EAI Endorsed Transactions
on Scalable Information System

Research Article EAI.EU

An Automated Recommender System for Educational Institute in India

Mamata Garanayak[1],*, Sipra Sahoo[2], Sachi Nandan Mohanty[3] and Alok Kumar Jagadev[1]

[1]School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, 751024, India

[2]Department of Computer Science & Engineering, Siksha o Anusandhan Deemed to be University, Bhubaneswar Odisha, 751030, India

[3]Department of Computer Science & Engineering, ICFAI Foundation for Higher Education, Hyderabad, Telangana, 50120, India

Abstract

This study aims to suggest a recommender System for undergraduate students who desire to seek admission into engineering courses in different Indian Institute of Technology (IITs) in India. Initially, the focus is to purpose a recommender system for admission into the top 10 IIT on a pilot basis in four common branches such as Electrical Engineering, Computer Science and Engineering, Mechanical Engineering, Civil Engineering. Data were collected from different authentic sources from 2016 to 2018. A model was built to predict the ranks for 2019 for each branch of every IITs. This paper illustrates prediction using Time Series Forecasting and recommendation algorithm using classification techniques. A comparative study of Random Forest Classification and K-Nearest Neighbor classification has been done. Finally, the recommendation algorithm shown reliable results with high accuracy in prediction model. It can be diversify and implement other streams as part of future work.

Keywords: Recommender system, Random Forest Classification, K-Nearest Neighbor (KNN).

Received on 27 September 2019, accepted on 10 February 2020, published on 14 February 2020

doi: 10.4108/eai.13-7-2018.163155

1. Introduction

As RS has emerged as a demanding tool a short time ago, it is extremely important to have a proper understanding of it. One must focus on few important aspects of recommender systems such as "building a recommender system", different techniques to build or develop the recommender system such as DM, collaborative filtering, Content-Base filtering and Context-Aware methods. Different aspects that affect the RSD are the users, domains, and interfaces.

One more important aspect is to safeguard the privacy of the user during this decision making process. The recommender system is used in a wide variety of fields such as music, movies, education, social networks, mobile computing, healthcare, insurance, e-commerce applications and many more.

In the Internet era, the biggest problem for a person who wants to buy something online is not only how to get enough information to make a decision, but also how to make a right decision with that enormous information.

Recommender systems give a piece of guidance about the information, products or services that the user might be fascinated with. Those are quick-witted applications to serve the user in a decision-making process where they wish to pick one item between a potentially enormous set of substitute products and they are presumably between the most eminent applications having a notable smack on

[*]Mamata Garanayak. Email: mamatagaranayak@gmail.com

the performance of e-commerce web sites and the sectors in prevalent.

In this paper the process of developing a recommender system for EI is being analyzed, the application domain of the Education Institutions is considered, algorithms are developed and architectures about recommender systems have been designed.

1.1. Recommender System

RS is a subspace of an information retrieval system that forecasts the "liking" or “rating” that a user would provide to a certain item. We can say it is a software tool /technique which suggests different products, services and information (which can be interpreted as items in short) based on the user's interest to help the user for successful decision making. User preferences and constraints are two keywords that play a vital role in a recommender system. A recommender system is an independent area of research in the Mid-1990s after databases and search engines. Research regarding recommender systems has increased tremendously with the availability of information in the digital world. With the availability of so many choices for the user, the user is confused while choosing the right option or taking the right decision. The recommender system has evolved as a blend to the information overburden problem (IOP) helping the user in the right decision making. The recommender system deals with PR and NPR.

1.2. Approaches of RS

The RS can be classified based on the below techniques adopted for a recommendation. Figure 1 depicts the different types of recommendation systems. At a high level, the recommender systems can be classified as two types such as personalized and non-personalized.

The personalized recommender systems are again classified into various categories such as techniques of collaborative filtering (item and user-based), techniques of content-based filtering, Demographics techniques of filtering, KB and HRS. The Collaborative filtering technique is implemented through two mechanisms. One is UB and another is the IB recommendation. All the techniques are applied as per the need and based on the type of problems at hand.

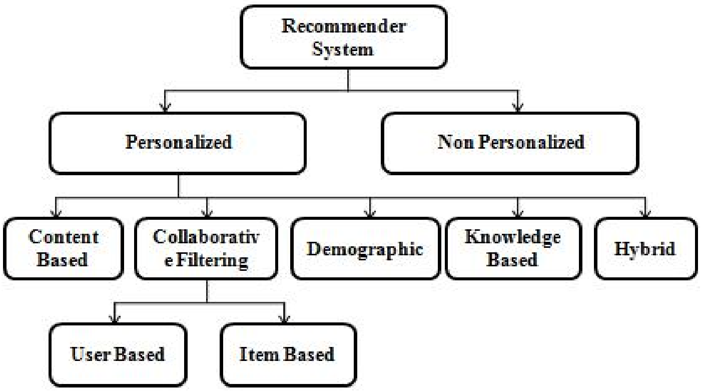


Figure 1. Different types of Recommendation System

CRS also is known as PPC and it recommends users who are active based on the items rated by the other users having the same flavour before. This is the most favoured technique and widely used by the system. CBRS used to suggest the items that are having the same flavour to the items tend to choose by the user before. DBRS has relied on the demographic information of the users. UBRS uses the user preference and rating and considers the convenience of the item for the user. The utility function is calculated for each user separately. KBRS uses knowledge of the users as well as the products. It identifies which products are suitable for which users. Based on the user's requirements the products are recommended. HRS is a combination of more than one of the above techniques to get a more efficient and better recommendation.

A personalized recommendation system suggests the products to a user based on their previous history but a non-personalized recommender system does not take into account the personal predilection of the user. The recommendations produced by these systems are alike for each customer. In the case of online E-Commerce websites, the recommendations can either be manually picked by the online retailer, based on the popularity of items or the recommendations can be the top-N brand new products. For example, Flipcart.com as an anonymous user it shows items that are currently viewed by other members. These systems recommend items to customers based on what other customers have said about the items.

1.3. Predictive Analysis

PA deals with the procedure of taking out useful information from the existing knowledge of data and predicts the future value by analyzing the previous value and trend. It may not result in actual value but can forecast what may happen in the future preserving the authenticity of the data. An analysis is done on the existing data especially with a large number of data and then optimization is done on the data. After that different model is applied a nd implemented to forecast future data. Predictive analytics is defined as the branch of analytics which deals with the prediction of unknown future events. Figure 2. represents the high-level view of how predictive analytics works.

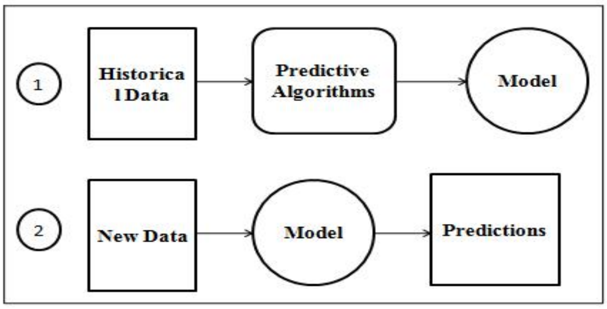


Figure 2. Predictive Analytics

The basic functionality of predictive analytics is that it takes the historical data as input, uses the predictive algorithms and applies the models to train it. It tests the model with the new data and predicts future outcomes. Figure.3. highlights the trend of the plot of value and difficulty in descriptive, predictive and prescriptive analytics.

