DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5898

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

CASE STUDY

Available Online at www.ijarcs.info

AUTOMATED STUDENT PERFORMANCE ANALYSER AND RECOMMENDER

Nisha Maria Arunkumar
Loyola-ICAM College Of Engineering
And Technology, Chennai, India

Angela Miriam. A
Loyola-ICAM College Of Engineering
And Technology, Chennai, India

Christina. J
Loyola-ICAM College Of Engineering And Technology,
Chennai, India

Abstract: Big data is a torrent of data streams that are complex such that traditional data-processing system software is inadequate to deal with this huge data. Dealing with big data consists of challenges such as capturing data, data processing, data storage, data analysis, search, sharing, transfer, visualization, querying and updating. Predictive modelling is used for analysing historical events or data to predict known or unknown facts. Neural networks is used to train models efficiently based on the input data sets consisting of parametric features provided. Educational institutions deal with voluminous student mark records that are utilized for understanding and interpreting the institutional academic competency with other institutions. Analysing the performance involves manual computations for report generation. In this paper we have introduced a student performance analyser and recommender that uses prediction algorithms and content based recommendation approach to predict the academic performances which is fully automated and reduces the manual calculations while enabling students to select from a range of recommended subjects based on multiple queries that are suitable to the caliber of the student. The prediction algorithm uses back propagation techniques that take multiple input parametric features to improve the performance in terms of reducing the error rate. Multiple parametric feature
inputs are integrated and compared to the performance when single input features are provided. This research paper helps in analysing the data by automating the tedious calculations including prediction of students performance and recommending papers for the upcoming year. It has been observed that based on the analysis reports that are automatically generated with prediction, future performances of students are found to be accurate due to the multiple input features that gave a higher accuracy and low error rate when compared to other traditional models.

Keywords: Predictive modelling, back propagation, neural networks, apriori algorithm, content based filtering, frequent itemsets, data visualisation.

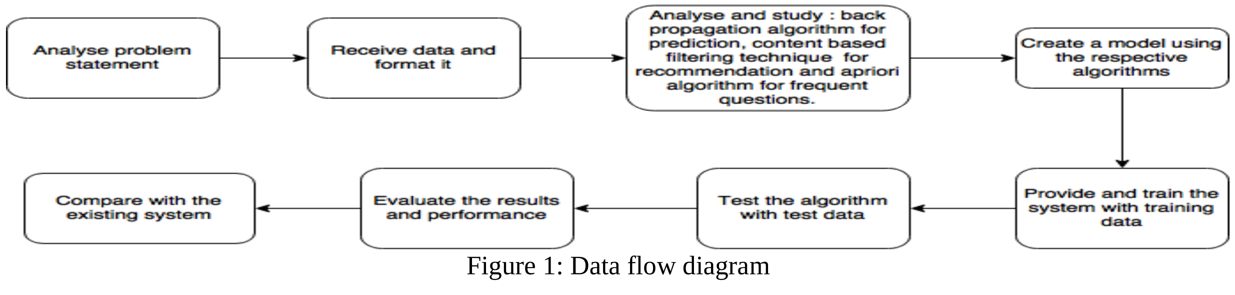
1. INTRODUCTION

Big data analytics is the trending term for the non-traditional methods and technologies that are required to gather, process, organise, and gather insights from data generated from various sources. While the issue of dealing with data that over runs the computing power or storage of a single computer is a known fact, the scale, pervasiveness applications and value of this type of computing has greatly developed in recent year. As universities are becoming highly competitive on the basis of student performances there is a demanding need to keep track of the progress of the institution. This is enabled by periodically monitoring and analysing the mark records of the university while manual processing is reduced and automation takes a leap ahead. Artificial neural networks replicates the processing style of the human brain as a base for developing algorithms that can be utilized for modeling prediction problems and complex patterns. The main advantages of using an ANN is the capability to achieve customized learning that enables the network of neurons to normalize, format and generalize their inputs for processing. Training is done on a dense or strongly connected electronic network of neurons. There exists three layers that constitute a neural network, they are the input layer, hidden layer and the output layer. The input layer is fed with the training data and is transferred to the hidden layer. The hidden layer is responsible for computing data and training the network by modifying the weights. The corresponding output is then directed to the output layer. There are various applications where neural networks have proved its stand in areas such as image recognition, recommendation systems, predictive modeling travelling salesman and various miscellaneous applications. This paper focuses on implementing a student performance analyser and recommender that uses predictive modeling and content based filtering recommendations to build a interactive application. Predictive modeling uses probability and data mining concepts to forecast outcomes. Each model consists of various predicting factors and variables that are likely to alter the outcomes in future. Predictive modeling can either employ a linear equation or a complex neural network that work on specialized software. Using predictive modeling techniques, data that is generated from various sources and fed into the model is said to increase the accuracy of the predicted outcome. Using the datasets prediction of expected outcomes can be carried out. The accuracy of the prediction usually depend on the size of the training data. Usually larger the training data more accurate the prediction outcomes tend to be, but there are various other cases where increased or decreased training data size leads to over fitting or under fitting data models. Thus selecting and analysing the optimal training data is important in prediction modelling. Back propagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. Back propagation is a special case of an older and more general technique called automatic differentiation. In the context of learning, back propagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function.

A recommender system is a subcategory of information filtering system whose main objective is to predict the "ratings" or "preference" that a user gives to an item.With the enormous amount of information generated on world wide web and with significant increase in the number of users, it definitely becomes important to retrieve, map and provide users with the relevant piece of information according to their interests. Various organisations nowadays are developing intelligent smart recommendation engines by studying historic behavior of their users, hence providing them recommendations and choices of their interests. Recommendation systems generate recommendations lists of users interests in one of the two ways –either through collaborative filtering or through content-based filtering (also known as the personality-based approach). Collaborative based approaches builds up a model based on the user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is used to predict item elements (or ratings for items) that the user may be interested in. Content-based filtering approaches uses a category of discrete item features in order to recommend additional items with similar characteristics. Deep Learning (DL) is one of the trending features that has been incorporated in Recommender Systems. In the recent years DNN have exhibited tremendous developmental success in speech recognition, computer vision, and natural language processing (NLP). Deep learning that consists of deep neural network methods are also emerging as powerful tool to tackle Recommender Systems tasks.The flow of the system as represented in Figure 1 is as follows: Initially need for the analyser and recommender is analysed and checked for its requirements. The key raw data is then collected and formatted to a csv format that can be processed by the analyser and recommender. The data includes the student mark files and google forms that consists of students preferences that are directly mapped to an excel file. Once the data is collected the appropriate algorithm that is suitable for the size and format of data is implemented. Since neural networks are used the data is then subjected to splitting of the data into training and testing data. The algorithms are then trained with the data using a number of input parameters and hidden layers for training. Once the data is trained the algorithm is exposed to the test data to analyse the accuracy level. Using the performance metrics the data is checked for best fitting or over fitting. Once the algorithm is evaluated and is trained well using the loss functions it is integrated to the user interface using the shiny package in R to create the students performance analyser and recommender system.

