UniPredict: A GRE-score based University Recommender System using Hybrid Model

Nishita Pali
Dept of IT,
Pune Institute of Computer Technology,
Affiliated to SPPU

Nikita Khivasara
Dept of IT,
Pune Institute of Computer Technology,
Affiliated to SPPU

Ashutosh Harkare
Dept of IT,
Pune Institute of Computer Technology,
Affiliated to SPPU

ABSTRACT
In recent times, it is seen that many graduate students are willing to learn in foreign universities. Various factors like better opportunities of research, post-graduation, PhD and wider exposure to grab work in a plethora of jobs drive fresh graduates and experienced people to apply for different universities. This situation is predominant in students from Indian sub-continent and Asian countries. These students aim to get admissions in many top universities in the USA. According to the data obtained, the scores of exams like GRE, IELTS, TOEFL and, GPA of UG along with the work experience play a pivotal role in the university admissions. The aim of the web based recommendation system is to suggest the users - top 3 recommended colleges based on their profiles and inputs. As students spend huge amounts of money on counseling for obtaining university recommendations, our UniPredict system acts as a complete cost affordable platform for accurate results and user preferences. Collaborative filtering and content-based filtering is used to form a hybrid model that will be in turn used with ensemble learning to predict the universities. This system can be financially very affordable and helpful for the test takers to send 4 universities free applications after taking their test according to the GRE policy as of 2021.

General Terms
Machine Learning, Recommender System

Keywords
Model based collaborative filtering, Content based filtering, Pearson's coefficient, Ensemble Learning, Neural Network Matrix factorization SVM, K-NN, Recommender systems, item-item, user-item, cosine similarity.

1. INTRODUCTION

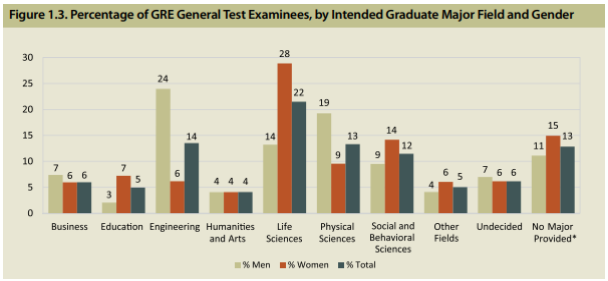

Fig 1 : Percentages of GRE General Test Examinees, by Intended Graduate Major Field and Gender [1]

According to a research analysis done by the official ETS, there are approximately 3.2L and 1L GRE test-takers in a one-year time frame from the USA and India respectively.[1] Statistics show that 44% of the total test-takers are under the age of 23 i.e. they plan for further studies after their graduation, whereas the remaining 56% go abroad after getting some work experience[1].

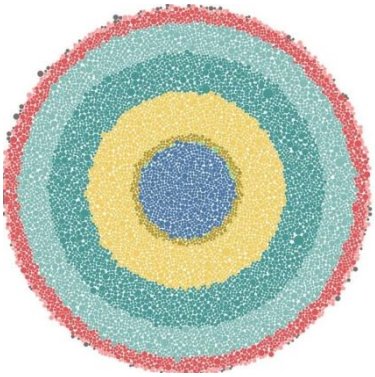

Fig 2 : Probability of acceptance based on distribution of GRE score, TOEFL/IELTS score, CGPA, Admitted/Rejected Status

In the above figure, the user's GRE score, TOEFL/IELTS score, work experience, cost of living affordability, preferred major, etc are considered. More the distance from the centre, the greater is the probability of acceptance in a particular University.

Owing to this data, it is evident that the large numbers of students take GRE for their future studies. With the surge in the number of students pursuing higher studies nowadays, and many universities available, the students often find themselves confused with the plethora of options to choose from.

Factors like the GRE score, TOEFL score, education fees, cost of living, university ranking, etc need to be considered. There are many firms available these days that offer a counseling plan to the students and help in choosing the ideal universities. However, these counseling firms cost a lot of money that would have otherwise been saved by the students and rather invest that amount in their university applications. Consequently, our model comes into picture and helps the user to decide the top-n universities based on the user's inputs regarding his scores, work experience and financial affordability.

To predict the list of universities best suitable for the user, Hybrid Filtering which is the combination of collaborative filtering and content-based filtering along with ensemble learning will be used. The user sees top-n universities on the screen as the output of the recommender.

