ABSTRACT

Educational Data Mining (EDM) is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in. New computer-supported interactive learning methods and tools, intelligent tutoring systems, simulations, games have opened up opportunities to collect and analyse student data, to discover patterns and trends in those data, and to make new discoveries and test hypotheses about how students learn. Data collected from online learning systems can be aggregated over large numbers of students and can contain many variables that data mining algorithms can explore for model building. Focus is upon on developing new tools and algorithms for discovering data patterns by applying methods and techniques from statistics, machine learning, and data mining to analyse data collected during teaching and learning. Similarly in this project the analysis has be done based on the employee and alumni data collected from various sources and advanced algorithms have been used to draw patterns and predict suited career to a computer science undergraduate based on his abilities, interests and opportunities. As students are going through their academics and pursuing their interested courses, it is very important for them to assess their capabilities and identify their interests so that they will get to know in which career area their interests and capabilities are going to put them in. This will help them in improving their performance and motivating their interests so that they will be directed towards their targeted career and get settled in that. Also recruiters while recruiting the candidates after assessing them in all different aspects, these kind of career recommender systems help them in deciding in which job role the candidate should be kept in based on his/her performance and other evaluations.

CHAPTER - 1

1.1 INTRODUCTION

Competition in today's society is heavily multiplying day by day. Especially it is too heavy in present day's technical world. So as to compete and reach the goal students need to be planned and organized from initial stages of their education. So it is very important to constantly evaluate their performance, identify their interests and evaluate how close they are to their goal and asses whether they are in the right path that directs towards their targeted. This helps them in improving themselves, motivating themselves to a better career path if their capabilities are not up to the mark to reach their goal and pre evaluate themselves before going to the career peek point.

Not only that recruiters while recruiting people into their companies evaluate candidates on different parameters and draw a final conclusion to select an employee or not and if selected, finds a best suited role and career area for him. There are many types of roles like Database administrator, Business Process Analyst, Developer, Testing Manager, Networks Manager, Data scientist and so on. All these roles require some pre-requisite knowledge in them to be placed in them. So, recruiters analyse these skills, talents and interests and place the candidate in the right job role suited for them. These kind of prediction systems make their recruitment tasks very easy because as the inputs are given, recommendation is done based on inputs.

Already these type of various career recommendation systems and job role recommendation, prediction systems are being used in various private performance

evaluation portals like Co-Cubes, AMCAT. They only take factors like technical abilities and psychometry of students into consideration. These portals asses the students technically and suggest the students and companies job roles suited on their performance. But here various factors including abilities of students in sports, academics and their hobbies, interests, competitions, skills and knowledge are also taken into consideration. Considering all the factors the total number of parameters that were taken into consideration as inputs are 39. And the final job roles are fixed to 15 in number. As the input parameters and final classes of output are large in number typical programming and normal algorithms cannot give the best possible output classification

and prediction. So advanced machine learning algorithms like KNN are used.

Machine Learning is a technique where the machines are trained in such a way that it gains the ability to respond to a particular input or scenario based on the previous inputs it has learnt. Simply it the giving computers the ability to learn by using statistical techniques. Machine learning helps the computers to act without explicitly being programmed. This aims at reducing the human intervention in the machine depend-able problems and scenarios. This helps in solving very com-plex tasks and problems very easily and without involving much human labour. Various applications of machine learning include NLP, classification, prediction, image recognition, medical diagnosis, algorithm building, self-driving cars and much more. In this paper classification and prediction are being done. Let us see what is classification and prediction. Majority of problems in machine learning can be solved using supervised and unsupervised learning. If the final class labels are previously know and all the other data items are to be assigned with one of the available class labels, then it is call supervised. And if the final output classes and sets are not known and it is done by identifying the similarity between data point and their characteristics and finally they are made into groups based on these characteristics then it is called un-supervised. Classification falls under supervised. Input parameters are given and based on their properties a predefined class label is assigned. There are other alternatives like clustering and regression. Based on the type of problem the apt model is chosen.

However here algorithms like KNN areused. After training and testing the data with these we take into consideration the mostaccurate results given algorithm for our further processing. So, initial task done is predicting the output using all algorithms proposed above and later analysing the resultsand there on continued with the most accurate algorithm. So finally, this paper deals with various advanced machine learning

algorithms that involves classification and prediction and are used to improve the accuracy for better prediction, reliability and analysing these algorithms performance.

1.2 CONCEPTS INVOLVED

1.2.1 EDUCATIONAL DATA MINING

Educational data mining (EDM) describes a research field concerned with the application of data mining, machine learning and statistics to information generated from educational settings like universities and intelligent tutoring systems. At a high level, the field seeks to develop and improve methods for exploring this data, which often has multiple levels of meaningful hierarchy, in order to discover new insights about how people learn in the context of such settings. In doing so, EDM has contributed to theories of learning investigated by researchers in educational psychology and the learning sciences. The field is closely tied to that of learning analytics, and the two have been compared and contrasted.

Educational data mining refers to techniques, tools, and research designed for automatically extracting meaning from large repositories of data generated by or related to people's learning activities in educational settings. Quite often, this data is extensive, fine-grained, and precise. For example, several learning management systems (LMSs) track information such as when each student accessed each learning object, how many times they accessed it, and how many minutes the learning object was displayed on the user's computer screen. As another example, intelligent tutoring systems record data every time a learner submits a solution to a problem; they may collect the time of the submission, whether or not the solution matches the expected solution, the amount of time that has passed since the last submission, the order in which solution components were entered into the interface, etc. The precision of this data is such that even a fairly short session with a computer-based learning environment may produce a large amount

of process data for analysis.

In other cases, the data is less fine-grained. For example, a student's university transcript may contain a temporally ordered list of courses taken by the student, the grade that the student earned in each course, and when the student selected or changed his or her academic major. EDM leverages both types of data to discover meaningful information about different types of learners and how they learn, the structure of domain knowledge, and the effect of instructional strategies embedded within various learning environments. These analyses provide new information that would be difficult to discern by looking at the raw data. For example, analysing data

from an LMS may reveal a relationship between the learning objects that a student accessed during the course and their final course grade. Similarly, analysing student transcript data may reveal a relationship between a student's grade in a particular course and their decision to change their academic major. Such information provides insight into the design of learning environments, which allows students, teachers, school administrators, and educational policy makers to make informed decisions about how to interact with, provide, and manage educational resources.

Fig 1: Overview of various Advanced Machine Learning Algorithms Used

1.2.2 ARTIFICIAL INTELLIGENCE

AI (artificial intelligence) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions) and self-correction. Particular applications of AI include expert systems, speech recognition and machine vision.

AI was coined by John McCarthy, an American computer scientist, in 1956 at The Dartmouth Conference where the discipline was born. Today, it is an umbrella term that encompasses everything from robotic process automation to actual robotics. It has gained prominence recently due, in part, to big data, or the increase in speed, size and variety of data businesses are now collecting. AI can perform tasks such as identifying patterns in the data more efficiently than humans, enabling businesses to gain more insight out of their data.

It is a branch of computer science that aims to create intelligent machines. It has become an essential part of the technology industry.

Research associated with artificial intelligence is highly technical and specialized. The core problems of artificial intelligence include programming computers for certain traits such as:

- Knowledge
- Reasoning
- Problem solving
- Perception

- Prediction
- Learning
- Planning
- Ability to manipulate and move objects

Knowledge engineering is a core part of AI research. Machines can often act and react like humans only if they have abundant information relating to the world. Artificial intelligence must have access to objects, categories, properties and relations between all of them to implement knowledge engineering. Initiating common sense, reasoning and problem-solving power in machines is a difficult and tedious task.

Machine learning is also a core part of AI. Learning without any kind of supervision requires an ability to identify patterns in streams of inputs, whereas learning with adequate supervision involves classification and numerical regressions. Classification determines the category an object belongs to and regression deals with obtaining a set of numerical input or output examples, thereby discovering functions enabling the generation of suitable outputs from respective inputs. Mathematical analysis of machine learning algorithms and their performance is a well-defined branch of theoretical computer science often referred to as computational learning theory.

Machine perception deals with the capability to use sensory inputs to deduce the different aspects of the world, while computer vision is the power to analyze visual inputs with a few sub-problems such as facial, object and gesture recognition. Robotics is also a major field related to AI. Robots require intelligence to handle tasks such as object manipulation and navigation, along with sub-problems of localization, motion planning and mapping.

1.2.3 MACHINE LEARNING

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. Intraditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allowfor computers to train on data inputs and use statistical analysis in order to output valuesthat fall within a specific range. Because of this, machine learning facilitates computersin building models from sample data in order to automate decision-making processes based on data inputs.

Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Optical character recognition (OCR) technology converts images of text into movable type. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learning to navigate may soon be available to consumers. Machine learning is a continuously developing field. Because of this, there are some considerations to keep in mind as you work with machine learning methodologies, or analyze the impact of machine learning processes.

Here in this thesis, we are providing basic info of the common machine learning methods of supervised and unsupervised learning, and common algorithmic approaches in machine learning, including the k-nearest neighbour algorithm, KNN learning, and

deep learning.

1.2.3.1 SUPERVISED LEARNING

In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed. Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the

algorithm with no labeled data in order to allow it to find structure within its input data. Let's explore these methods in more detail.

The majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output $Y = f(X)$. The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data. Techniques of Supervised Machine Learning algorithms include linear and logistic regression, multi-class classification, KNNs and support vector machines. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers.For example, a classification algorithm will learn to identify animals after being trainedon a dataset of images that are properly labeled with the species of the animal and someidentifying characteristics. Supervised learning problems can be further grouped into Regression and Classificationproblems. Both problems have as goal the construction ofa succinct model that can predict the value of the dependent attribute from the attributevariables. The difference between the two tasks is the fact that the dependent attribute is numerical for regression and categorical for classification.