2. RELATED WORKS

Analysis of student's performance is very essential these days as the number of educational institutions keeps increasing day by day. A number of system designs for this analysis have been proposed in the past using different models such as decision trees, neural networks, association rules, Naive-Bayes algorithm, K-nearest neighbour algorithm and support vector machine. As the number of attributes or factors considered for classification of data is important , a detailed analysis of the various key factors that affect the student's performance is done in [1], [3]. In [1], it is said that the number of attributes that are to be considered for inference such that the appropriate algorithm can be applied and the various algorithms that can be applied based on whether the data is a categorical response type or continuous response type. Further, data type must be checked for applying decision tree algorithms such that data becomes compatible to implement. In [3], the paper does a comparative study on the performance analyser of students based on 2 factors: the attributes used for prediction and the method or algorithms used in prediction. Neural Network is a supervised learning method that gave the highest prediction accuracy of 98% followed by decision trees with an accuracy of 91% because of the influence from main attributes. The latest advancement in this field of artificial intelligence is the birth of deep neural networks and convolution neural networks that works on much more complicated data as in [7]. The attributes used in our ANN model are the result of hybridization of two features, which are internal and external assessments. A survey of work done in existing systems for student performance analysis and monitoring along with the survey to understand and analyze the existing system and algorithms that are used in it is done in [4].


Analyse problem statement
Receive data and format it
Analyse and study : back propagation algorithm for prediction, content based filtering technique for recommendation and apriori algorithm for frequent questions.
Create a model using the respective algorithms
Provide and train the system with training data
Test the algorithm with test data
Evaluate the results and performance
Compare with the existing system
Figure 1: Data flow diagram

In [5], an analysis of the various comparisons between algorithms such as ID3 a supervised learning technique that is based on the concept of concept learning system and C4.5 that focuses on data splitting based on attributes.C4.4 has an advanced version C5 that is more efficient.

A system that will analyze student performance and will guide them by displaying the areas where they need improvement, in order to contribute to a student's overall development by generating a scorecard for the same as mentioned in [1], [3], [4]. Further, in order to validate our model we have used the multiple cross-validation test as mentioned in [8] to assess the efficiency and accuracy of our predictions. A recommendation system is used to recommend papers to students and to recommend areas of improvement to the students.

In [2], we have a comparison of recommender system techniques with traditional regression methods such as logistic/linear regression by using educational data for intelligent tutoring systems.

Experimental results in [6] show that rather than a single approach, combined features of content based filtering, collaborative filtering and associative rules provides better results. The main goal of the recommendation is to recommend based on the past choices of individual students and the respective opinions on subjects and decision factors. Thus instead of collaborative filtering the best approach for our project is found to be the content based filtering as mentioned in [2].

In [9] the paper improves the algorithm using interest sets and frequency thresholds to improve the performance. In [10] the traditional back propagation algorithm is modified to reduce the error rate of prediction.