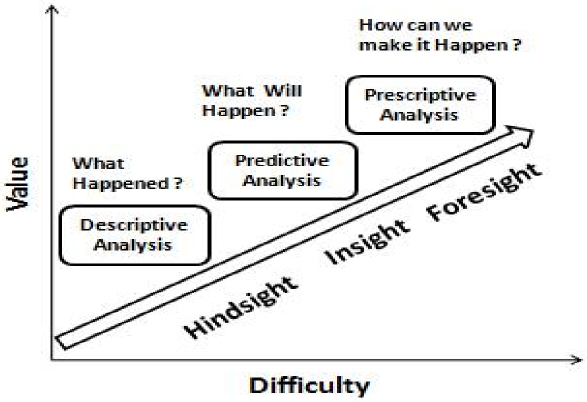


Figure 3. Predictive Analytics value-difficulty plot

1.4. Evaluation metrics for RS

The standards or the grades of the RS are classified into two groups.

1. Metrics for statistical accuracy
2. Metrics for decision support accuracy

The first one evaluates the accuracy of the RS filtering technique by comparing the forecast ratings in a straight line with the real rating. It is calculated by the formula;

$$\text{MAE} = \frac{1}{N} \sum_{u,i} \left| P_{u,i} - r_{u,i} \right| \qquad (1)$$

Where N= Total rating on the group of items.
$P_{u,\,i}$ = Predicted rating for a user on an item.
$R_{u,\,i}$ = Real rating value of the user for an item.

RMSE is a measure of RS and it is calculated by the formula:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{u,i}(P_{u,i} - r_{u,i})^2} \qquad (2)$$

The Precision and Recall is the decision support accuracy matrix which is calculated by,

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives + false negatives}} \qquad (3)$$

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives + false positives}} \qquad (4)$$

In section 2 earlier literature and existing research have been highlighted and discussed as part of the literature survey. Section 3 defines the proposed methodology to achieve the recommending the educational institutes for the students based on the rank. It also describes the tools and techniques used to carry out this piece of work. Section 3 defines the problem statement and objective of the paper. Section 4 explains the stepwise procedure and implementation part. It describes the lab recommendation and software implementation of the proposed system. Section 5 explains the outcome of the implementation, analysis of the result, comparison of the result and accuracy of the result. Section 6 highlights the inference or conclusion of the present work.

2. Related Work

Though the work is done in this area is not very huge about the recommendation for educational institutes in India, still significant amount of research has been done in the field of recommendation techniques, course recommendation, student's academic performance, an online recommendation for shopping, embedding emotional context in the recommendation and many more.

F. Ricci, L. Rokach, and B Shapira-2007 [1] described the recommender system, its role in recent times, and various aspects of the recommendation system. It also highlights the various domains of recommendation systems, evaluation of its quality and applications of a recommender system.

Ken Goldberg and et al. Aug 2000 [2] proposed an innovation- the recommender system not only uses user's liking and attentiveness but also the user's emotional intellectual aspects.

Fuzheng Zhang and et al. 2016 [3] explored how to grip the assorted information based on knowledge to enhance the rank of the systems of recommendation. First, by utilizing the knowledge base recommendation, they design 3 constituents to extricate item semantic depictions from the textual content, constitutional content and optical content. It shows the joint learning of the latent constituents in collaborative filtering and also items semantic constituents from the filtering of knowledge base methods.

Ray and Sharma [4] suggested an RS method that assembles the approaches of collaborative filtering for the generation of courses that are elective as a recommendation. They assemble the utilization of both user and item base RS filtering techniques, which is applied to real-time data for the forecast of courses (elective). Outcomes are built on MAE, calculated for each elective course.

Arazy, Shapira [5] suggested an RS for course enrolment that is done online, which uses ancient data to manifest the factors which impact student's choice of course selection of elective courses. They use the techniques of collaborative filtering and based on precision and recall performed the designed RS.

Osmanbegovic and Suljic [6] dispensed the techniques for data mining for molding predictions of students based on their performance. Several data mining techniques were compared for the design of the prediction model, which in return calculate the student victory. By performing the survey during the semester the data was collected. The rate of success of students was calculated based on grades acquired by students in their end examinations.

Willium W. Guo [7] dispensed the neural network method by taking the use of methods of statistics. In this, the setting up of dynamic models techniques are incorporated which lend a hand in forecasting satisfaction of student course. Besides the entire model applied, MLP outperforms in the generation of near practical results.

Thilina Ranbaduge [8] proposed a model rely on the techniques of DM. The main aim of this article is to make a survey in the curriculum for the generation of approaches that can make predictions based on each student's past performance in their academic curriculum.

Al-Badarenah and Alsakran [9] presented a collaborative filtering RS using the technique of clustering for the generation of elective courses by making use of rules of association to suggest courses based on liking parameters.

Cakmak [10] designed an RS technique that is used for the estimation of student course grades. They also increase the quality of result generation by implementing automated outlier removal.

Tran et al. [11] presented a model in educational scenarios on the calculation of student performance. They use the DM techniques and different regressions for the implementation of the suggested algorithm. They also implemented a combination of all the suggested techniques and based on the accuracy of the model they calculated the results.

Mueen et al. [12] hand over analysis for DM application methods for the prediction of the performances of the students. For this, they use the real-time data gathered from an undergraduate student.

Bienkowski et al. [13] dispensed amplified learning and teaching through educational DM along with learning analytics mock-up to predict the performance of the student based on different influencing factors to hold-up the online learning system.

Upendran and Chatterjee [14] designed a course RS that undertook students as the basis of their ancient performances and ability of learning. By using the data of the former student as the input they constructed the model. The main subsequent thing for the technique here is that if a student with certain expertise is capable of outright the course successfully then a new student with some expertise will also be capable of outright the course.

Castro and Vellido [15] come up with efficient and effective particulars about contemporary research and applications of DM in e-learning such as to know about the lack of success of the students, students' performances based on classifications, e-learning counsel, clustering of students, etc.

Affendy et al. [16] put forward a model that relies on DM methods. Based on constituents that can affect the student's overall grades, they use predictions for calculating their academic performance. The key purpose of this article is to provide a position to the affecting constituents to alert the students so that they are capable to keep up their grades.

Bydžovská [17] based on former achievements of similar students, presented a prediction model using collaborative filtering, regression, and classification techniques for forecasting concluding grades of students.

Ramesh et al. [18] presented statistical and DM methods. The main aim of this paper is to generate predictions of the student based on their performance in the academic curriculum. The influencing constituents that can affect student performance and their concluding grades also considered here.

Gaddam, poha, and Balagani [19] come up with a model for identifying anomalies by combining the k-mean clustering and ID3 tree classification. They also compare the performance of classification with the individual ID3 decision tree and K-mean clustering.

