

A K-Nearest Neighbour Algorithm-Based Recommender System for the Dynamic Selection of Elective Undergraduate Courses

Adewale Opeoluwa Ogunde^{*}, Emmanuel Ajibade

Department of Computer Science, Redeemer's University, Ede, Nigeria

Email address:

ogundea@run.edu.ng (A. O. Ogunde)

^{*}Corresponding author

To cite this article:

Adewale Opeoluwa Ogunde, Emmanuel Ajibade. A K-Nearest Neighbour Algorithm-Based Recommender System for the Dynamic Selection of Elective Undergraduate Courses. *International Journal of Data Science and Analysis*. Vol. 5, No. 6, 2019, pp. 128-135.

doi: 10.11648/j.ijdsa.20190506.14

Received: October 15, 2019; **Accepted:** November 14, 2019; **Published:** November 21, 2019

Abstract: The task of selecting a few elective courses from a variety of available elective courses has been a difficult one for many students over the years. In many higher institutions, guidance and counsellors or level advisers are usually employed to assist the students in picking the right choice of courses. In reality, these counsellors and advisers are most times overloaded with too many students to attend to, and sometimes they do not have enough time for the students. Most times, the academic strength of the student based on past results are not considered in the new choice of electives. Recommender systems implement advanced data analysis techniques to help users find the items of their interest by producing a predicted likeliness score or a list of top recommended items for a given active user. Therefore, in this work, a collaborative filtering-based recommender system that will dynamically recommend elective courses to undergraduate students based on their past grades in related courses was developed. This approach employed the use of the k-nearest Neighbour algorithm to discover hidden relationships between the related courses passed by students in the past and the currently available elective courses. Real-life students' results dataset was used to build and test the recommendation model. The new model was found to outperform existing results in the literature. The developed system will not only improve the academic performance of students; it will also help reduce the workload on the level advisers and school counsellors.

Keywords: Collaborative Filtering, Elective Undergraduate Courses, K-nearest Neighbour Algorithm, Recommender Systems

1. Introduction

In many educational institutions, undergraduates usually face a difficult time in choosing electives for many reasons. The number of available electives are sometimes many. There is no direct method of determining whether the student will pass the course. There is also no way the students can also assess their capacity to pass a specific number of extra elective courses. This paper, therefore, proposes a recommender system that would dynamically advise undergraduate students on which electives they are to pick for optimum academic performance. Recommender systems help with searching for suitable web resources, recommend the right solutions to improve students' knowledge or analyse

data obtained from quizzes and provide feedback to instructor to modify a quiz [1]. These systems can enhance the standard and process of teaching and learning. The collaborative filtering recommendation techniques have been proven to provide satisfying recommendations to users. They rely on the experiences of similar users, i.e., users who share the same preferences. Most of collaborative filtering-based algorithms use k-nearest-Neighbourhood algorithm to recommend items. The nearest-Neighbour users are those that exhibit the most substantial relevance to the target user. The k-Nearest Neighbours algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on

whether k-NN is used for classification or regression [2]. In this work, the k-Nearest Neighbours algorithm was deployed to find correlations and hidden knowledge from students' previous grades. The discovered knowledge was used as a knowledge-base to recommend the elective courses to students. The proposed system will help students to dynamically choose the most appropriate electives they can do in a semester for better performance.

2. Recommender Systems

A recommender system is an information filtering system technique that attempts to recommend information items (books, movies, courses, etc.) that are likely to fit a user's taste. They try to predict the rating or preference that a user would give to an item. A recommender system can be used to filter any sort of information, and with any kind of preference. The main methods recommender system utilizes to propose an item to a user include analyzing user data, extracting useful information, and finally predicting items to users. The availability of practically limitless online information makes recommender system absolutely necessary [3]. A lot of research work has been carried out yet the challenges in the field of recommender models persist, and there is plenty of scope for improvement. Aside this, recommender systems have been successful in many fields including the educational environment. Recommender systems have helped with searching for suitable web resources, recommend the right solutions to improve students' knowledge and in general, enhance the teaching and learning processes. The use of efficient and accurate recommendation techniques is essential for a system that will provide good and useful recommendation to its users. This explains the importance of understanding the features and potentials of different recommendation techniques.