3. SYSTEM DESIGN

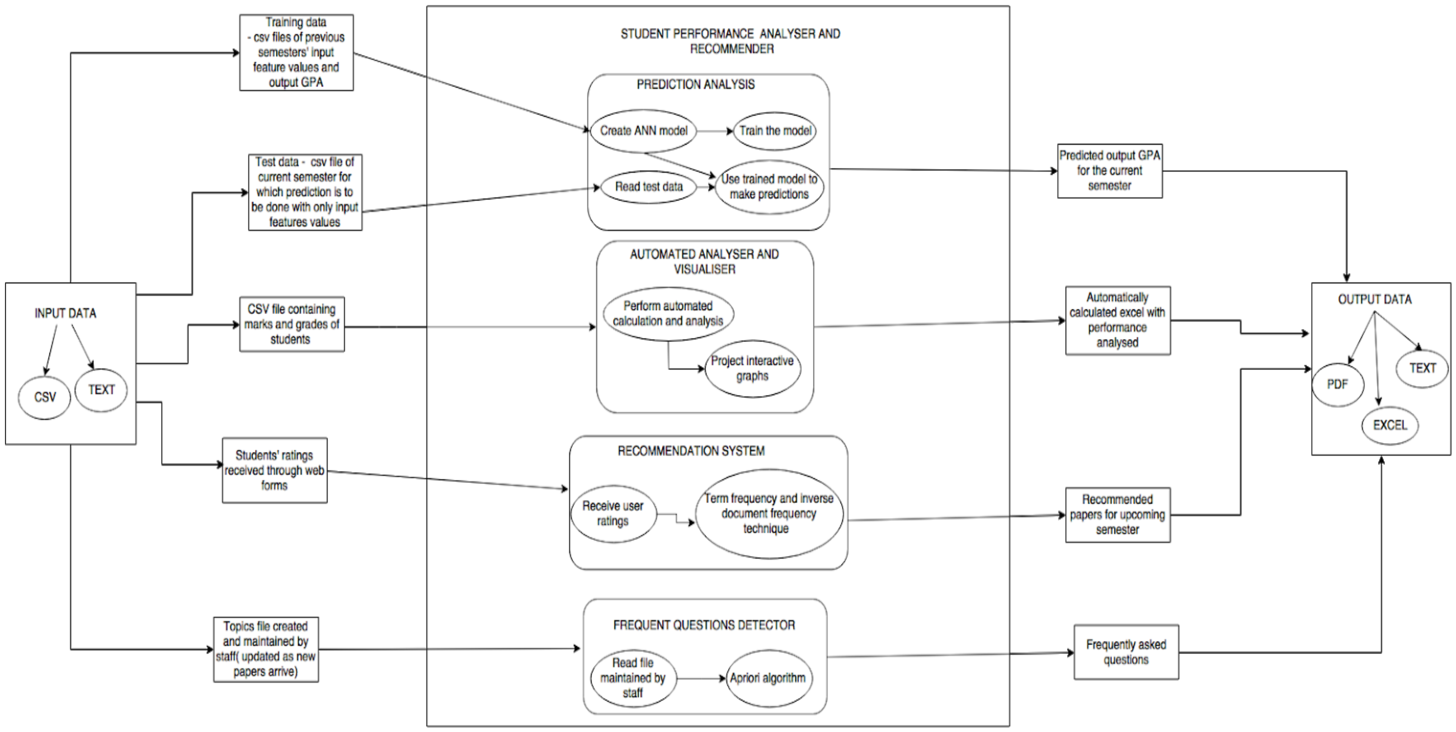

INPUT DATA
CSV
TEXT
Training data - csv files of previous semesters' input feature values and output GPA
Test data - csv file of current semester for which prediction is to be done with only input features values
CSV file containing marks and grades of students
Students' ratings received through web forms
Topics file created and maintained by staff(updated as new papers arrive)
STUDENT PERFORMANCE ANALYSER AND RECOMMENDER
PREDICTION ANALYSIS
Create ANN model
Train the model
Read test data
Use trained model to make predictions
AUTOMATED ANALYSER AND VISUALISER
Perform automated calculation and analysis
Project interactive graphs
RECOMMENDATION SYSTEM
Receive user ratings
Term frequency and inverse document frequency technique
FREQUENT QUESTIONS DETECTOR
Read file maintained by staff
Apriori algorithm
Predicted output GPA for the current semester
Automatically calculated excel with performance analysed
Recommended papers for upcoming semester
Frequently asked questions
OUTPUT DATA
PDF
TEXT
EXCEL

Figure 2: System architecture

The system we have proposed is a multipurpose system which is useful to both staff and students in an educational institution. The staff can use it to automate the manual process of analysing student's performance to determine their levels of success and provide them with extra coaching in their areas of weakness. Students can use the system to get recommendations of papers that they could opt for based on an individual's area of interest and once he/she has opted for a paper they can view the frequently asked questions from the previous years' question papers which is also automated thereby avoiding the manual work of going through each question paper before the exam. The flow of the data in our system is represented in Figure 1 whereas the overall system architecture with its modules is shown in Figure 2. Hence, there are four modules:

1. A predictive analyser, that provides a prediction of the future performance of students based on a number of decision factors.

2. An automated data visualiser, that can provide an automated analysis of student’s performance and a visual representation to view their strengths and weaknesses.
3. A recommender, that can be used to suggest subject papers to students based on their interests.
4. A frequent questions detector, that can give the most frequently asked questions in a given subject to the students.

4. PREDICTIVE ANALYSIS:

Neural Networks:Neural networks is defined by a structured series of layers, each layer has a number of neurons. The network resembles the human brain and its working. Each neuron in the network works similar to the neurons in the human brain and acts as the basic functional unit of the network by producing an activation based on the outputs of the previous layers and a set of weights. The two basic algorithms used in neural networks are the forward propagation algorithm which uses feed forward neural networks and the back propagation algorithm.

Back Propagation Algorithm:
The backpropagation algorithm is the fastest algorithm which employs the idea of supervised learning for multilayer feed forward networks used to compute gradients, much more accurately and faster than the earlier approaches to learning, in order to solve problems which had been previously unsolvable. The basic principle behind a backpropagation algorithm is to model a function by changing the internal weights of input signals to derive at the expected output. The system is trained under supervision, where the error between the output and the expected output is used to modify the system’s internal state. The error defined for a single perceptron is generalised to include all the squared error at the outputs k = 1,....,K.

$$E_p = (1/2) \sum_{k=1}^{K} [d_{pk} - o_{pk}]$$

where p is the pth pattern and dp is the output for the pth pattern. A standard network structure used in this algorithm has one input layer, one or many hidden layer and one output layer. Backpropagation can be used for both classification and regression problems.
The basic steps in backpropagation algorithm are:

1. Initialize the neural network.
2. Do forward propagation using feed forward networks.
3. Propagate the error backwards.
4. Train the network with the training dataset.
5. Predict the results for the provided test input.

Every neural network has a number of neurons. Each neuron has a set of weights, such that there is one weight for each input and an additional weight for the bias. The network can have any number of hidden layers in between the input and output layer and computations take place at these layers. The network weights are initialised to small random values. In forwards propagation, we calculate the output from a neural network by propagating an input signal through each layer until the values are seen at the output layer. Using this technique, we can train the network and correct errors, so that predictions can be made on new data. The forward propagation has three steps: neuron activation, neuron transfer and forward propagation. The first step is to calculate the activation of each neuron given an input.
Neuron activation is calculated as:
activation = [∑ weight(i) ∗ input(i)] + bias
where weight is a neuron weight, input is a the input signal or the intermediate results from the hidden layer, i is the index of a weight or an input and bias is a special weight that has no input to multiply with. Once the neuron gets activated, there is need to apply different transfer functions to transfer the activation and see what the neuron output is actually. The sigmoid activation functions which is also called the logistic function can take any input value and produce a number between 0 and 1. Since it can easily calculate the derivative used to propagate errors , it is the most commonly used transfer function. Other transfer functions are the tanh function (hyperbolic tangent) to transfer outputs and the rectifier transfer function. Next, we create the feed-forward network by calculating the outputs for each neuron. Error is calculated between the expected outputs and the outputs forward propagated from the network based on calculus. First we calculate the slope of a neuron , given an output value from the neuron.

The derivative is calculated as follows:
derivative = output * (1.0 − output) .

The error for a given neuron can be calculated as follows:
error = (actual output − output) ∗ derivative(output)
where actual output is the expected or original output value for the neuron, output is the calculated or obtained output value for the neuron and derivative calculates the slope of the neuron’s output value. This error calculation is used for neurons in the output layer where the expected value is the class value itself. The error signal for a neuron in the hidden layer is calculated as the weighted error of each neuron in the output layer and the errors travel back along the weights of the output layer to the neurons in the hidden layer. The back-propagated error signal is accumulated and then used to determine the error for the neuron in the hidden layer, as follows: error = weight(n) ∗ error(m) ∗ derivative(output)
where error(m) is the error signal from the mth neuron in the output layer, weight(n) is the weight that connects the nth neuron to the current neuron and output is the output for the current neuron.
Once the modelling of the network is done, training of the model created is to be carried out using stochastic gradient descent. This process involves multiple iterations of exposing a training dataset to the network and for each row of data the following steps are repeated: providing the inputs to the model, assessing the error, backpropagating the error and updating the network weights.
Network weights are updated as follows:
weight = weight + (learning − rate) ∗ error ∗ input

where weight is a given weight, learning-rate is a parameter that the user must specify and is used to control the rate at which the weights are to be changed in order to correct the error, error is the error calculated by the backpropagation procedure for the neuron and input is the input value that caused the error. The same procedure can be used for updating the bias weight, with the exception that there is no input term. Once the network model is trained, we use it to make predictions where the output values are given directly as the predicted outcome.