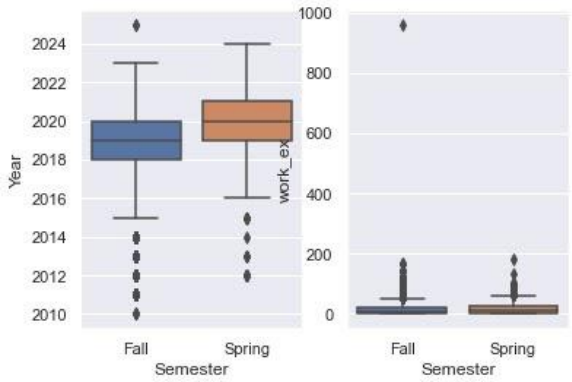

Fig 3 : Box Plots to compare the applications between spring and fall semester

The first plot represents semester wise average, minimum, maximum years by which it can be deduced that in recent years, more applicants are applying for spring semester as compared to the fall semester.

Whereas, the second plot represents semester wise distribution of work experience through which it can be understood that people who have applied for Spring semester have more work experience (in months), than those in Fall semester.

2. LITERATURE SURVEY

[1] explains different algorithms and approaches used for recommender systems. It has classified approaches into three types – Content-based filtering, Collaborative and hybrid model. It also provides a summary of different algorithms to be used.

[2] has a course wise recommender system for students in various colleges. It uses a unique user-based collaborative filtering algorithm implemented. Results of this algorithm are compared with different existing approaches.

[3] provides a Bayesian model which is good in accuracy and the results of this model can also be explained effectively. Compares results with different existing systems like matrix factorization.

[4] has implemented a model of educational services recommendations. It explains about the approach of, Multi-Layer-Perceptron (MLP) which takes the N-dimensional and non-linear features to implement the algorithm.

[5] explains TF_IDF Vectorization technique for parsing user LinkedIn profiles and through their resumes. It also provides a detailed explanation about the university recommendation system. It takes the data from user profiles and recommends it to 10 colleges. It uses cosine similarity for recommendation engine.

[6] explains in detail about cosine similarity and Pearson's coefficient for collaborative filtering approach in recommendation engine. It uses these algorithms to predict graduate schools for students. Implementation is also done on USA graduate schools' dataset with accuracy results provided.

[7] Has a detailed explanation about Multi criteria Collaborative Filtering. It also proves it by taking a survey on one dataset. For hard, intricate and huge datasets, Multi criteria Collaborative Filtering (MC-CF) gives better accuracy as well as performance and top-quality recommendations for users considering all varied features of items and users. CF algorithms often need continuous updating because of a constant increase in load of information, ways of retrieving that information, scalability and sparseness in the rating matrix.[1] Dimensionality Reduction techniques like: Matrix Factorization and Tensor Factorization techniques have been evaluated.

[8] consists of comparison with existing models like cosine similarity, Pearson's coefficient, Jaccard, mean squared difference for product recommendation using collaborative filtering. A novel approach of triangle similarity is discussed and elaborated with the help of six commonly used datasets.

[9] has explained opinions about the existing algorithms kike collaborative filtering and mass diffusion. The authors provided a novel Cov-covariance recommendation method based on correlation coefficients. This approach also provides precision and accuracy of results. This method first expresses the positive and negative correlation among random samples without knowing the distribution of items in the dataset. Then it sorts and segregates popular items in the recommendation list. Usability of these approaches was also shown by implementing those in movie recommendations.

[10] This paper shows an innovative approach and lists popular algorithms to be used in recommendation systems. It also uses ANN with complex neural networks for improving the precision of results on various attributes of datasets. It uses a student course and stream selection-based recommendation engine to reduce the dropout rate. It showed the use of random forest, k-means, multi level perceptron, support vector machines to evaluate the dataset and including every attribute in the prediction. Encoding technique was also used to combine two to three attributes and form a score for all three combined.

3. FRAMEWORK

3.1 Content based filtering

Content filtering recommendation technology plays an important role in the method of information filtering. The working of content filtering can be described as mentioned here: Technologies like machine learning and probability statistics are ordinarily used for filtering, where a user interest vector is initially utilized to represent the user's information request. Followed by segmentation, indexing, weighing of word frequency statistics are carried out in the text collection to generate a text vector. At last, the similarity between the text vector and user vector is calculated and the resource the entries with high similarities are sent to registered users of the user model. Content filtering based recommendations do not need to use the user's rating information in the learning system. Instead, they need to get the user and resource project description files. The usage of user and product description files can better solve the cold start problem. The problem of the paucity of the system score data can also be addressed because user's score data are not needed. So basically, in this approach of recommendation systems, the selected user preference or features play an important role. The recommendation is based on the similarity of features selected by related or matched users of the system. For example, if two users of the same interest are using the book finder and one user selects the science fiction books as its genre, then the other user might also get science fiction as the recommendation.