Content-Based filtering recommendation system, which recommends items that are similar in content to items the user has liked in the past or matched to attributes of the user, was applied to solve the problem presented in this paper. It usually bases its predictions around the information provided by the user and ignore other contributions from other users. The content-based filtering system can be defined as methods that provide recommendations by comparing representations of content describing an item to representations of content that interests the user [4]. Systems implementing a content-based recommendation approach analyse a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the objects rated by that user.

2.1. K-nearest Neighbour

K-nearest Neighbour is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of the 1970's as a non-parametric technique. K-NN can be used for both classification and regression

predictive problems. It is a non-parametric method. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its Neighbours, with the object being assigned to the class most common among its k nearest Neighbours (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest Neighbour. Whenever we have a new point to classify, we find its K nearest Neighbours from the training data. The similarity metrics do not consider the relation of attributes which result in inaccurate distance and then impact on classification precision. The wrong classification due to presence of many irrelevant attributes is often termed as the presence of many irrelevant attributes is often termed as the curse of dimensionality [5].

In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest Neighbours. Both for classification and regression, it can be useful to assign weight to the contributions of the Neighbours, so that the closer Neighbours contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each Neighbour a weight of $1/d$, where d is the distance to the Neighbour.

2.2. Other Related Works

Four algorithms suitable for the course recommendation were proposed by Hana [1]. The first algorithm searches for the most frequently enrolled courses. The second algorithm utilizes similarities of students based on courses of their interests. The third algorithm recommends courses of students' favorite teachers. The last algorithm calculates the social ties among students and selected courses which were attractive to students' friends. A new collaborative recommendation system that employed association rules algorithm to recommend university elective courses to a target student based on what other similar students have taken was proposed by Al-Badarenah, and Alsakran [2]. The experiments showed that association rule is a popular tool for making recommendation to a target student. Through their experiments, it was noticed that the patterns of influence of different parameters on the performance of the system. The confidence and match of a rule have a significant impact on the performance, but the highest confidence or match may not be the best choice. By choosing a relatively high confidence or match, better performance can be achieved.

Recommender systems open new opportunities for retrieving personalized information on the Internet [6]. It also helps to alleviate the problem of information overload which is a widespread phenomenon with information retrieval systems and enables users to have access to products and services which are not readily available to users on the system. Their work discussed the two traditional recommendation techniques and highlighted their strengths and challenges with diverse kind of hybridisation strategies used to improve their performances. Various learning algorithms used in generating recommendation models and evaluation metrics used in measuring the quality and

performance of recommendation algorithms were discussed.

Systems that made only top-n recommendations have also been proposed in the literature [7, 8]. Two collaborative filtering approaches for predicting the grade a student will obtain in different courses based on their performance in earlier courses was compared by Sanjog and Sharma [9]. They used the collaborative filtering approach to recommend grades students will get in a course if they select some elective courses out of the range of available elective courses [9]. Experimental evaluation using the collaborative filtering approach on a real-life data set showed good results.

Two essential works deployed neural networks to build relationships between users and products, considering matrix factorisation for determining user preferences [10, 11]. An auto-encoder based collaborative filtering technique which employed original and partially perceived vectors was proposed by Ouyang et al. [12]. Another research focused on YouTube recommendations using a Multilayered Perceptron technique, which produced good results [13]. A related work that recommended elective subjects based on the neural network and association rules using the Rapid Miner Tool was presented by Samrit and Thomas [14]. Their system worked on real database obtained from the university. It analysed the past behaviour of students concerning their elective subject choices. More explicitly, it formalises association rules that were implicit before. The accuracy of the system and other performance metrics for the system were not presented in the study.

Collaborative metric learning, which maximised the distance between users and their disliked items and minimised the gap between the users and their liked items was proposed by Hsieh et al. [15]. A survey of the literature of course recommender systems for course selection was carried out by Unelsrod [16]. One of the goals was to develop and test a recommender system that uses a combination of collaborative filtering and content-based filtering to give recommendations. It was achieved by using memory-based collaborative filtering with k-nearest Neighbour search to find closely correlated users, and content filtering based on the most popular courses with older students and which teachers the user has shown a preference for.

Other recommender systems have been built to solve problems in several domains such as online libraries, e-commerce sites, LinkedIn, Google, Facebook, Twitter and other social media [17-19]. A very recent research also developed a system that recommended potential industrial training organizations to students using the J48 algorithm and a decision tree-based model [20]. The Placement predictor and course recommender system tool was developed to improve the quality of education and enhance school resource management using C4.5 algorithm, which gave a much better prediction than any other classification algorithms used in the work [21].