In our project, we use the backpropagation algorithm to make predictions of student's GPA based on a number of decision factors. This prediction can help staff to identify if extra coaching must be given to specific students. We train the neural network using training datasets of previous semesters, the prediction algorithm processes the input students mark data provided by the user as a training set and for a new semester's input data (test data) predictions are made. Multiple decision factors such as gender, internal exam scores, past semester GPA's, paper difficulty level are considered for the prediction as a student's GPA depends on not just their internal scores. The graph in Figure 3 shows the neural network model generated for the prediction and contains the computed weights of each neuron along with the bias.

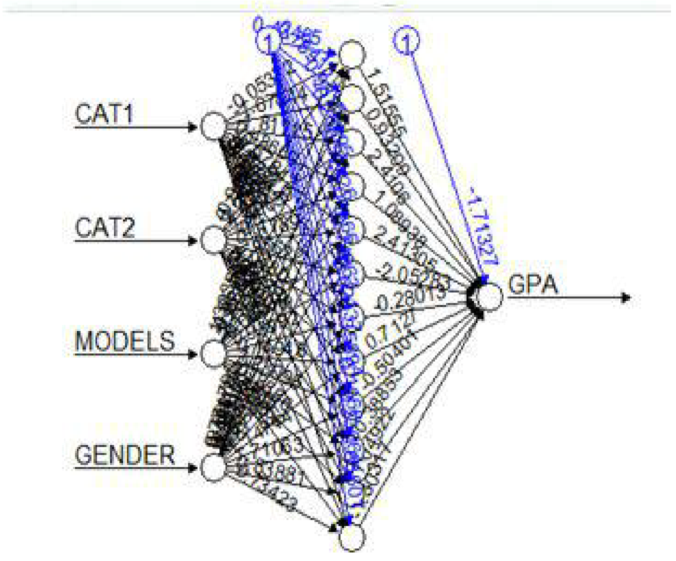


Figure 3: Neural network model used in prediction of GPA

5. AUTOMATED DATA VISUALISATION:

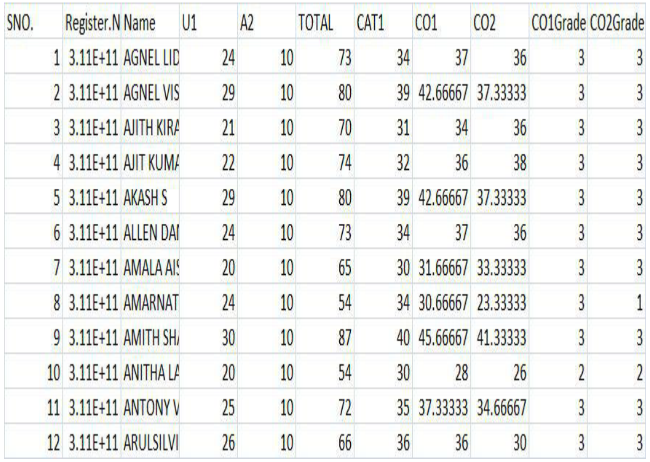
Data visualization is essential to interpret and understand the significance of data by enabling it in a visual block context. Most often correlations, patterns and trends might go undetected in traditional text-based data can be exposed and vulnerable to be recognized easier with data visualization techniques.

There following are the steps to visualize data:

1. A clear idea about the problem is proposed that includes the various datasets and data that are to be compared.(avoid plotting variables of different type on the same chart)
2. Visualize data using bar chart, line chart, flow chart, scatter plot, surface plot, map, networks etc depending on what data are available.
3. Identify messages of the visualization, and generate the most effective indicator.
4. Choose the right chart type.
5. Use colour, size, shapes, scales and labels to direct attention to the key messages.

A. Overall Automation:

The analyser and recommendation interface is implemented using Rshiny, a interactive web application R package. It consists of a user and server code that helps integrate the modules proposed. Shiny has the advantage of combining the computational power of R with the interactivity of modern web. The analyser and recommender application consists of a prediction, analysis ,frequent question analyser, individual students performance analyser and a recommender module. Interaction with the application involves users to upload the files to be analysed in a csv format. The input files are linked to the respective modules based on the users requirements. By automating the modules the manual task of calculating and computing student records are reduced. automated documents such as a analysed class performance ,individual students performance, predictive gpa and question sets and its corresponding graphical visualisation are generated as text and pdf documents. An interesting feature of the application is the ability to be flexible subjected to user give parameters that are to be visualized and analysed.



SNO.	Register.N	Name	U1	A2	TOTAL	CAT1	CO1	CO2	CO1Grade	CO2Grade
1	3.11E+11	AGNEL LID	24	10	73	34	37	36	3	3
2	3.11E+11	AGNEL VIS	29	10	80	39	42.66667	37.33333	3	3
3	3.11E+11	AJITH KIRA	21	10	70	31	34	36	3	3
4	3.11E+11	AJIT KUMA	22	10	74	32	36	38	3	3
5	3.11E+11	AKASH S	29	10	80	39	42.66667	37.33333	3	3
6	3.11E+11	ALLEN DAI	24	10	73	34	37	36	3	3
7	3.11E+11	AMALA AIS	20	10	65	30	31.66667	33.33333	3	3
8	3.11E+11	AMARNAT	24	10	54	34	30.66667	23.33333	3	1
9	3.11E+11	AMITH SHA	30	10	87	40	45.66667	41.33333	3	3
10	3.11E+11	ANITHA LA	20	10	54	30	28	26	2	2
11	3.11E+11	ANTONY V	25	10	72	35	37.33333	34.66667	3	3
12	3.11E+11	ARULSILVI	26	10	66	36	36	30	3	3