1.2.3.1.1 CLASSIFICATION

As the name suggests, Classification is the task of "classifying things" into sub-categories.But, by a machine. If that doesn't sound like much, imagine your computer being able to differentiate between you and a stranger. Between a potato and a tomato. Between an A grade and a F.In Machine Learning and Statistics, Classification is the problem of identifying to which of a set of categories (sub populations), a new

observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

Types of Classification

Classification is of two types:

- **Binary Classification :** When we have to categorize given data into 2 distinct classes. Example – On the basis of given health conditions of a person, we have to determine whether the person has a certain disease or not.
- **Multiclass Classification :** The number of classes is more than 2. For Example

  – On the basis of data about different species of flowers, we have to determine which specie does our observation belong t**o**

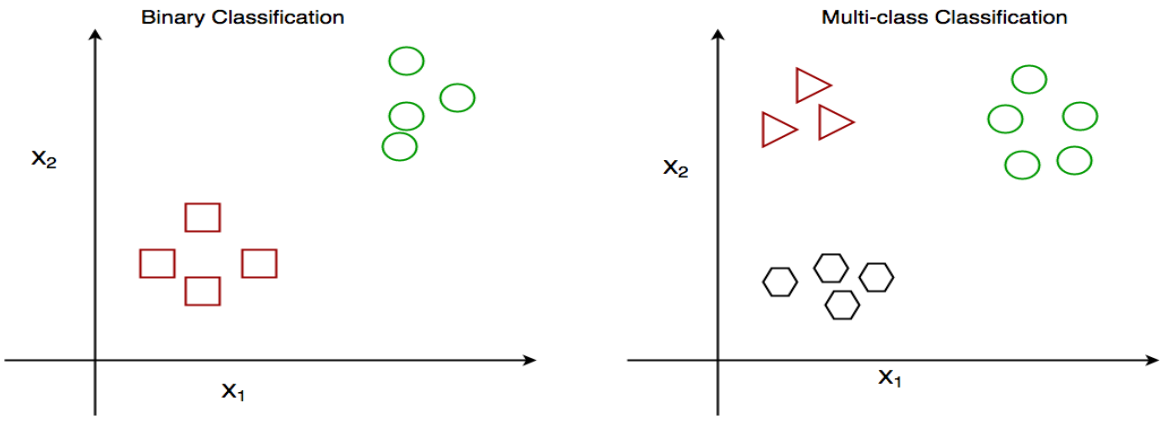


Fig 2 : Binary and Multiclass Classification. Here x1 and x2 are our variables upon which the class is predicted.

Suppose we have to predict whether a given patient has a certain disease or not, on the basis of 3 variables, called features.

Which means there are two possible outcomes:

1. The patient has the said disease. Basically a result labelled “Yes” or “True”.
2. The patient is disease free. A result labelled “No” or “False”.

This is a binary classification problem.We have a set of observations called training data set, which comprises of sample data with actual classification results. We train a model, called Classifier on this data set, and use that model to predict whether a certain patient will have the disease or not.

The outcome, thus now depends upon :

1. How well these features are able to “map” to the outcome.
2. The quality of our data set. By quality I refer to statistical and Mathematical qualities.
3. How well our Classifier generalizes this relationship between the features and the outcome.
4. The values of the x1 and x2.

Following is the generalized block diagram of the classification task.

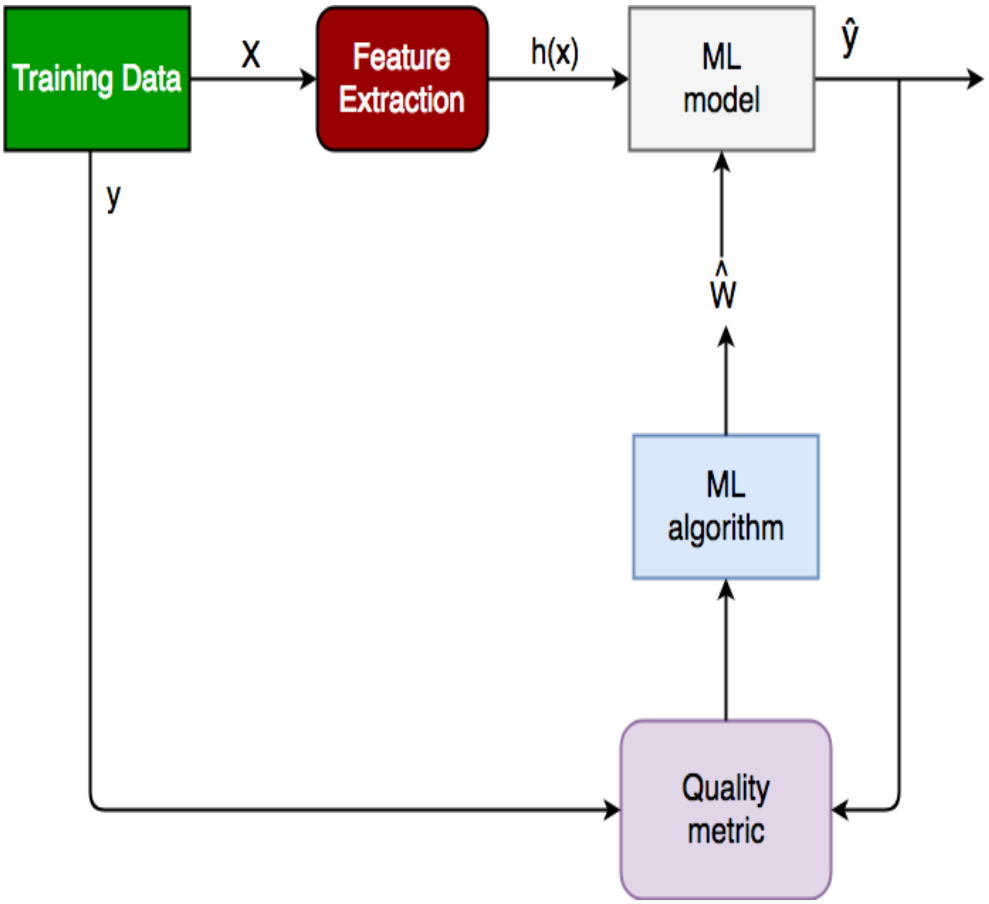


Fig 3: Generalized Classification Block Diagram.

1. X : pre-classified data, in the form of a N*M matrix. N is the no. of observations and M is the number of features
2. y : An N-d vector corresponding to predicted classes for each of the N observations.
3. Feature Extraction : Extracting valuable information from input X using a series of transforms.
4. ML Model : The “Classifier” we’ll train.
5. y’ : Labels predicted by the Classifier.
6. Quality Metric : Metric used for measuring the performance of the model.
7. ML Algorithm : The algorithm that is used to update weights w’, which update the model and “learns” iteratively.

Types of Classifiers (Algorithms)

There are various types of classifiers. Some of them are :

- Linear Classifiers : Logistic Regression
- Tree Based Classifiers : KNN Classifier
- Support Vector Machines
- Artificial Neural Networks
- Bayesian Regression
- Gaussian Naive Bayes Classifiers

- Stochastic Gradient Descent (SGD) Classifier
- Ensemble Methods : Random Forests, AdaBoost, Bagging Classifier, Voting Classifier, ExtraTrees Classifier

Practical Applications of Classification

- Google's self driving car uses deep learning enabled classification techniques which enables it to detect and classify obstacles.
- Spam E-mail filtering is one of the most widespread and well recognized uses of Classification techniques.
- Detecting Health Problems, Facial Recognition, Speech Recognition, Object Detection, Sentiment Analysis all use Classification at their core.

1.2.3.1.2 REGRESSION

A regression problem is when the output variable is a real or continuous value, such as "salary" or "weight". Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

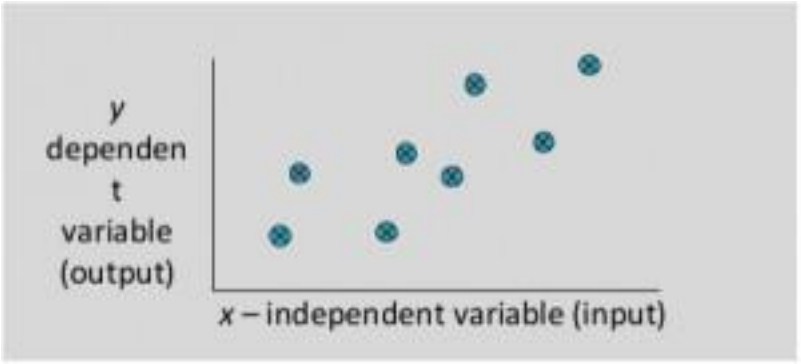


Fig 4: Linear Regression Model

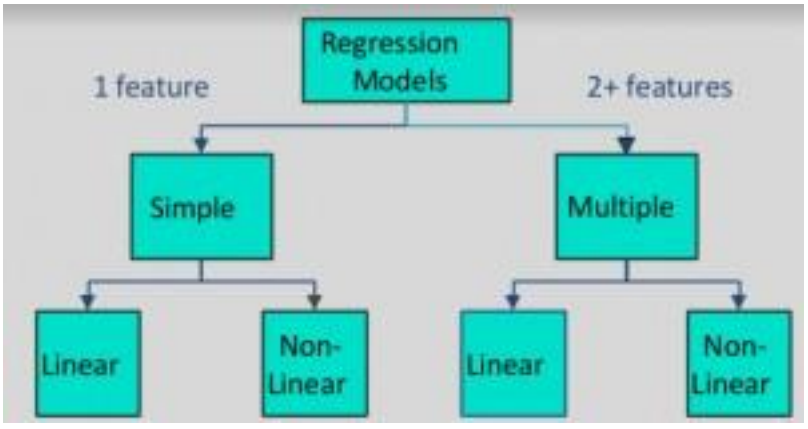


Fig 5: Types Of Regression Models

1.2.3.2 UNSUPERVISED LEARNING

Unsupervised learning is where we only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to

discover and present the interesting structure in the data. Unsupervised learning problems can be further grouped into clustering and association problems.