In a very recent work [22], a course recommender system that aims to improve students' career readiness by suggesting relevant skills and courses based on their unique career interests was presented. The course recommender system

used content-based filtering and an ensemble learning algorithm of k-means clustering and TF-IDF to suggest relevant skills and courses based on students' career interests. More information, analysis and comparison of several methods used for building course recommendation systems from educational data set was presented by Thanh-Nhan et al. [23].

3. Methodology

3.1. Data Collection and Description

The data used in this work was collected from the Department of Computer Science, Redeemer's University. The data consist of student grades in both compulsory and elective courses for five academic sessions, that is, 2010/2011 to 2015/2016 session. The data was made up of the students' grades in which A was the maximum grade, and F was the minimum grade. The status of course can either be compulsory (C) or elective (E). The list of both elective and compulsory courses was also obtained. The attributes described here are shown in Table 1.

Table 1. Description of Data.

Variable	Description	Possible Values
C	Compulsory Course	
E	Elective Course	
GR	Grade	A, B, C, D, E and F

To simplify the work for analysis, domain values were associated with each of the attributes used. The domain values for the variables defined for the present investigation are as follows:

1. C - Compulsory Course
2. E - Elective Course
3. CMP - Computer
4. GST - General Study
5. CHE - Chemistry
6. STA - Statistics
7. MAT - Mathematics
8. PHY - Physics
9. S - Score
10. GR - Grade

The final grade is represented by the values A, B, C, D, E and F. These values are expanded below.

- A. Values from 70 and above.
- B. Values ranging from 60 to 69.
- C. Values ranging from 50 to 59.
- D. Values ranging from 45 to 49.
- E. Values ranging from 40 to 44.
- F. Values below 40.

The data to be trained is a collection of students results for five academic sessions in various courses (2011/2012, 2012/2013, 2013/2014, 2014/2015 and 2015/2016 sessions). All Sessions consist of students' results in both elective and compulsory courses. The detailed characteristics of the students' results obtained are shown in Table 2. Altogether, a total of ten thousand six hundred and one data instances were

obtained.

Table 2. Data Characteristics.

Session	Instances
2011/2012	1506
2012/2013	1785
2013/2014	2400
2014/2015	2340
2015/2016	2570
TOTAL	10, 601

3.2. Design of the Elective Course Recommender System

No.	1: semester	2: session	3: coursecode	4: status	5: score	6: grade
1	1	2.014201SE7	MAT101	C	45.0	D
2	1	2.014201SE7	MAT101	C	50.0	C
3	1	2.014201SE7	CMP101	C	63.0	B
4	1	2.014201SE7	MAT101	C	51.0	C
5	1	2.014201SE7	CMP101	C	50.0	C
6	1	2.014201SE7	MAT101	C	40.0	E
7	1	2.014201SE7	MAT101	C	73.0	A
8	1	2.014201SE7	CMP101	C	63.0	B
9	1	2.014201SE7	MAT101	C	56.0	C
10	1	2.014201SE7	CMP101	C	60.0	B
11	1	2.014201SE7	CMP101	C	55.0	C
12	1	2.014201SE7	MAT101	C	55.0	C
13	1	2.014201SE7	CMP101	C	70.0	A
14	1	2.014201SE7	MAT101	C	46.0	D
15	1	2.014201SE7	MAT101	C	40.0	E
16	1	2.014201SE7	MAT101	C	52.0	C
17	1	2.014201SE7	CMP101	C	64.0	B
18	1	2.014201SE7	CMP101	C	51.0	C
19	1	2.014201SE7	MAT101	C	40.0	E
20	1	2.014201SE7	MAT101	C	55.0	C
21	1	2.014201SE7	CMP101	C	55.0	C
22	1	2.014201SE7	CMP101	C	55.0	C
23	1	2.014201SE7	MAT101	C	66.0	B
24	1	2.014201SE7	CMP343	C	48.0	D
25	1	2.014201SE7	MAT201	C	46.0	D
26	1	2.014201SE7	CMP345	C	52.0	C
27	1	2.014201SE7	MAT317	C	19.0	F
28	1	2.014201SE7	CMP333	C	55.0	C
29	1	2.014201SE7	CMP349	C	55.0	C

Figure 1. Sample data in .arff format.