Figure 4: Excel sheet containing the automated results

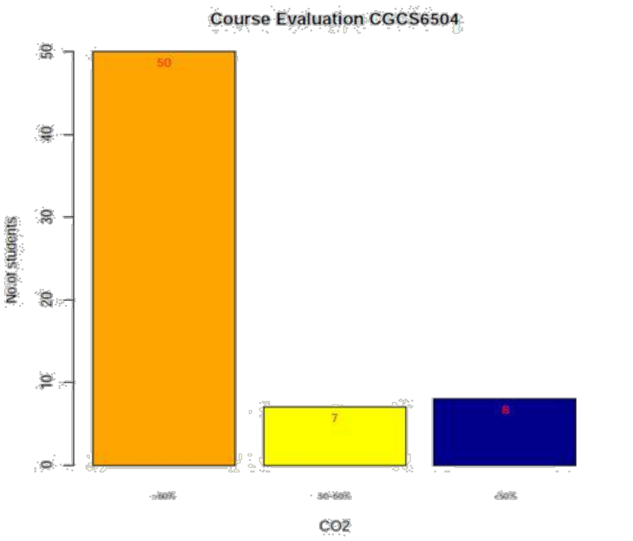


Figure 5: Graph for evaluation of course outcome 2

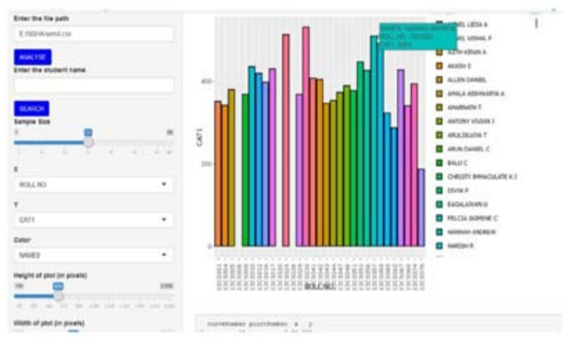


Figure 6: Interactive graph for overall analysis

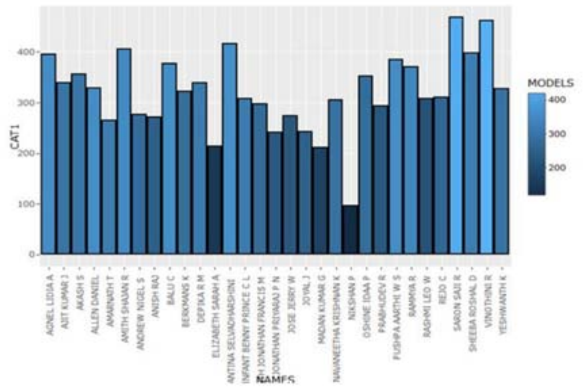


Figure 7: Interactive graph for analysis of student's model exam performance

Figure 4 contains the output of our automation module. This automation module takes as input the unit test and CAT marks, performs an automated analysis and provides an excel sheet as output which is shown in the figure. Further we also have an automation module to calculate the values of GPA and CGPA of students when their grades are provided. Figure 5 is one of the graphs used for visualisation that is used in the result analysis process. This helps to keep track of the student's performance in an easy manner.

B. Individual Analysis:

The individual analysis consists of the following modules:

Overall Analysis Module:The analysis module computes the overall performance of the students data.This generates a course outcome document that analysis the pass percentage and outcomes of course specific objectives.

Figures 6 and 7 are the interactive graphs that get generated for the overall performance analysis. These graphs are user friendly and dynamically changes based on the user's clicks and selections.

Individual Analysis Module:

This module analyses and compares the individual student performance with the overall performance. This enables users to manipulate with the parametric features to obtain a correlation between the factors. Interactive graphs are built that enables individual student records to be visualized. The analyser consists of options that enable the users to increase or decrease the

sample size that is to be visualized, adjusting the parametric values alone the x and y axis. Clicking on the graph student respective details are displayed enabling easy and quick analysis of student data. Figure 8 shows an interactive graph that provides a particular student's performance analysis when his/her name is entered.

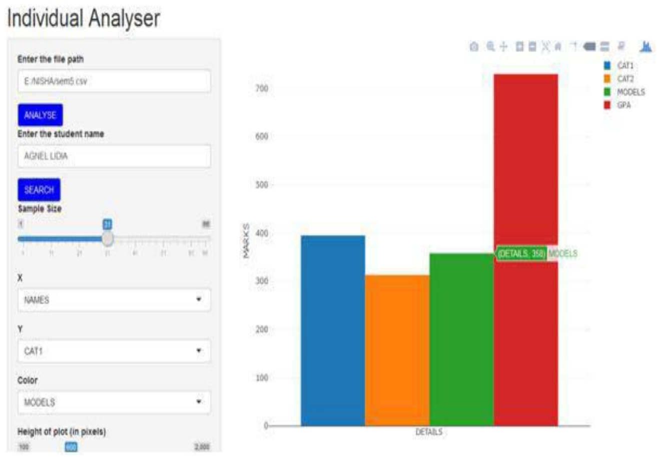


Figure 8: Individual student's interactive graph

6. RECOMMENDATION SYSTEM

Recommender systems are typically information filtering and retrieval systems which recommend and personalize the information that users are interested in and shown to the user, relevance of information and so on based on the user's profile. Recommendation systems are used to recommend different features such as movies, books etc., which is of high advantage to both service providers and users as they reduce transaction costs of finding and selecting items. There are two types of filtering: content based filtering and collaboration based filtering. A hybrid approach that contains a combination of both content and collaborative approaches is also used widely these days.

Content based filtering:
A content based recommendation system uses data provided by the user either through ratings or by clicking on a link , that is either implicitly or explicitly. Using this input data, a user profile is created that is used to make recommendations to the user based on his/her area of interest. The accuracy improves as more inputs in the form of actions are provided by the user to the system.

Term frequency and Inverse document frequency algorithm:
The most famous and traditionally used techniques in content based filtering are Term Frequency (TF) and Inverse Document Frequency (IDF) to identify the relative importance of the items to be recommended.
Term Frequency counts the frequency of the words in a document whereas document frequency is the count of the frequency of a document among a group of documents. Inverse of this value gives the Inverse document frequency. is simply the frequency of a word in a document. IDF gives the inverse of the document frequency among the whole collection of documents. TF-IDF is the most commonly used technique because it negates the effect of the high frequency words based on the importance of the item or document making the values more comparable to each other.
A simple calculation used are:

$$w_{t,d} = 1 + \log_{10} tf_{t,d} \qquad , \text{if } tf_{t,d} > 0$$

The value of $w_{t,d}$ becomes 0 otherwise. In this
$w_{t,d}$ is the weighted term frequency that is a logarithmically scaled frequency and is based on the term frequency $tf_{t,d}$.

$$idf(t,D)=\log \frac{N}{|d\varepsilon D:t\varepsilon T|}, \text{if } tf_{t,d} > 0$$

where N is the number of documents in the corpus and $d \varepsilon D:t \varepsilon T$ is the number of documents that $idf_{t,D}$ contain the term t. If $tf_{t,d}$ is not equal to 0 , the value of changes by modifying the denominator as $1 + |\{d\varepsilon D : t\varepsilon T \}|$.
The final term frequency-inverse document frequency is got by the formula:

$$tfidf_{t,d,D} = w_{t,d} * idf(t,D)$$

To determine if certain items are close to each
other or to the user profile, we use the vector space model which computes the proximity based on the angle between the vectors. In this model, we determine the similarity between the various items and users by the following method. Each item considered for recommendation is stored as a vector of attributes in an n-dimensional space and user profile vectors are created based on his/her actions. The angles between the vectors - item vectors and user profile vectors are calculated and this provides the details of similarity between an item and a user. The same method can be used to determine the similarity between the various items by calculating the angles between the various item vectors. The user's likes/dislikes or the area of interest of a given user can be determined by taking the cosine of the angle between the user profile vector and the document vector. We have used the cosine value as cosine increases the value with decreasing angle value which shows more similarity.
In order to perform computations to recommend items based on user's interest, an easier way of computation is by normalizing the values in the vector. After normalization, the vectors become vectors of length 1 and then the cosine value is simply the sum product of vectors. We can also convert the data into a binary representation (i.e., as 1/0 form) which provides metrics based on the context of use. Another metric used is the count representation which is more commonly used in complex systems such as search query engines upon the context of use.
In our project, we have implemented the recommendation system that uses content based filtering technique to determine subject papers for students based on their areas of interest. It can determine the subject papers based on their strengths and weaknesses. As the latest regulation under Anna University provides students with the privilege of opting for papers in a given semester, this recommender serves as an assistant to students, helping them choose papers based on their areas of interest. Students were made to rate papers already chosen to determine their area of interest and recommendations of papers in those fields were provided as output using the term frequency-inverse document frequency technique.