1.2.3.2.1 CLUSTERING

It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For example, the data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.

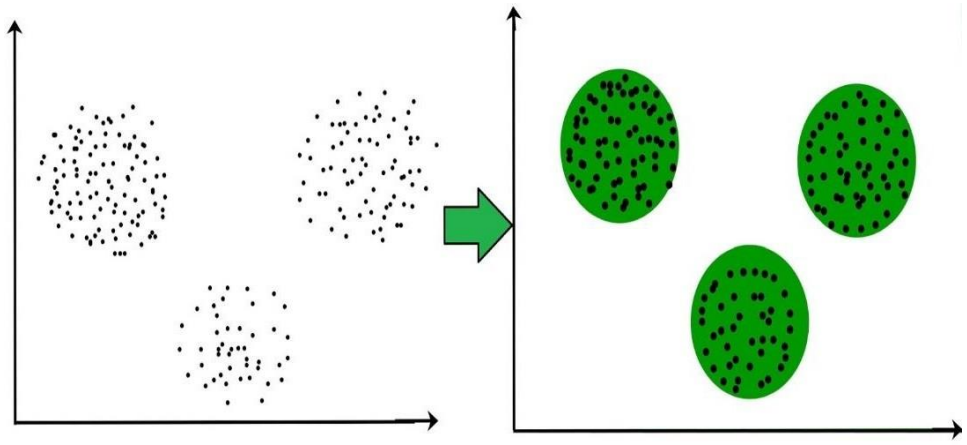


Fig 6: Clustering overview

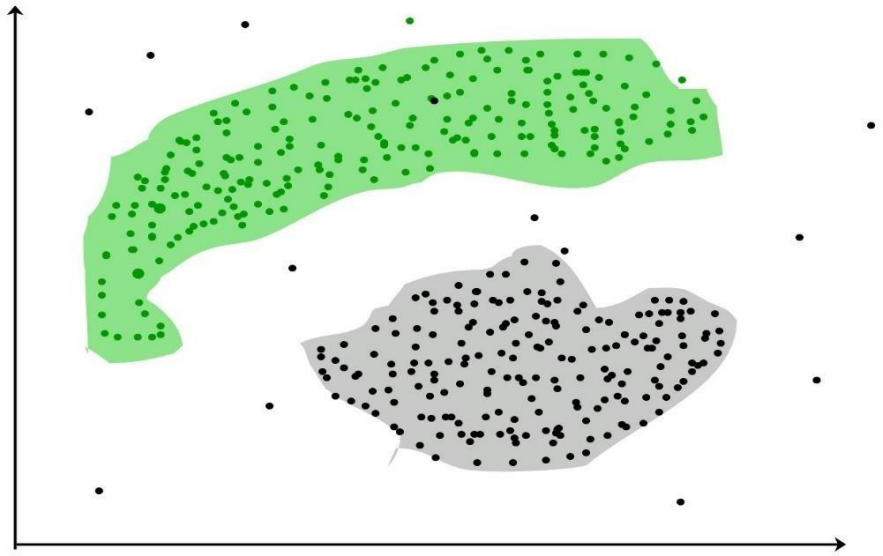


Fig 7: Clustering Example

These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for calculation of the outliers.

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

Clustering Methods :

1. Density-Based Methods : These methods consider the clusters as the dense region

having some similarity and different from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters.Example *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise) , OPTICS (Ordering Points to Identify Clustering Structure) etc.

2. Hierarchical Based Methods : The clusters formed in this method forms a tree type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category

- Agglomerative (bottom up approach)
- Divisive (top down approach) .

3. Partitioning Methods **:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example K-means, CLARANS (Clustering Large Applications based upon randomized Search) etc.

4. Grid-based Methods **:** In this method the data space are formulated into a finite number of cells that form a grid-like structure. All the clustering operation done on these grids are fast and independent of the number of data objects example STING (Statistical Information Grid), wave cluster, CLIQUE (CLustering In Quest) etc.

Clustering Algorithms :

- K-Means Clustering.
- Mean-Shift Clustering for a single sliding window.
- The entire process of Mean-Shift Clustering.
- DBSCAN Smiley Face Clustering.

- EM Clustering using GMMs.

- Agglomerative Hierarchical Clustering.

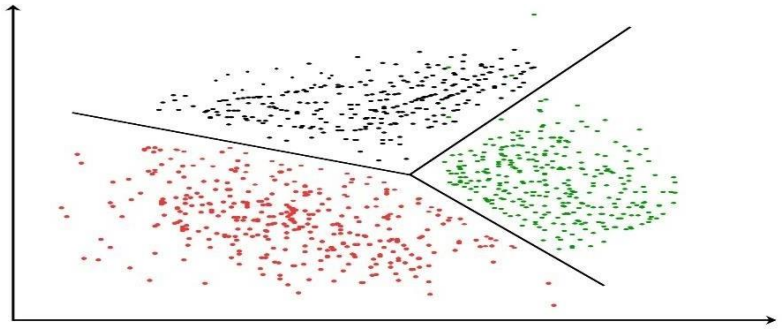


Fig 8: k-means clustering

1.2.4 ALGORITHMS USED

1.2.4.1 KNNS

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression).

Methods like KNNs, random forest, gradient boosting are being popularly usedin all kinds of data science problems. Hence, for every analyst it's important to learn these algorithms and use them for modelling. KNN is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiatorin input variables.

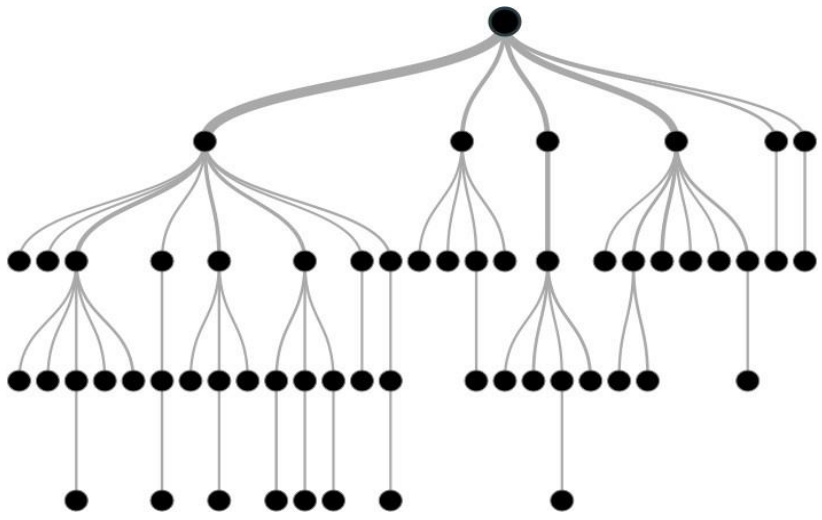


Fig 9: A Typical KNN Structure

For instance, let's say we have a sample of 30 students with three variables Gender (Boy/ Girl), Class(IX/ X) and Height (5 to 6 ft). 15 out of these 30 play cricket in

leisure time. Now, I want to create a model to predict who will play cricket during leisure period? In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

This is where KNN helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other). In the snapshot below, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.

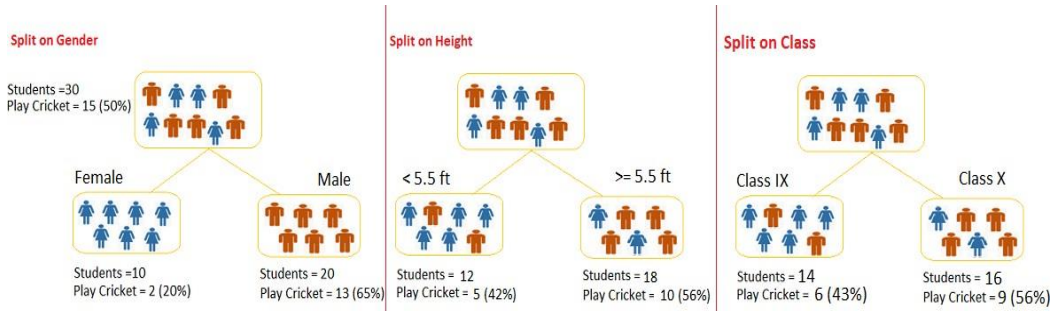


Fig 10: Example of a Tree classification

As mentioned above, KNN identifies the most significant variable and it's valuethat gives best homogeneous sets of population. Now the question which arises is, howdoes it identify the variable and the split? To do this, KNN uses various algorithms, which we will shall discuss in the following section.

Types of KNNs:

Types of KNN is based on the type of target variable we have. It can be of twotypes:

1. Categorical Variable KNN**:** KNN which has categorical targetvariable then it called as categorical variable KNN. Example:- In abovescenario of student problem, where the target variable was “Student will play cricket or not” i.e. YES or NO.
2. Continuous Variable KNN**:** KNN has continuous target variable then it is called as Continuous Variable KNN.

For instance, let's say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that income of customer is a significant variable but insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a KNN to predict customer income based on occupation, product and various other variables. In this case, we are predicting values for continuous variable.

Important Terminology related to KNNs:

Here is the info about the basic terminology used with KNNs:

- Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.
- Leaf/ Terminal Node: Nodes do not split is called Leaf or Terminal node.