The recommender system, in this work, was designed to contain a knowledge-base that has accumulated experience and a set of rules for applying the knowledge-base to each particular situation that was described. Collaborative filtering recommendation approach was applied to analyse the data. This filtered information by using the recommendations of other students that passed the same elective courses to predict future cases. This was premised on the idea that people who passed certain elective courses in the past are likely to pass similar elective courses in the future. The collaborative filtering system used in this work applied the Neighbourhood-based technique. In the Neighbourhood-based approach, a number of elective courses were selected based on their similarity to the active student. A prediction for the active course was made by calculating a weighted average of the ratings of the selected courses. Using k -Nearest Neighbour classification, the training dataset was used to classify each member of the "target" dataset. The structure of the data is that there is a classification

(categorical) variable of interest and a number of additional predictor variables.

WEKA, a popular data mining tool, was used to generate the recommendation model and the k -nearest Neighbour decision tree. The primary data used for the analysis was imported from a database in CSV format and converted to ARFF format, which is suitable for analysis by WEKA. Data preprocessing was done using the filtering algorithm. The selected data in CSV format was pre-processed by the WEKA ARFF viewer. The converted data was stored as "reco.arff". A sample of the converted data is shown in figure 1. The field attributes of figure 1 are serial number, semester, session, coursecode, course status, score and grade.

The K -Nearest Decision tree was generated from the stored .arff data format in WEKA explorer. The attributes in the data include semester, session, coursecode, status, score and grades. All these were used to generate rules for elective course recommendation. The K -Nearest rules generated using WEKA, and the results classifier are shown in figures 2 and 3 respectively.

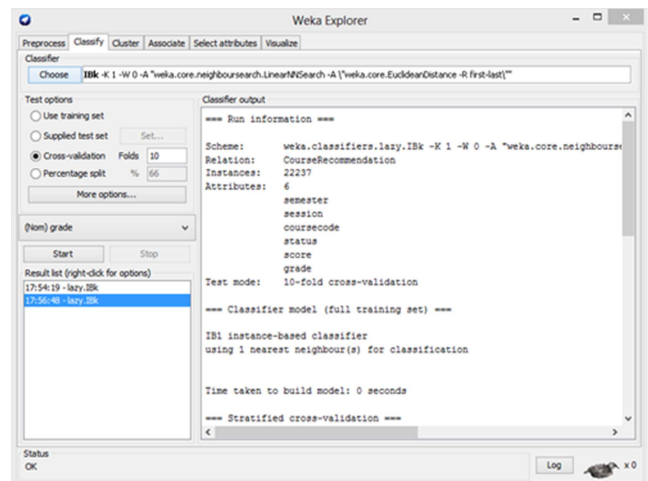


Figure 2. K-Nearest Rules Generated using WEKA.

A classifier with cross-validation option (10 folds) was used to classify the data and result is shown in figure 2.

Figure 3 shows the results of the classifier. Correctly classified data instances were 95.6469% while incorrectly classified data instances were 4.3531%. This proved a high level of accuracy for the model generated. Kappa statistic measures the agreement of prediction with the actual class. In figure 2, the kappa statistic was shown to be 0.9467, which means a higher agreement between the predicted values and the actual class values. Mean Absolute Error (MAE), which is a measure of the average magnitude of the errors in a dataset without considering their direction, was 0.0146. The small value of the MAE indicated that the error associated with this classification was very minimal. Root Mean Squared Error (RMSE) was 0.1178, which is a measure of the average magnitude of the error. The RMSE is expected to be greater or equal to the MAE, which is the case in this work. This implied a

variance in the individual errors of the dataset. The True Positive (TP) value signifies the correct predictions. TP had an average value of 0.956, which implied correct prediction for majority of the class labels. The False Positive (FP) value is the number of instances predicted positive but was actually negative. In this work, FP had an average value of 0.010, which implied very few cases of incorrect predictions. The Recall was an average value of 0.956, which was an indication that majority of the class instances predicted to be positive was actually positive. The average value of Precision, in this work, was 0.956, which corroborated the fact that many cases that were predicted positive were positive. In this work, the average value of the F-measure was 0.956, which implied that the recall and the precision were evenly weighted. The average value of the MCC was 0.947. This indicated that the model was predicting with a very high level of accuracy.

