ELOAN-RECOMMENDER SYSTEM

A system which deals with student's academic records with their past few months banking transaction (optional) and recommend a loan for their higher study purpose.

Project guide: Dr. Tanmoy Halder

Name: Souvik Ganguly

Roll no: 32301218027

Department: Bachelor of Computer Application (B.C.A)

Dr. B.C. Roy Engineering College, Academy of Professional Courses

Acknowledgement

I would like to express my deep sense of gratitude and deep regards to my faculty and project mentor Dr. Tanmoy Halder as well as our Principal Dr. Arunava Mukherjee who gave me the golden opportunity to do this wonderful project on **“E-Loan Recommender System”.** Dr. Tanmoy Halder sir helped me a lot with his exemplary guidance and constant sublime throughout the making of this project. I am feeling oblige to express that the blessings, supports and encouragement which I have got from them will strongly carry me a long way in my journey of life on which I am about to embark.

Minimum software requirements in this project

1. ***Language****: Python 3.7*
2. ***Open Source Distribution****: Anaconda -3.5*
3. ***Notebook****: Jupyter notebook*
4. ***Database****: MS-Excel*
5. ***Operating System****: Windows 10*

Minimum hardware requirements in this project

For ML algorithms:

1. Desktop or laptop used with i3 processor and 4 gb of Ram.
2. Desktop or Laptop used with 100 gb free of space.

For DL algorithms (optional)

1. Desktop or laptop used with i5 processor and 8 gb of ram.
2. Desktop or laptop used with 100 gb free of space.

Mouse: Any Standard

Keyboard: Any Standard

ABSTRACT

Recommender systems have become omnipresent in our lives. Project on recommender systems is become more favorable in today's world in Machine Learning. In this project, we attempt to understand the association systems and recommendation systems and compare their performance on the "Student Education Loan" dataset. We attempt to upgradable model to perform this analysis. We start by uploading the model in "Jupyter" notebook and load the dataset while we are performing on the project and then we compare the various models on a smaller dataset of more chances for a student to get a loan. We attempt to analyze the dataset and took those columns which we have needed. Then we found the all student details with their academic records and their previous 6 months transactions through bank which have done by their parents' or guarantors'. We attempt to compare the both student dataset and bank dataset on your needs. Then we scale the model by using python and we found for the loan on which bank offered more loan for a student and finally we kept a look on the recommend details and chances to get a loan as per the previous data.

INTRODUCTION

A recommendation system is a type of information filtering system which attempts to predict the performances of a user, and make suggestions based on these performances. These are a wide variety of applications for recommendation systems. They have become increasingly popular over the few years and are now utilized in various platforms that we use. The content of such platforms varies from movies, music's, books, loans and videos to friends and store on various platforms for future use.

AIM

Load the dataset into our model one by one. At first attempt, we load the student dataset and bank transaction dataset with their academic records and filtering the chances of a particular student to get the exact loan. After that we load our bank dataset consists of bank name and loan offered by the bank and perform the association rule mining tasks to get the more accurate list of result whereas the system will accurately predict the final output.

ALGORITHM USED:

1. **Supervised learning algorithm*:*** Supervised learning is when the model is getting trained on a labeled dataset. Labeled dataset is one which has both input and output parameters. In this type of learning both training and validation datasets are labeled.
   - ***i.*** **CLASSIFICATION:** It is a supervised learning task where output is having defined (discrete values).

ID	SC%	HSC%	OUTPUT
1	72	70	1
2	72	71	1
3	67	54	0
4	78	70	??

In the above figure output has defined labels i.e. 0 and 1. 1 means the student has good SC and HSC percentage and 0 means the student has SC and HSC percentage but not satisfied. The goal is here to predict discrete values belonging to a particular class and evaluate on the basis of accuracy.
In this table the exact outcome or output of id 4 would be 1 because the SC% and HSC% of id (1, 2, 4) are >= 72% and 70%.
TRAINING PROCESS: while training the model, data is usually split in the ratio of 80:20 or 70:30 i.e. 80% as training data and 20% is testing data. In training data, we feed input as well as output for 80% data. The model learns from training data only. We use different machine learning algorithms to build our model. Once the model is ready

then it is good to be tested. At the time of testing input is fed from remaining 20% data which the model has never seen before, the model will predict some value and we will compare it with actual output and calculate the accuracy.

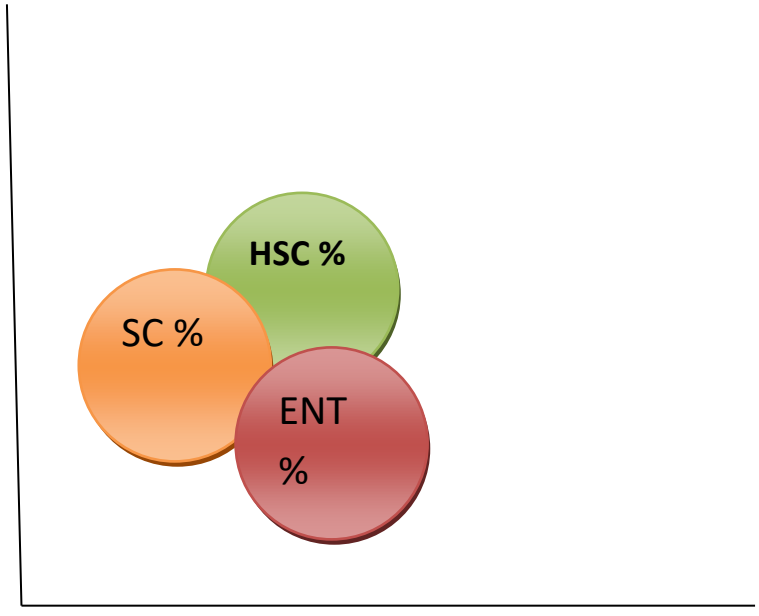
ii. **REGRESSION:** It is a supervised learning task where output is having continuous value.