7. AUTOMATED FREQUENT QUESTIONS DETECTOR

Association Rules and Apriori intro:
The Apriori algorithm is an effective algorithm for mining frequent itemsets for boolean association rules. It uses a "bottom" up approach where existing subsets that are frequent are extended one item at a time. Apriori uses breadth first search and a hash tree structure to count candidate itemsets efficiently. The frequent item sets determined by Apriori can be used to determine association rules. This has applications in domains such as stock market analysis etc. With this concept this paper extends the concept to finding frequent questions with higher occurrence probability. After that, it scans the transaction database to determine frequent item sets among the candidates. In this paper the apriori algorithm is implemented to find the frequent questions from the previous years question papers. The question topics are given as a item input formatted according to the algorithm specifications as a text file. On the given input the frequency function is applied that generates the frequent itemsets of possible questions as a text document.

Item frequency function:
It derives a function that is generic in nature 'itemFrequency' and S4 methods to get the frequency/support for all individual items of the itemset based on a itemMatrix.
itemFrequency(x, type, weighted = FALSE)

The algorithm pseudo code below consists of a transaction database T and a support threshold usually consisting of theoretical notations that are employed, note that T is a multiset. C(k) is the candidate set for level k. At each step, it is assumed to be that the algorithm generates candidate sets from a larger item sets of the preceding level, heeding the downward closure lemma. C accesses a data structure field that represents candidate set C, which is assumed to be zero at the initial stage.

Apriori Algorithm:

```
Begin
   I <- itemset
   S <- support
   L1 <- {frequent item1 - itemset}
   K <- 2
   While L(k-1) != NULL
    Temp <- Candidate I(L(k-1))
    C(k) <- Frequency Of I (Temp)
    L(k) <- Compare I with min(S){C(k),minsup}
    K <- K+1
End while
Return L
End
```

Using the algorithm the frequent questions are iterated through a series of itemfrequency functions to obtain a text document of the question that are the frequent itemsets. It also generates a graph that illustrates the relationship and hierarchy of subject topics that are linked to help students analyse the flow of concepts and the related questions that are likely to appear.

8. EXPERIMENTAL RESULTS

1. Prediction analyser:

The prediction analyser is used for the computation of student's results-GPA based on a number of input factors.

We have integrated a number of input factors such as internal marks, attendance, gender etc., which play a backbone in assessing student's performance. The inputs given are continuous assessment 1 marks, continuous assessment 2 marks, model examination marks and the gender of the student. Training data of semesters before the final year are used to model the network and the final model is tested with the seventh semester data. Computation of the individual student's GPA is done and returned as an excel file.

Table 1: Comparison of the predicted GPA with actual GPA.

REG .NO.	CAT 1	CAT 2	MODELS	GENDER	GPA	PREDICTED GPA
4	339	340	301	M	7.5	7.08
10	405	363	350	M	7	7.081
11	288	285	263	F	5.8	5.781
31	240	245	230	M	6.2	6.909
45	207	247	180	M	5.3	5.778
47	409	390	360	F	7.04	7.08
12	337	292	323	M	6.8	4.67118
32	273	254	227	M	5.2	7.081

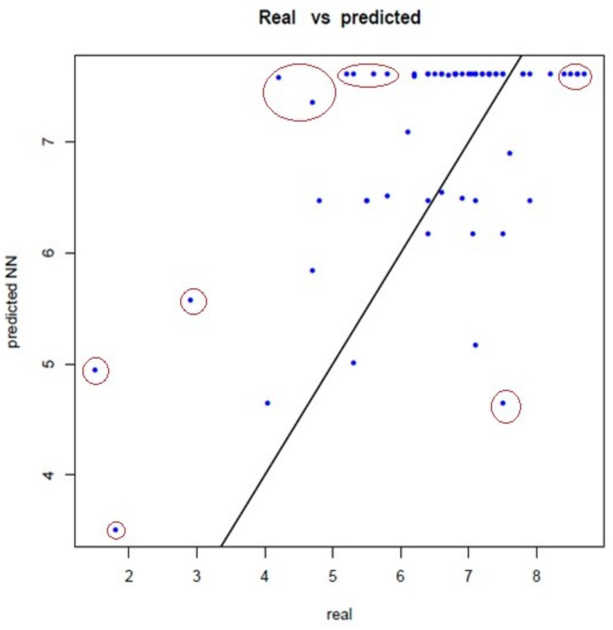
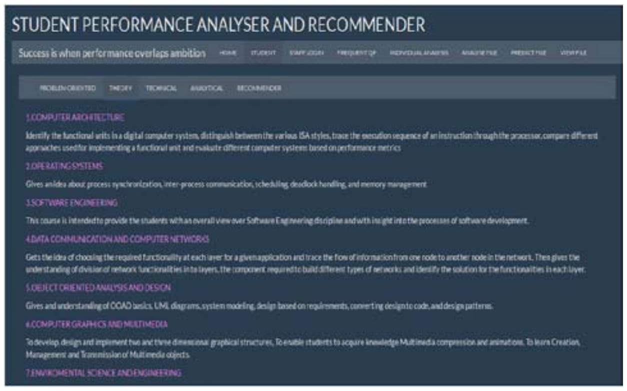


Figure 9: Comparison graph - real versus predicted GPA

A sample set of test data is shown in Table 1 from which a comparison of the predicted GPA with actual values can be obtained. Figure 9 contains the graph of real versus predicted value. A few outliers are found in the graph but a major set of points lie about the absolute line.

2. Student's subject recommender:



1.COMPUTER ARCHITECTURE

Identify the functional units in a digital computer system, distinguish between the various ISA styles, trace the execution sequence of an instruction through the processor, compare different approaches used for implementing a functional unit and evaluate different computer systems based on performance metrics

2.OPERATING SYSTEMS

Gives an idea about process synchronization, inter-process communication, scheduling, deadlock handling, and memory management

3.SOFTWARE ENGINEERING

This course is intended to provide the students with an overall view over Software Engineering discipline and with insight into the processes of software development.