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      21269      95.6469 %
Incorrectly Classified Instances     968         4.3531 %
Kappa statistic                     0.9467
Mean absolute error                  0.0146
Root mean squared error              0.1178
Relative absolute error              5.3466 %
Root relative squared error          31.9016 %
Coverage of cases (0.95 level)      96.0831 %
Mean rel. region size (0.95 level)  16.8091 %
Total Number of Instances          22237

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
	0.968	0.006	0.973	0.968	0.970	0.964
	0.957	0.012	0.950	0.957	0.954	0.942
	0.970	0.014	0.958	0.970	0.964	0.952
	0.887	0.010	0.905	0.887	0.896	0.885
	0.955	0.010	0.951	0.955	0.953	0.943
	0.971	0.001	0.995	0.971	0.983	0.981
Weighted Avg.	0.956	0.010	0.956	0.956	0.956	0.947

```

=== Confusion Matrix ===
  a  b  c  d  e  f  <-- classified as
3678 116  3  1  2  0 | a = A
100 4180 84  1  1  0 | b = B
  1 100 5437 60  6  0 | c = C
  1  2 134 1981 113  2 | d = D
  2  3 16 139 3632 10 | e = E
  0  0  0  6 65 2361 | f = F

```

Figure 3. Result of the Classifier.

4. Implementation and Results

The rules generated by the classifier were used as a knowledge-base for building a front-end web-based recommendation system, which was thoroughly tested to ensure that it complies with the rules. Some of the development tools include Adobe Dreamweaver and Sublime Text. The Admin is the user that performs administrative activities on the system such as editing, adding and deleting

of courses from the system. The home page of the elective course recommendation system is shown in figure 4.

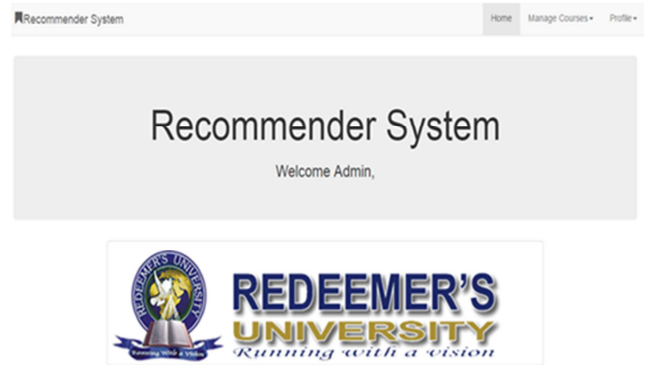


Figure 4. Home Page.

The login page for the recommender system is shown in figure 5. This page allows a student to either present his/her login credentials consisting of the student's email address and the student's password or head to the sign-up page to create an account.

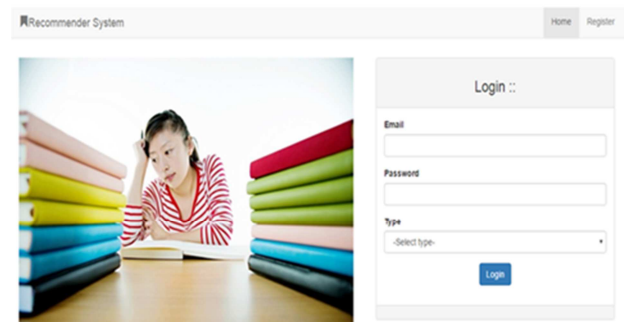


Figure 5. Login Page.

The flowchart in figure 6 describes the steps a student must follow to view recommendations concerning the choice of elective courses.

The student needs to register or log in if already registered. If the login is valid, the student uploads his/her past results into the recommender system after which the student selects his/her department, level and the current semester in which the recommendation is needed. After that, the students can now view the recommendations of elective courses he/she can register in the current semester.

The developed recommendation system contains several useful pages to students. Some of them are: Update Past Results Page, View Recommendation Page, Recommendation Page etc. The Update Past Result page (figure 7) enables the user to update his/her past results and when they are done the system stores the information for future use. This increases the knowledge in the knowledge-base from time to time and improves the quality of recommendations with time.