6_M_T	SC%	HSC%	ENT%
20,000	72	70	66
2000	72	78	79
500	71	68	51
7530	83	50	60

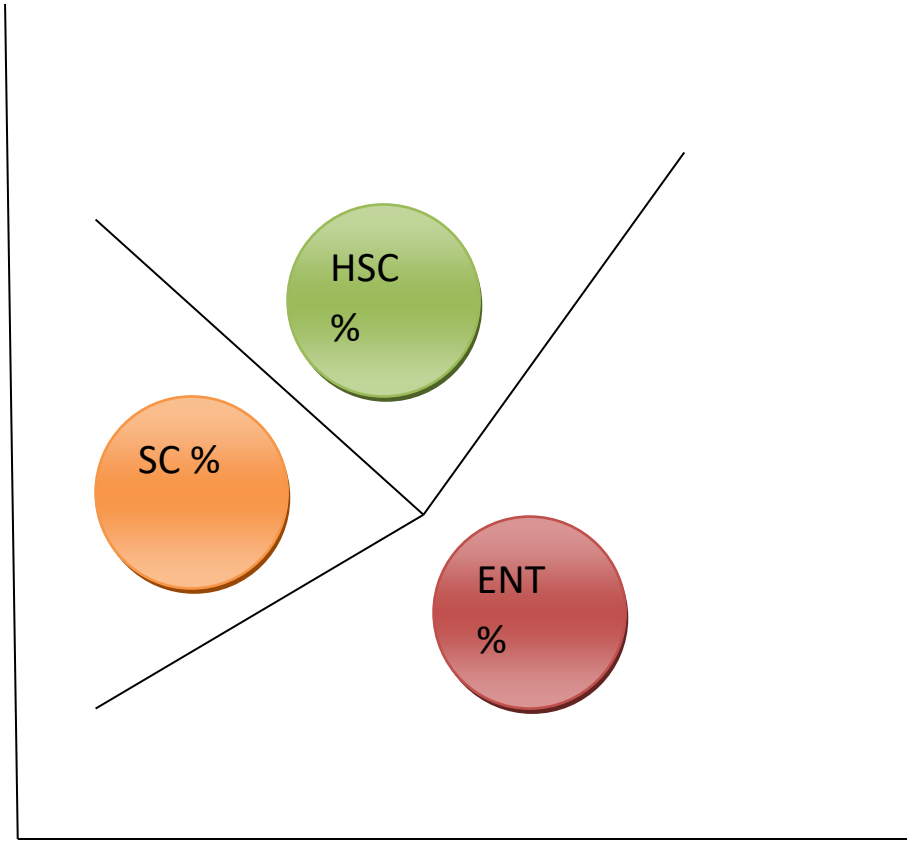
In the above diagram, output- ENT% has no discrete value but it is continuous in the particular range. The goal is here to predict a value as much closer to the actual output value as our model can and then evaluation is done by calculating error value. "**The smaller the error the greater the accuracy of our regression model.**"

2. **Unsupervised learning algorithm:** Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data. Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our self.

I. **Clustering:** A clustering problem is where you want to discover the inherent groupings in the data, such as grouping students by taking loans.



In the above diagram sc%, hsc% and ent% all are in same place. Hence we need to cluster or grouping themselves.



Hence the data of sc%, hsc% and ent% are partially grouped and being separated by their behavior.

II. Association: An association rule learning problem is where you want to discover rules that describe large portions of our data, such as people that take Loan A also tend to take loan B that can be offered to a person also tend to offer another loan.

On the above example a user wants to take Loan X. While he/she has already took the loan now he/she will be taking the Loan Y. Here, the Loan Y is associated with the Loan X. So in this example it is clearly visible that the user frequently took the recommended loans. Association rule based algo are viewed as two step approach:

1. Find all frequent item sets with support>= pre determined min_support count.
2. List all association rules from frequent item-sets. Calculate support and confidence for all rules. Prune rules that fail min_support and min_confidence thresholds.

What is recommendation?

Recommender system is Machine Learning systems that help users discover new products and services. It collects the data from any resources and analyzes that data and generates customized

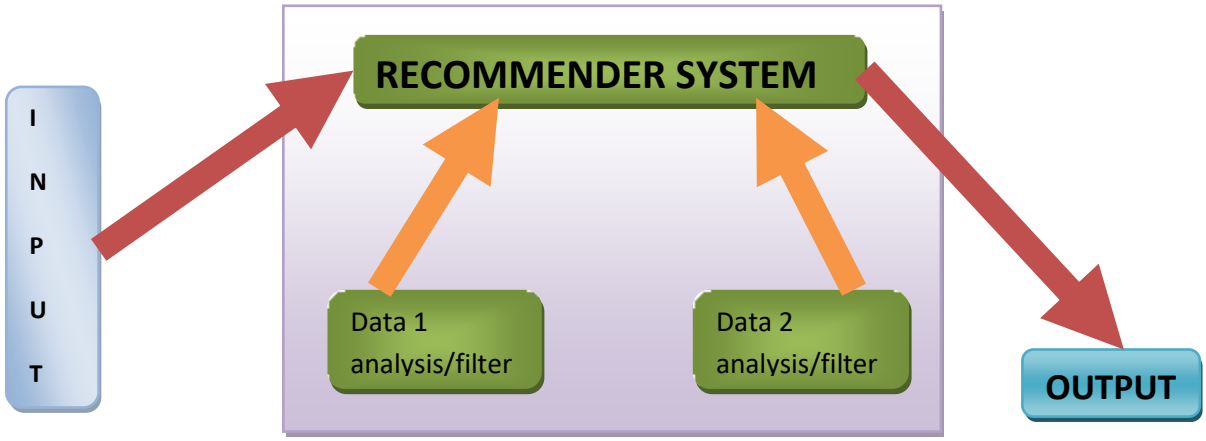
recommendations for their users/ customers. The application of recommender systems are movies, programs, books, documents etc.

On the above example a customer/user go to take item A associated with item B but there is one more item which is item C (recommended item).

Hybrid Recommendation

Hybrid recommendation systems combine two or more recommendation strategies/ analysis in different ways to benefit their advantages.

Only a single recommendation component



Apriori algorithm

To make this project we have used **APRIORI** algorithm. Apriori algorithm assumes that any subset of a frequent item set must be frequent.

Let's take an example:

Say, a transaction containing {loan1, loan2, loan3} also contains {loan1, loan2}. So, according to the principal of Apriori if {loan1, loan2, loan3} is frequent then {loan1, loan2} must also be frequent.

There is a dataset consisting of four transactions, each transaction is a combination of 0's and 1's, where 0 represents the absence of an item and 1 represents the presence of it.

Transaction id	ELoan 1	ELoan 2	ELoan 3
1	1	1	1
2	1	0	1
3	0	1	0
4	1	1	1

In order to find out interesting rules out of multiple possible rules from this small educational loan scenario, we will be using the following metrics:

- **Supports**(ELoan)=(transactions involving ELoan1)/(total transactions)

Here transaction will be 3/4

- **Confidence**({ELoan1,ELoan2}=>{ELoan3})=support(ELoan1,ELoan2,ELoan3)/support(ELoan1, ELoan2)

Here transactions will be (2/4) / (2/4)

- **Lift:**

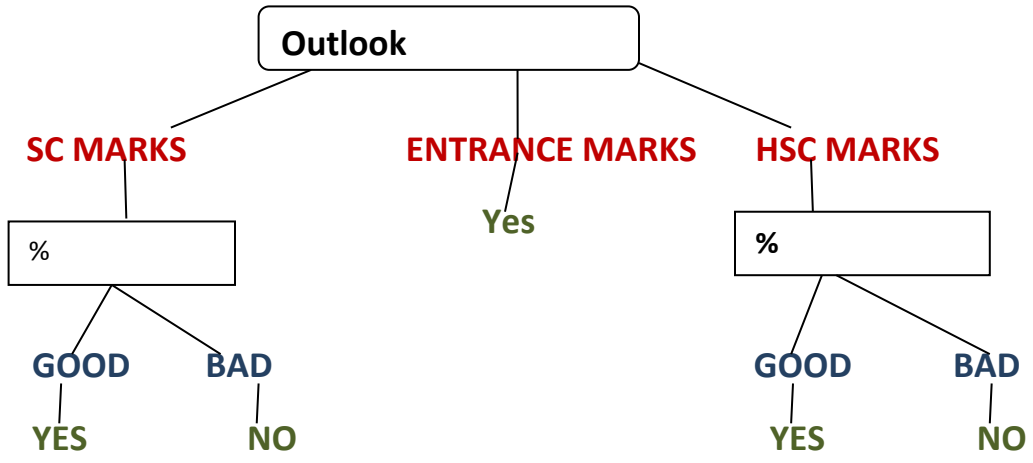
Lift (A=>B) =1 means there is no co relation within the dataset

Lift (A=>B) >1 means there is positive co relation between the item set and took A, B together.

