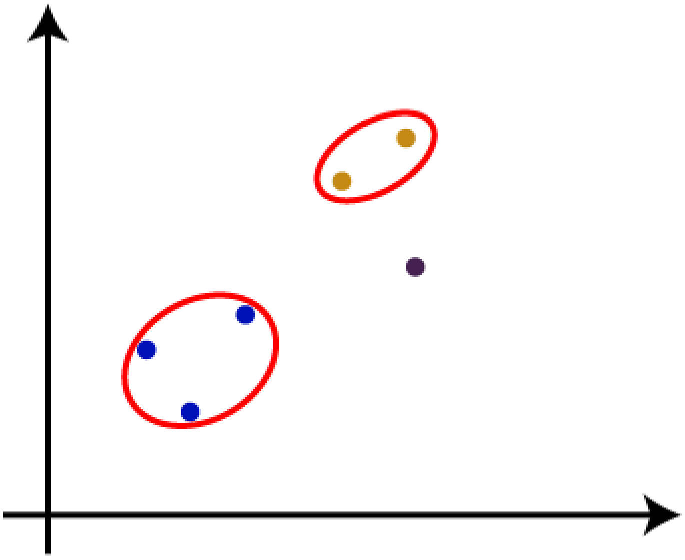
- **Step-2:** Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.

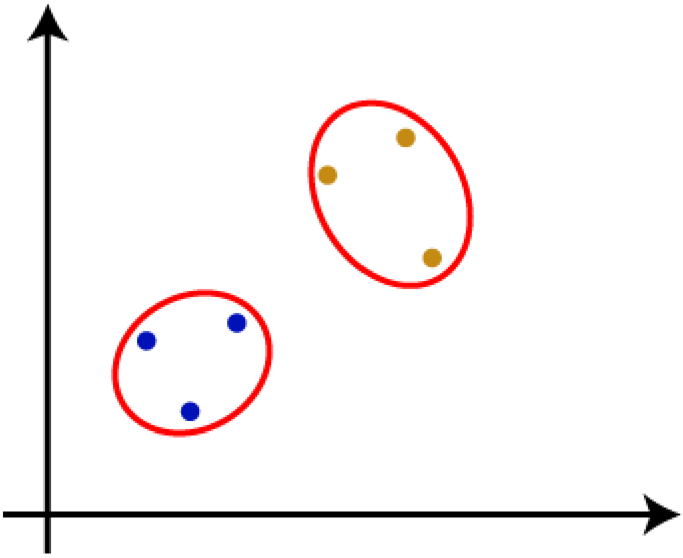


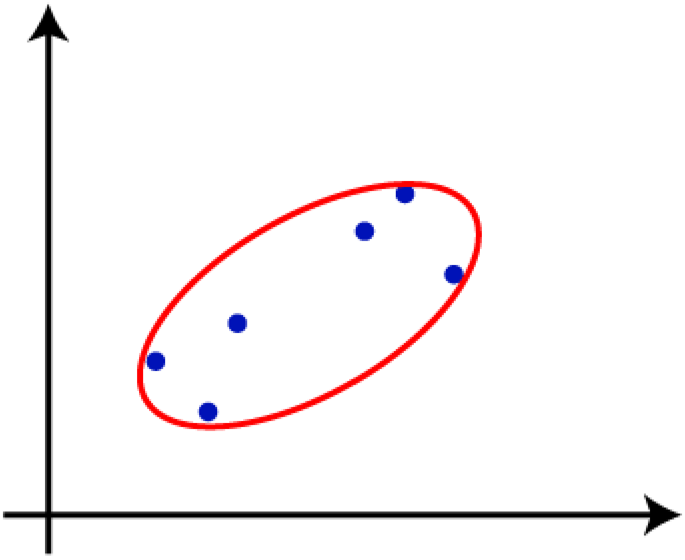
- **Step-3**: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



- **Step-4:** Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:





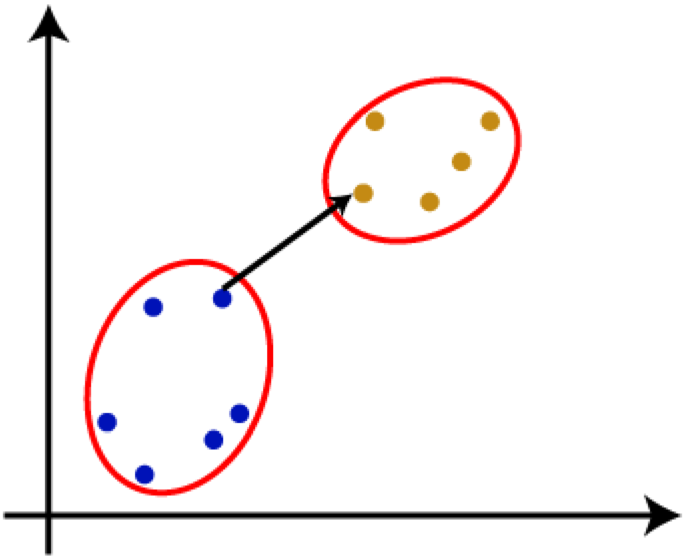


- **Step-5:** Once all the clusters are combined into onebig cluster, develop the dendrogram to divide the clusters as per the problem.

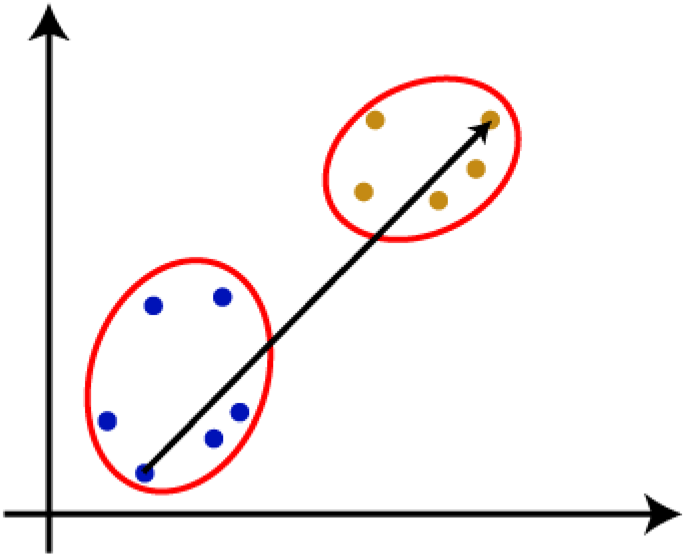
Measure for the distance between two clusters

As we have seen, the **closest distance** between the two clusters is crucial for the hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

1. **Single Linkage:** It is the Shortest Distance between the closest points of the clusters. Consider the below image:

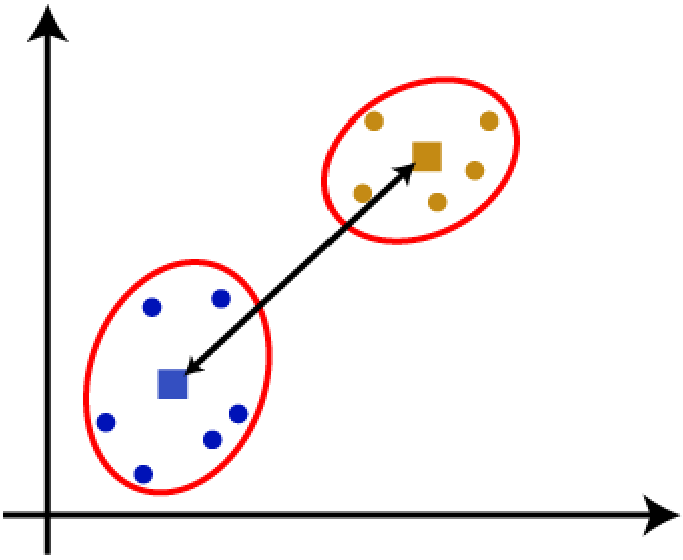


2. **Complete Linkage:** It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



3. **Average Linkage**: It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

4. **Centroid Linkage:** It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:



From the above-given approaches, we can apply any of them according to the type of problem or business requirement.

K-means Clustering

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

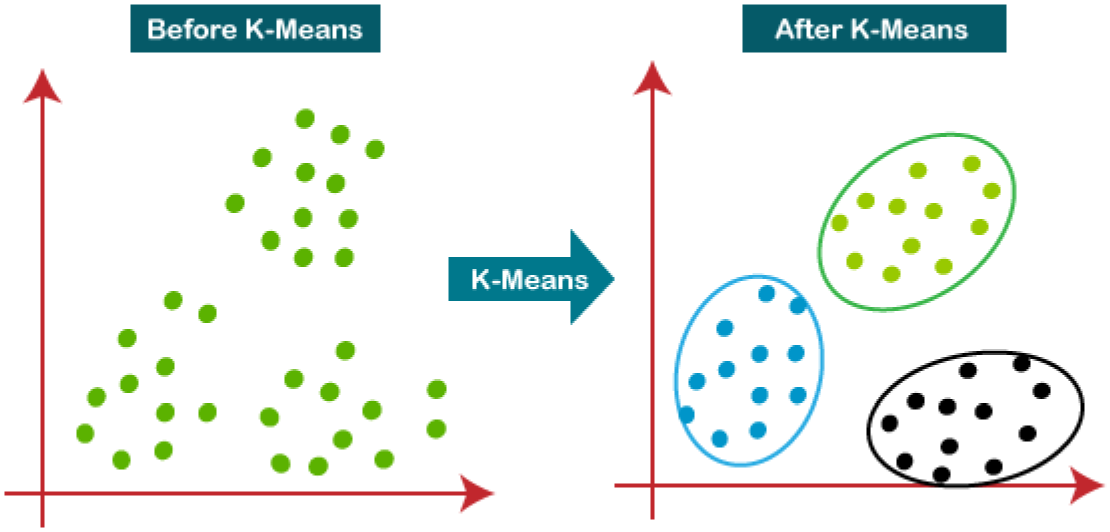
It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

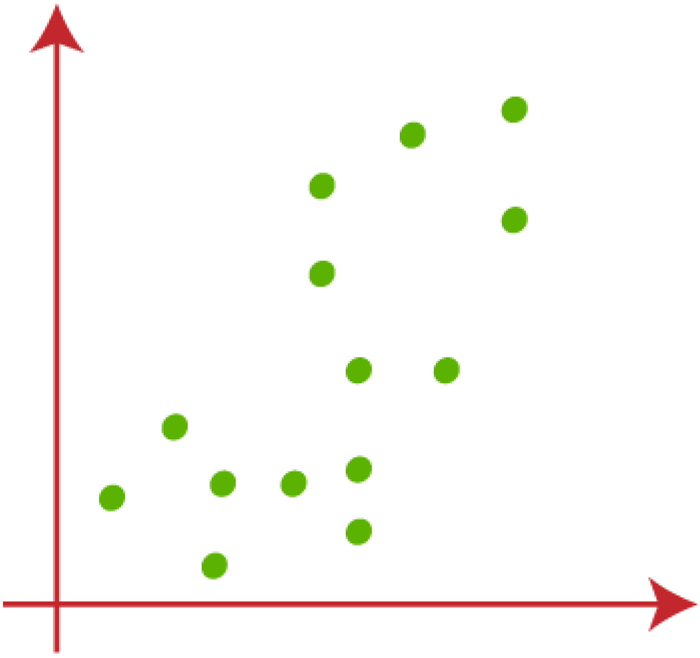
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