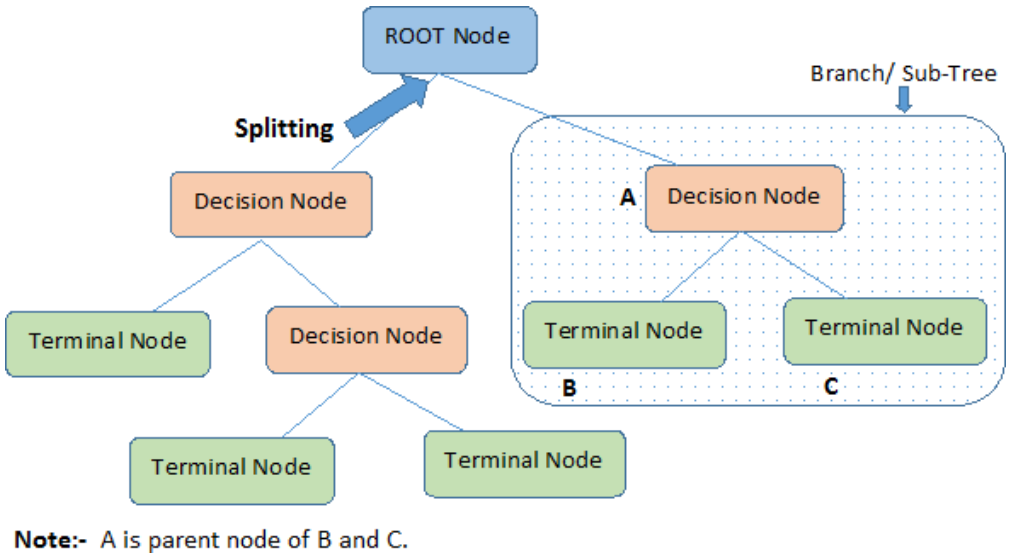


Fig 11: Process undergone in a KNN

- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- Branch / Sub-Tree: A sub section of entire tree is called branch or sub-tree.

- Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

Advantages

- Easy to Understand: KNN output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.
- Useful in Data exploration**:** KNN is one of the fastest way to identify most significant variables and relation between two or more variables. With thehelp of KNNs, we can create new variables / features that has better power to predict target variable. It can also be used in data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there KNN will help to identify most significantvariable.
- Less data cleaning required**:** It requires less data cleaning compared to some other modelling techniques. It is not influenced by outliers and missing values to a fair degree.
- Data type is not a constraint: It can handle both numerical and categorical variables.
- Non Parametric Method: KNN is considered to be a non-parametric method. This means that KNNs have no assumptions about the space distribution and the classifier structure.

Now, Let us know how tree knows where to split the branch.The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria is different for classification and regression trees.KNNs use multiple algorithms to decide to split a

node in two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the nodeincreases with respect to the target variable. KNN splits the nodes on all available variables and then selects the split which results in most homogeneous sub- nodes.The algorithm selection is also based on type of target variables. Let's look at thefour most commonly used algorithms in KNN:

Gini Index

Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.

1. It works with categorical target variable “Success” or “Failure”.
2. It performs only Binary splits
3. Higher the value of Gini higher the homogeneity.
4. CART (Classification and Regression Tree) uses Gini method to create binary splits.

Steps to Calculate Gini for a split

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2).
2. Calculate Gini for split using weighted Gini score of each node of that split

Referring to example used above, where we want to segregate the students based on target variable (playing cricket or not). In the snapshot below, we split the population using two input variables Gender and Class. Now, I want to identify which split is producing more homogeneous sub-nodes using Gini index.

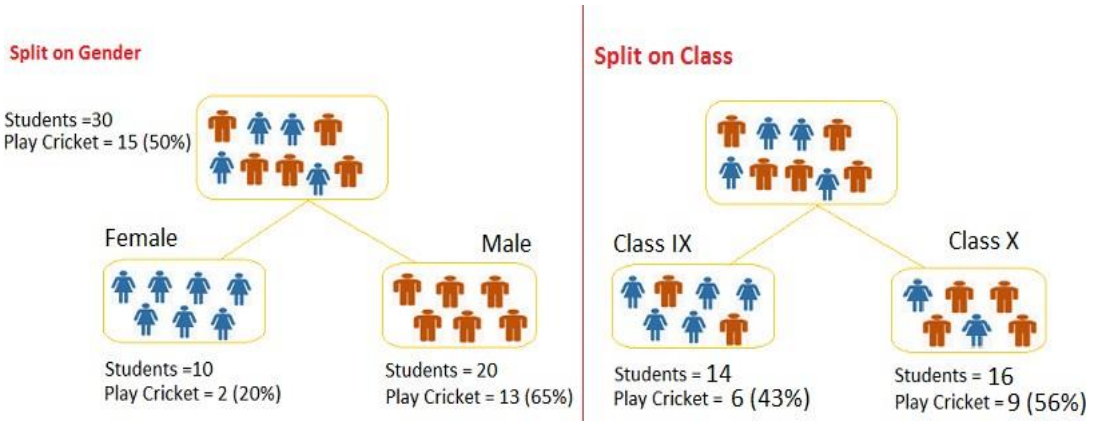


Fig 12: Example for tree classification

Split on Gender:

1. Calculate, Gini for sub-node Female = (0.2)*(0.2)+(0.8)*(0.8)=0.68
2. Gini for sub-node Male = (0.65)*(0.65)+(0.35)*(0.35)=0.55
3. Calculate weighted Gini for Split Gender = (10/30)*0.68+(20/30)*0.55 = 0.59

Similar for Split on Class:

1. Gini for sub-node Class IX = (0.43)*(0.43)+(0.57)*(0.57)=0.51

2. Gini for sub-node Class X = (0.56)*(0.56)+(0.44)*(0.44)=0.51

3. Calculate weighted Gini for Split Class = (14/30)*0.51+(16/30)*0.51 = 0.51

Above, you can see that Gini score for *Split on Gender* is higher than *Split on Class*, hence, the node split will take place on Gender.

Chi-Square

It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node. We measure it by sum of squares of standardized differences between observed and expected frequencies of target variable.

1. It works with categorical target variable "Success" or "Failure".
2. It can perform two or more splits.
3. Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.
4. Chi-Square of each node is calculated using formula,
5. Chi-square = ((Actual – Expected)^2 / Expected)^1/2
6. It generates tree called CHAID (Chi-square Automatic Interaction Detector)

Steps to Calculate Chi-square for a split:

1. Calculate Chi-square for individual node by calculating the deviation for Success and Failure both

2. Calculated Chi-square of Split using Sum of all Chi-square of success and Failure of each node of the split

Now, Let's work with above example that we have used to calculate Gini.

Split on Gender:

1. First we are populating for node Female, Populate the actual value for "Play Cricket" and "Not Play Cricket", here these are 2 and 8 respectively.
2. Calculate expected value for "Play Cricket" and "Not Play Cricket", here it would be 5 for both because parent node has probability of 50% and we have applied same probability on Female count(10).

3. Calculate deviations by using formula, Actual – Expected. It is for "Play Cricket" (2 – 5 = -3) and for "Not play cricket" (8 – 5 = 3).
4. Calculate Chi-square of node for "Play Cricket" and "Not Play Cricket" using formula with formula, = ((Actual – Expected)^2 / Expected)^1/2. You can refer below table for calculation.
5. Follow similar steps for calculating Chi-square value for Male node.
6. Now add all Chi-square values to calculate Chi-square for split Gender.

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
Female	2	8	10	5	5	-3	3	1.34	1.34
Male	13	7	20	10	10	3	-3	0.95	0.95
							Total Chi-Square	4.58	

Table 1: Explaining example for Gini Index based tree classification

Split on Class:

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
IX	6	8	14	7	7	-1	1	0.38	0.38
X	9	7	16	8	8	1	-1	0.35	0.35
							Total Chi-Square	1.46	

Table 2: Explaining split on class classification tree

Performing similar steps of calculation for split on Class and we will come up with below table. Above, we can see that Chi-square also identify the Gender split is more significant compare to Class.

Information Gain:

We can easily say by looking at below figure which node can describe further process

easily, that is C because it requires less information as all values are similar. On the other hand, B requires more information to describe it and A requires the maximum information. In other words, we can say that C is a Pure node, B is less Impure and A is more impure.

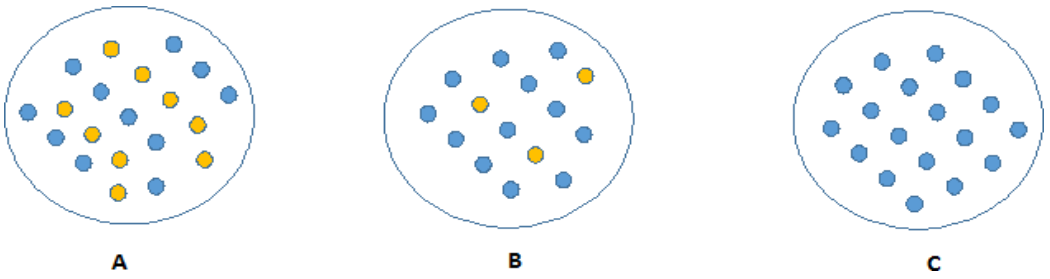


Fig 13: Example of best classification

Now, we can build a conclusion that less impure node requires less information to describe it. And, more impure node requires more information. Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% – 50%), it has entropy of one.

Entropy can be calculated using formula

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

Here p and q is probability of success and failure respectively in that node. Entropy is also used with categorical target variable. It chooses the split which has lowest entropy compared to parent node and other splits. The lesser the entropy, the better it is.

Steps to calculate entropy for a split:

1. Calculate entropy of parent node
2. Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

Example: Let's use this method to identify best split for student example.