3.2 Collaborative filtering

Collaborative filtering (CF) method sheds light upon identifying other users with similar interests and then utilizes their ratings to recommend items. Recently, E-commerce giant Amazon.com has used item-to-item collaborative filtering successfully to generate real-time recommendations to shoppers. This recent non-traditional collaborative filtering (CF) technology is able to scale to Amazon's millions of

consumers and products and also reduce the computational costs. Today, one of the most widely used personalized recommendation technology is collaborative filtering (CF) in fields like both E-commerce and the E-learning world. For learning, CF can be used to decrease the information overload by suggesting to learner resources that are most relevant to them. Automated collaborative filtering (ACF) systems have been developed to predict a person's predilection towards items or information by connecting that person's recorded interests with the recorded interests of a large set of people and sharing ratings between like-minded individuals. So basically, in this approach similarity between the users or the items is taken into consideration. For example, if two users of the book review system have rated the two books with similar ratings, then the non-similar book item recommendation of one user might be based on the ratings or interest shown by another user.

4. SYSTEM ARCHITECTURE

Design Principles, Methodology and Workflow

The web architecture is used to provide a User Centric dashboard which uses the implemented algorithms and provides the user with an efficient and simple interface. This is divided into -

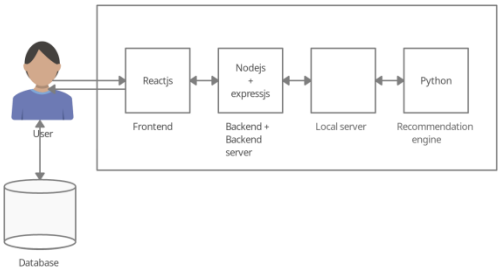

Fig 4 : System Architecture overview

1. Database : Contains all the data of the user like GRE score, GPA, work experience and it also contains University dataset comprising their ranks, financial expenses, tuition fees and acceptance rates.

2. Front-end : React Js is a front-end framework for making single-page applications. We are working with React Js for providing all GUI functions on the user side. It is then made to be directly in contact with the Node Js server. The frontend responds to the user actions and requests or sends data from/to node Js server.

3. Node JS Server : Node Js is an open-source, cross platform JavaScript run-time environment which executes JavaScript code outside the browser. We are using Express with Node Js for creating a request-response server.

4. Recommendation Engine: The engine is created by pre-processing the obtained data and running the python model on it in order to give the user recommendation as its output which will then be displayed back at the frontend.

4.1 Design Principles

This platform aims to develop Web-based College recommendation architecture with the interactive and graphical content display, it should follow some principles.

● Serve for the Recommendation dashboard. This platform mainly provides the detailed information through graphs and charts about colleges and best-suited options to choose from, so it should follow the principle of thorough and clear information display and accurate recommendation is the key idea of developing this platform.

● Simplicity and brevity. The structure of the platform, color and font should be simple and brief, the navigation of the web should be clear and definite, an instructive map should be detailed, and the style of whole page should be unified.

● Easily operated. The user may have a low skill of educational technology, so the platform should be simple and easily operated, and they don't need the training to use the platform. The platform should run fast, provide the detailed and appropriate prompt message.

● Safe, stable and easily maintained. The platform should run stable, and assure the safety of the content, and the platform can be easily updated and extended because of the technology.

4.2 Methodology

The implementation aims to build a web application which will recommend top 10 universities based on user profile and preferences. In this process, the aim is to use a hybrid recommendation algorithm which will consist of collaborative filtering followed by content-based filtering.

Database is obtained from yocket.in where the students have entered their past information. Different algorithms can be used to parse the data and extract relevant information from the profiles. Database will fetch and store the data from these profiles. The database can also provide data to the data cleaning phase. Users can also provide the information through a form which will be provided after the successful login of the student to our web portal.

The data from the user is then transmitted to the data cleaning phase. The output of this phase will prepare the data for model training. According to data parsed in phase 1, model based collaborative filtering algorithms will work to reduce the data weight. Considering different parameters of the user data, the memory based collaborative filtering algorithm will match this data to the data of professors and university data. It will narrow down the recommendation options. This is then provided as an input to the content-based filtering algorithm.

According to the data fields in all the datasets, the content-based filtering algorithm will match the best option according to the items in the datasets. The output of this phase will give the user top 3 recommendations of their preference and qualifications.