Lift (A=>B) <1 means there is negative co relation.

Decision Tree

A decision tree is a flow chart like structure in which each internal node represents a test on a feature (whether a coin flip comes up heads or tails), each leaf node represents a class label and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules.



Above diagram illustrate the basic flow of decision tree for decision making with labels (loan offered (yes)), no loan offered (No).

Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. This algorithm used for both classification and regression tasks.

CART: Classification and Regression trees

Entropy: Entropy is nothing but the measure of disorder.

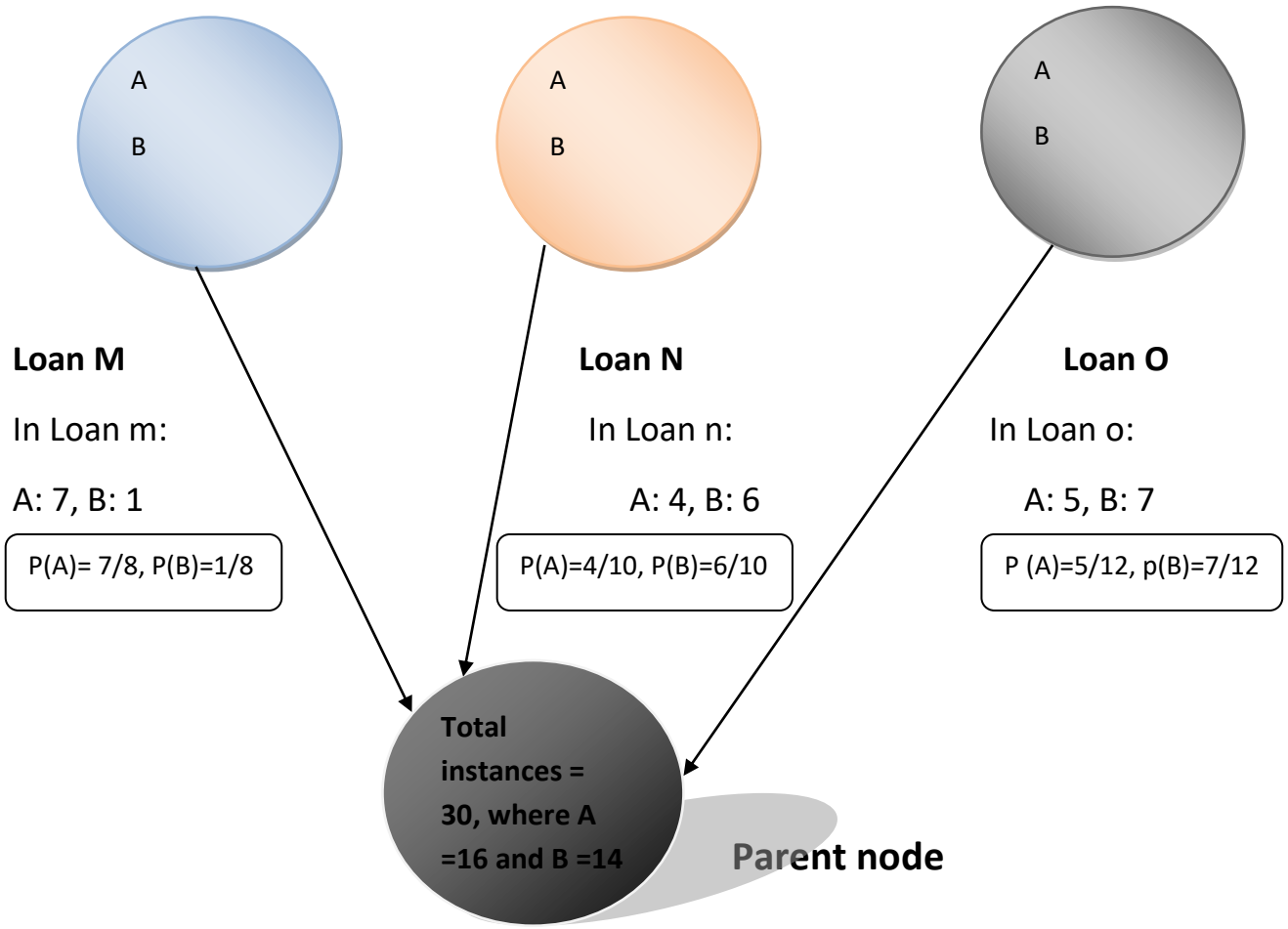
$$E(S) = \sum_{i=1}^{c}(-\text{Pi } \log 2 \text{ Pi})$$

Where 'Pi' is simply the frequentist probability of an element/class 'i' in data. If we have two classes (+ ve and – ve), therefore 'i' here could be either +ve or –ve. As we have 100 data points. Among the 30% belonging from +ve class and 70% belonging from –ve. So, 'P+' would be 3/10 and 'P-'would be 7/10.

- (3/10) * log2 (3/10) – (7/10) * log2 (7/10) which is approximately 0.88(a high level disorder). Entropy is measured between 0 and 1 (depending on the number of classes in the dataset). Entropy can be greater than 1 but it means a high level disorder.

Information gain: Information gain is the reduction in entropy by transforming a dataset and is used in training decision trees. IG is calculated by comparing the entropy of the dataset before and after transformation.

Information gain of y to x (y, x): IG(y, x) = E(y) –E(y/x)



E (loan M) = - (7/8) log2 (7/8) – (1/8) log2 (1/8) = approx. 0.54

E (loan N) = - (4/10) log2 (4/10) – (6/10) log2 (6/10) = approx. 0.97

E (loan O) = - (5/12) log2 (5/12) – (7/12) log2 (7/12) = approx. 0.98

Weighted average of entropies of each node:

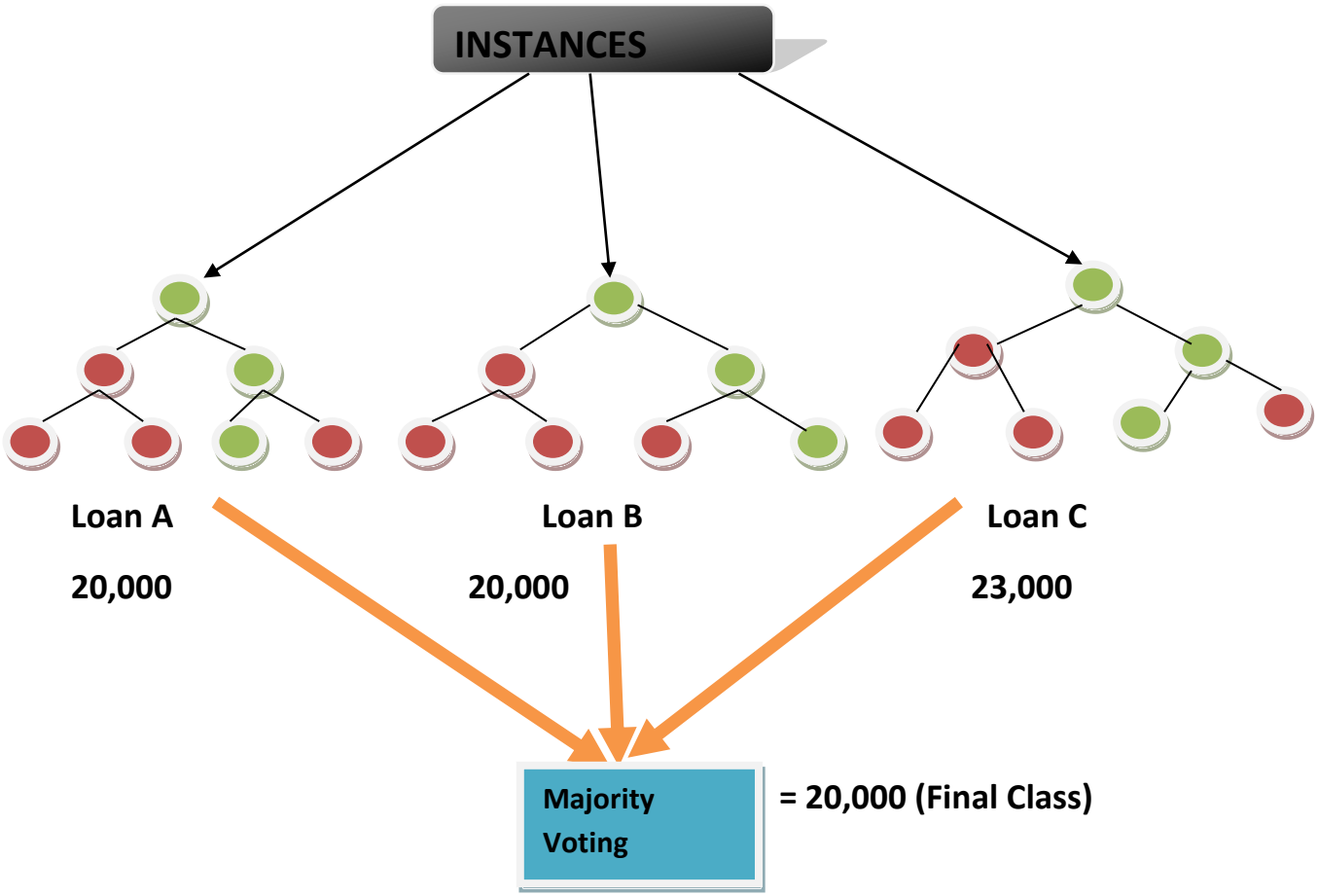
E (loans) = (8/30) * 0.54 + (10/30) * 0.97 + (12/30) * 0.98 = 0.86

Information gain: E (Parent, loans) = E (Parent) – E (loans) = 1-0.86 = 0.14

Random Forest

Random forest is an ensemble learning algorithm. The basic premise of the algorithm is that building a small decision tree with few features is a computationally cheap process. It can be used for both classification and regression.

Bagging: Ensemble is a Machine Learning technique in which multiple models are generally trained using the same ML algo. Bagging is the way to reduce the variance in the prediction by generating additional data from training dataset. The training algorithm for random forests applies the general technique of bagging, to tree learners.



Let a training set of loan x_train with responses y. bagging repeatedly selects a random sample with replacement of the training sets and fits tree to these samples.

Training sample x_train and y_train and train a classification or a regression tree on x_train and y_train. Let times N.

$$(f)\hat{} = 1/N \sum_{n=1}^{n} \mathrm{f(n)}\left(\mathrm{x\_train}\right)'$$

N_estimators: The number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions.

Max_features: Maximum number of features random forest considers splitting a node.

Min_sample_leaf: It determines the minimum number of leafs required to split the internal node.

Random_state: It makes the model's output replicable. The model will always produce the same results when it has a definite value of random_state and it has been given the same training data.

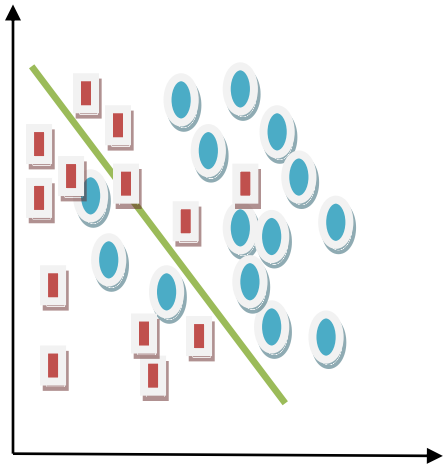
Underfitting:

A statistical model or a machine learning algorithm is said to have Underfitting when it cannot capture the underlying trend of the data. Underfitting destroys the accuracy of the machine learning model. Its occurrences simply mean that the model or algorithm does not fit the data well enough. It usually happens when we have fewer amounts of data to build an accurate model and also when we try to build a linear model with a nonlinear data.

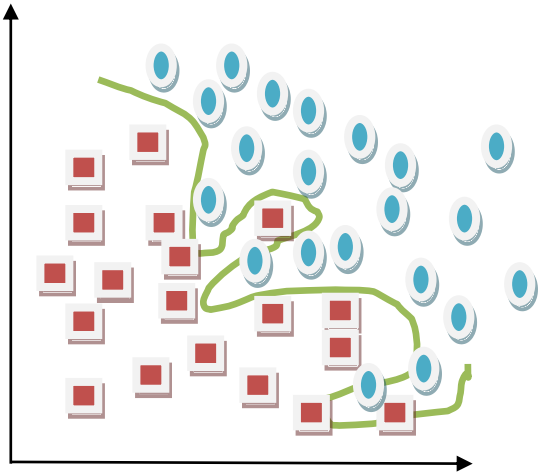
Overfitting:

A statistical model is said to be overfitted, when we train it with a lot of data. When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set. Then the model does not categorize the data correctly, because of too much of details and noise. The cause of Overfitting are the non parametric and non linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset therefore they can really build unrealistic model.

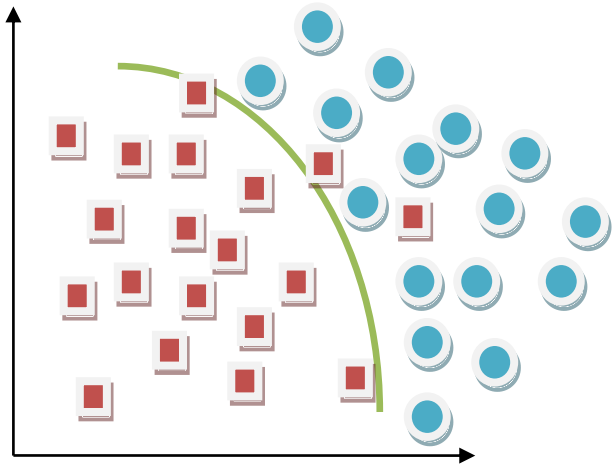
This problem occurs in both **Decision Tree** and **Random Forest** algorithms.