1. Entropy for parent node = -(15/30) log2 (15/30) – (15/30) log2 (15/30) = 1. Here 1 shows that it is a impure node.
2. Entropy for Female node = -(2/10) log2 (2/10) – (8/10) log2 (8/10) = 0.72 and for male node, -(13/20) log2 (13/20) – (7/20) log2 (7/20) = 0.93

3. Entropy for split Gender = Weighted entropy of sub-nodes = (10/30)*0.72 + (20/30)*0.93 = 0.86
4. Entropy for Class IX node, -(6/14) log2 (6/14) – (8/14) log2 (8/14) = 0.99 and for Class X node, -(9/16) log2 (9/16) – (7/16) log2 (7/16) = 0.99.
5. Entropy for split Class = (14/30)*0.99 + (16/30)*0.99 = 0.99

Above, we can see that entropy for Split on Gender is the lowest among all, so the tree will split on gender. We can derive information gain from entropy as 1- Entropy.

1.2.4.2 SUPPORT VECTOR MACHINE:

Support Vector Machine (KNN) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well .

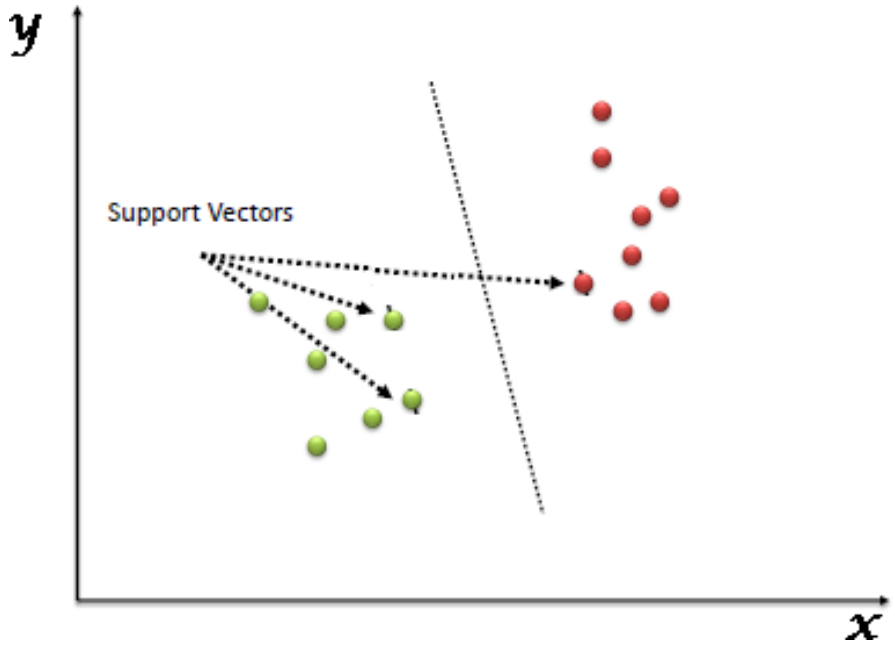


Fig 14: Support vector machine

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

Working of an KNN:

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

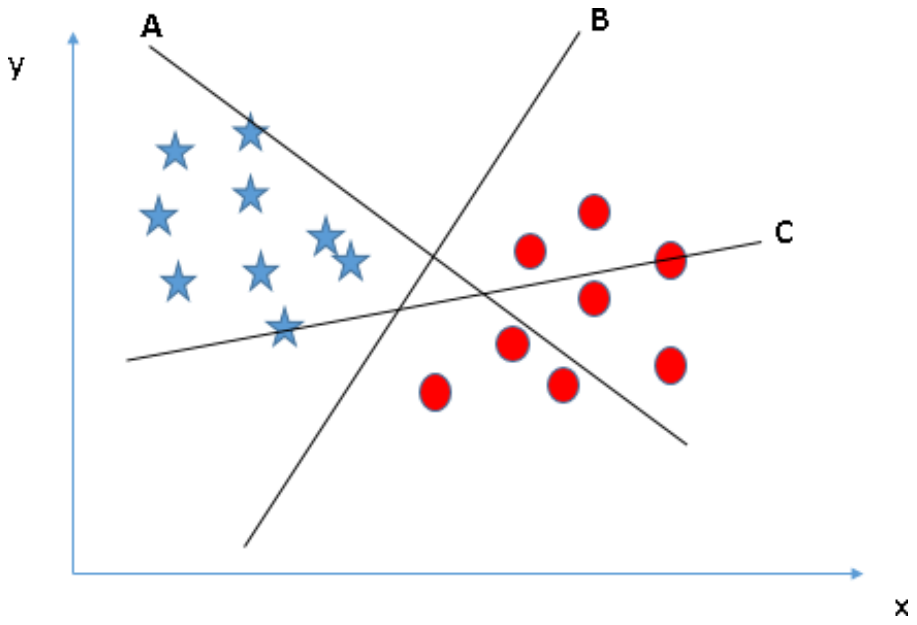
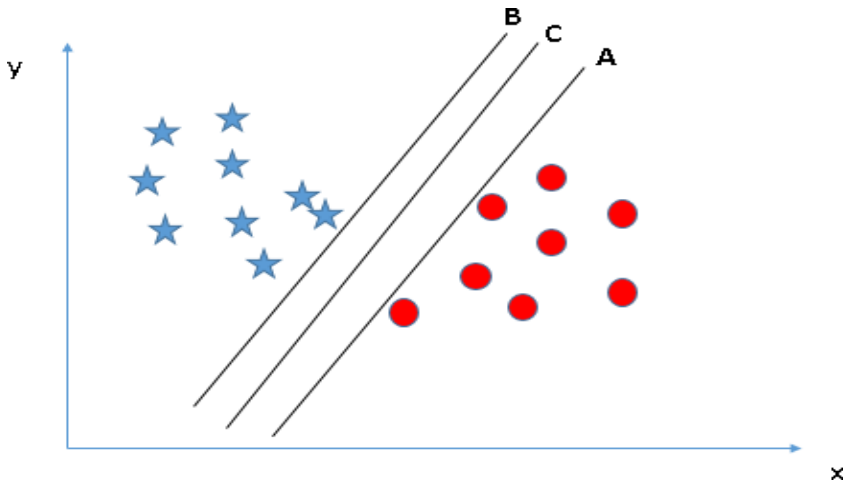


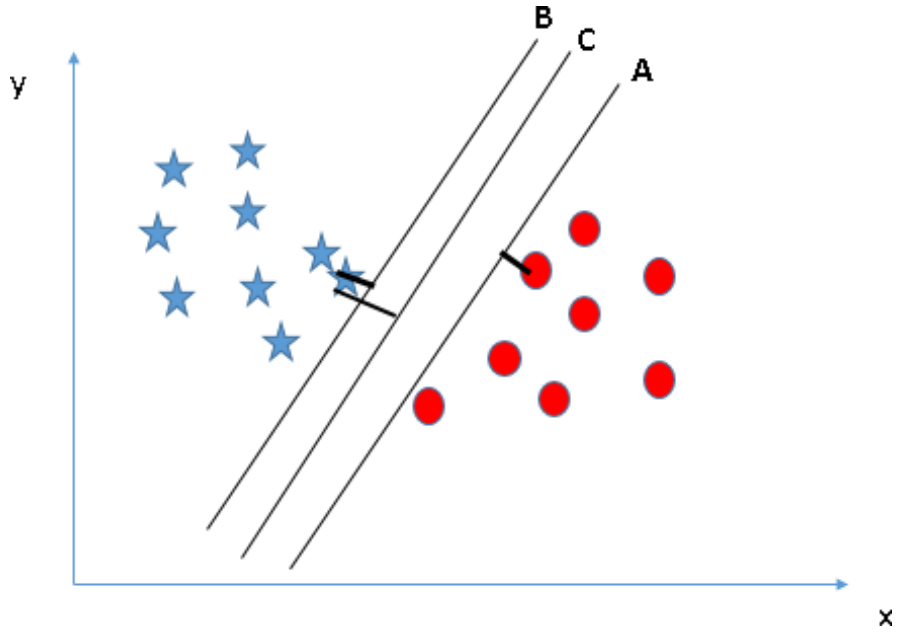
Fig 15: Steps involved in Working of an KNN

In this scenario, hyper-plane "B" has excellently performed this job.

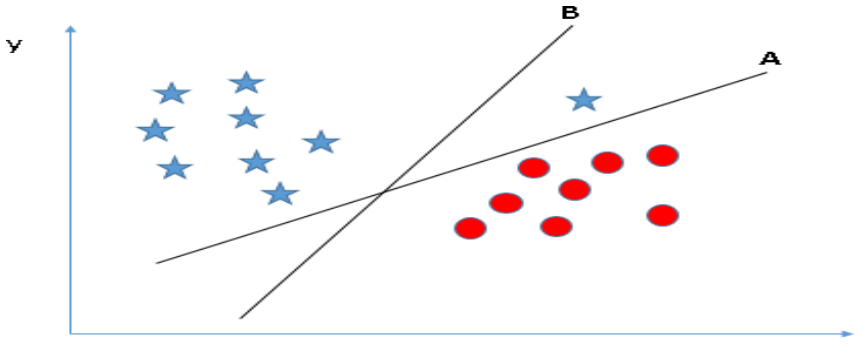
- Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?