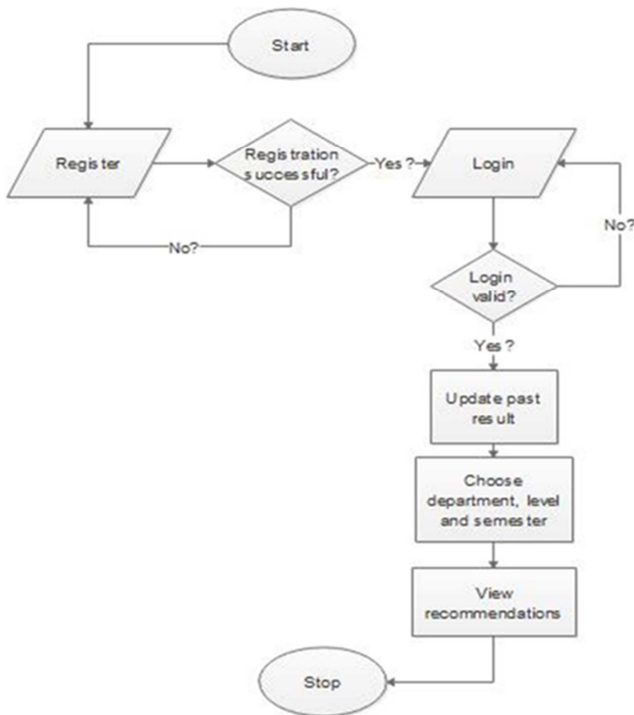


Figure 6. Recommendation steps.

Figure 7. Update Results Page.

The Prepare Recommendation (figure 8) helps the student to select his/her details and the current semester in which the recommendation is needed and after that 'View Recommendation' button is clicked.

Figure 8. Prepare Recommendation Page.

When the 'view recommendation' button is clicked, the application triggers the recommendation engine based on the value inputted, and results of the recommendation are presented to the user (figure 9).

Figure 9. Sample Recommendation.

4.1. Class Accuracy and Accuracy Model for Course Recommendation

The true positive, false positive, and the correct precision in percentages for the course recommendation system based on the six-grade outcome categories are shown in table 3.

Table 3. Class-wise accuracy for the recommendation system.

GRADE	True Positive (TP)	False Positive (FP)	Correct Precision (%)
A	0.968	0.006	79.4%
B	0.957	0.012	70.4%
C	0.970	0.014	82.7%
D	0.887	0.010	71.0%
E	0.955	0.010	82.7%
F	0.971	0.010	89.7%

It was observed that grade F had the highest correct precision (89.7%), while grade B had the least correct precision (70.4%). This implies that the recommendation system is working at a very high level of accuracy in predicting all the grades.

4.2. Overall Accuracy Percentage of the Model

Table 4 shows the accuracy percentage for the k-nearest Neighbour classification of all the data instances. The overall accuracy percentage of the model is 95.6469%. This outperformed all the related works described in section two and further confirmed that the model for the developed recommender system is very efficient and predicting at a very high level of accuracy.

Table 4. Accuracy Percentages.

Algorithm	Correctly Classified Instances	Incorrectly Classified Instances
K-Nearest Neighbour	95.6469%	4.3531%.

5. Conclusions and Future Works

A study was conducted on the dynamics of recommending elective courses to undergraduate students based on the past performances of other students in the course and the results of the student under study using a machine learning technique. From the class-wise accuracy, it was clear that the true positive rates for the grades A, B, C, D, E and F were 79.4%, 70.4%, 82.7%, 71.0%, 82.7% and 89.7% respectively. A decision tree-based model from which prediction rules were extracted and implemented in a front-end web application. It could be concluded that this model with similar functionalities can aid schools in effectively monitoring students' progress. The university can advise students on steps to take to ensure they come out successful in their examinations. The effectiveness and efficiency of this machine learning application have made it possible to solve real-world problems in the form of decision making in higher institutions. It is highly recommended that institutions should start adopting similar systems in their educational network to vastly improve educational experiences and render timely and effective decision-making solutions to students. Future works will include experiments using other types of decision tree algorithms. A comparative analysis of the performances of the algorithms would be carried out. Finally, data from a single discipline was used for the analysis in this work; future works will consider the development of a generic system with data from many disciplines.

Acknowledgements

The authors would like to appreciate the department of computer science, for providing the data used in executing this research.