Under Fitting



Over Fitting



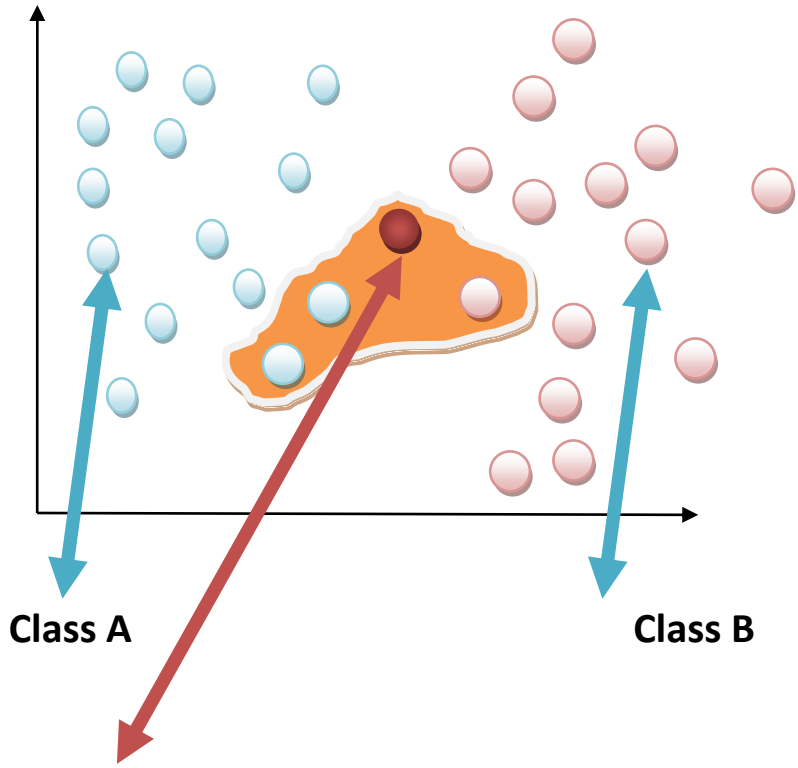
Appropriate Fitting

Avoid Overfitting:

1. **Cross Validation**: A standard way to find out of sample prediction error is to use 5-fold cross validation.
2. **Early stopping**: its rule provides us the guidance as to how much iteration can be before learner begins to over fit.
3. **Pruning**: Pruning is extensively used while building related models. It simply removes the nodes which add little predictive power for the problem in hand.

Regularization: It introduces a cost term for bringing in more features with the objective function. Hence it tries to push the co efficient for many variables to zero and hence reduce cost term.

K Nearest Neighbor Algorithm: This is a simple algorithm which predicts data point with its K Nearest neighbors. The value of k is a critical factor here regarding the accuracy of prediction. It determines the nearest by calculating the distance using distance functions like Euclidean. However, this algorithm needs high computation power and we need to normalize data initially to bring every data point to same range.

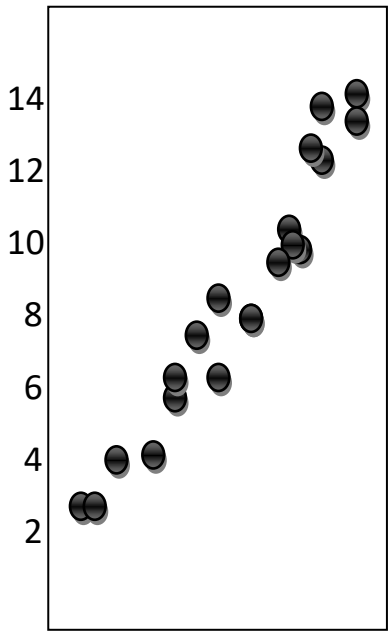


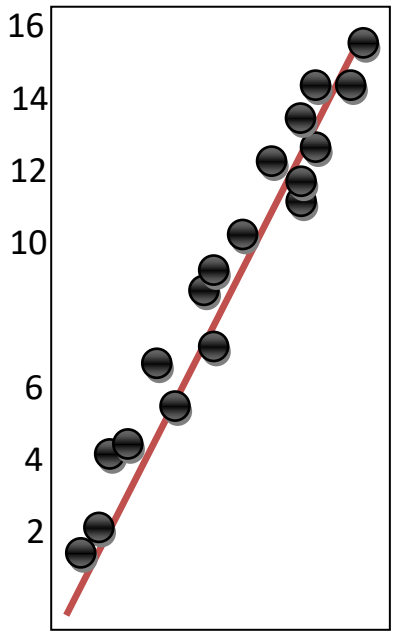
K point =3 so that it can find 3 data points from two classes which are just near able from k point.

Gradient Boosting algorithm: Gradient boosting algorithm uses multiple weak algorithms to create a more powerful accurate algorithm. Instead of using single estimator, having multiple will create a more stable and robust algorithm.

XGBOOST: XGBoost is a decision tree based ensemble Machine Learning algorithm that uses a gradient boosting framework. The applications of this algorithm are to solve regression task, classification task etc. XGBoost as gradient boosting on **'steroids'**, it is a perfect combination of software and hardware optimization techniques to yield superior results using less computing resources in the shortest amount of time. Suppose in a dataset the accuracy of a model is 90% on decision tree algorithm and 91% on random forest algorithm then the accuracy of xgboost algorithm must be 94%. XGBoost algorithm has the best combination of prediction performance and processing time compared to other algorithms.

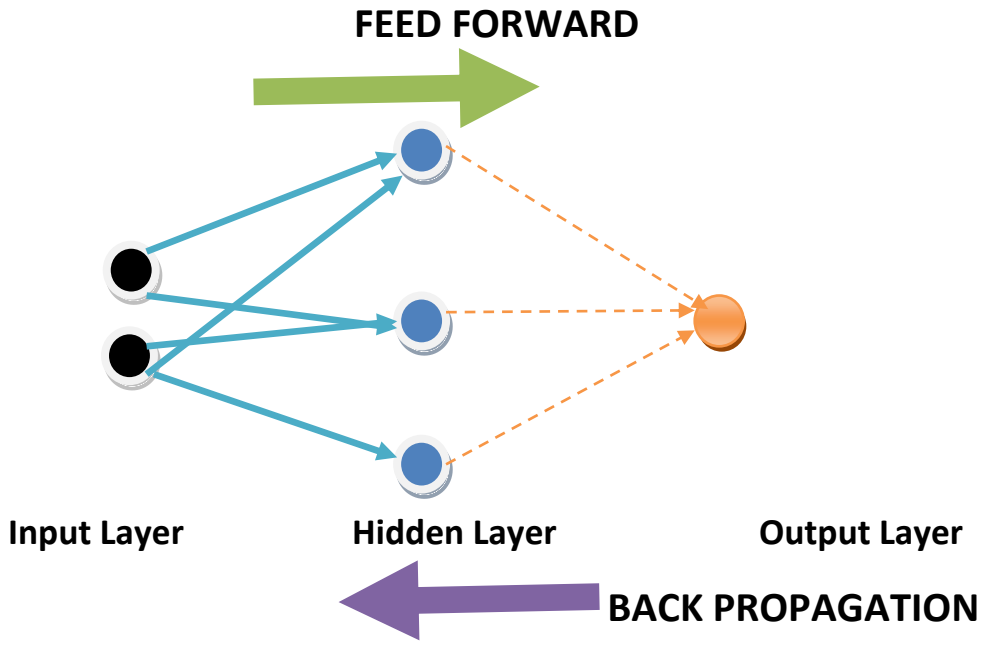
Linear Regression: Linear regression is used for predictive analysis. Linear regression is a linear approach for modeling the relationship between the criterion or the scalar response and the multiple predictors or exemplary variables. Linear regression focuses on the conditional probability distribution of the response given the values of the predictors.