B
C
A
y
x

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



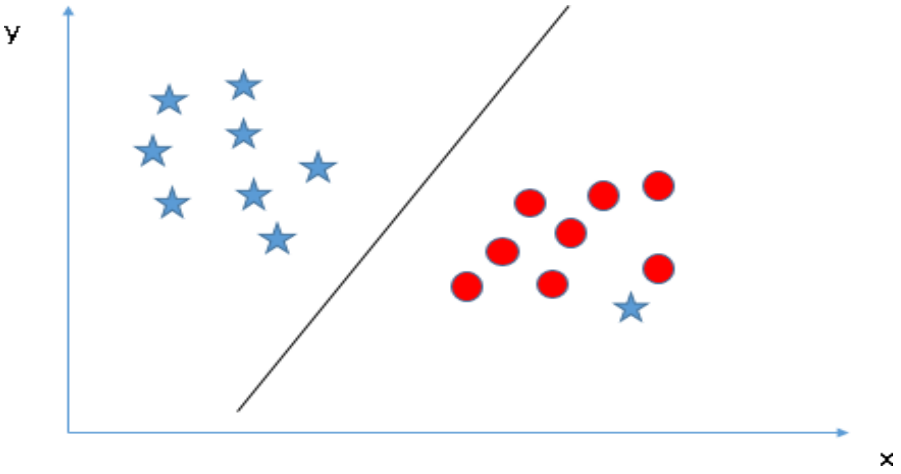
- Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.
- Identify the right hyper-plane (Scenario-3)**:**Use the rules as discussed in previous section to identify the right hyper-plane



Some

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

- Can we classify two classes (Scenario-4): Below, I am unable to segregate the twoclasses using a straight line, as one of star lies in the territory of other(circle) classas an outlier.
- As I have already mentioned, one star at other end is like an outlier for star class. KNN has a feature to ignore outliers and find the hyper-plane that has maximum margin. Hence, we can say, KNN is robust to outliers.



- Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, wecan't have linear hyper-plane between the two classes, so how does KNN classify these two classes? Till now, we have only looked at the linear hyper-plane.
- KNN can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot thedata points on axis x and z:

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

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

1.2.4.3 ONE HOT ENCODING:

OneHot Encoding is a technique by which categorial values present in the data collected are converted into numerical or other ordinal format so that they can be provided to machine learning algorithms and get better results of prediction. Simply OneHot encoding transforms categorial values into a form that best fits as input to feed to various machine learning algorithms. This algorithm works fine with almost all machine learning algorithms. Few algorithms like KNNhandle categorical values very well. In such cases OneHot encoding is not required.

Process of OneHot encoding may seem difficult but most modern day machine learning algorithms take care of that. The process is easily explained here: For example in a data if there are values like yes and no., integer encoder assigns values to them like 1 and 0.This process can be followed as long as we continue the fixed values for yes as 1 and no as 0. As long as we assign or allocate these fixed numbers to these particular labels this is called as integer encoding. But here consistency is very important because if we

invert the encoding later, we should get back the labels correctly from those integer values especially in the case of prediction. Next step is creating a vector for each integer value. Let us suppose this vector is binary and has a length of 2 for the two possible integer values. The ‘yes’ label encoded as 1 will then be represented with vector [1,1] where the zeroth index is given the value 1.Similarly ‘no’ label encoded as ‘0’ will be represented like [0,0] which represents the first index is represented with value 0.

For example [pillow, rat, fight, rat] becomes [0,1,2,1]. This is here imparting an ordinal property to the variable , i.e. pillow < ret < fight. As this is ordinally characteristic and is usually not required and desired and so OneHot encoding is re-quired for correct representation of distinct elements of a variable. It makes representation of categorial variables to be more expressive.

1.2.4.4 KNN

KNNis an open-source software library which provides the gradient boosting framework for C++, Java, Python, R, and Julia. It works on Linux, Windows, and macOS. From the project description, it aims to provide a "Scalable, Portable and Distributed Gradient Boosting (GBM, GBRT, GBDT) Library". Other than running on a single machine, it also supports the distributed processing frameworks Apache Hadoop, Apache Spark, and Apache Flink. It has gained much popularity and attention recently as it was the algorithm of choice for many winning teams of a number of machine learning competitions.

KNNinitially started as a research project by Tianqi Chen as part of the Distributed (Deep) Machine Learning Community (DMLC) group. Initially, it began as a terminal application which could be configured using a libKNN configuration file. After winning the Higgs Machine Learning Challenge, it became well known in the ML competition circles. Soon after, the Python and R packages were built and now it has packages for many other languages like Julia, Scala, Java, etc. This brought the library to more developers and became popular among the Kaggle community where it has been used for a large number of competitions.

It soon became used with multiple other packages making it easier to use in the respective communities. It now has integrations with scikit-learn for Pythonusers, and also with the caret package for R users. It can also be integrated into Data Flow frameworks like Apache Spark, Apache Hadoop, and Apache Flink using the abstracted Rabit and XGBoost4J. The working of KNNhas also been published byTianqi Chen and Carlos Guestrin.

KNNdenotes eXtreme Gradient Boosting. KNNis implementation of gradientboosting algorithms. It is available in many forms like tool, library et cetera. It mainly

focus-es on model performance and computational time. It greatly reduces the time and greatly lifts the performance of the model. It's implementation has the features of scikit-learn and R implementations and also have a newly added features like regularization. Regularized gradient boosting means gradient boosting with both L1 and L2 type regularizations. The main best features that the implementation of the algorithm provides are: Automatic handling of missing values with sparse aware implementation, and it provides block structure to promote parallel construction of tree and continued training which supports further boost an already fitted model on the fresh data. Gradient boosting is a technique where new models are made that can predict the errors or remains of previous models and then added together to make the final prediction. they use gradient descendent algorithms to reduce loss during adding of new models. They support both classification and regression type of challenges. In the training part generally an objective function is defined. Define an objective function and try to optimize it.

obj=i=1 Σl(yi,yi^(t))+ i=1 Σ Ω(fi)

CHAPTER – 2

2.1 SOFTWARES USED

2.1.1 PIP

PIP is a free and open source distribution of the Python and R programminglanguages for data science and machine learning related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. PIP Distribution is used by over 6 million users, andit includes more than 250 popular data science packages suitable for Windows, Linux,and MacOS.

Python is a high-level programming language devised by Guido van Rossum & first released in 1991. It's the most popular coding language used by software developers to build, control, manage and for testing. It is also an interpreter which executes Python programs. The python interpreter is called python.exe on Windows.

Python Packages

Packages or additional libraries help in scientific computing and computational modelling. In Python, the packages are not the part of the Python standard library. Few major packages are –

- numpy (NUMeric Python): matrices and linear algebra
- scipy (SCIentific Python): many numerical routines

- matplotlib: (PLOTting LIBrary) creating plots of data
- sympy (SYMbolic Python): symbolic computation
- pytest (Python TESTing): a code testing framework

Together with a list of Python packages, tools like editors, Python distributions include the Python interpreter. PIP is one of several Python distributions. PIP is a new distribution of the Python and R data science package. It wasformerly known as Continuum Analytics. PIP has more than 100 new packages.

This work environment, PIP is used for scientific computing, data_science, statistical analysis, and machine learning. The latest version of PIP 5.0.1 isreleased in October 2017.

The released version 5.0.1 addresses some minor bugs and adds useful features, such as updated R language support. All of these features weren't available in the original 5.0.0 release.

This package manager is also an environment manager, a Python distribution, and a collection of open source packages and contains more than 1000 R and Python Data Science Packages.

PIP can help with

- Installing Python on multiple platforms
    - Separating out different environment
- Dealing with not having correct privileges and
- Getting up and running with specific packages and libraries

2.1.2 IPYTHON NOTEBOOKS

IPython is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history. IPython provides the following features:

- Interactive shells (terminal and Qt-based).
- A browser-based notebook interface with support for code, text, mathematical expressions, inline plots and other media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into one's own projects.
- Tools for parallel computing.

Parallel computing in Ipython:

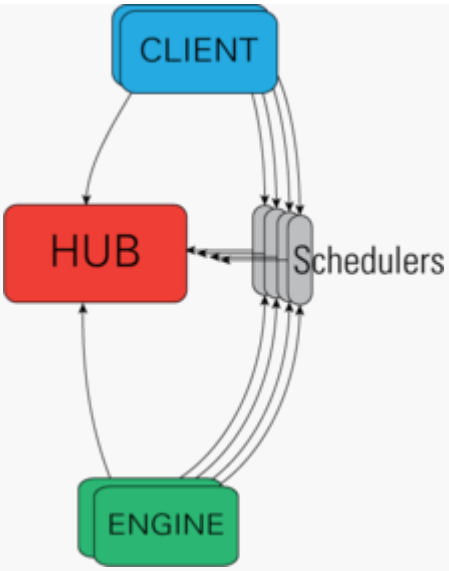


Fig 16: Architectural View of IPython's parallel machinery

IPython is based on an architecture that provides parallel and distributed computing.

IPython enables parallel applications to be developed, executed, debugged and monitored interactively. Hence, the I (Interactive) in IPython.[3]This architecture abstracts out parallelism, which enables IPython to support many different styles of parallelism[4]including:

- Single program, multiple data (SPMD) parallelism
- Multiple program, multiple data (MIMD) parallelism
- Message passing using MPI

- Task parallelism
- Data parallelism
- Combinations of these approaches
- Custom user defined approaches

With the release of IPython 4.0, the parallel computing capabilities have been made optional and released under the ipyparallel python package.

IPython frequently draw from SciPy stack[5] libraries like NumPy and SciPy, often installed alongside from one of many Scientific Python distributions. IPython provide

integration some library of the SciPy stack like matplotlib, like inline graph when in used with the COLAB notebook. Python libraries can implement IPython specific hooks to customize object Rich object display. SymPy for example implement rendering of Mathematical Expression as rendered LaTeX when used within IPython context.

Other features:

IPython also allows non-blocking interaction with Tkinter, PyGTK, PyQt/PySide and wxPython (the standard Python shell only allows interaction with Tkinter). IPython can interactively manage parallel computing clusters using asynchronous status call-backs and/or MPI. IPython can also be used as a system shell replacement. Its default behaviour is largely similar to Unix shells, but it allows customization and the flexibility of executing code in a live Python environment. Using IPython as a shell replacement is less common and it is now recommended to use Xonsh which provide most of the IPython feature with better shell integrations.