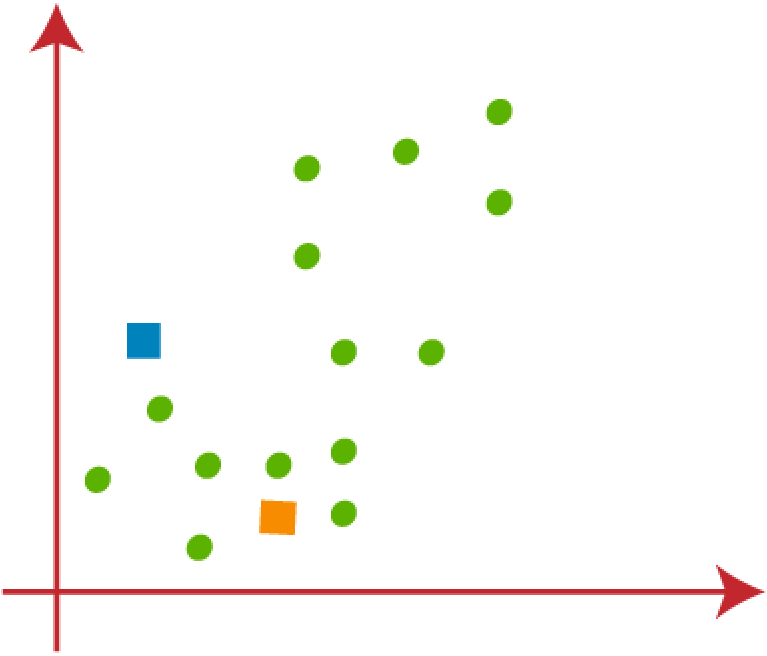
Let's understand the above steps by considering the visual plots:

Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:

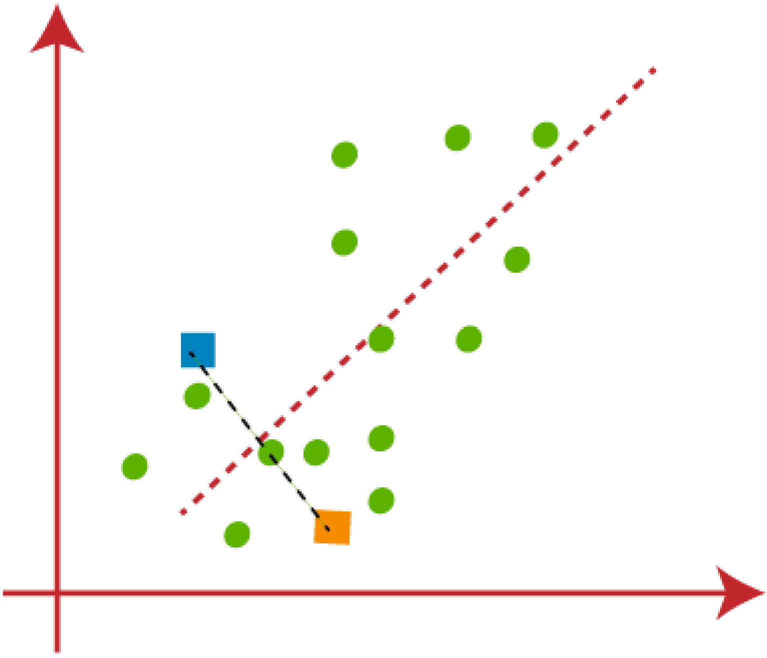


- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset.

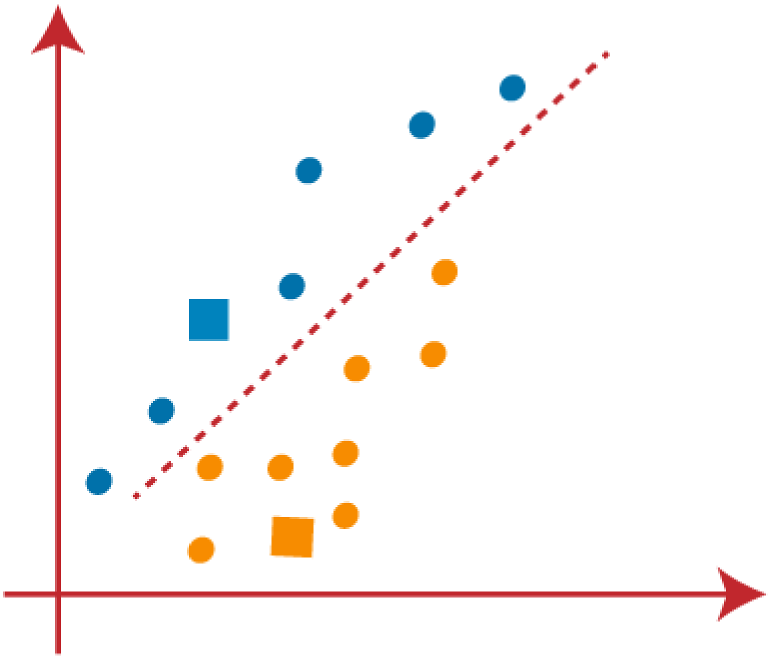
Consider the below image:



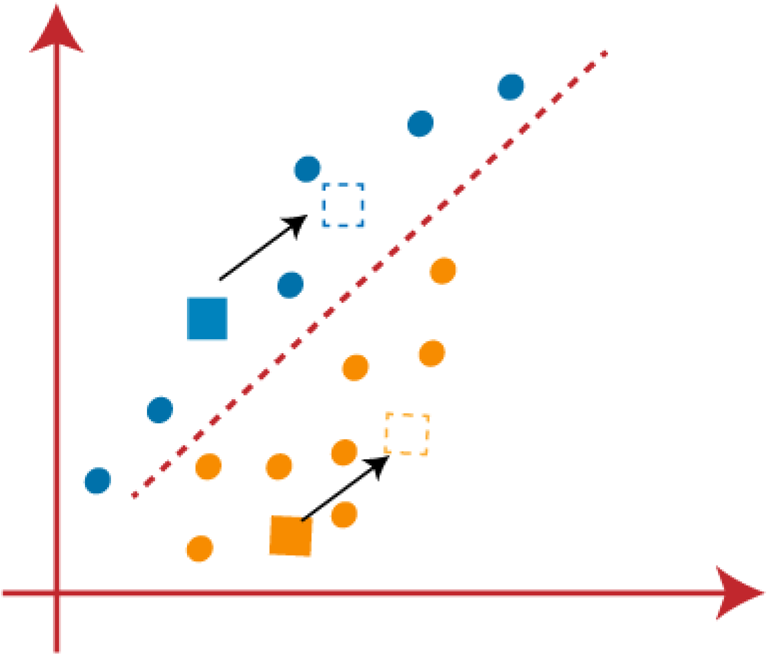
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



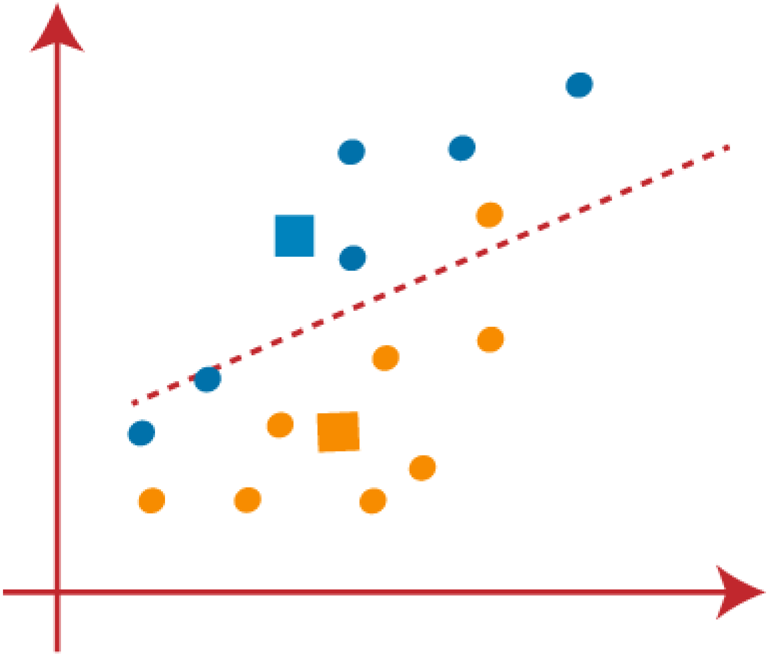
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



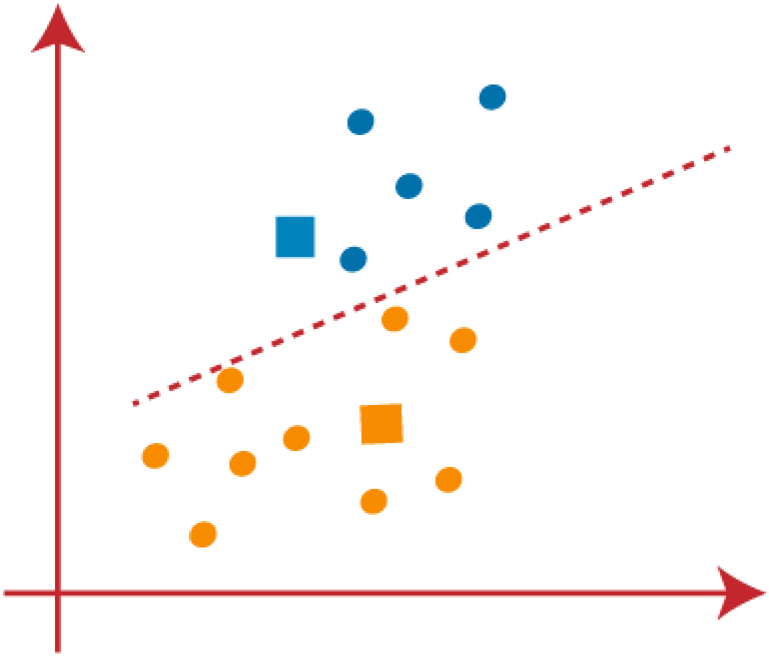
- As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



- Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

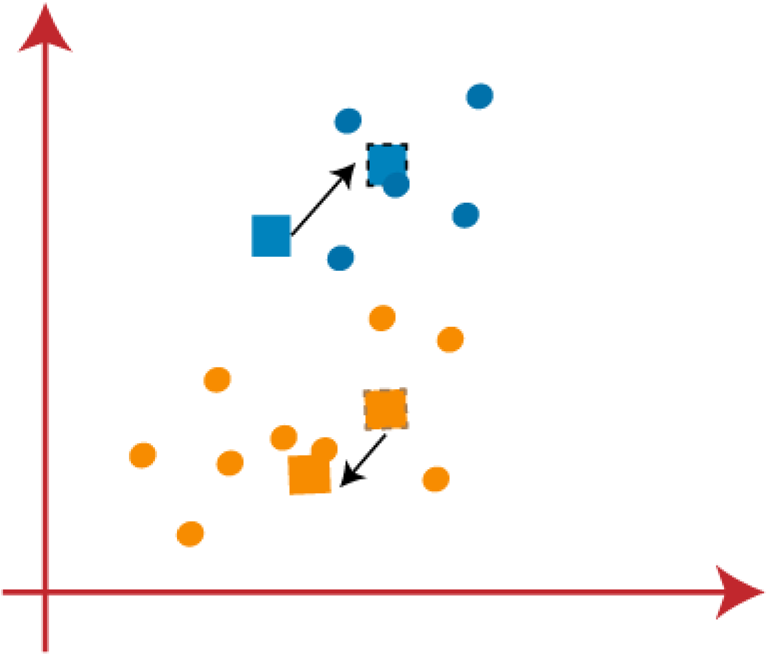


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

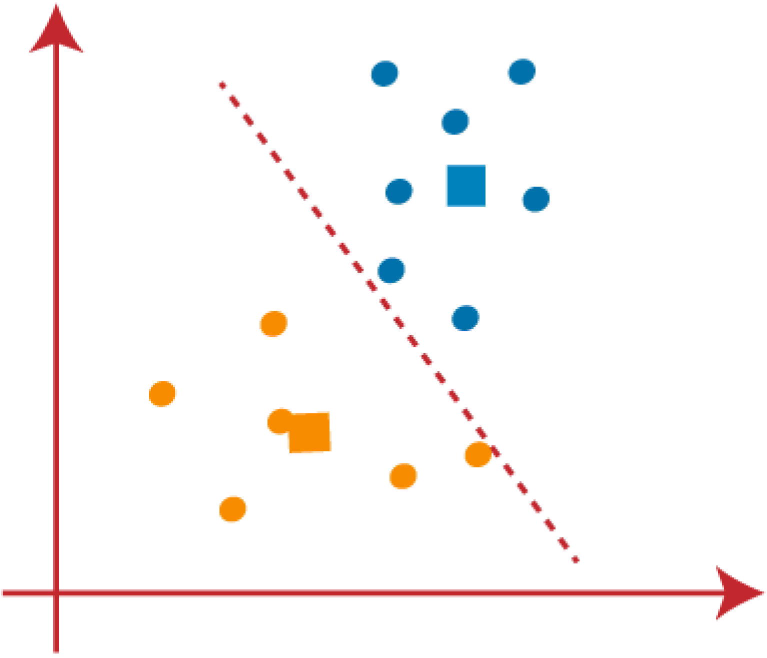


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

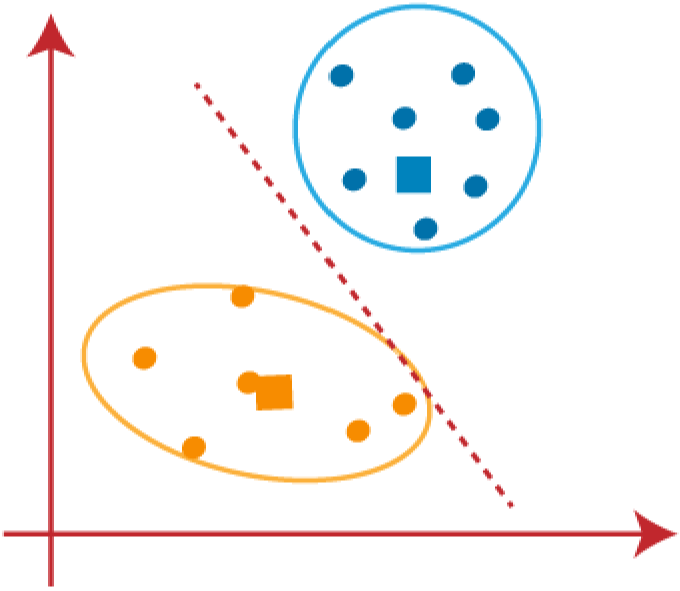
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



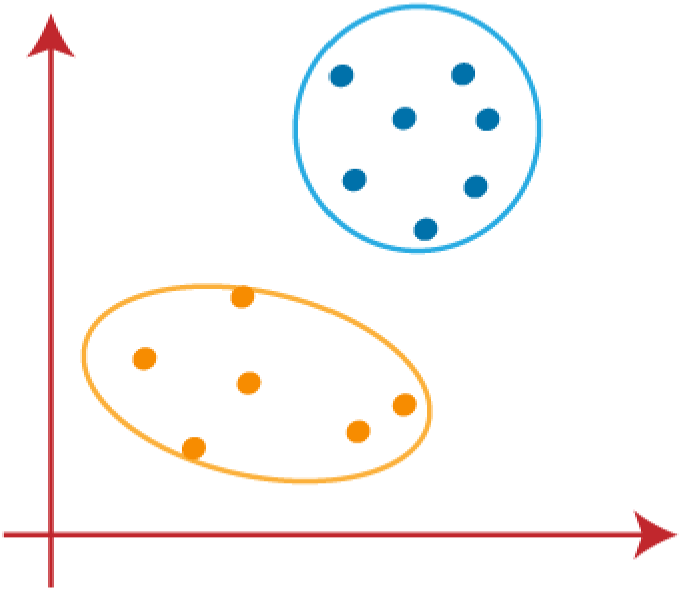
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



- We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:

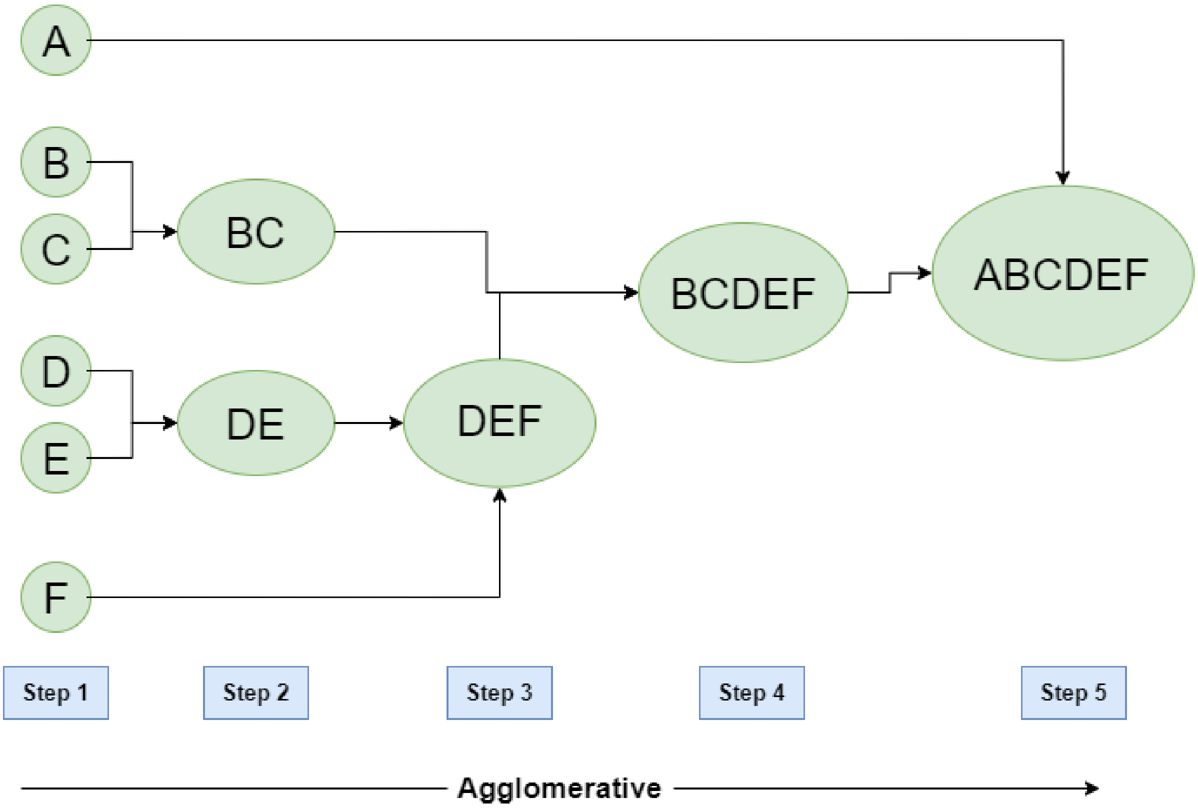


As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



Agglomerative clustering

It is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all data.



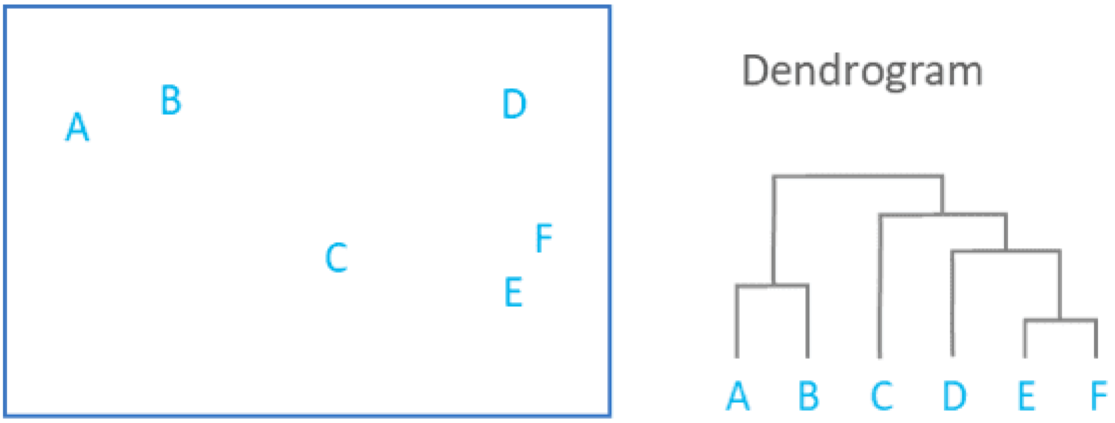
Steps:

- Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.
- In the second step, comparable clusters are merged together to form a single cluster. Let's say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly to cluster (D) and (E) and at last, we get the clusters [(A), (BC), (DE), (F)]
- We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]
- Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We're now left with clusters [(A), (BCDEF)].

- At last, the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

Dendrogram

A *dendrogram* is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from *hierarchical clustering*. The main use of a dendrogram is to work out the best way to allocate objects to clusters. The dendrogram below shows the hierarchical clustering of six *observations* shown on the *scatterplot* to the left. (Dendrogram is often miswritten as dendogram.