It's a way to model the relationship between two variables. The equation has the form **Y= ax + b**, where Y is the dependent variable (the variable goes on the Y axis) and x is the independent variable (plotted on the x axis) , b is the slope of the line and Y is the intercept.

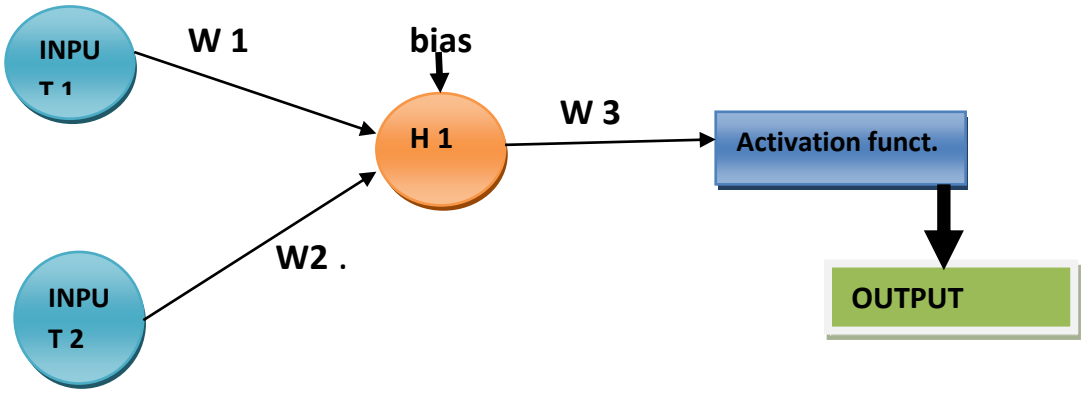
Deep Neural Networks: It is a group of multiple perceptron or multiple neurons at each layer. It is an Artificial Neural Network with multiple layers between the input and output layers. It is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. An ANN is based on a collection of connected units called artificial neurons. Each connection between neurons can transmit a signal and then signal downstream neurons connected to it. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases and functions.



Input is processed in the **forward direction**.

Weights and Biases referred to as W and B, are the learnable parameters of a machine learning model. Weights control the signal between two neurons. In other words, a weight decides how much influence the input will have on the output. Biases are constant and an additional input into the next layer. That will always have the value of 1. Bias units are not influenced by the previous layer but they do have outgoing connections with their own weights.

$$Y = \sum_{n=1}^{\infty}(\mathbf{input} * \mathbf{weight}) + \mathbf{bias}$$

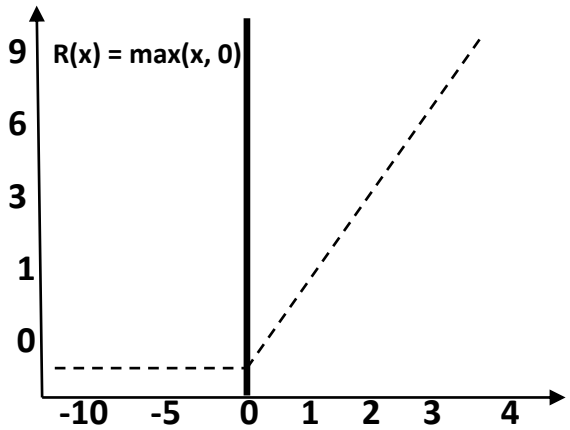


OUTPUT= (input 1*w1+input 2*w2) +bias

Activation Function: An activation function in neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.

RELU: it stands for rectified linear unit. It is the most widely used activation function. It's implemented in hidden layers of neural networks.

Value range: [0, infinite), Equation: A(x) = max (0, x), the output gives x if x is positive and 0 otherwise. It is faster than sigmoid and tanh function.

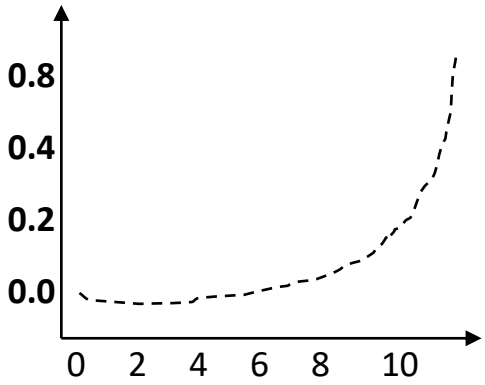


Softmax: the Softmax function is used for when we are trying to handle classification problems. The Softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of outputs.

Transform values into probabilities. Value is 1 in a list [1, 2].

Probability: p1= exp (1) / (exp (1) + exp (2))

P2= exp (2) / (exp (1) + (exp (2)),,,,,, sum= (p1 +p2)



1. **Input layer:** this layer accepts input features. It provides information from the outside world to the network. In other words, it is a layer whereas we can give our inputs to the model.
2. **Hidden layer:** Hidden layer performs all sort of computation on the features entered through the input layer and transfer the result to the output layer.
3. **Output layer:** This layer gives the output which is being performed by the network to the outer world.

Back propagation: Back propagation plays an important role in Machine Learning. It is the central mechanism by which neural networks learn. It works like messenger telling the network whether or not the net made any mistake when it made a prediction. In other words it is a mathematical tool for improving the accuracy of model. It works until and unless the desired output matched the actual output.

Error rate= $\sum 1/2\,(\text{actual output} - \text{desired output})\ \text{^}2$

Correct error= we need to change the weight i.e., (input 1 * w2 + input2 * w1) +w3

New weight= old weight – learning rate x (derivative of error with respect to weight)

W3 = W3 – learning rate (derivative of error / derivative of W3)

Update of W3 = W3 – learning rate x W3 - δh1

A small explanation of neural network techniques:

Dropout: it is a technique to omit the over fitting, during each iteration of GD we draw a randomly selected nodes. Drop the node which does not exist.

Dense: Dense layer is the regular deeply connected neural network layer; it means all the neurons in a layer are connected to those in the next layer. It is the most common and frequently used layer.

Learning rate: In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving towards a minimum loss function.

OPTIMIZATION Algorithm: The process of minimizing any mathematical expression is called optimization. Optimizers are the algorithms used to change the attributes of the neural networks such as weights and learning rate to reduce the losses. It is used to solve optimization problems by minimizing the function. It can also update the various parameters that can reduce the loss in much less effort. Suppose in a neural network W is a vector called weights and B is a scalar called bias. The weights and bias are called the parameters of the model. All we need to do is estimate the value of W and B from the given set of data such that the resultant hypothesis produces the least cost J which is defined by the following cost.

$$J(W,B) = 1/2m \sum_{i=1}^{m} (Yi + h(Xi))\text{^}2$$

Where m is the number of data points in the given dataset. This cost function is also called **Mean Squared Error.** For finding the optimized value of the parameters for which J is minimum, optimization algorithm is being used.