Mishra et al. [20] presented an analysis of methods and trends followed in PA in the domain of knowledge discovery. They made use of data of business intelligence for modeling and forecasting.

Felfernig et. al. [21] suggested the fundamental techniques in RS.

Zaiane [22] proposed the systems of web mining which described on-line learning exercises in course web pages, because of learners' history to intensify the course route and assist the process of online learning. By using association rule mining, they designed an RS agent.

3. Problem Statement

Despite many types of research around the field of academics and recommender systems, the research around the recommendation to choose the institutes in higher

Table 2. Input Data Set

S. No	CSE-IIT Bombay	Rank
1	2016	60
2	2017	62
3	2018	59

education in India is very limited. Because of this, students who are seeking admissions into reputed institutions in India are still confused about making a decision on which institute to choose. With the progressing trend of fascination towards an engineering career, students and parents are having a keen desire to choose the IIT based on rank. Still, they are not very sure to choose the college and not getting confidence while seeking admissions into different colleges. This research fills the above gap and assists the students and guardians in choosing the college based on their ranks in JEE advance.

Here data (closing ranks of JEE advance) from different authentic sources are collected for the past three years such as 2016-2018. A model was built to predict the ranks for 2019 for each branch of every IIT. This paper illustrates prediction using Time Series Forecasting and recommendation using classification techniques. A comparative study of Random Forest Classification and KNN classification has been done. Finally, a recommendation is done based on the accuracy level.

4. Methodology

4.1 Data Set Description

As part of data collection, data for the past three years (2018, 2017 and 2016) have been collected for the top ten IITs from various authentic sources. Appendix 1 reveals the collected data where each data set contains the year, college name, branch, and rank for different IITs.

4.2 Feature Selection

College Name, Branch, Rank and Year are the different attributes or features selected for this work.

4.3 Data Pre-processing

Data from different sources are collected for the years 2016, 2017, and 2018. All the closing ranks for IIT JEE advances are collected for all the IITs in India. After analyzing the data, the top ten IITs are filtered based on the closing ranks of their different branches. Finally, four branches are filtered out to be the common branches (CSE, Electrical, Mechanical, and Civil). Data optimization is done by cleaning data set with missing value rows.

4.4 Methodology for Prediction

There exist many methodologies and statistical approaches, which can be applied to predict future rank based on historical data. The following approach is explored and analyzed to fit into our problem so that a clear idea of the methodology along with the efficiency and limitations were captured.

Time Series Forecasting

A time series is taken at successively uniformly distributed spaced tips in time. Hence it is an order of discrete-time data. The four main components of time series data are (1) Seasonal disparities that redo over a particular period such as day, week, month and season (2) current disparities that move up or down in a reasonably foreseeable pattern, (3) Cyclical disparities that coincide with business or economic boom-bust cycle and (4) Irregularity.

This method is the chosen method for our problem to predict the future year rank based on the past three years of historical data. As there is only one independent variable, time series forecasting is giving the best result while predicting future data. Figure 4. below shows a simple time series plot in R for ranks in different years (Table 2).

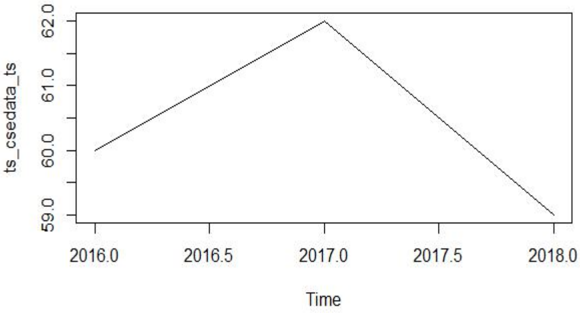


Figure 4. Time series plot of data set

ARIMA Model

ARIMA is a well-liked and ductile category of a forecasting model that employs ancient information to construct forecasting. This type of erection is a basic forecasting proficiency that can be used as a footing for many complex constructions.

ARIMA is described by the three parameters order: (p- order of the AR term, d- order of the MA term, and q- To make the time series stationary the no. of differencing needed). The auto regressive (AR (p)) parameter is mentioned to the use of ancient values in the regression equation for the series Y. Auto-regressive part p defines the number of lags utilized in the model and y_t depends only on its lags i.e. y_t - function of the lags of y_t. As an example, AR (2) or ARIMA (2, 0, 0), is represented by

$$y_t = a_1y_{t-1} + a_2y_{t-2} + \epsilon_t \qquad (5)$$

where y_{t-1} is the lag 1 of the series, a_1 is the coefficient of lag1, y_{t-2} is the lag 2 of the series, a_2 is the coefficient of lag 2 and ϵ_t is the error term that the model estimate .

The d is the degree of differencing in the integrated component (I (d)). Differencing a series includes simply the d times of subtracting its current and previous values. When the stationary thought is not encountered, differencing is used to stabilize the series.

The moving average (MA (q)) part contains the error of the mock-up as a coalition of previous error phrase (e_t). The number of terms to be included in the model is determined by the order q., Autoregressive, differencing and moving average constituents are incorporated in a non-seasonal ARIMA model.

Here the model is narrated by two sets of order parameters: (p, d, and q) and parameters relating the seasonal constituents of m periods. These models straightly based on ancient values, and therefore work best on series that are long and stable. ARIMA normally approximates ancient patterns and hence does not focus to explain the structure of the elementary mechanism of data.

If someone wants to choose the model of his/her choice, he/she can apply the arima () function in R. There is another function arima () in R which does not permit for the constant c unless d=0 and it does not return everything required for other functions in the forecast package to function properly. At last, this does not allow the estimated model to be applied to the new data that is useful for checking forecast accuracy. The exceptional cases of ARIMA model are listed in Table 3.

Table 3. Exceptional cases of ARIMA

White noise	ARIMA (0, 0, 0)
Random walk	with no constant : ARIMA (0, 1, 0)
Random walk with drift	with a constant : ARIMA (0, 1, 0)
Auto regression	ARIMA (p, 0, 0)
Moving average	ARIMA (0, 0, q)

Modelling Procedure of ARIMA

The procedure for putting an ARIMA model to a set of non-seasonal time series data is listed below.

Step1: First is to intrigue the data and recognize any unusual inspections/observations.
Step2: Alter the data using the Box-Cox transformation if required to stabilize the variance.
Step3: In case if the data are immotile, take the differences of the data until the data are stationary first.
Step4: Inspect the ACF/PACF that is an ARIMA (p, d, 0) or ARIMA (0, d, q) model relevant?
Step5: Apply the model
Step6: Examine the residuals from the model by plotting the ACF of the residuals. If they do not match with white noise, try another model.
Step7: Once the residuals match with white noise calculate the forecasts.

The Hyndman-Khandakar procedure only takes care of steps 3 to 5. Hence even if we use this procedure, we will need to look after the supplementary steps our self.