We have shown the detailed flow through the diagram.

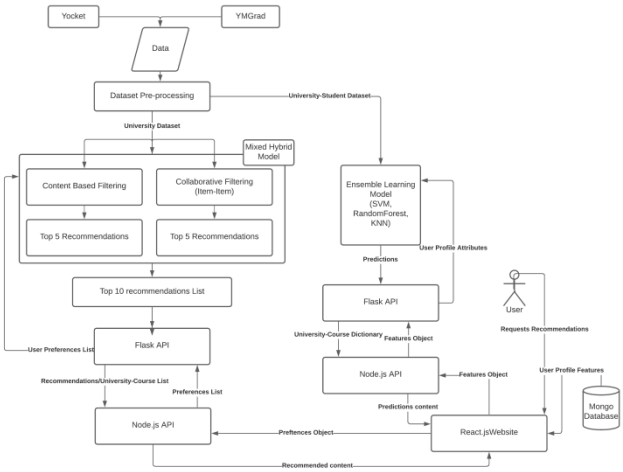

Fig 5 : Flowchart of the hybrid recommendation system

4.3 Algorithms used

4.3.1 University Course Predictor

Let U be the set of universities available and U(i) be the i-th instance of U. C represents the courses offered by them and C(i) be the i-th instance of C. Let the list containing the U(i)-C(i) pair be represented using UC. Each U(i) be represented using a unique id. For every user profile feature values including GRE score, TOEFL score, Work experience, CGPA, Semester let it be referred to as a vector S. Let M1 be the model built using the SVM algorithm. Let M2 be the model built using the KNN algorithm. Let M3 be the model built using the RandomForestClassifier algorithm. Let p1(i) represent the predicted value for every U(i)-C(i) pair by M1. Let p2(i) represent the predicted value for every U(i)-C(i) pair by M2. Let p3(i) represent the predicted value for every U(i)-C(i) pair by M3.

Let p(i) represent the final predicted value for every U(i)-C(i) pair.

Let the university-course list be represented using P.

Algorithm : To predict P

Input : *S*
Output : *P*

FOR each i in UC:
    p1(i)=M1.predict(U(i),C(i),S)
    p2(i)=M2.predict(U(i),C(i),S)
    p3(i)=M3.predict(U(i),C(i),S)
    p(i)=mode(p1(i),p2(i),p3(i))
    IF p(i)=1 THEN
        P←U(i),C(i)
    ENDIF
ENDFOR
RETURN P

4.3.2 Collaborative Recommender Item-Item filtering

Let U be the set of universities available and U(i) be the i-th instance of U. For every user profile preferences list would be taken and let it be referred to as a vector UP. Let M be the user-item matrix prepared for every user and university based on UP and M(i)(j) represents the rating, here binary for j-th university by i-th user.

Now, let S(i) be the similarity score for every university recommended based on UP. Let N be top 10 nearest neighbor to university U(i)

Let R represent the recommended university list.

Algorithm : To recommend R

Input: UP
Output: R

Normalize(M)
FOR i=1 to U.length():
    FOR each i in U:
        Sparse(M)
        S(i)←cosine_similarity(U(i),U)
    ENDFOR
ENDFOR
FOR each i in U:
    sort(U(i)) on S(i)
    FOR i=1 to 10:
        N←U(i)
    ENDFOR
ENDFOR
FOR each in UP:
    FOR each in N:
        S(i)←cosine_similarity(N(i),UP(i))
        R←U(i)
    ENDFOR
ENDFOR
R←top10[sort(R) on S(i)]
RETURN R

4.3.3 Content Recommender

Let U be the set of universities available and U(i) be the i-th instance of U. Each U(i) be represented using a unique id. Each U(i) has features associated with it including Rank, Type, Tuition Fees, Living Expenses, Average GPA, Minimum TOEFL score, Minimum IELTS score, GRE score, State. For every user profile preferences list would be taken and let it be referred to as a vector UP.

Now, let S(i) be the similarity score for every university recommended based on UP. Let R1 represent the recommended university list :

Algorithm: To recommend R1

Input: UP
Output: R1

FOR each in UP:
    FOR each i in U:
        IF U not equal to UP THEN
            S(i)←cosine similarity(UP(i),U(i))
        ENDIF
    ENDFOR
    R1.append(top3[sort(U(i)) on S(i)])
ENDFOR
RETURN R1

4.3.4 Hybrid Recommender

Let R2 be the final list of recommended universities.