ADAM: Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning model. **Adaptive Gradient Descent:** It maintains a per-parameter learning rate that improves performance on problems with sparse gradients. **Root Mean Squared Propagation:** It also maintains the per parameter learning rates that are adapted based on the average of recent magnitudes of the gradients for the weight that means how quickly it is changed.

Working Areas:

- First made a dataset using Excel format.
- Upload the dataset into Jupyter Notebook.
- Read the dataset through Pandas library.
- All the data analysis has been done through mathematical expression/ libraries.
- Set the model into different classification and regression model to find out the accuracy after training.
- Find out the chances of a student to get the loan or which student is applicable to get the educational loan.

- Find out the error percentage of the dataset of each model.
- An overlook, which model is more preferable for this data.
- Under fitting problem
- Filtering the data on the basis of student marks
- Data visualization
- Clustering
- Recommendation

SEQUENCE DIAGRAM

Sequence diagrams can be useful reference diagrams for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how tasks are moved between objects or components of a process.
- Plan and understand the detailed functionality of an existing or future scenario.

Popular Sequence Diagram Uses:

Usage Scenario – A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system. **Method Logic** - Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to Usage Scenario - A usage scenario is a diagram of how your system could potentially be used. It's a great explore the logic of any function, procedure, or complex process.
Service Logic - If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