References

- [1] B. Hana, Course Enrollment Recommender System. Proceedings of the 9th International Conference on Educational Data Mining, (2016), pp 312-317.
- [2] A. Al-Badarenah, J. Alsakran, An Automated Recommender System for Course. International Journal of Advanced Computer Science and Applications, Vol. 7, No. 3, (2016), pp 166-175.
- [3] K. D. Gupta, A Survey on Recommender System, International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 14, Number 14, (2019) pp. 3274-3277.
- [4] P. Melville, V. Sindhwani, Recommender Systems, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, (2010), pp 1-18.
- [5] D. Siddharth, Weighted K Nearest Neighbour, CS 8751, (2009).
- [6] F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh, Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, Vol. 16, (2015), pp 261-273.
- [7] Z. Xu, C. Chen, T. Lukasiewicz, Y. Miao, X. Meng, "Tag-Aware Personalized Recommendation using a Deep-Semantic Similarity Model with Negative Sampling," Proc. 25th ACM International on Conference on Information and Knowledge Management, (2016), pp. 1921-1924.
- [8] A. M. Elkahky, Y. Song, X. He, "A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems," Proc. 24th International Conference on World Wide Web, (2015), pp. 278-288.
- [9] R. Sanjog, A. Sharma, A collaborative filtering based approach for recommending elective courses, Springer-Verlag Berlin Heidelberg, (2011), pp. 30-39.
- [10] G. K. Dziugaite, D. M. Roy, "Neural Network Matrix Factorization," arXiv 103 preprint arXiv: 1511.06443, 2015.
- [11] X. He, L. Liao, H. Zhang, L. Y. Nie, X. Hu, T. S. Chua, "Neural Collaborative Filtering," Proc. 26th International Conference on World Wide Web, (2017), pp. 173-182.
- [12] Y. Ouyang, W. Liu, W. Rong, Z. Xiong, "Autoencoder-Based Collaborative Filtering," 21st International Conference on Neural Information Processing, (2014), pp. 284-291.
- [13] P. Covington, J. Adams, E. Sargin, "Deep Neural Networks for YouTube Recommendations," Proc. 10th ACM Conference on Recommender Systems, (2016), pp. 191-198.
- [14] N. B. Samrit, A. Thomas, A Recommendation System for Prediction of Elective Subjects, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 5 Issue 4, (2017), pp 36-43.
- [15] C. K. Hsieh, L. Yang, Y. Cui, T. Y. Lin, S. Belongie, D. Estrin, "Collaborative metric learning," Proc. 26th International Conference on World Wide Web, (2017), pp. 193-201.
- [16] H. F. Unelsrød, Design and Evaluation of a Recommender System for Course Selection, Published Master of Science in Computer Science Project, Norwegian University of Science and Technology Department of Computer and Information Science, (2011), pp 1-43.
- [17] R. He, J. McAuley, "VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback," Proc. 13th AAAI Conference on Artificial Intelligence, (2016), pp. 144-150.
- [18] B. Bai, Y. Fan, W. Tan, J. Zhang, "DLTSR: A Deep Learning Framework for Recommendation of Long-tail Web Services," IEEE Transactions on Services Computing, (2017), pp. 1-11.

- [19] T. Donkers, B. Loepp, J. Ziegler, "Sequential User-Based Recurrent Neural Network Recommendations," Proc. 11th ACM Conference on Recommender Systems, (2017), pp. 152–160.
- [20] A. O. Ogunde, J. O. Idialu, A recommender system for selecting potential industrial training organizations. Engineering Reports. 2019; e12046. <https://doi.org/10.1002/eng2.12046>.
- [21] P. P. Wadekar, Y. P. Pillai, M. U. Roy, N. Phadnis, Placement Predictor and Course Recommender System, International Research Journal of Engineering and Technology, Volume: 05 Issue: 03, (2018), pp 3960-3965.
- [22] S. Shahab, "Next Level: A Course Recommender System Based On Career Interests" (2019). Master's Projects. 684. DOI: <https://doi.org/10.31979/etd.z4v2-k6gg>.
- [23] H. Thanh-Nhan, H. Nguyen, N. Thai-Nghe, Methods for building course recommendation systems, Eighth International Conference on Knowledge and Systems Engineering (KSE), 2016, DOI: 10.1109/KSE.2016.7758047.