Hybrid recommender built using mixed technique, combines top 5 results from both recommenders.

Algorithm: To recommend R2

Input: R,R1
Output: R2

1. R←top5[R]
2. R1←top5[R1]
3. R2.append(R)
4. R2.append(R1)
5. RETURN R2

4.4 Workflow

4.4.1 Dataset Gathering

The datasets are taken from :

1. Yocket : The data was scraped in order to retrieve the Student Dataset.

2. YMGrad : The data was scraped in order to obtain the University Dataset.

Student dataset and the University dataset will be in the CSV format and will be implemented in python using Pandas data frame. The user data and the details of the colleges were done by scraping the data from Yocket.in and YMGrad.com

4.4.2 Dataset Preparation

The Student and University datasets were preprocessed with the removal of NULL values, conversion of features into required format single scale values and data type needed was done.

4.4.3 University-Course Prediction

The Student dataset file was processed for conversion of categorical features into continuous features using binary encoding technique. Ensemble learning model was created using SVM, KNearestNeighbour, RandomForestClassifier algorithm built models using VotingClassifier technique. The ensemble model was then trained on Student dataset. To predict university-course to user profile feature values will be used. For every university-course pair the model will predict 'Status' either accept or reject and a list would be appended. This list would then be made available to the user.

4.4.4 University-Course Prediction

The University dataset will be used for building content based filtering recommender. University dataset contains features that describe the university. For every university cosine similarity technique will be used to calculate the most similar university to the user preferred universities. A list would be generated for every university that shows 3 most similar universities. The final list would predict 12 universities based on user preferences.

$$\cos(\theta)=A.B/(||A||*||B||)$$

4.4.5 University-Course Prediction

For a collaborative filtering recommender, item-item filtering technique would be used. A user-item matrix would be generated from the user preferences list. Binary rating used as the user is only specifying preferences. This binary encoded user-item matrix would be used for calculating item-item similarity using cosine techniques based on 12 nearest neighbors.

4.4.6 University-Course Prediction

Using recommendations from both the content based filtering and collaborative filtering recommenders; a hybrid recommender was built using mixed techniques where top five recommendations from both will be selected to provide a final recommendation list.

5. RESULTS

In this section, we will illustrate the performance of the proposed system for binary encoding technique used and the built ensemble learning model.
The ensemble learning model includes a model built using KNN algorithm and the accuracy variation is shown using line plot.

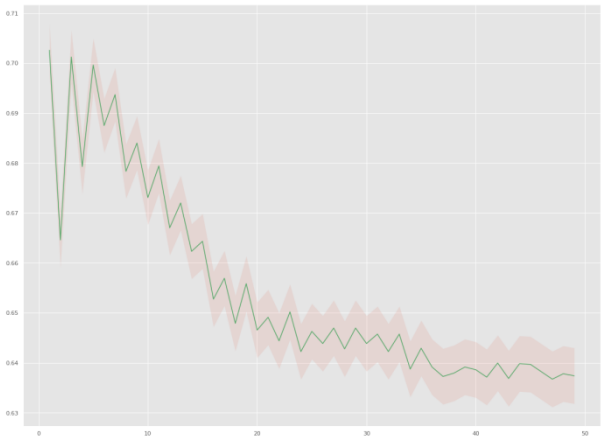

Fig 6 : Accuracy of the model with varying k-values

This leads to a conclusion that the built model will have highest accuracy for k=1 and accuracy equals 70.54%.

Another model was built using the SVM algorithm. The following table shows the accuracy of the model with respect to different kernels.

Table 1: Accuracy of model using different kernels

Sr. No.	SVM	Accuracy (%)
I	RBF	68.59%
II	POLYNOMIAL	67.82%
III	LINEAR	60.04%
IV	SIGMOID	51.92%

Maximum accuracy was obtained while using the rbf kernel and accuracy equals 68.59%.

Ensemble learning techniques were used to further increase the accuracy of the prediction. The following table shows the variation of accuracy with respect to these different techniques.

Table 2: Accuracy obtained by using different ensemble technique

Sr. No.	Ensemble Technique	Accuracy(%)
I	Simple - Max Voting	79.72%
II	Bagging • Bagged Decision Trees • RandomForest	65.09% 73.11%
III	Boosting • AdaBoost • GradientBoosting • XGBoost • CatBoost	63.45% 63.45% 64.18% 76.80%

This states that with max voting technique where prediction models include KNN built model, SVM built model, RandomForest built model gives a maximum accuracy equals 79.72%.