4.DATA COMMUNICATION AND COMPUTER NETWORKS

Gets the idea of choosing the required functionality at each layer for a given application and trace the flow of information from one node to another node in the network. Then gives the understanding of division of network functionalities in to layers, the component required to build different types of networks and identify the solution for the functionalities in each layer.

5.OBJECT ORIENTED ANALYSIS AND DESIGN

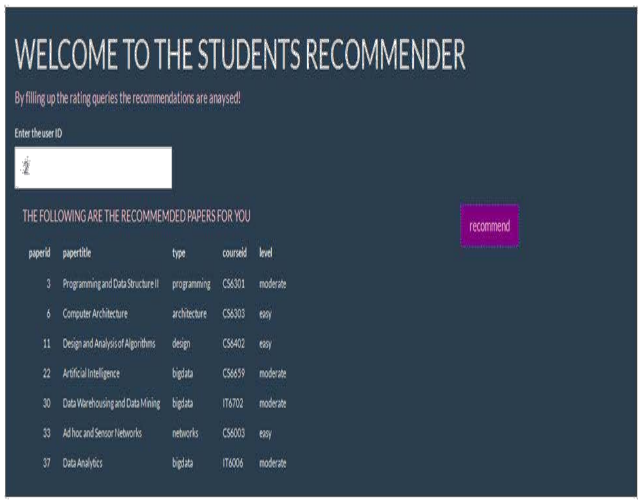
Gives and understanding of OOAD basics, UML diagrams, system modeling, design based on requirements, converting design to code, and design patterns.

6.COMPUTER GRAPHICS AND MULTIMEDIA

To develop, design and implement two and three dimensional graphical structures. To enable students to acquire knowledge Multimedia compression and animations. To learn Creation, Management and Transmission of Multimedia objects.

7.ENVIROMENTAL SCIENCE AND ENGINEERING

Figure 10: User interface of recommender module



By filling up the rating queries the recommendations are anaysed!

Enter the user ID

2

THE FOLLOWING ARE THE RECOMMEMDED PAPERS FOR YOU

paperid	papertitle	type	courseid	level
3	Programming and Data Structure II	programming	CS6301	moderate
6	Computer Architecture	architecture	CS6303	easy
11	Design and Analysis of Algorithms	design	CS6402	easy
22	Artificial Intelligence	bigdata	CS6659	moderate
30	Data Warehousing and Data Mining	bigdata	IT6702	moderate
33	Ad hoc and Sensor Networks	networks	CS6003	easy
37	Data Analytics	bigdata	IT6006	moderate

Figure 11: Recommendation module with recommended papers for user

The recommendation module is used by students to see what papers can be opted by them based on their areas of interest and ratings of previously selected papers. A general idea of all the papers is provided to students so that they can have a detailed idea of the subjects as in Figure 10. Further, based on ratings collected through web forms papers are recommended to a particular student or user as provided in Figure 11.

3.Frequent questions detector:

The frequent questions detector provides assistance to students in finding out the most frequently asked questions from the previous years' papers. This could save time from the manual work of skimming and scanning each and every paper during preparation for e examinations. Figure 12 has the graph that contains the list of topics with their frequency of occurrence ranked from the most frequently asked to the least frequently asked question. For this we have used the apriori algorithm and Figure 12 contains the output for the paper - Programming and Datastructures.

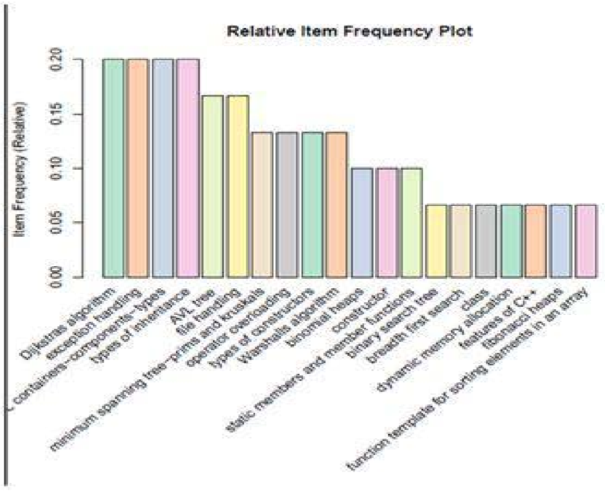


Figure 12: Graph with questions and their frequency of occurrence

9. PERFORMANCE METRICS:

The prediction analyzer is used for the computation of student's results-GPA based on a number of input factors. The inputs given are continuous assessment 1 marks, continuous assessment 2 marks, model examination marks and the gender of the student. Training data of semesters before the final year are used to model the network and the final model is tested with the seventh semester data. Computation of the individual student's GPA is done and returned as an excel file. A sample set of test data is shown in Table 1 from which a comparison of the predicted GPA with actual values can be obtained. Figure 9 contains the graph of real versus predicted value. A few outliers are found in the graph but a major set of points lie about the absolute line.

It can be seen that the last three rows of Table1 has predictions with varied difference from the actual values, these are the outliers in the prediction graph Figure 9. It can also be inferred from the graphs in Figure 14 and Figure 15 that the error rate is reduced as multiple input features are added and also through the use of neural networks the accuracy is improved when compared to the predictions done through the use of decision trees.

The efficiency and effectiveness of our predictor is observed using the performance metrics evaluation of cross validation. Cross validation typically uses the k-fold cross-validation method to perform cross-validation. In k-fold cross-validation, the input data is split into k subsets of data (also known as folds). Training the model is done on all but one (k-1) of the subsets, and then evaluated on the subset that was not used for training. This process is repeated k times, with a different subset reserved for evaluation (and excluded from training) each time. The outcome of this evaluation has proved that our predictor provides greater accuracy with the hybridized combination of input attributes.