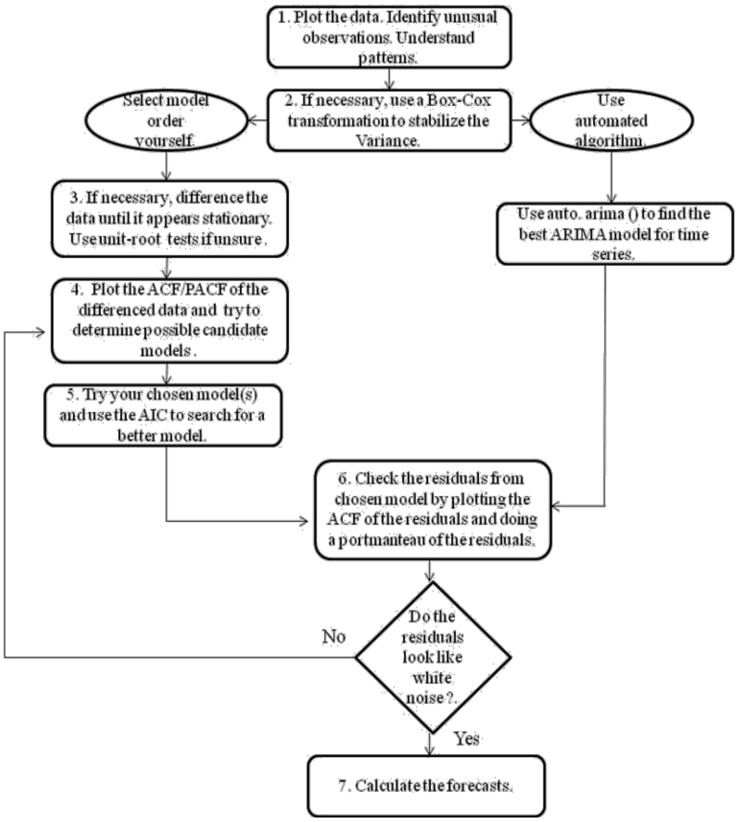

1. Plot the data. Identify unusual observations. Understand patterns.
Select model order yourself.
2. If necessary, use a Box-Cox transformation to stabilize the Variance.
Use automated algorithm.
3. If necessary, difference the data until it appears stationary. Use unit-root tests if unsure.
Use auto. arima () to find the best ARIMA model for time series.
4. Plot the ACF/PACF of the differenced data and try to determine possible candidate models.
5. Try your chosen model(s) and use the AIC to search for a better model.
6. Check the residuals from chosen model by plotting the ACF of the residuals and doing a portmanteau of the residuals.
Do the residuals look like white noise?
No
Yes
7. Calculate the forecasts.

Figure 5. Modelling Procedure for ARIMA

4.5 Rank Prediction using Time Series Forecasting

The approach for predicting rank was chosen to be the "Time Series Forecasting" using ARIMA model.

Proposed Algorithm

Input: Optimized data set for each branch of each IIT having year and rank
Output: Predicted rank

Procedure:

Step1: The optimized data set was prepared for each IIT separately as shown above (Table 2. Input Data Set) was employed as the input to the algorithm.
Step2: Reading the data set in .CSV format
Step3: Applying time series to the data with frequency 1 starting with the year 2016.
Step4: Plotting the data set as per time series
Step5: Applying ARIMA model on the time series data obtained in step 3 and storing the result in another variable.
Step6: Call forecast method with the result obtained in step5 as an argument. It will return the forecasted data for the subsequent years.

Input Data Set

Table 4. Input Data Set for CSE branch of IIT Bombay

CSE-IIT Bombay	Rank
60	2016
62	2017
59	2018

Output of Time Series Forecasting using ARIMA

Points	Forecasts	Lo 80	Hi 80	Lo 95	Hi 95
2019	62.32921	61.23427	63.42415	60.65465	64.00377
2020	59.24058	57.74700	60.73415	56.95635	61.52480

Table 5 shows the predicted rank for 2019 for CSE branch of IIT Bombay.

Table 5. Predicted rank for CSE branch of IIT Bombay (Output Data)

CSE-IIT Bombay	Rank
2016	60
2017	62
2018	59
2019	62

The above procedure was repeated individually for each branch of each IIT and the predicted ranks were calculated for 2019 using ARIMA forecasting. Finally, outputs obtained from each dataset are put in a table called "predicted rank table for 2019" which is shown below. Table 6 below is the "predicted rank table for 2019".

Table 6. Predicted rank

Name of the IITs	CSE	Electrical	Mechanical	Civil
IIT Bombay	62	281	827	2566
IIT Delhi	104	625	1077	3178
IIT Madras	202	651	1270	3984
IIT Kanpur	213	818	1672	3845
IIT Kharagpur	278	1341	1802	4164
IIT Roorkee	474	1933	2564	4985
IIT Guwahati	630	2146	2972	5934
IIT Hyderabad	913	2377	3086	6138
IIT Indore	1613	3983	5085	7432
IIT Bhubaneswar	3262	5268	5446	8355
IIT Patna	3672	5836	7636	9031

Figure 6. shows the graphical representation of Table 6 predicted rank for the year 2019 for each branch of each IIT.

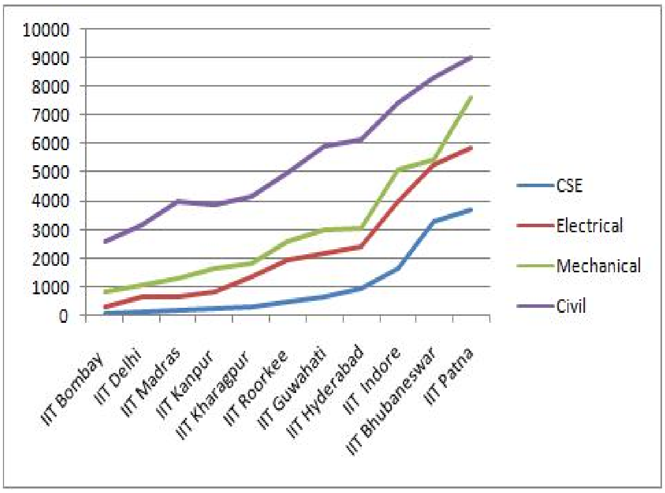


Figure 6. Predicted ranks for 2019

Looking at the plot, a student can get an idea of the college and branch he may get admitted to as per his current rank. This prediction may help the student to get an idea at a high level.

4.6 Random Forest Classification

A forest can be made from constructing numerous decision trees. A random forest works in the following path: First, random samples are created by using the Bagging (Bootstrap Aggregating) algorithm. A new data set D2 is created from the given D1 data set having n rows and p columns by random sampling of n cases with replacement from the original data. The one-third of the rows from D1 is left out which is known as Out of Bag sample. Then, the model train on D2. Out of bag sample is used to determine the unbiased estimate of the error.

The P << p columns are picked at each node in the data set out of p columns. Randomly the P columns are chosen. For the regression tree, the by default choice of P is p/3and P is sqrt (p) for classification tree. Unlike a tree, in a random forest, no pruning takes place. In decision trees, to circumvent over fitting one method is there known as pruning. Selecting a sub tree that leads to the low test error rate is termed as Pruning. To obtain the test error rate of a sub tree the cross-validation is used. Various trees are grown and the final estimation is procured by voting or averaging. On a different sample of actual data each tree is grown. The random forest has the quality to internally enumerate OOB error; cross validation doesn't build many perceptions in a random forest. It gives a more definitive average output. For this, the random forest is robust to correlated predictors.