6. CONCLUSION

UniPredict, a GRE-score based University recommender system uses different algorithms based on collaborative filtering and content-based filtering to thus create a hybrid model. Various data sources for our recommendation engine are also discussed. They include GRE Scores, TOEFL/IELTS Scores, Work experience, GPA for the Student dataset whereas University Acceptance rates, ranks, financial in our website. Dataset of universities are also taken from varied sources in order to match the user with his/her respective requirements. We tried to summarize the process of recommendation engine for top n recommendations with different algorithms and then select the one with the maximum accuracy and minimal error.

7. ACKNOWLEDGEMENTS

The authors wish to thank the contributors of the previous research work as well as thank our family members for their continuous support throughout. Sincere gratitude towards the faculty of Pune Institute of Computer Technology and our project mentor Mr. Ravi Murumkar for their inputs and support.

8. REFERENCES

[1] Ashutosh C. Harkare; Nishita Pali; Nikita Khivasara; Ishita Jain; Ravi Murumkar. "Personalized College Recommender - A System for graduate students based on different input parameters using hybrid model" - International Journal of Science and Research Technology, 2021.

[2] MahdiJalili, SajadAhmadian, MalihehIzad (2018) "Evaluating Collaborative Filtering Recommender Algorithms": A Survey. In: IEEE Access 2018 date of publication November 28, 2018. Volume 6, 2018.

[3] Ling Huang, Chang-Dong Wand, Hong-Yang Chao, Jian-Huang Lai. "A Score Prediction Approach for Optional Course Recommendation via Cross-User-Domain Collaborative Filtering". In. IEEE Access 2019 date of publication February 7, 2019. Volume 7, 2019.

[4] Priscila Valdiviezo-Diaz, Fernando Ortega, Eduardo Cobos, and Raúl Lara-cabrera "A Collaborative Filtering Approach Based on Naïve Bayes Classifier". In. IEEE Access date of publication August 5, 2019. Volume 7, 2019.

[5] Farhan Ullah, Bofeng Zhang, Rehan Ullah Khan, Tae-Sun Chung, Muhammad Attique, Khalil Khan, Salim El Khediri, And Sadeeq J "Deep Edu: A Deep Neural Collaborative Filtering for Educational Services Recommendation". In. IEEE Access 2020 date of publication June 15, 2020. Volume 8, 2020.

[6] Vidish Sharma, TarunTrehan, Rahul Chanana, Suma Dawn "StudieMe: College Recommendation System". In 2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE). 978-1-7281-2068-3/19/$31.00 ©2019 IEEE.

[7] Madhav Iyengar, Ayanava Sarkar, Shikhar Singh "A Collaborative Filtering based Model for Recommending Graduate Schools". In IEEE Conference 2017. 978-1-5090-5454-1/17/$31.00 ©2017 IEEE.

[8] Dheeraj kumarBokde, Dheeraj kumarBokde, Debajyoti Mukhopadhyay "An Approach to A University Recommendation by Multi-Criteria Collaborative Filtering and Dimensionality Reduction Techniques". In 2015 IEEE International Symposium on Nanoelectronic and Information Systems. 978-1-4673-9692-9/15 $31.00 © 2015 IEEE.

[9] Arta Iftikhar1, Mustansar Ali Ghazanfar1, MubbashirAyub, Zahid Mehmood And Muazzam Maqsood "An Improved Product Recommendation Method for Collaborative Filtering". In IEEE Access 2020 date of publication June 30, 2020. Volume 8, 2020.

[10] Biao Cai, Yusheng Huang "Personalised recommendation algorithm based on covariance". In. The 3rd Asian Conference on Artificial Intelligence Technology (ACAIT 2019). J. Eng., 2020, Vol. 2020 Iss. 13, pp. 577-583.

[11] Antonio Jesús Fernández-García, Roberto Rodríguez-Echeverría, Juan Carlos Preciado, José María Conejero Manzano, And Fernando Sánchez-Figueroa "Creating a Recommender System to Support Higher Education Students in the Subject Enrollment Decision". In. IEEE Access 2020 dateofpublicationOctober16,2020. Volume 8, 2020.

[12] NacimYanes, Ayman Mohamed Mostafa, (Member, IEEE), Mohamed Ezz,, And Saleh NaifAlmuayqil "A Machine Learning-Based Recommender System for Improving Students Learning Experiences". In IEEE Access 2020 date of publication November 5, 2020. Volume 8, 2020.

[13] Dataset from: https://yocket.in/ and https://www.ymgrad.com/