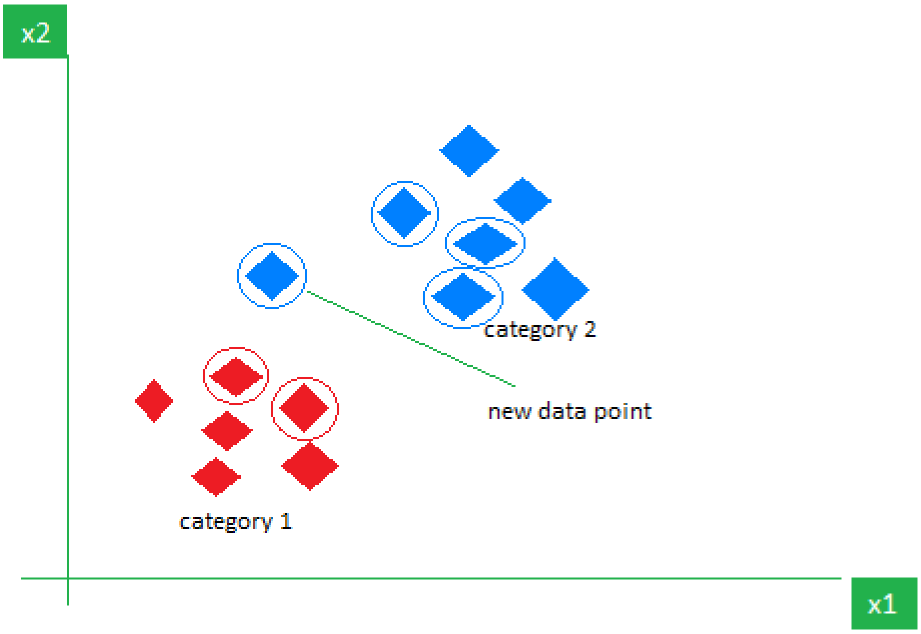


K- Nearest neighbors

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

As an example, consider the following table of data points containing two features:



Distance Metrics Used in KNN Algorithm

As we know that the KNN algorithm helps us identify the nearest points or the groups for a query point. But to determine the closest groups or the nearest points for a query point we need some metric. For this purpose, we use below distance metrics:

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance

Euclidean Distance

This is nothing but the cartesian distance between the two points which are in the plane/hyperplane. Euclidean distance can also be visualized as the length of the straight line that joins the two points which are into consideration. This metric helps us calculate the net displacement done between the two states of an object.

Manhattan Distance

This distance metric is generally used when we are interested in the total distance traveled by the object instead of the displacement. This metric is calculated by summing the absolute difference between the coordinates of the points in n-dimensions.

Minkowski Distance

We can say that the Euclidean, as well as the Manhattan distance, are special cases of the Minkowski distance.

From the formula above we can say that when p = 2 then it is the same as the formula for the Euclidean distance and when p = 1 then we obtain the formula for the Manhattan distance.

The above-discussed metrics are most common while dealing with a Machine Learning problem but there are other distance metrics as well like Hamming Distance which come in handy while dealing with problems that require overlapping comparisons between two vectors whose contents can be boolean as well as string values.

How to choose the value of k for KNN Algorithm?

The value of k is very crucial in the KNN algorithm to define the number of neighbors in the algorithm. The value of k in the k-nearest neighbors (k-NN) algorithm should be chosen based on the input data. If the input data has more outliers or noise, a higher value of k would be better. It is recommended to choose an odd value for k to avoid ties in classification. Cross-validation methods can help in selecting the best k value for the given dataset.

Applications of the KNN Algorithm

- **Data Preprocessing** – While dealing with any Machine Learning problem we first perform the EDA part in which if we find that the data contains missing values then there are multiple imputation methods are available as well. One of such method is KNN Imputer which is quite effective ad generally used for sophisticated imputation methodologies.
- **Pattern Recognition** – KNN algorithms work very well if you have trained a KNN algorithm using the MNIST dataset and then

performed the evaluation process then you must have come across the fact that the accuracy is too high.

- **Recommendation Engines** – The main task which is performed by a KNN algorithm is to assign a new query point to a pre-existed group that has been created using a huge corpus of datasets. This is exactly what is required in the recommender systems to assign each user to a particular group and then provide them recommendations based on that group's preferences.

Advantages of the KNN Algorithm

- **Easy to implement** as the complexity of the algorithm is not that high.
- **Adapts Easily** – As per the working of the KNN algorithm it stores all the data in memory storage and hence whenever a new example or data point is added then the algorithm adjusts itself as per that new example and has its contribution to the future predictions as well.
- **Few Hyperparameters** – The only parameters which are required in the training of a KNN algorithm are the value of k and the choice of the distance metric which we would like to choose from our evaluation metric.

Disadvantages of the KNN Algorithm

- **Does not scale** – As we have heard about this that the KNN algorithm is also considered a Lazy Algorithm. The main significance of this term is that this takes lots of computing power as well as data storage. This makes this algorithm both time-consuming and resource exhausting.
- **Curse of Dimensionality** – There is a term known as the peaking phenomenon according to this the KNN algorithm is affected by the curse of dimensionality which implies the algorithm faces a hard time classifying the data points properly when the dimensionality is too high.
- **Prone to Overfitting** – As the algorithm is affected due to the curse of dimensionality it is prone to the problem of overfitting as well. Hence generally feature selection as well as dimensionality reduction techniques are applied to deal with this problem.

Principal Components Analysis

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are ***image processing, movie recommendation system, optimizing the power allocation in various communication channels.*** It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a non-zero vector v is given. Then v will be eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Steps for PCA algorithm

1. **Getting the dataset**

   Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

2. **Representing data into a structure**

   Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.

3. **Standardizing the data**

   In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.

4. **Calculating the Covariance of Z**

   To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.

5. **Calculating the Eigen Values and Eigen Vectors**

   Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.

6. **Sorting the Eigen Vectors**

   In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.

7. **Calculating the new features Or Principal Components**

   Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.

8. **Remove less or unimportant features from the new dataset.**

   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed out.

Applications of Principal Component Analysis

- PCA is mainly used as the dimensionality reduction technique in various AI applications such **as computer vision, image compression, etc.**
- It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.

Association

Association rules allow you to establish associations amongst data objects inside large databases. This unsupervised technique is about discovering interesting relationships between variables in large databases.

For example, people that buy a new home most likely to buy new furniture.

Other Examples:

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

Supervised vs. Unsupervised Machine Learning

Parameters	Supervised machine learning technique	Unsupervised machine learning technique

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

Collaborative filtering

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users.

It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user.

What is a Recommendation system?

There are a lot of applications where websites collect data from their users and use that data to predict the likes and dislikes of their users. This allows them to recommend the content that they like. Recommender systems are a way of suggesting similar items and ideas to a user's specific way of thinking.

There are basically two types of recommender Systems:

Collaborative Filtering: Collaborative Filtering recommends items based on similarity measures between users and/or items.

- The basic assumption behind the algorithm is that users with similar interests have common preferences.

Content-Based Recommendation:

- It is supervised machine learning used to induce a classifier to discriminate between interesting and uninteresting items for the user.

What is Collaborative Filtering?

In Collaborative Filtering, we tend to find similar users and recommend what similar users like. In this type of recommendation system, we don't use the features of the item to recommend it, rather we classify the users into clusters of similar types and recommend each user according to the preference of its cluster.

There are basically four types of algorithms o say techniques to build Collaborative filtering based recommender systems:

- Memory-Based
- Model-Based
- Hybrid
- Deep Learning

- Memory-based

Memory-based methods use user rating historical data to compute the similarity between users or items. The idea behind these methods is to define a similarity measure between users or items, and find the most similar to recommend unseen items.

- Model-based

Model-based CF uses machine learning algorithms to predict users' rating of unrated items.

There are many model-based CF algorithms, the most commonly used are matrix factorization models such as to applying a SVD to reconstruct the rating matrix, latent Dirichlet allocation or Markov decision process based models.

- Hybrid

These aim to combine the memory-based and the model-based approaches. One of the main drawbacks of the above methods, is that you'll find yourself having to choose between historical user rating data and user or item attributes.

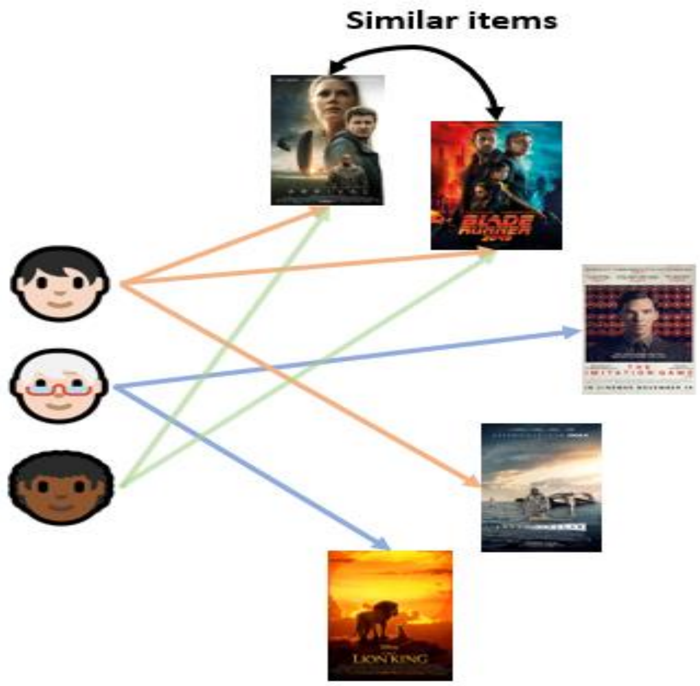
Hybrid methods enable us to leverage both, and hence tend to perform better in most cases. The most widely used methods nowadays are factorization machines.

Memory-based CF

There are 2 main types of memory-based collaborative filtering algorithms: User-Based and Item-Based. While their difference is subtle, in practice they lead to very different approaches, so it is crucial to know which is the most convenient for each case. Let's go through a quick overview of these methods:

- **Item-Based**

The idea is similar, but instead, starting from a given movie (or set of movies) we find similar movies based on other users' preferences.



Cold start is a potential problem in computer-based information systems which involves a degree of automated data modelling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information.

Also, since a *single item is enough* to recommend other similar items, this method will not suffer from the cold-start problem.

The algorithm for **used-based CF** can be summarised as:

1. *Compute the similarity between the new user with all other users (if not already done)*
2. *Compute the mean rating of all movies of the* ***k*** *most similar users*
3. *Recommend the top* ***n*** *rated movies by other users unseen by the user*

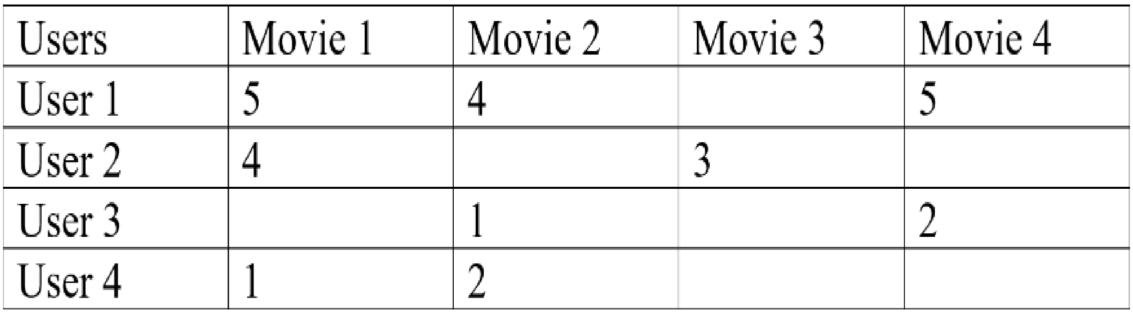
Advantages of Collaborative Filtering-Based Recommender Systems

As we know there are two types of recommender systems the content-based recommender systems have limited use cases and have higher time complexity.