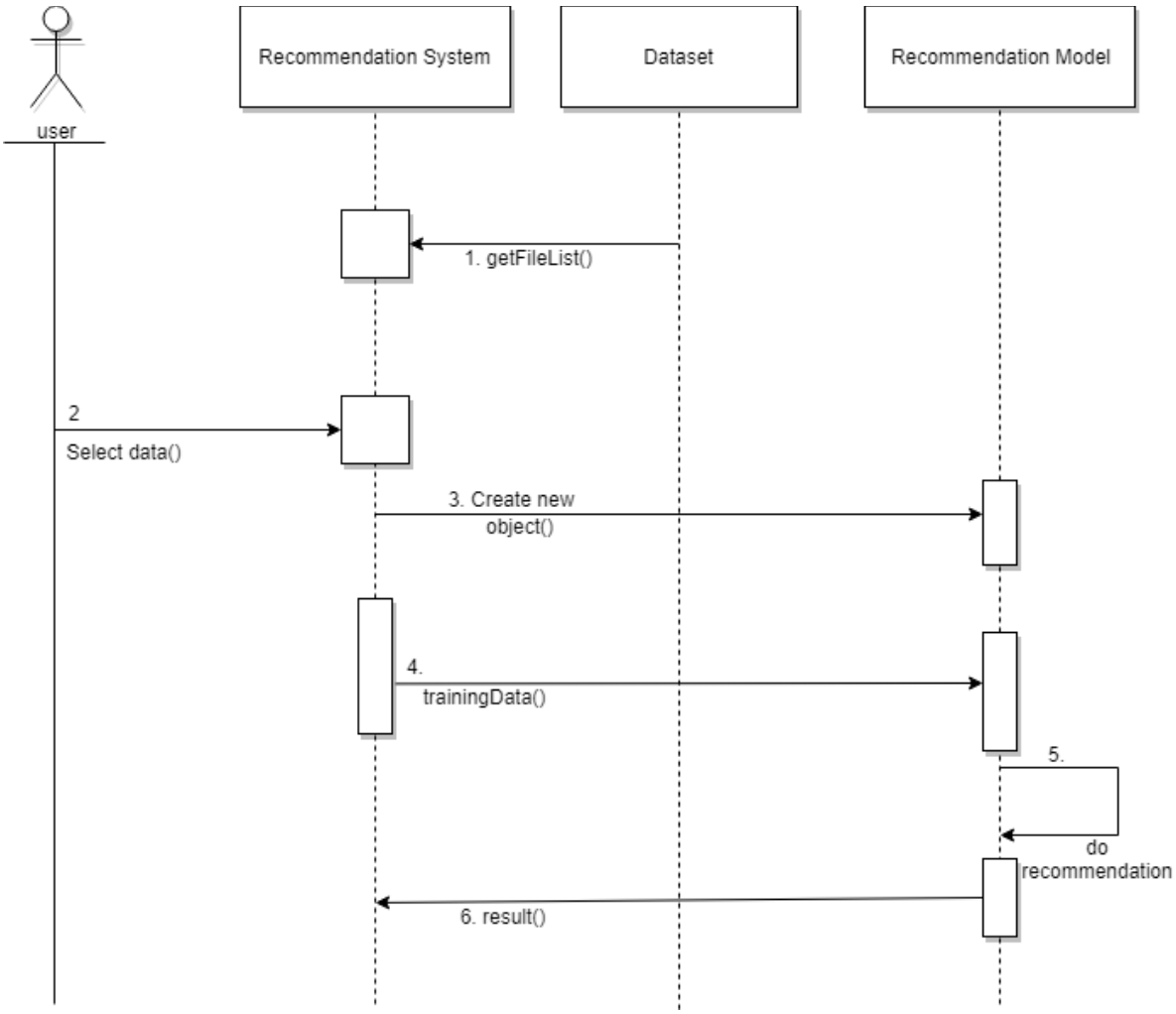


Fig: Sequence Diagram

ACTIVITYDIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to

show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

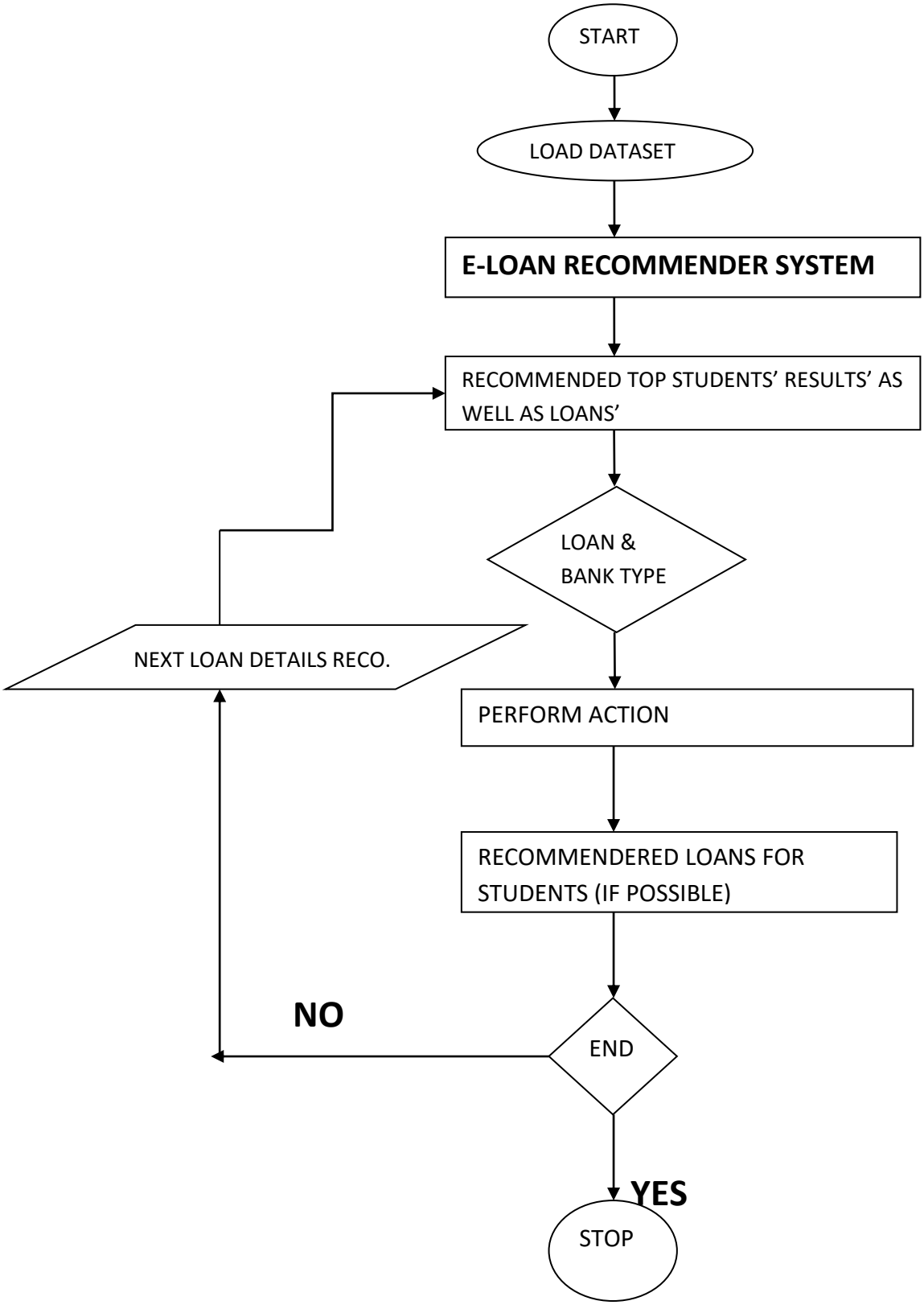


Fig: Activity Diagram

USE CASE DIAGRAM

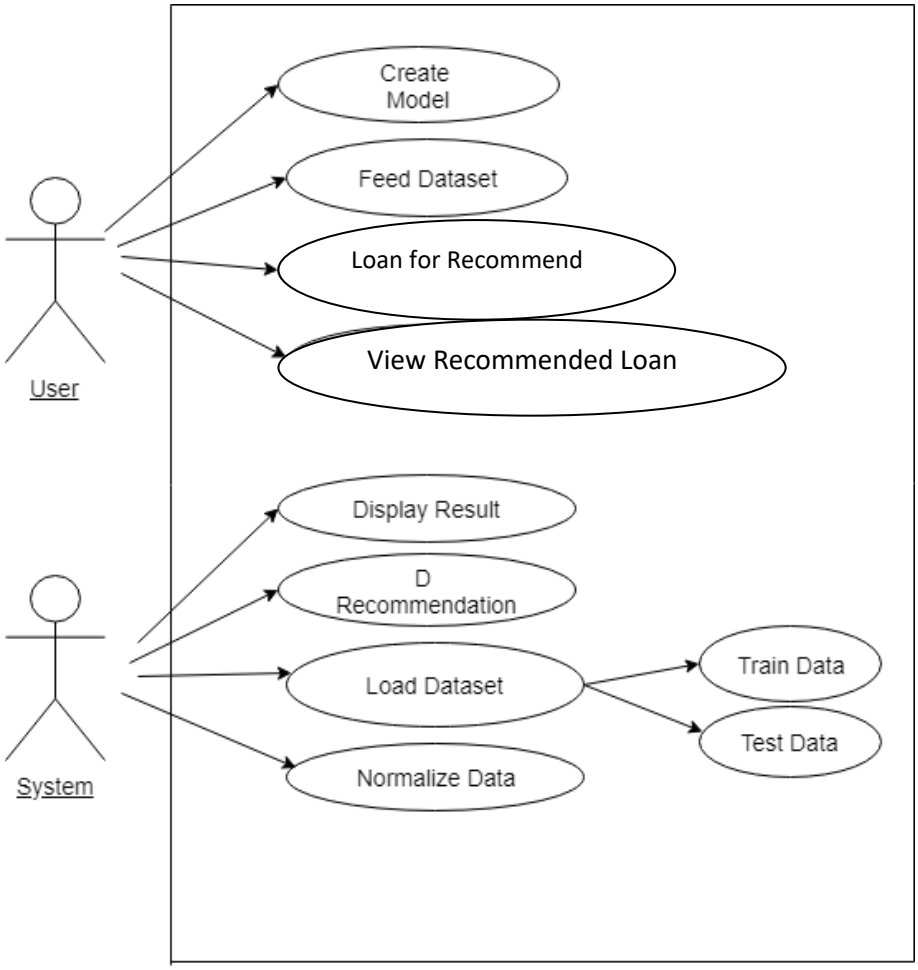


Fig: Use Case Diagram

Testing Steps

★ Unit Testing

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

★ Integration Testing

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

Tested By:	Souvik Ganguly
Test Type	Unit Testing and integration testing
Test Case Number	1
Test Case Name	Recommendation of Educational Loan for a Student
Test Case Description	The user should enter their academic records s well as last 6 months transactions. The system checks to which bank offer the loans for a particular student. The system after that recommends the loans from the banks based on the academic records.

Item(s) to be tested	
1	Verification of recommendation provided by the system.

Specifications

Input	**Expected Output/Result**
1) Enter the student academic records	1) Chances of a student to get the loan
2) Enter bank details along with student academic records	2) Recommend the loan to a student

FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work

Effort and time spent on it: Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

Technical feasible: Technically, this project is very feasible because of use of current and latest technologies.

Economical feasible: This project work is economically feasible as it does not take into account any additional costs. Whatever data is extracted, it is done without any charges.

Operational feasible: Operationally, this project is very easy to use.

FUTURE SCOPE AND FURTHER ENHANCEMENTS

Recommender system is basically designed to identify the items that a user will like or find useful based on their preference.

Using of this project, it is very easy to view that a student will be eligible for a loan or not. Now many companies are building up

this type of recommend system. So Recommendation system has a very wide scope in future, though it has become too much popular but if we talk about future then in future it may be become very popular for the selling, booking, loan prediction purpose and also e- commerce.

CONCLUSION

Thus the recommender system was successfully implemented. We found that **APRIORI** algorithm and **classification** tasks are the best as the accuracy is the higher in its case as compared to the other methods. For working on large dataset, it was an approach in implementing the algorithm and making it a Recommender system. Recommender system has become popular. People use them to find books, movies, smart phones, loans etc. Nearly every products, service, or type of information has recommenders to help people select from among the alternatives the few they would most like and appreciate. So this project is very grateful towards me with creative ideas and techniques.