2.1.3 COLAB NOTEBOOK

It is a web based interface that allows for rapid prototyping and sharing of data-related projects. It works with many kernels (this is the name given to the code env that it can run), including but not limited to Python and R (even though its more famous and suited for Python).Previously it used to be called IPython Notebook but has beenrenamed and moved to the COLAB project. COLAB is an open source project aiming atcreating a better work experience for (data) scientists.

Unique features:

- Supports all imports and exports
- Great for sharing and collaborative work
- Supports variety of data types within the same window such as text, code, graphs, videos, pictures
- Great for Visualizations
- Enables parallel computing
- Presentation feature in the COLAB and ipython notebook where you can makethe presentation directly from your notebook. One such extension is “RISE”.

2.1.4 SPYDER

Spyder (formerly Pydee) is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates NumPy, SciPy, Matplotlib and IPython, as well as other open source software.[4][5] It is released under the MIT license.

Spyder is extensible with plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylinth and Rope. It is available cross-platform through PIP, on Windows with WinPython and Python (x,y), on macOSthrough MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder makes use of Qt either through the binding PyQt or PySide. This flexibility is reached through a small abstraction layer called QtPy.

Features include:

- editor with syntax highlighting and introspection for code completion
- support for multiple Python consoles (including IPython)
- the ability to explore and edit variables from a GUI

- M5Rules (M5' algorithm presented in terms of mathematical function without a tree)
- DecisionStump (same as M5' but with a single number output in each node)

- M5P (splitting domain into successive binary regions and then fit linear models to each tree node)
- RandomForest (several model trees combined)
- RepTree (several model trees combined)
- ZeroR (the average value of outputs)
- DecisionRules (splits data into several regions based on a single independent variable and provides a single output value for each range)
- LinearRegression
- SMOreg (support vector regression)
- SimpleLinearRegression (uses an intercept and only 1 input variable for multivariate data)
- MultiLayerPerceptron (neural network)
- GaussianProcesses

CHAPTER – 3

3.1 INSTALLATION PROCEDURES

SOFTWARE DESCRIPTION:

PYTHON 3.7:

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object- oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in manya reason most platforms and

may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off- line as well. For a

description of standard objects and modules, see library-index. Reference-index gives a more formal definition of the language. To write extensions in C or C++, read extending-index and c-api-index. There are also several books covering Python in depth. This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most notes worthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in library-index. If you do much work on computers, eventually you find that there's some task you'd like

to automate. For example, you may wish to perform a search-and-replace over a large number of text files, or rename and rearrange a bunch of photo files in a complicated way. Perhaps you'd like to write a small custom database, or a specialized

GUI application or a simple game. If you're a professional software developer, you may have to work with several C/C++/Java libraries but find the usual write/compile/test/re-compile cycle is too slow. Perhaps you're writing a test suite for such a library and find writing the testing code a tedious task. Or maybe you've written a program that could use an extension language, and you don't want to design and implement a whole new language for your application.

Typing an end-of-file character (Control-D on Unix, Control-Z on Windows) at the primary prompt causes the interpreter to exit with a zero exit status. If that doesn't work, you can exit the interpreter by typing the following command: quit(). The interpreter's line-editing features include interactive editing, history substitution and code completion on systems that support read line. Perhaps the quickest check to see whether command line editing is supported is typing Control-P to the first Python prompt you get. If it beeps, you have command line editing; see Appendix Interactive Input Editing and History Substitution for an introduction to the keys. Ifnothing appears to happen, or if ^P is echoed, command line editing isn't available; you'll only be able to use backspace to remove characters from the current line. The interpreter operates somewhat like the Unix shell: when called with standard input connected to a tty device, it reads and executes commands interactively; when called with a file name argument or with a file as standard input, it reads and executes a script from that file. A second way of starting the interpreter is python -c command [arg] ..., which executes the statement(s) in command, analogous to the shell's -c option. Since Python statements often contain spaces or

other characters that are special to the shell, it is usually advised to quote commands in its entiretywithsinglequotes.SomePythonmodulesarealsousefulasscripts. These can be invoked using python-m module [arg]...,which executes the source file for the module as if you had spelled out its full name on the command line. When a script file is used, it is sometimes useful to be able to run the script and enter interactive mode afterwards. This can be done by passing -i before the script.

There are tools which use doc strings to automatically produce online or printed documentation or to let the user interactively browse through code; it's good practice to include doc strings in code that you write, so make a habit of it. The execution of a function introduces a new symbol table usedfor the local variables of the function. More precisely, all variable assignments in a functions to read the value in the local symbol table; whereas variable references first look in the local symbol table, then in the local symbol tables of enclosing functions, then in the global symbol table, and finally in the table of built-in names. Thus, global variables cannot be directly assigned a value within a function (unless named in a global statement), although they may be referenced. The actual parameters (arguments) to a function call are introduced in the local symbol table of the called function when it is called; thus, arguments are passed using call by value (where the value is always an object reference, not the value of the object).1 When a function calls another function, a new local symbol table is created for that call. A function definition introduces the function name in the current symbol table. The value of the function name has a type that is recognized by the interpreter as a user-defined function. This value can be assigned to another name which can then also be used as a function.

Annotations are stored in the annotations attribute of the function as a dictionary and haven o effect on any other part of the function. Parameter annotations are

defined by a colon after the parameter name, followed by an expression evaluating to the value of the annotation. Return annotationsare defined by a literal ->, followed by an expression, between the parameter list and the colon denoting the end of the def statement.

The comparison operators in and not in check whether a value occurs (does not occur) in a sequence. The operator is and does not compare whether two objects are really the same object; this only matters for mutable objects like lists. All comparison operators have the same priority, which is lower than that of all numerical operators. Comparisons can be chained. For example,a<b==ctestswhetheraislessthanbandmoreoverbequalsc. Comparisons may be combined using the Boolean operators and the outcome of a comparison (or of any other Boolean expression) may be negated with not. These have lower priorities than comparison operators; between them, not has the highest priority and or the lowest, so that A and not B or C is equivalent to (A and (not B)) or C. As always, parentheses can be used to express the desired composition. The Boolean operators and are so-called short-circuit operators: their arguments are evaluated from left to right, and evaluation stops as soon as the outcome is determined. For example, if A and C are true but Bis false, A and B and C does not evaluate the expression C. When used as a general value and not as a Boolean, the return value of a short-circuit operator is the last evaluated argument.

Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its class) for modifying its state. Compared with other programming languages, Python's class mechanism adds

classes with a minimum of new syntax and semantics. It is a mixture of the class mechanisms found in C++ and Modula-3. Python classes provide all the standard features of Object Oriented Programming: the class inheritance mechanism allows multiple base classes, a derived class can override any methods of its base class or classes, and a method can call the method of a base class with the same name. Objects can contain arbitrary amounts and kinds of data. As is true for modules, classes partake of the dynamic nature of Python: they are created at runtime, and can be modified further after creation. In C++ terminology, normally class members (including the data members) are public (except see below Private Variables), and all member functions are virtual. A sin Modula-3, there are no short hands for referencing the object's members from its methods: the method function is declared with an explicit first argument representing the object, which is provided implicitly by the call. A sin Small talk, classes themselves are objects. This providesSemantics for importing and renaming. Unlike C++ and Modula-3, built-in types can be used as base classes for extension by the user. Also, like in C++, most built-in operators with special syntax (arithmetic operators, sub scripting etc.) can be redefined for class instances.(Lacking universally accepted terminology to talk about classes, I will make occasional use of Smalltalk and C++ terms. I would use Modula-3 terms, since its object- oriented semantics are closer to those of Python than C++, but I expect that few readers have heard of it.)

Objects have individuality, and multiple names (in multiple scopes) can be bound to the same object. This is known as aliasing in other languages. This is usually not appreciated on a first glance at Python, and can be safely ignored when dealing with immutable basic types (numbers, strings, tuples).However, aliasing has a possibly surprising effect on these mantic of Python code involving mutable objects such as

lists, dictionaries, and most other types. This is usually used to the benefit of the program, since aliases behave like pointers in some respects. For example, passing an object is cheap since only a pointer is passed by the implementation; and if a function modifies an object passed as an argument, the caller will see the change — this eliminates the need for two different argument passing mechanisms as in Pascal.

A namespace is a mapping from names to objects. Most name spaces are currently implemented as Python dictionaries, but that's normally not noticeable in any way (except for performance), and it may change in the future. Examples of name spaces are: these to f built-in names (containing functions such as abs(), and built-in exception names); the global names in a module; and the local names in a function invocation. In a sense the set of attributes of an object also form a namespace. The important thing to know about namespaces is that there is absolutely no relation between names in different namespaces; for instance, two different modules may both define a function maximize without confusion — users of the modules must prefix it with the module name. By the way, I use the word attribute for any name following a dot — for example, in the expression z. real, real is an attribute of the object z. Strictly speaking, references to names in modules are attribute references: in the expression modname.funcname, modname is a module object and funcname is an attribute of it. In this case there happens to be a straight forward mapping between the module's attributes and the global names defined in the module: they share the same namespace!1 Attributes may be read-only or writable. In the latter case, assignment to attributes is

possible. Module attributes are writable: you can

write modname.the_answer = 42. Writable attributes may also be deleted with the del statement. For example, del mod name .the_ answer will remove the attribute the_answer from the object named by mod name. Namespaces are created at different moments and have different lifetimes. The namespace containing the built-in names is created when the Python interpreter starts up, and is never deleted. The global namespace for a module is created when the module definition is read in; normally, module namespaces also last until the interpreter quits.The statements executed by the top-level invocation of the interpreter, either read from a script file or interactively, are considered part of a module called main, so they have their own global namespace.(The built-in names actually also live in a module; this is called built ins.) The local namespace for a function is created when the function is called, and deleted when the function returns or raises an exception that is not handled within the function. (Actually, forgetting would be a better way to describe what actually happens.) Of course, recursive invocations each have their own local namespace.