Also, this algorithm is based on some limited content but that is not the case in Collaborative Filtering based algorithms.

One of the main advantages that these recommender systems have is that they are highly efficient in providing personalized content but also able to adapt to changing user preferences.

Measuring Similarity



Users	Movie 1	Movie 2	Movie 3	Movie 4
User 1	5	4		5
User 2	4		3	
User 3		1		2
User 4	1	2		

Social Media Analytics

Social media analytics is the ability to gather and find meaning in data gathered from social channels to support business decisions — and measure the performance of actions based on those decisions through social media.

Social media analytics is broader than metrics such as likes, follows, retweets, previews, clicks, and impressions gathered from individual channels. It also differs from reporting offered by services that support marketing campaigns such as LinkedIn or Google Analytics.

- Social media analytics uses specifically designed software platforms that work similarly to web search tools.
- Data about keywords or topics is retrieved through search queries or web ‘crawlers’ that span channels. Fragments of text are returned, loaded into a database, categorized and analyzed to derive meaningful insights.

Why is social media analytics important?

Social media analytics helps companies address these experiences and use them to:

- Spot trends related to offerings and brands
- Understand conversations — what is being said and how it is being received

- Derive customer sentiment towards products and services
- Gauge response to social media and other communications
- Identify high-value features for a product or service
- Uncover what competitors are saying and its effectiveness
- Map how third-party partners and channels may affect performance

Key capabilities of effective social media analytics

From there, topics or keywords can be selected and parameters such as date range can be set.

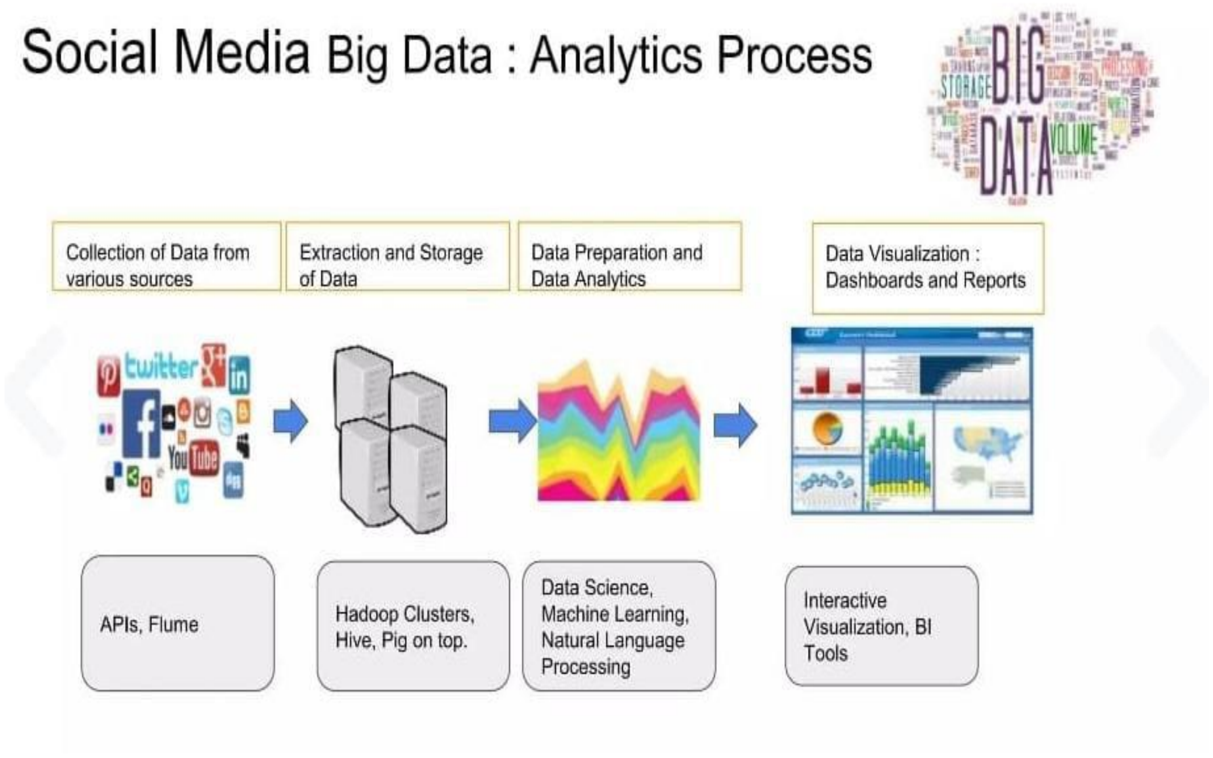
Sources also need to be specified — responses to YouTube videos, Facebook conversations, Twitter arguments, Amazon product reviews, comments from news sites.

- **Natural language processing and machine learning** technologies identify entities and relationships in unstructured data — information not pre-formatted to work with data analytics. Virtually all social media content is unstructured. These technologies are critical to deriving meaningful insights.
- **Segmentation** is a fundamental need in social media analytics. It categorizes social media participants by geography, age, gender, marital status, parental

status and other demographics. It can help identify influencers in those categories. Messages, initiatives and responses can be better tuned and targeted by understanding who is interacting on key topics.

- **Behavior analysis** is used to understand the concerns of social media participants by assigning behavioral types such as user, recommender, prospective user and detractor. Understanding these roles helps develop targeted messages and responses to meet, change or deflect their perceptions.
- **Sentiment analysis** measures the tone and intent of social media comments. It typically involves natural language processing technologies to help understand entities and relationships to reveal positive, negative, neutral or ambivalent attributes.
- **Share of voice** analyzes prevalence and intensity in conversations regarding brand, products, services, reputation and more. It helps determine key issues and important topics. It also helps classify discussions as positive, negative, neutral or ambivalent.
- **Clustering analysis** can uncover hidden conversations and unexpected insights. It makes associations between keywords or phrases that appear together frequently and derives new topics, issues and opportunities. The people that make baking soda, for example, discovered new uses and opportunities using clustering analysis.

- **Dashboards and visualization** charts, graphs, tables and other presentation tools summarize and share social media analytics findings — a critical capability for communicating and acting on what has been learned. They also enable users to grasp meaning and insights more quickly and look deeper into specific findings without advanced technical skills.



Social Media Big Data : Analytics Process

Collection of Data from various sources

Extraction and Storage of Data

Data Preparation and Data Analytics

Data Visualization : Dashboards and Reports

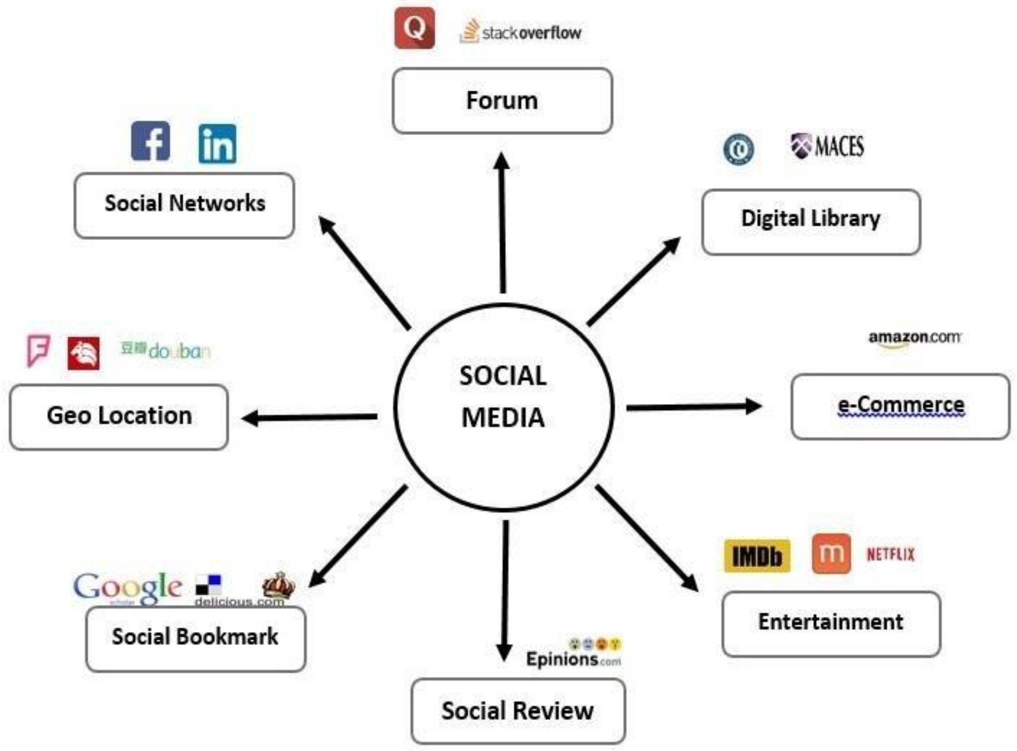
APIs, Flume

Hadoop Clusters, Hive, Pig on top.

Data Science, Machine Learning, Natural Language Processing

Interactive Visualization, BI Tools

Social Media Types :



***Power of Data Science in Social Media Analytics* :**

***1.* Sentiment Analysis**

***2.* Social Network Analysis**

***3.* Identification of Top Influencers**

***4.* Identification of most related words**

***5.* Understanding the main concerns of customers**

***6.* Tracking public sentiments real time**

***7.* Identification of social network**

***8.* Tracking sentiments for rival products/services**

How big is it ?

- Its massive size, high update speed and range of content modalities are frequently cited as a textbook example of just what constitutes “big data” in today’s data drenched world.
- We hold up social media platforms today as the epitome of “big data.” However, the lack of external visibility into those platforms means that nearly all of our assessments are based on the hand picked statistics.
- those companies choose to report to the public and the myriad ways those figures, such as “active users,” are constantly evolved to reflect the rosiest image possible of the growth of social media as a whole.
- ***Advantages of social media***

Useful for educational purposes

2. **Build your brand**

3. **Reach a large audience**

4. **Target audiences based on their interests.**

5. **Stay up to date.**

6. **Get connected to new people.**

7. **Create your audiences.**

8. **Free to use.**

9. **Builds relationships.**

10. **Get new visitors to website**

Disadvantages of social media

- **Spending a lot of time on social media.**
- **Decrease in Communication skills.**
- **Fake news.**

- **Social media can cause sleeplessness**
- **Content on social media is not appropriate for children.**
- **Cyber attacks are becoming more prevalent in today's world.**
- **Lack of Confidence.**
- **Fear of missing out (FOMO).**
- **No privacy**
- **Getting close to Depression**

Mobile Analytics

Mobile analytics involves measuring and analysing data generated by mobile platforms and properties, such as mobile sites and mobile applications. AT Internet's analytics solution lets you track, measure and understand how your mobile users are interacting with your mobile sites and mobile apps.