4.7 K- Nearest Neighbour Classification

KNN can be used in predictive problems such as regression and classification. This method is mainly used in the industry for classification problems. For the evaluation of any technique, we must focus on 3 features:

1. Interpret output easily
2. Time of calculation
3. Predictive Power

KNN algorithm fairs across all parameters of considerations. It is generally used for its ease of interpretation and minimum calculation time.

4.8 Accuracy, Recall and Precision

Accuracy is the ratio of the correctly ranked Indian institute of technology where a student can get admission from the whole pool of Indian institute of technologies. It predicts how many IITs we correctly ranked out of all the IITs. Precision is a measure of exactness that determines the fraction of the relevant rank of IITs retrieved out of all IITs ranked, for example, the proportion of recommended IITs that are good. The recall is a measure of completeness, determines the fraction of rank of IITs retrieved out of all relevant IITs.

5. Implementation of the proposed algorithm and result

5.1 Proposed Model

To implement the proposed system, at first, the design has been made. The following diagram represents the architectural diagram of the high-level end-user view of the proposed system. Based on the student's rank and branch of his/her choice, it recommends the college.

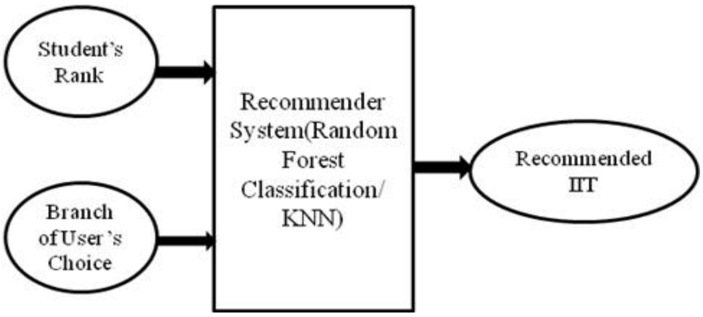


Figure 7. Architectural diagram of the proposed model

5.2 Tools/Technology

To implement the proposed system, the tool/technology used was 'R'. R studio was used. Coding has been done in R to develop the model and the model has been applied for predicting the rank and recommending the college. For developing the code in R, the required package and library were installed.

5.3 Implementation

The implementation of the entire work has been done in four parts.

A) Prediction of rank using Time Series Forecasting (ARIMA)
B) Recommendation by classification using Random Forest Classification
C) Recommendation by classification using KNN classification
D) Comparison of the Random Forest Classification and KNN for choosing the best model.

Rank Prediction

Installing the packages and adding a library

For predicting the rank, time series forecasting implemented with ARIMA model. As a pre-requisite to developing R code to run for implementing this forecast method, the "forecast" package was installed and the corresponding library was added.

Reading the input file

Once the package and library were added, the R studio was ready to implement the time series forecasting. Then the input data set was kept in an excel sheet.CSV extension. The file was then read and the time series method was applied to the input file. The dataset was then stored in another variable and then they were plotted.

The below plot shows the data set for CSE, IIT Bombay for the year 2016, 2017 and 2018.

The lines of code were executed 44 times to read all the data sets and R plots were obtained for all.

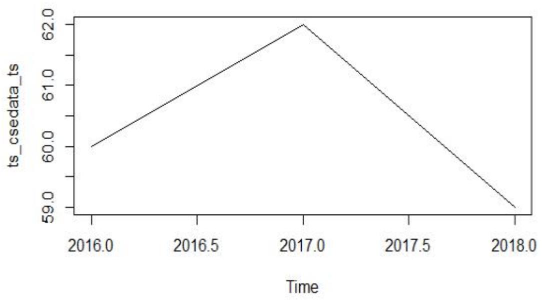


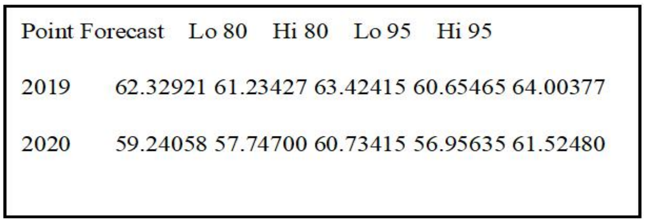
Figure 8. R plot of the input data set

Predicting the rank using time series forecasting

Once the data set was read, the next job was to apply ARIMA model to predict the future rank.

The codes were executed in R to get the predicted rank for each data set. This was also run 44 times to run 44 data sets.

The output of ARIMA forecast is obtained as shown below.



```
Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95

2019      62.32921 61.23427 63.42415 60.65465 64.00377

2020      59.24058 57.74700 60.73415 56.95635 61.52480
```

Figure 9. Output data for predicted rank

The above result when plotted in R by taking the years and ranks, gives the following graph as viewed in Figure 10.

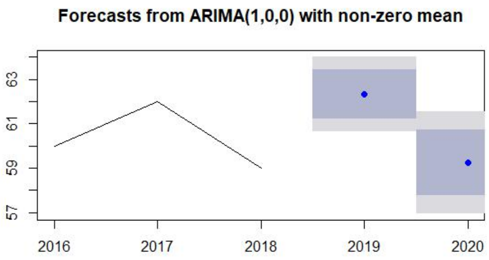


Figure 10. R plot of output from ARIMA forecast

Recommendation using Random Forest Classification

For random forest classification, the input data are the predicted ranks obtained for 2019 in the above step. At first, to implement a random forest, the library (random Forest) was added. Then input data file was read. Then factorization of data was done. Also, data were normalized. There were 173 observations with 3 variables. Partitioning of data was done. Then training set and testing set of data were segregated. The model was built and trained at first. Then the model was validated with the testing set of data. Out of all the observations, the code demonstrated 47 observations with 3 variables. The model was trained and tested multiple times and results were obtained. This classification technique gave 80% accuracy. The confusion matrix was obtained with a 20% error rate.

Table 7. Result from random forest

	Rank	College	Branch
1	1	IITB	CSE
2	3	IITB	CSE
3	10	IITB	CSE
4	20	IITB	CSE
5	30	IITB	CSE
6	40	IITB	CSE

Here 47 obs. of 3 variables are taken as follows:
Rank : int 1 3 10 20 30 40 50 260 270 280 ...
College: Factor w/ 3 levels "IITB","IITD",..: 1111111111...
Branch : Factors w/ 3 level "CSE","EEE","MECH": 1 1 1 1 1 1 1 2 2 2 ...