To speed uploading modules, Python caches the compiled version

of each module in the __pycache____directory under the name

module.version.pyc, where the version encodes the format of the compiled

file; it generally contains the Python version number. For example, in CPython release 3.3 the compiled version of spam.py would be cached as

pycache/spam.cpython-33.pyc. This naming

convention allows compiled modules from different releases and different versions

of Python to coexist. Python checks the modification date of the source against the compiled version to see if it's out of date and needs to be recompiled. This is a completely automatic process. Also, the compiled modules are platform-independent, so the same library can be shared among systems with different architectures. Python does not check the cache in two circumstances. First, it always recompiles and does not store the result for the module that's loaded directly from the command line. Second, it does not check the cache if there is no source module. To support anon-source (compiled only) distribution, the compiled module must be in the source directory, and there must not be a source module. Some tips for experts:

- You can use the -O or -OO switches on the Python command to reduce the size of a compiled module. The -O switch removes assert statements, the -OO switch removes both assert statements and doc

strings. Since some programs may rely on having these available, you should only use this option if you know what you're doing. "Optimized"

modules have an opt- tag and are usually smaller. Future releases may change the effects of optimization.

- A program doesn't run any faster when it is read from a .pyc file than when it is read from a .py file; the only thing that's faster about .pyc files is the speed with which they are loaded.
- The module compile all can create .pyc files for all modules in a directory.
- There is more detail on this process, including a flow chart of the decisions

KNN ALGORITHM:

- 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.

Working of KNN:

- **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.

CHAPTER – 4

4.1 PROCESS FLOW AND IMPLEMENTATION

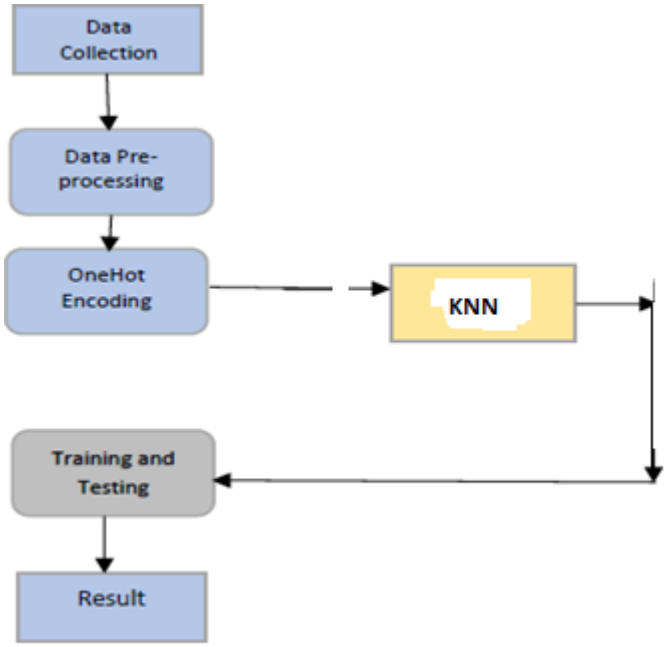


Fig 17: Process Flow

1. Data Collection:

Collection of data is one of the major and most important tasks of any machine learning projects. Because the input we feed to the algorithms is data. So, the algorithms efficiency and accuracy depends upon the correctness and quality of data col-lected. So as the data same will be the output. For student ca-reer prediction many parameters are

required like students academic scores in various subjects, specializations, programming and analytical capabilities, memory, personal details like relationship, interests, sports, competitions, hackathons, workshops, certifications, books interested and many more. As all these factors play vital role in deciding stu-dent's progress towards a career area, all these are taken in-to consideration. Data is collected in many ways. Some data is collected from employees working in different organizations, some amount of data is collected through LinkedIn api, some amount of data is randomly

generated and other from college alumni database. Totally nearly 20 thousand records with 36 columns of data is collected.

2. Data Pre-processing:

Collecting the data is one task and making that data useful is an-other vital task. Data collected from various means will be in an unorganized format and there may be lot of null values, in-valid data values and unwanted data. Cleaning all these data and replacing them with appropriate or approximate data and removing null and missing data and replacing them with some fixed alternate values are the basic steps in pre processing of data. Even data collected may contain completely garbage val-ues. It may not be in exact format or way that is meant to be. All such cases must be verified and replaced with alternate values to make data meaning meaningful and useful for further processing. Data must be kept in a organized format.

3. Application Of Algorithms:

The next step is alogorithms are applied to data and results are noted and observed.The algorithms are applied in the fashion menction in the diagram so as to improve accuracy at each stage.

4. Training and Testing:

Finally after processing of data and training the very next task is obviously testing. This is where performance of the algorithm, quality of data, and required output all appears out. From the huge data set collected 80 percent of the data is utilized for train-ing and 20 percent of the data is reserved for testing. Training as discussed before is the process of making the machine to learn and giving it the capability to make further predictions based on the training it took.Where as testing means already having a predefined data set with output also previously labelled and the model is tested whether it is working properly or not and is giving the right prediction or not. If maximum number of predictions are right then model will have a good accuracy percentage and is reliable to continue with otherwise better to change the model.

4.2 SOURCE CODE

CHAPTER – 5

5.1 RESULT

The data is trained and tested with all three algorithms and out of all KNN gave more accuracy with 90.3 percent and then the KNN with 88.33 percent accuracy. As KNN gave the highest accuracy, all further data predictions are chosen to be followed with KNN. So, finally a web application is made to give the input parameters of the student and the final prediction is generated and displayed. The background algorithmbeing used is KNN and the new prediction are keep on adding to the dataset for furthermore accuracy.

5.3 RESULT SCREENSHOTS

Fig 23: KNNimplementation in COLAB

In [18]:

```
#----------------calculating confusion vector values matrix and accuracy------------#
cm = confusion_matrix(y_test,y_pred)
accuracy = accuracy_score(y_test,y_pred)+da
print("confusion matrics=",cm)
print("  ")
print("accuracy=",accuracy*100)
```

```
confusion matrics= [[2 7 1 ..., 3 6 2]
 [2 2 2 ..., 3 3 4]
 [2 2 1 ..., 1 2 4]
 ...,
 [4 5 2 ..., 3 0 7]
 [4 5 3 ..., 8 3 4]
 [4 4 1 ..., 4 3 3]]

accuracy= 86.48
```

CHAPTER – 6

FUTURE SCOPE:

A powerful web application can be developed where inputs are not given directly instead student parameters are taken by evaluating students through various evaluations and examining. Technical, analytical, logical, memory based, psychometry and general awareness, interests and skill based tests can be designed and parameters are collected through them so that results will be certainly accurate and the system will be more reliable to use.

Also KNNs have few limitations like overfitting, no pruning, lack of capabilityto deal with null and missing values and few algorithms have problem with huge number of values. All these can be taken into consideration and even more reliable andmore accurate algorithms can be used. Then the project will be more powerful to dependupon and even more efficient to depend upon.

CHAPTER – 7

REFERENCES

[1] P.KaviPriya**, "**A Review on Predicting Students' Academic Performance Earlier, Using Data Mining Techniques", International Jour-nal of Advanced Research in Computer Science and Software Engineering

[2] Ali Daud, Naif Radi Aljohani, "Predicting Student Performance using Advanced Learning Analytics", 2017 International World Wide Web Conference Committee (IW3C2).

[3] Marium-E-Jannat,SaymaSultana,Munira Akther, "A Probabilistic Machine Learning Approach for Eligible Candidate Selection", Inter-national Journal of Computer Applications (0975 – 8887)Volume 144 – No.10, June 2016

[4] Sudheep Elayidom, Dr. Sumam Mary Idikkula, "Applying Data mining using Statistical Techniques for Career Selection", International Journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009.

[5] Dr. Mahendra Tiwari ,Manmohan Mishra, "Accuracy Estimation of Classification Algorithms with DEMP Model", International Jour-nal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 11, November 2013.

[6] Ms. Roshani Ade,Dr. P. R. Deshmukh, "An incremental ensemble of classifiers as a technique for prediction of student's career choice", 2014 First International Conference on Networks & Soft Compu-ting

[7] Nikita Gorad ,Ishani Zalte, "Career Counselling Using Data Mining", International Journal of Innovative Research in Computer and Communication Engineering.

[8] Bo Guo , Rui Zhang, “Predicting Students Performance in Educa-tional Data Mining”,2015 International Symposium on Educational Technology

[9] Ali Daud , Naif Radi Aljohani , “Predicting Student Performance using Advanced Learning Analytics”

[10] Rutvija Pandya Jayati Pandya , “C5.0 Algorithm to Improved KNN withFeature Selection and Reduced Error Pruning“, Inter-national Journal of Computer Applications (0975 – 8887) Volume 117 – No. 16, May 2015.

[11] Comparative Analysis of KNN Algorithms: ID3, C4.5 and KNNShiju Sathyadevan and Remya R. Nair

[12] Yu Lou, Ran Ren, “A Machine Learning Approach for Future Ca-reer Planning”

[13] Gareth James ,Daniela Witten ,Trevor Hastie, ”An Introduction to Statistical Learning with Applications in R”

[14] Anuj Karpatne, Gowtham Atluri, “Theory- Guided Data Science: A New Paradigm for Scientific Discovery from Data”, IEEE trans-actions on knowledge and data engineering, vol.29, no. 10, october 2017.

PUBLISHED RESEARCH PAPER