Why do companies use mobile analytics?

Mobile analytics gives companies unparalleled insights into the otherwise hidden lives of app users. Analytics usually comes in the form of software that integrates into companies' existing websites and apps to capture, store, and analyze the data.

This data is vitally important to marketing, sales, and product management teams who use it to make more informed decisions.

Without a mobile analytics solution, companies are left flying blind. They’re unable to tell what users engage with, who those users are, what brings them to the site or app, and why they leave.

Why are mobile analytics important?

- Mobile usage surpassed that of desktop in 2015 and smartphones are fast becoming consumers’ preferred portal to the internet. Consumers spend 70 percent of their media consumption and screen time on mobile devices, and most of that time in mobile apps.
- This is a tremendous opportunity for companies to reach their consumers, but it’s also a highly saturated market. There are more than 6.5 million apps in the major mobile app stores, millions of web apps, and more than a billion websites in existence.
- Companies use mobile analytics platforms to gain a competitive edge in building mobile experiences that stand out. Mobile analytics tools also give teams a much-needed edge in advertising.
- As more businesses compete for customers on mobile, teams need to understand how their ads perform in detail, and whether app users who interact with ads end up purchasing.

How do mobile analytics work?

Mobile analytics typically track:

- Page views
- Visits
- Visitors
- Source data
- Strings of actions
- Location
- Device information
- Login / logout
- Custom event data

Companies use this data to figure out what users want in order to deliver a more satisfying user experience.

For example, they’re able to see:

- What draws visitors to the mobile site or app
- How long visitors typically stay
- What features visitors interact with
- Where visitors encounter problems
- What factors are correlated with outcomes like purchases
- What factors lead to higher usage and long-term retention

How different teams use mobile analytics:

- **Marketing:** Tracks campaign ROI, segments users, automates marketing
- **UX/UI:** Tracks behaviors, tests features, measures user experience
- **Product:** Tracks usage, A/B test features, debugs, sets alerts
- **Technical teams**: Track performance metrics such as app crashes

~~How to implement mobile analytics~~

~~Mobile analytics platforms vary widely in features and functionality. Some free applications have technical limitations and struggle with tracking users as they move between mobile websites and apps. **A top tier mobile analytics platform should be able to:**~~

- ~~**Integrate easily:** With a codeless mobile feature, for instance~~
- ~~**Offer a unified view of the customer:** Track data across operating systems, devices, and platforms~~
- ~~**Measure user engagement:** For both standard and custom-defined events~~
- ~~**Segment users:** Create cohorts based on location, device, demographics, behaviors, and more~~
- ~~**Offer dashboards:** View data and surface insights with customizable reporting~~
- ~~**A/B test:** Test features and messaging for performance~~
- ~~**Send notifications:** Alert administrators and engage users with behavior-based messaging such as push notifications and in-app messages~~
- ~~**Out-of-the-box metrics**: Insights with minimal client-side coding~~
- ~~**Real-time analytics**: Proactively identify user issues~~
- ~~**Reliable infrastructure**: Guaranteed uptime for consistent access to the platform~~

The actual installation of mobile analytics involves adding tracking code to the sites and SDKs to the mobile applications teams want to track. Most mobile analytics platforms will be set up to automatically track website visits.

Platforms with codeless mobile features will be able to automatically track certain basic features of apps such as crashes, errors, and clicks, but you'll want to expand that by manually tagging additional actions for tracking. With mobile analytics in place, you'll have deeper insights into your mobile web and app users which you can use to create competitive, world-class products and experiences.

The Challenges of Mobile Analytics

Because mobile analytics is a fairly new field of analytics and continues to change with rapidly changing consumer expectations, there are many challenges to be faced in implementing it.

Collecting the data necessary for successful mobile analytics is often the greatest challenge organizations face when attempting to understand consumer behavior on mobile devices. Many devices do not allow for cookies to track actions or do not use Javascript which can also help with website data tracking.

Another challenge to ensuring the usefulness of mobile analytics is correctly segmenting users based on their mobile behavior. Organizations must guarantee quality data and tracking so that all the necessary attributes are correctly captured and never duplicated, including network, device, entry page, time spent on page, etc. It is possible for dirty data from mobile analytics, or even algorithmic bias, to lead to incorrect conclusions. To combat this risk, businesses should put measures in place to eliminate bad data, biases, and opportunities for misrepresentation of the data.

Finally, implementing mobile analytics may require a culture shift if your company is not yet fully data driven. To receive a return on your investment in mobile analytics, make sure to follow best practices for onboarding new technologies and gaining and maintaining buy-in from key stakeholders.

The Benefits of Mobile Analytics

Despite the challenges of mobile analytics, it’s essential for modern businesses to invest in and can lead to many opportunities for the business.

- **Improved UI/UX**: Mobile analytics can help businesses develop content and campaigns that best meet their target audience’s needs. With metrics on user engagement, designers can ensure a smooth, consistent user experience that is responsive to user behavior.
- **Optimized Websites and Apps**: Mobile analytics is also useful for product managers and app developers. It can be used to optimize the performance of mobile websites and apps and therefore enhance consumers’ interactions with your business on mobile devices.
- **More Customer Loyalty**: Every company is always on the lookout for ways to improve customer retention. Mobile analytics can help organizations identify those areas for improvement. User retention on a business’ mobile app shows strong brand loyalty and can increase the likelihood of repeat customers once downloaded. It’s also a great way to analyze customer feedback and implement changes rapidly for more agile app development.
- **Analytics Insights**: Mobile analytics looks at easy-to-understand metrics and often provides user-friendly dashboards that non-technical users like marketers and product managers can quickly assess and use to improve performance. This can lead to more targeting marketing and accurate segmentation for overall improved business results.

Big Data Analytics with BigR

Big Data analytics is the process of examining large and complex data sets that often exceed the computational capabilities. R is a leading programming language of data science, consisting of powerful functions to tackle all problems related to Big Data processing.

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R is an important tool for Data Science. It is highly popular and is the first choice of many statisticians and data scientists.

Features of R – Data Science

Some of the important features of R for data science application are:

- R provides extensive support for statistical modelling.
- R is a suitable tool for various data science applications because it provides aesthetic visualization tools.
- R is heavily utilized in data science applications for ETL (Extract, Transform, Load). It provides an interface for many databases like SQL and even spreadsheets.
- R also provides various important packages for data wrangling.
- With R, data scientists can apply machine learning algorithms to gain insights about future events.
- One of the important feature of R is to interface with NoSQL databases and analyze unstructured data.

Most common Data Science in R Libraries

- **Dplyr**: For performing data wrangling and data analysis, we use the dplyr package. We use this package for facilitating various functions for the Data frame in R. Dplyr is actually built around these 5 functions. You can work with local data frames as well as with remote database tables. You might need to:
  Select certain columns of data.
  Filter your data to select specific rows.
  Arrange the rows of your data into order.
  Mutate your data frame to contain new columns.
  Summarize chunks of your data in some way.

- **Ggplot2:** R is most famous for its visualization library ggplot2. It provides an aesthetic set of graphics that are also interactive.The ggplot2 library implements a “grammar of graphics” (Wilkinson, 2005). This approach gives us a coherent way to produce visualizations by expressing relationships between the attributes of data and their graphical representation.
- **Esquisse:** This package has brought the most important feature of Tableau to R. Just drag and drop, and get your visualization done in minutes. This is actually an enhancement to ggplot2.It allows us to draw bar graphs, curves, scatter plots, histograms, then export the graph or retrieve the code generating the graph.
- **Tidyr:** Tidyr is a package that we use for tidying or cleaning the data. We consider this data to be tidy when each variable represents a column and each row represents an observation.
- **Shiny:** This is a very well known package in R. When you want to share your stuff with people around you and make it easier for them to know and explore it visually, you can use shiny. It’s a Data Scientist’s best friend.
- **Caret:** Caret stands for classification and regression training. Using this function, you can model complex regression and classification problems.
- **E1071:** This package has wide use for implementing clustering, Fourier Transform, Naive Bayes, SVM and other types of miscellaneous functions.
- **Mlr:** This package is absolutely incredible in performing machine learning tasks. It almost has all the important and useful algorithms for performing machine learning tasks. It can also be termed as the extensible framework for classification, regression, clustering, multi-classification and survival analysis.

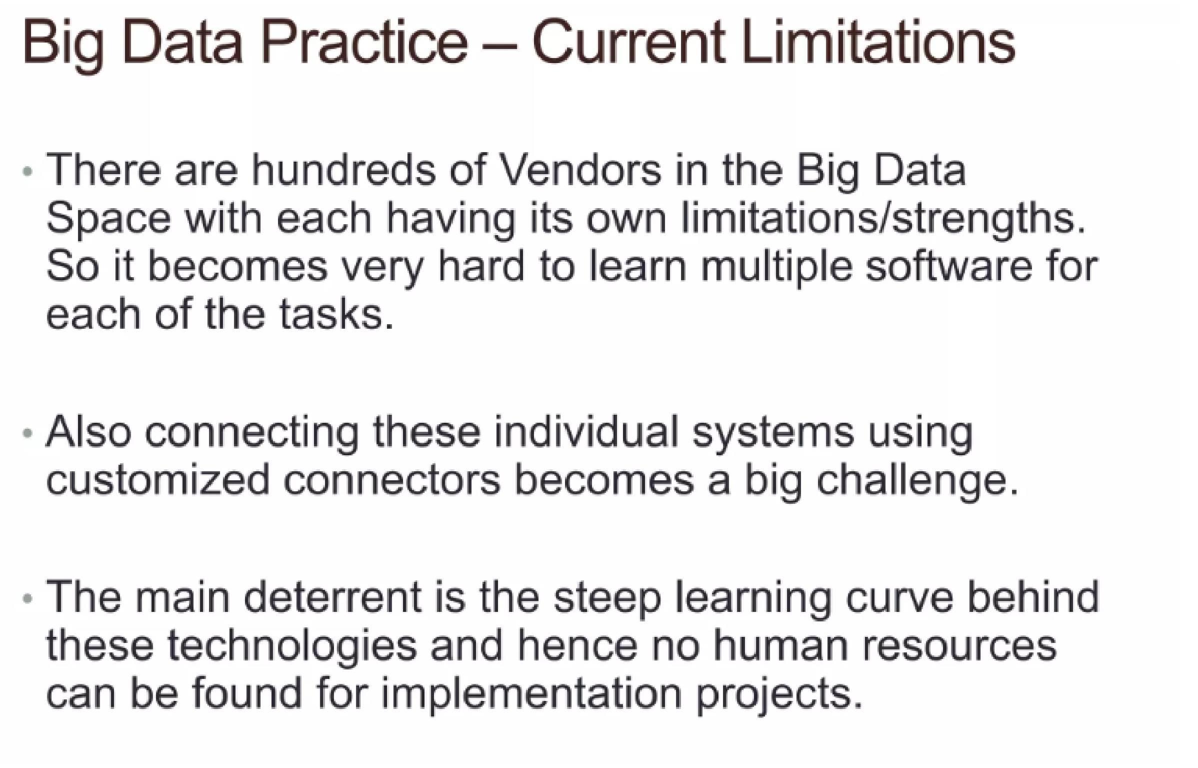
Other worth mentioning R libraries:

- Lubridate
- Knitr
- DT(DataTables)
- RCrawler
- Leaflet
- Janitor
- Plotly

Applications of R for Data Science

Top Companies that use R for Data Science:

- **Google:** At Google, R is a popular choice for performing many analytical operations. The Google Flu Trends project makes use of R to analyze trends and patterns in searches associated with flu.
- **Facebook** Facebook makes heavy use of R for social network analytics. It uses R for gaining insights about the behavior of the users and establishes relationships between them.
- **IBM:** IBM is one of the major investors in R. It recently joined the R consortium. IBM also utilizes R for developing various analytical solutions. It has used R in IBM Watson – an open computing platform.
- **Uber:** Uber makes use of the R package shiny for accessing its charting components. Shiny is an interactive web application that's built with R for embedding interactive visual graphics.