Here type of random forest: classification, Number of trees: 100, Number of variables tried at each split : 1
OOB estimate of error rate: 20%

Table 8. Data Set of 47 obs. of 3 variables

The confusion matrix, train data set and predicted college obtained are shown in is Table 9, 10 and 11 respectively

Table 9. Confusion Matrix

	Rank	College	Branch		Rank	College	Branch
1	1	IITB	CSE	25	230	IITM	MECH
2	3	IITB	CSE	26	240	IITM	MECH
3	10	IITB	CSE	27	250	IITM	MECH
4	20	IITB	CSE	28	310	IITM	EEE
5	30	IITB	CSE	29	320	IITM	EEE
6	40	IITB	CSE	30	330	IITM	EEE
7	50	IITB	CSE	31	340	IITM	EEE
8	260	IITB	EEE	32	350	IITM	EEE
9	270	IITB	EEE	33	110	IITD	CSE
10	280	IITB	EEE	34	120	IITD	CSE
11	290	IITB	EEE	35	130	IITD	CSE
12	300	IITB	EEE	36	140	IITD	CSE
13	360	IITB	MECH	37	150	IITD	CSE
14	370	IITB	MECH	38	160	IITD	MECH
15	380	IITB	MECH	39	170	IITD	MECH
16	390	IITB	MECH	40	180	IITD	MECH
17	400	IITB	MECH	41	190	IITD	MECH
18	60	IITM	CSE	42	200	IITD	MECH
19	70	IITM	CSE	43	410	IITD	EEE
20	80	IITM	CSE	44	420	IITD	EEE
21	90	IITM	CSE	45	430	IITD	EEE
22	100	IITM	CSE	46	440	IITD	EEE
23	210	IITM	MECH	47	450	IITD	EEE
24	220	IITM	MECH				

IITB IITD IITM Class Error

IITB	13	0	0	0.0000000
IITD	3	7	1	0.3636364
IITM	2	1	8	0.2727273

Table 10.Train Data Set

	IITB	IITD	ITM
IITB	13	3	2
IITD	0	7	1
IITM	0	1	8

Table 11. Predicted College

College Pred	IITB	IITD	IITM
IITB	4	0	3
IITD	0	3	0
IITM	0	1	1

Predicted Final Data

```
   1        2
 IITM     IITD
Levels:   IITB     IITD     IITM
```

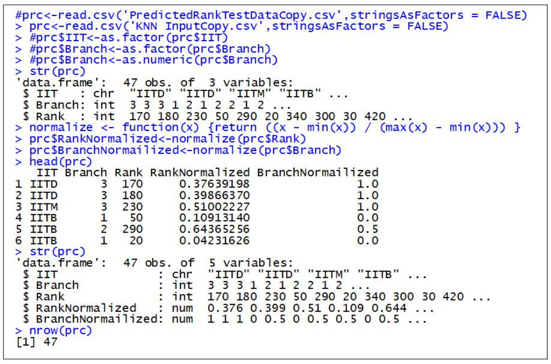
Recommendation using KNN classification

The predicted ranks obtained from section 5.3.1 were given as input to the KNN model.

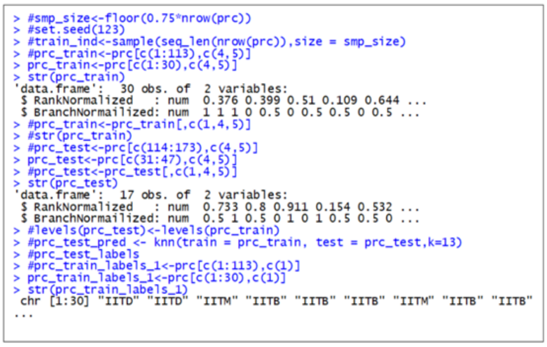
The “Branch” column was encoded with integer value for ease of implementation.

Data were arranged in an unordered way so the "Branch" and the "Rank" columns were normalized using the technique of max-min normalization (Appendix II).

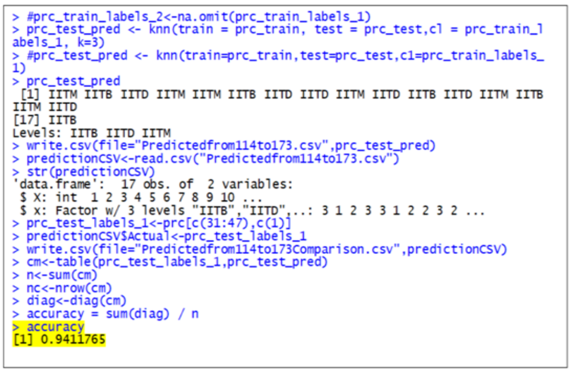
Result obtained from KNN classification



```
#prc<-read.csv('PredictedRankTestDataCopy.csv',stringsAsFactors = FALSE)
> prc<-read.csv('KNN InputCopy.csv',stringsAsFactors = FALSE)
> #prc$IIT<-as.factor(prc$IIT)
> #prc$Branch<-as.factor(prc$Branch)
> #prc$Branch<-as.numeric(prc$Branch)
> str(prc)
'data.frame':  47 obs. of  3 variables:
 $ IIT   : chr  "IITD" "IITD" "IITM" "IITB" ...
 $ Branch: int  3 3 3 1 2 1 2 2 1 2 ...
 $ Rank  : int  170 180 230 50 290 20 340 300 30 420 ...
> normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x))) }
> prc$RankNormalized<-normalize(prc$Rank)
> prc$BranchNormailized<-normalize(prc$Branch)
> head(prc)
   IIT Branch Rank RankNormalized BranchNormailized
1 IITD      3  170     0.37639198               1.0
2 IITD      3  180     0.39866370               1.0
3 IITM      3  230     0.51002227               1.0
4 IITB      1   50     0.10913140               0.0
5 IITB      2  290     0.64365256               0.5
6 IITB      1   20     0.04231626               0.0
> str(prc)
'data.frame':  47 obs. of  5 variables:
 $ IIT              : chr  "IITD" "IITD" "IITM" "IITB" ...
 $ Branch           : int  3 3 3 1 2 1 2 2 1 2 ...
 $ Rank             : int  170 180 230 50 290 20 340 300 30 420 ...
 $ RankNormalized   : num  0.376 0.399 0.51 0.109 0.644 ...
 $ BranchNormailized: num  1 1 1 0 0.5 0 0.5 0.5 0 0.5 ...
> nrow(prc)
[1] 47
```



```
> #smp_size<-floor(0.75*nrow(prc))
> #set.seed(123)
> #train_ind<-sample(seq_len(nrow(prc)),size = smp_size)
> #prc_train<-prc[c(1:113),c(4,5)]
> prc_train<-prc[c(1:30),c(4,5)]
> str(prc_train)
'data.frame':  30 obs. of  2 variables:
 $ RankNormalized   : num  0.376 0.399 0.51 0.109 0.644 ...
 $ BranchNormailized: num  1 1 1 0 0.5 0 0.5 0.5 0 0.5 ...
> #prc_train<-prc_train[,c(1,4,5)]
> #str(prc_train)
> #prc_test<-prc[c(114:173),c(4,5)]
> prc_test<-prc[c(31:47),c(4,5)]
> #prc_test<-prc_test[,c(1,4,5)]
> str(prc_test)
'data.frame':  17 obs. of  2 variables:
 $ RankNormalized   : num  0.733 0.8 0.911 0.154 0.532 ...
 $ BranchNormailized: num  0.5 1 0.5 0 1 0 1 0.5 0.5 0 ...
> #levels(prc_test)<-levels(prc_train)
> #prc_test_pred <- knn(train = prc_train, test = prc_test,k=13)
> #prc_test_labels
> #prc_train_labels_1<-prc[c(1:113),c(1)]
> prc_train_labels_1<-prc[c(1:30),c(1)]
> str(prc_train_labels_1)
 chr [1:30] "IITD" "IITD" "IITM" "IITB" "IITB" "IITB" "IITM" "IITB" "IITB"
...
```



```
> #prc_train_labels_2<-na.omit(prc_train_labels_1)
> prc_test_pred <- knn(train = prc_train, test = prc_test,cl = prc_train_l
abels_1, k=3)
> #prc_test_pred <- knn(train=prc_train,test=prc_test,cl=prc_train_labels_
1)
> prc_test_pred
 [1] IITM IITB IITD IITM IITM IITB IITD IITD IITM IITD IITB IITD IITM IITB
IITM IITD
[17] IITB
Levels: IITB IITD IITM
> write.csv(file="Predictedfrom114to173.csv",prc_test_pred)
> predictionCSV<-read.csv("Predictedfrom114to173.csv")
> str(predictionCSV)
'data.frame':  17 obs. of  2 variables:
 $ X: int  1 2 3 4 5 6 7 8 9 10 ...
 $ x: Factor w/ 3 levels "IITB","IITD",..: 3 1 2 3 3 1 2 2 3 2 ...
> prc_test_labels_1<-prc[c(31:47),c(1)]
> predictionCSV$Actual<-prc_test_labels_1
> write.csv(file="Predictedfrom114to173Comparison.csv",predictionCSV)
> cm<-table(prc_test_labels_1,prc_test_pred)
> n<-sum(cm)
> nc<-nrow(cm)
> diag<-diag(cm)
> accuracy = sum(diag) / n
> accuracy
[1] 0.9411765
```