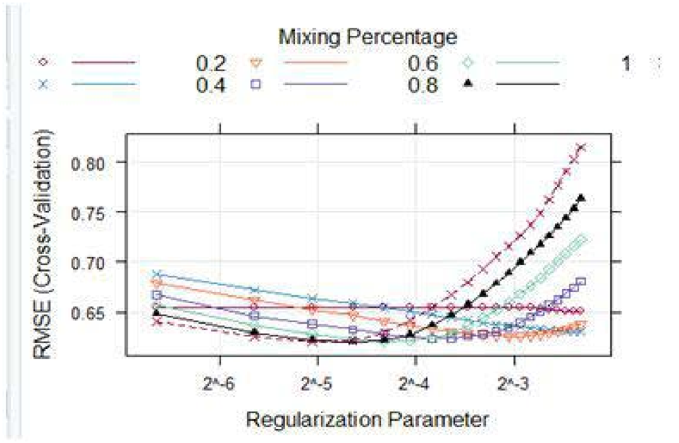


Figure 13: Regularization parameters versus RMSE(cross validation)

The entire dataset was split into 80% of training data and 20% of the testing data. Alpha and lambda are the tuning parameters that were used for training the data. Based on the model the MAE (Mean Absolute error) and the RMSE (Root mean squared error) were used to evaluate the performance. On training the data, the values of alpha and lambda that gave the best RMSE and MAE of 0.7290701 and 0.5710924 respectively were 0.4 and 0.13 respectively. A comparison between a single input feature and multiple input features was also done to prove an increase in accuracy and reduced error rate when using multiple input feature model. Using multiple input features, the RMSE and the MAE results were 0.6640597 and 0.5523700 respectively which is lesser than the corresponding single input feature values of 0.7290701 and 0.5710924 respectively.
The graph in Figure 13 has been plotted with regularization parameters along the x axis and the RMSE (cross-validation) along the y axis. It has been observed that the training considers different percentages of mixing data samples. It was found that according to the data size the mixing percentage of 0.2 resulted in a minimum RMSE that reduces the error rate and hence establishes higher accuracy when compared to the other algorithm models.

Table 2: Comparison of performance metrics between existing system and proposed system

Performance Metric	Performance metrics of system that uses decision trees		Performance metrics of the proposed system using neural networks
	ID3	C4.5	
RMSE	0.7778	0.75145	0.6640597
MSE	0.604	0.5646	0.5523700

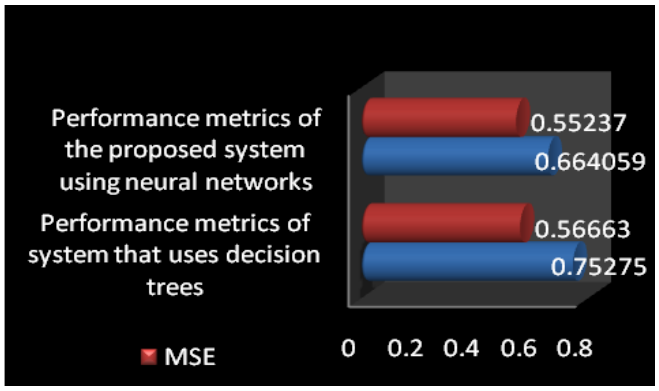


Figure 14: Comparison of performance metrics between existing system and proposed system.

Table 3: Comparison of performance metrics between system with single input feature and system with multiple input features.

Performance Metrics	ANN Model trained with only internal marks	ANN Model trained with additional input features
RMSE	0.7290701	0.6640597
MAE	0.5710924	0.5523700

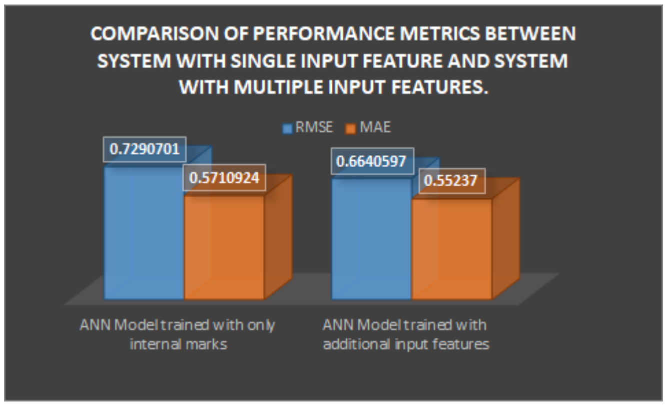


Figure 15: Comparison of performance metrics between system with single input feature and system with multiple input features.

10. CONCLUSION & FUTURE WORKS

In this paper, we have proposed a system that will enable automated analysis and visualization of the academic performance of students. It will enable staff to keep track of subjects they are handling, perform an analysis of the results generated, avoid manual calculation and aggregation of student marks for generating reports. Further, a predictive system using artificial neural networks has been used to predict the student's GPA which assists staff and students in making efforts to improve their GPAs. Also, a recommendation system that assists students in suggesting the various papers that they can opt for thereby assisting them in precisely choosing the subjects they are strong in by integration of standardized tools for students psychological predictions. Finally, an automated system to identify the frequently occurring questions in the previous years' papers has been proposed to assist students in preparing for their examinations. This system could be improvised in the future by automating many more processes such as automatic attendance checker etc., and integrating it with the above proposed system.

The proposed approach is compared with the previous standard systems and a performance analysis is done to prove that the system works in an intelligent manner with all the integrated modules and is beneficial to both staff and students.

REFERENCES

[1] Brijesh Kumar Baradwaj and Saurabh Pal on "Mining Educational Data to Analyse Students' Performance",(IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 2, No. 6, 2011.

[2] Thai-Nghe, Lucas Drumond, Artus Krohn-Gimberghe and Lars Schmidt-Thieme on "Recommender System for Predicting Student Performance", Elsevier, Procedia Computer Science1 (2010) 2811–2819.

[3] Amirah Mohamed Shahiria, Wahidah Husaina, Nur'aini Abdul Rashida on "A Review on Predicting Student's Performance using Data Mining Techniques", Elsevier, Procedia Computer Science 72 (2015) 414 – 422.

[4] Snehal Kekane, Dipika Khairnar, Rohini Patil, Prof. S. R. Vispute, Prof. N. Gawande on "Automatic Student Performance Analysis and Monitoring" ,International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 1, January 2016.

[5] Badr HSSINA,Abdel Karim MERBOUHA, Hanane EZZIKOURI,Mohammed ERRITALI on "A Comparative Study of Decision Tree ID3 and C4.5".

[6] Praveena Mathew, Bincy Kuriakose and Vinayak Hegde on " Book Recommendation System through content based and collaborative filtering method", IEEE Xplore.

[7] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel and Demis Hassabis on "Mastering the game of Go with deep neural networks and tree search",Nature volume529, pages484–489 (28 January 2016).

[8] Burkhard Rost, Chris Sander on "Combining evolutionary information and neural networks to predict protein secondary structure", Proteins,Volume 19, Issue 1, May1994.

[9] Libing Wu,Kui Gong,Yanxiang He on "A study of improving Apriori Algorithm" IEEE international systems and applications(ISA) (27 May 2010).

[10] W.A.M Ahmed,E.S.M Saad and E.S.A Aziz on " Modified back propagation algorithm for learning artificial neural networks" ,Radio science conference IEEE 27-29 March 2010.