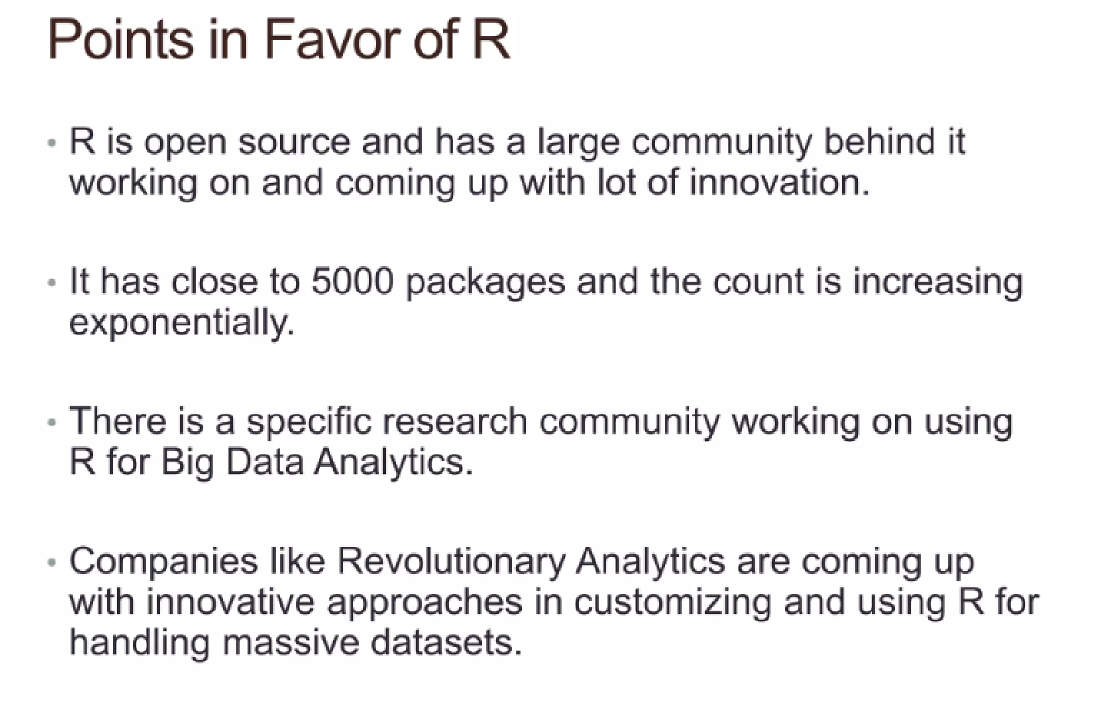


Big Data Practice – Current Limitations

- There are hundreds of Vendors in the Big Data Space with each having its own limitations/strengths. So it becomes very hard to learn multiple software for each of the tasks.

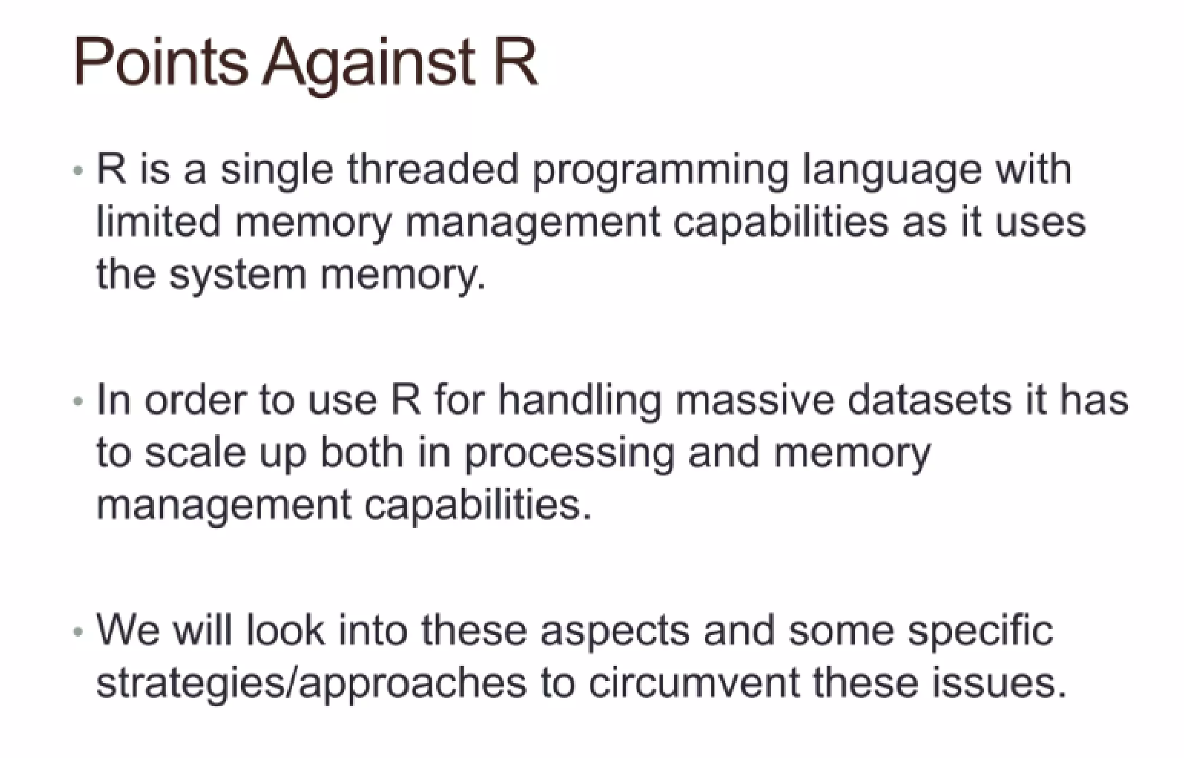
- Also connecting these individual systems using customized connectors becomes a big challenge.

- The main deterrent is the steep learning curve behind these technologies and hence no human resources can be found for implementation projects.



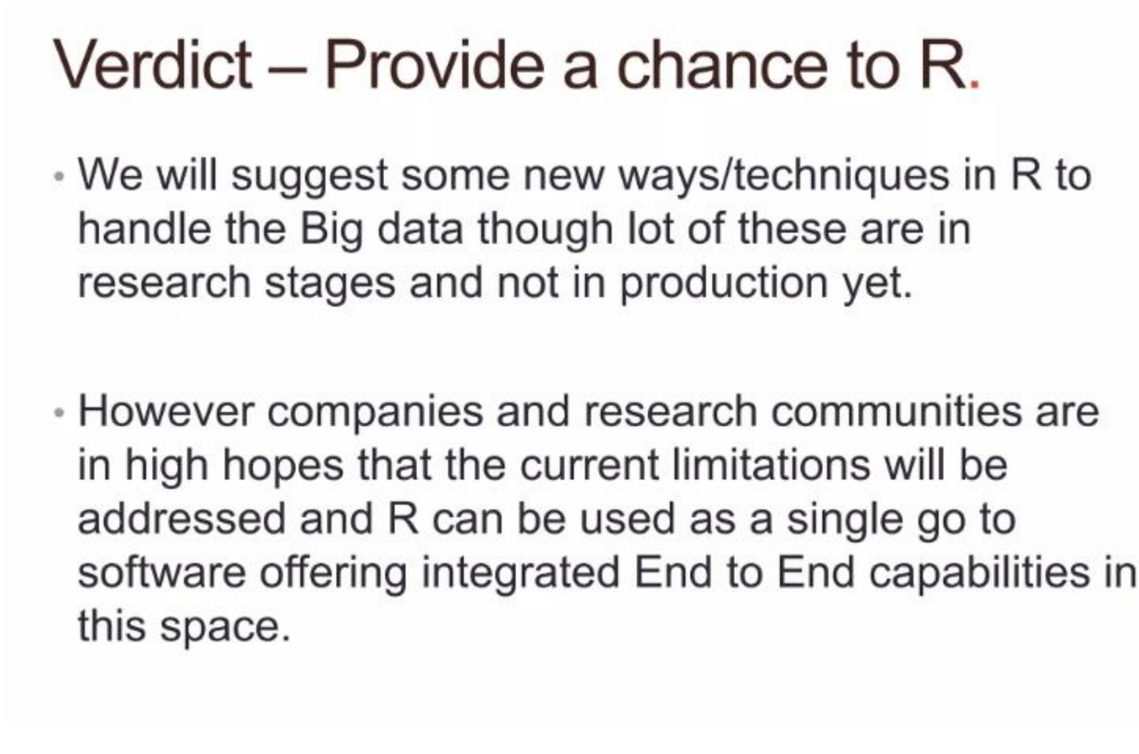
Points in Favor of R

- R is open source and has a large community behind it working on and coming up with lot of innovation.
- It has close to 5000 packages and the count is increasing exponentially.
- There is a specific research community working on using R for Big Data Analytics.
- Companies like Revolutionary Analytics are coming up with innovative approaches in customizing and using R for handling massive datasets.