Figure 11. Result of KNN

5.4 Discussion

Once ranks were predicted, a recommendation was done by adopting both the classification techniques such as "Random Forest Classification" and "KNN" classification. Random Forest uses most of the voting while KNN searches the nearest neighbor class to arrange the given data. Out of the above-described techniques, the accuracy level of Random Forest Classification is 80% (From the result of the random forest: OOB estimate of error rate: 20%) whereas the accuracy level of KNN is found to be 94.11% (Figure 11). Hence the proposed model which is foremost suited for recommending the college is KNN with 94.11% accuracy.

6 Performance Analysis

This work presents with two classification techniques, the performance measure in terms of Accuracy of classification, in terms of Recall and Precision. See (Table 12) .

Table 12. Comparative analysis between KNN and Random Forest Classification

Classification Technique	Accuracy (%)	Recall	Precision
Random Forest Classification	80.00	72.10	57.06
K-Nearest Neighbor	94.11	53.97	23.24

7 Conclusion and Future work

As the accuracy in KNN is 94.11% which is better than the Random Forest Classification having accuracy 80%, so the proposed model is KNN. In this paper, currently, only the top ten IITs have been taken into consideration on a pilot basis for recommending the students to get admissions to the above-specified branches only for general candidates excluding all the reservations. As it was completely a new piece of work about the study of recommendation for educational institutes, it has its limitations such as it is limited to only the general category and four common branches of IITs. This paper can be further enhanced and exercised for other categories and other institutes.

This piece of work can be enhanced further for implementing the functionality for the other category and also, for other graduate curriculums. Due to the limitations of data availability and time constraint, this work was limited to rank only. This study can be further enhanced and exercised for including other factors such as course fees, quality of the study, placement, ambiance, and distance from home city.

Appendix A

Table 1. Synthetic Data

Year	Name of the IITs	CSE	Electrical	Mechanical	Civil
2018	IIT Bhubaneswar	2092	4492	5102	8116
2018	IIT Bombay	59	265	848	2544
2018	IIT Delhi	100	462	1025	3107
2018	IIT Guwahati	554	1923	2942	5890
2018	IIT Hyderabad	777	2090	3192	5988
2018	IIT Indore	1264	3417	4905	7207
2018	IIT Kanpur	213	803	1611	3707
2018	IIT Kharagpur	272	1101	1745	4078
2018	IIT Madras	200	568	1333	3748
2018	IIT Patna	2731	5841	7053	8674
2018	IIT Roorkee	416	1735	2485	4756
2017	IIT Bhubaneswar	2695	5029	5522	7857
2017	IIT Bombay	62	266	738	2308
2017	IIT Delhi	104	416	944	2869
2017	IIT Guwahati	670	1987	2923	5284
2017	IIT Hyderabad	975	2357	3280	5773
2017	IIT Indore	1500	3770	4771	7258
2017	IIT Kanpur	206	816	1409	3533
2017	IIT Kharagpur	262	1174	1578	4071
2017	IIT Madras	177	641	1222	3632
2017	IIT Patna	3298	6151	7237	8281
2017	IIT Roorkee	449	1905	2343	4582
2016	IIT Bhubaneswar	3082	4763	4971	6294
2016	IIT Bombay	60	583	901	2251
2016	IIT Delhi	102	329	778	2181
2016	IIT Guwahati	666	2127	3052	5371
2016	IIT Hyderabad	988	2609	3588	4940
2016	IIT Indore	2074	3724	4250	6561
2016	IIT Kanpur	207	674	1192	2674
2016	IIT Kharagpur	300	1114	1377	3154
2016	IIT Madras	186	605	1257	2905
2016	IIT Patna	3483	5733	5955	6661
2016	IIT Roorkee	559	1844	2084	3419

Appendix B

Table 7. Normalized Data after Min- Max operation Branch wise

Name of the IITs	Branch	Rank
IITB	3	370
IIT Kanpur	4	3402
IITM	4	3590
IITB	1	40
IITD	4	2718
IIT Kanpur	1	200
IITM	2	330
IIT Bhubaneswar	3	4810
IITD	4	2959
IIT Kanpur	4	3382
IITB	2	270
IIT Bhubaneswar	2	4201
IITM	1	100
IIT Kanpur	3	1400
IIT Kanpur	4	3503
IIT Kanpur	4	3304
IITB	2	280
IITM	3	210
IITM	4	3520
IITM	3	220
IIT Kanpur	1	213
IITD	1	150
IITB	1	50
IITM	1	90
IITD	4	3107
IIT Bhubaneswar	3	4911
IIT Bhubaneswar	4	8116
IITB	3	380
IITB	1	10
IIT Bhubaneswar	1	2180
IITB	4	2410
IITD	3	190
IIT Kanpur	2	762
IITM	1	70
IITD	1	130
IITB	3	360
IITM	3	250
IITD	4	2869
IIT Kanpur	1	195
IIT Bhubaneswar	2	4492
IITM	1	60
IIT Kanpur	2	780
IIT Bhubaneswar	2	4525
IITB	2	300
IITD	3	160
IIT Bhubaneswar	4	7028
IIT Bhubaneswar	4	8018
IITM	3	230
IITD	1	110
IIT Bhubaneswar	4	7857
IITB	4	2350
IIT Kanpur	1	154
IIT Kanpur	4	3410
IITB	1	1
IITB	2	290
IITM	2	320
IITB	1	20
IITB	1	50
IITM	1	80
IIT Bhubaneswar	2	4306
IITM	4	3670
IITD	2	440