Points Against R

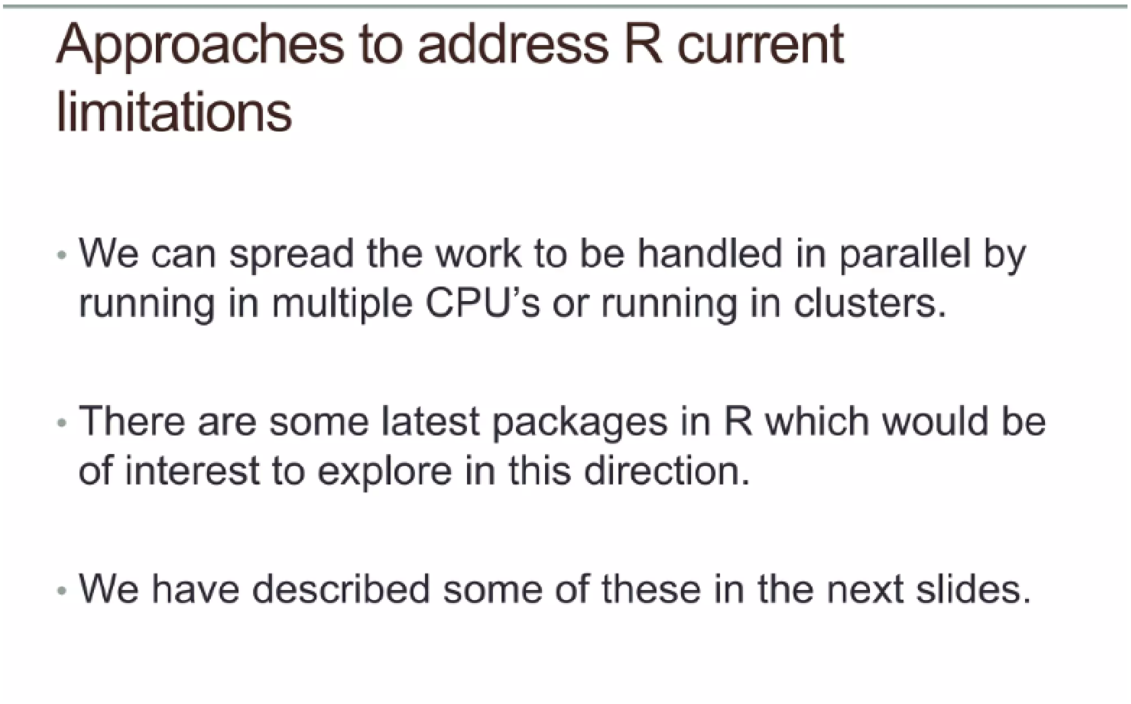
- R is a single threaded programming language with limited memory management capabilities as it uses the system memory.
- In order to use R for handling massive datasets it has to scale up both in processing and memory management capabilities.
- We will look into these aspects and some specific strategies/approaches to circumvent these issues.



Verdict – Provide a chance to R.

- We will suggest some new ways/techniques in R to handle the Big data though lot of these are in research stages and not in production yet.

- However companies and research communities are in high hopes that the current limitations will be addressed and R can be used as a single go to software offering integrated End to End capabilities in this space.

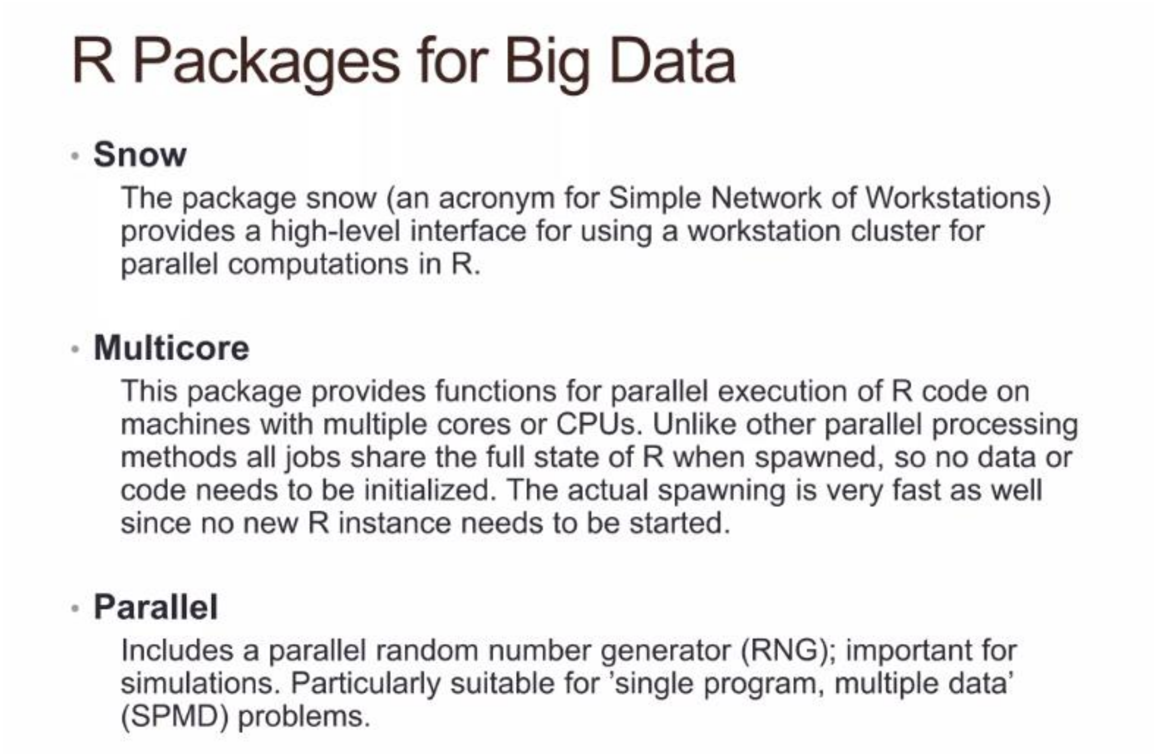


Approaches to address R current limitations

- We can spread the work to be handled in parallel by running in multiple CPU’s or running in clusters.

- There are some latest packages in R which would be of interest to explore in this direction.

- We have described some of these in the next slides.



R Packages for Big Data

- **Snow**

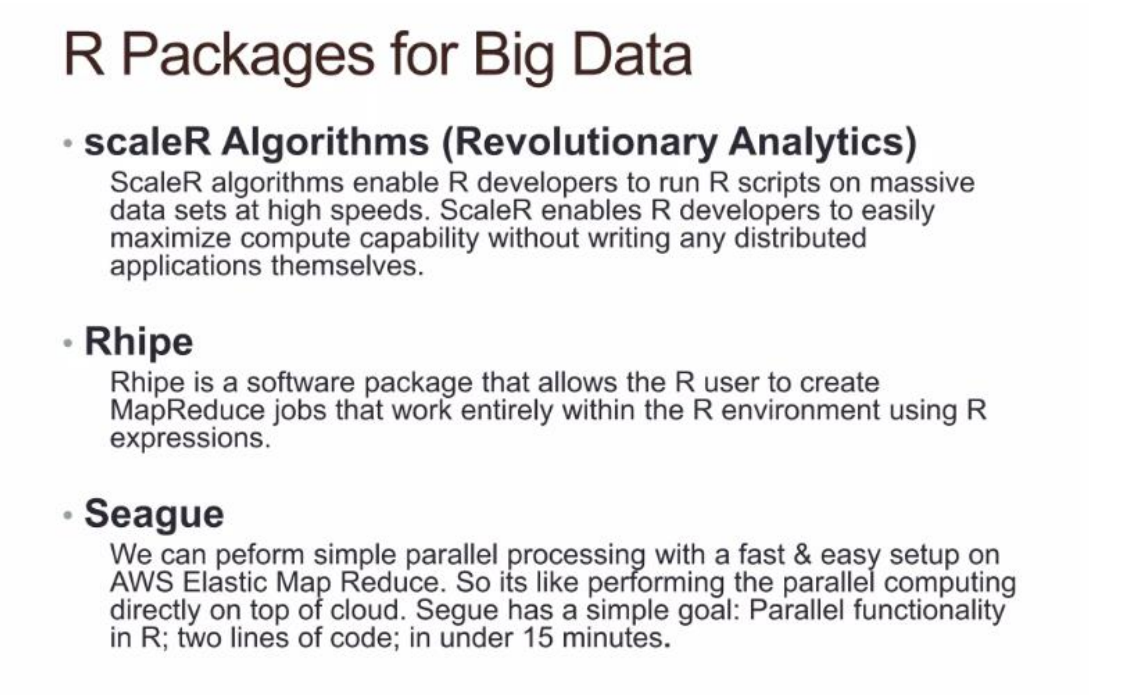
  The package snow (an acronym for Simple Network of Workstations) provides a high-level interface for using a workstation cluster for parallel computations in R.

- **Multicore**

  This package provides functions for parallel execution of R code on machines with multiple cores or CPUs. Unlike other parallel processing methods all jobs share the full state of R when spawned, so no data or code needs to be initialized. The actual spawning is very fast as well since no new R instance needs to be started.

- **Parallel**

  Includes a parallel random number generator (RNG); important for simulations. Particularly suitable for 'single program, multiple data' (SPMD) problems.



R Packages for Big Data

- **scaleR Algorithms (Revolutionary Analytics)**

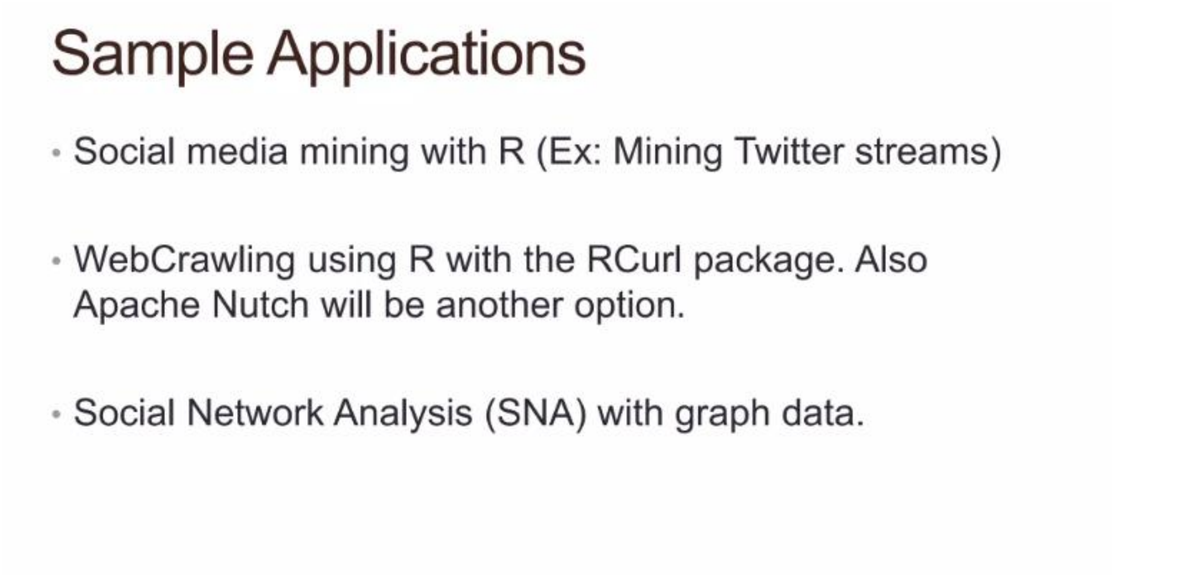
  ScaleR algorithms enable R developers to run R scripts on massive data sets at high speeds. ScaleR enables R developers to easily maximize compute capability without writing any distributed applications themselves.

- **Rhipe**

  Rhipe is a software package that allows the R user to create MapReduce jobs that work entirely within the R environment using R expressions.

- **Seague**

  We can peform simple parallel processing with a fast & easy setup on AWS Elastic Map Reduce. So its like performing the parallel computing directly on top of cloud. Segue has a simple goal: Parallel functionality in R; two lines of code; in under 15 minutes.



Sample Applications

- Social media mining with R (Ex: Mining Twitter streams)

- WebCrawling using R with the RCurl package. Also Apache Nutch will be another option.

- Social Network Analysis (SNA) with graph data.