IIT Bhubaneswar	3	5088
IIT Kanpur	2	816
IIT Bhubaneswar	3	5102
IIT Kanpur	2	701
IITB	3	400
IITB	4	2210
IITD	2	420
IIT Bhubaneswar	1	1802
IITB	4	2490
IITD	2	410
IITB	2	260
IITD	3	170
IIT Bhubaneswar	4	7650
IIT Kanpur	3	1317
IITD	1	140
IITM	4	3610
IITD	2	430
IITM	4	3690
IIT Bhubaneswar	2	4729
IITD	4	2824
IITB	1	3
IITB	1	30
IITB	3	390
IIT Bhubaneswar	1	2240
IITM	2	350
IITD	3	180
IITM	2	310
IITM	2	340
IITD	1	120
IIT Kanpur	3	1611
IITD	3	200
IITD	2	450
IITB	4	2290
IIT Bhubaneswar	1	2380
IIT Bhubaneswar	1	2092
IIT Kanpur	2	802
IIT Bhubaneswar	3	4958
IIT Kanpur	1	174
IITM	3	240
IIT Kanpur	3	1404
IIT Kanpur	3	1389

Appendix C

Abbreviations

RS	Recommender System
DM	Data Mining
RSD	Recommender System Design
EI	Educational Institutions
IOP	Information Overburden Problem
PR	Personalized Recommendation
NPR	Non Personalized Recommendation
KB	Knowledge Base
HRS	Hybrid Recommendation system
UB	User Base
IB	Item Base
CRS	Collaborative Recommender System
PPC	People to People Correlation
CBRS	Content-Base Recommender System
DBRS	Demographic Base Recommender System
UBRS	Utility Base Recommender System
KBRS	Knowledge-Based Recommender System
PA	Predictive Analysis
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
IIT	Indian Institute of Technology
JEE	Joint Entrance Examination
KNN	K- Nearest Neighbor
CSE	Computer Science & Engineering
ARIMA	Auto Regression Integrated Moving Average Model
ACF	Auto Correlation Function
PACF	Partial Auto Correlation Function
SVM	Support Vector Machine

References

[1] **Book Chapter**: F. Ricci, L. Rokach, and B Shapira , “Recommender Systems Introduction and Challenges”, book chapter, Springer, 2007, pp-1-34.

[2] **Journal article**: Ken Goldberg and Theresa Roeder and Dhruv Gupta and Chris Perkins, Eigen state, “A constant time collaborative filtering algorithm”, ResearchGate, Aug 2000, pp-133-151.

[3] **Journal article**: Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian Xing Xie, Wei-Ying Ma, “Collaborative Knowledge Base Embedding for Recommender Systems”, ACM digital library, 2016, pp-353-362.

[4] **Conference**: S. Ray and A. Sharma, “A collaborative filtering based approach for recommending elective courses,” in International Conference on Information Intelligence Systems, Technology and Management, Springer, 2011, pp. 330-339.

[5] **Journal article**: Arazy, O., Kumar, N., Shapira, B.: Improving social recommender systems. IT Professional, 11(4),2009, pp-38–44.

[6] **Journal article**: Edin Osmanbegović and Mirza Suljić, “Data mining approach for predicting student performance”, – Journal of Economics and Business, Vol. X, Issue 1, 2012, pp-3-12.

[7] **Journal article**: William W. Guo, “Incorporating statistical and neural network approaches for student course satisfaction analysis and prediction”, journal of Expert Systems with Applications, Volume 37 Issue 4, April 2010, pp- 3358-3365.

[8] **Journal article**: Thilina Ranbaduge, “Use of Data Mining Methodologies in Evaluating Educational Data”, International Journal of Scientific and Research Publications, Volume 3, Issue 11, November 2013, pp-1-12.

[9] **Journal article**: A. Al-Badarenah and A. Jamal, “An automated recommender system for course selection,” International Journal of Advanced Computer Science and Applications, vol. 7, no. 3, 2016, pp.1166-175.

[10] **Journal article**: A. Cakmak, “Predicting student success in courses via collaborative filtering,” International Journal of Intelligent Systems and Applications in Engineering, vol. 5, no. 1, 2017, pp.10-17.

[11] **Journal article**: T. Tran, H. Dang, V. Dinh, T. Truong, T. Vuong and X. Phan, “Performance prediction for students: a multi-strategy approach,” Cybernetics and Information Technologies, vol. 17, no. 2, 2017, pp. 164-182.

[12] **Journal article**: A. Mueen, B. Zafar and U. Manzoor, “Modeling and predicting students' academic performance using data mining techniques,” International Journal of Modern Education and Computer Science, vol. 8, no. 11, 2016, pp. 36-42.

[13] **Journal article**: M. Bienkowski , M. Feng and B. Means, “Enhancing teaching and learning through educational data mining and learning analytics: an issue brief”, U.S. Department of Education, Office of Education Technology, vol. 1, 2012, pp. 1-57.

[14] **Journal article**: D. Upendran, S. Chatterjee, S. Sindhumol and K. Bijlani, “Application of predictive analytics in intelligent course recommendation,” Procedia Computer Science, vol. 93, 2016, pp. 917-923.

[15] **Journal article**: F. Castro, A. Vellido, À. Nebot and F. Mugica, “Applying data mining techniques to e-learning problems,” Evolution of teaching and learning paradigms in intelligent environment, Springer, 2007, pp. 183-221.

[16] **Journal article**: L.S. Affendey, I.H.M. Paris, N. Mustapha, M.N. Sulaiman and Z. Muda, “Ranking of influencing factors in predicting students’ academic performance,” Information Technology Journal, vol. 9, no. 4, 2010, pp. 832-837.

[17] **Conference**: H. Bydžovská, “A comparative analysis of techniques for predicting student performance,” in Proceedings of the 9th International Conference on Educational Data Mining, 2016, pp. 306-311.

[18] **Journal article**: V. Ramesh, P. Parkavi and K. Ramar, “Predicting student performance: a statistical and data mining approach,” International Journal of Computer Applications, vol. 63, no. 8, 2013, pp. 35-39.

[19] **Journal article**: S. R. Gaddam, V. V. Phoha and K. S. Balagani, “K-Means + ID3: A novel method for supervised anomaly detection by cascading K-Means clustering and ID3 decision tree learning methods,” IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 3, 2007, pp. 345-354.

[20] **Journal article**: A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, S. Reiterer and M. Stettinger, “Basic approaches in recommendation systems,” in Recommendation Systems in Software Engineering, Springer, 2014, pp. 15-37.

[21] **Journal article**: N. Mishra and S. Silakari, “Predictive analytics: a survey, trends, applications, opportunities & challenges,” International Journal of Computer Science and Information Technologies, vol. 3, no. 3, 2012, pp. 4434-4438.

[22] **Conference**: Osmar R. Za¨ıane, “Building a Recommender Agent for e-Learning Systems”, Proceedings of the International Conference on Computers in Education (ICCE’02) 0-7695-1509-6/02 $17.00 © 2002 IEEE.