An Improved Course Recommendation System Based on Historical Grade Data Using Logistic Regression

Idowu Dauda Oladipo[1], Joseph Bamidele Awotunde[1](✉), Muyideen AbdulRaheem[1], Oluwasegun Osemudiame Ige[1], Ghaniyyat Bolanle Balogun[1], Adekola Rasheed Tomori[2], and Fatimoh Abidemi Taofeek-Ibrahim[3]

[1] Department of Computer Science, University of Ilorin, Ilorin, Nigeria
{odidowu,awotunde.jb,muyideen,balogun.gb}@unilorin.edu.ng,
16-52ha043@students.unilorin.edu.ng

[2] Directoriate of Computer Sciences and Information Technology,
University of Ilorin, Ilorin, Nigeria
tomori@unilorin.edu.ng

[3] Department of Computer Science, Federal Polytechnic, Offa, Nigeria

Abstract. Elective course selection is very important to undergraduate students as the right courses could provide a boost to a student's Cumulative Grade Point Average (CGPA) while the wrong courses could cause a drop in CGPA. As a result, institutions of higher learning usually have paid advisers and counsellors to guide students in their choice of courses but this method is limited due to factors such as a high number of students and insufficient time on the part of advisers/counsellors. Another factor that limits advisers/counsellors is the fact that no matter how hard we try, there are patterns in data that are simply impossible to detect by human knowledge alone. While many different methods have been used in an attempt to solve the problem of elective course recommendation, these methods generally ignore student performance in previous courses when recommending courses. Therefore, this paper, proposes an effective course recommendation system for undergraduate students using Python programming language, to solve this problem based on grade data from past students. The logistic regression model alongside a wide and deep recommender were used to classify students based on whether a particular course would be good for them or not and to recommend possible electives to them. The data used for this study was gotten from records of the Department of Computer Science, University of Ilorin only and the courses to be predicted were electives in the department. These models proved to be effective with accuracy scores of 0.84 and 0.76 and a mean-squared error of 0.48.

Keywords: Course recommendation · Grade data · Logistic regression · Classification · Wide and deep recommender · Machine learning

1 Introduction

The computing environment for learning is changing rapidly, due to the emergence of new information and communication technology such as big data [1]. Learning methods

H. Florez and M. F. Pollo-Cattaneo (Eds.): ICAI 2021, CCIS 1455, pp. 207–221, 2021.
https://doi.org/10.1007/978-3-030-89654-6_15

are also changing every day and so e-learning systems need to develop more techniques and tools to meet the increased need of learners around the world. The choice of elective courses to register is a problem for many undergraduate students in universities as elective courses can help students to improve their Cumulative Grade Point Average (CGPA) if they are chosen wisely; on the other hand, if poor choices are made in the selection of elective courses, students run the risk of a drop in their CGPA. Sometimes, the sheer number of possible electives that can be chosen can cause a student to be confused as to which ones would be best for him/her. Also, the fact that there is no certain way to know how well one would perform in elective courses before selecting them is another factor that makes elective course selection difficult.

Artificial intelligence methods that were developed at the beginning of research are now being applied to information retrieval systems [2]. Recommendation systems provide a promising approach to information filtering as they help users to find the most appropriate items without much effort. Based on the needs of each user, recommendation systems can generate a series of personalized suggestions. Despite the high impact and usefulness of course recommendation systems, they are limited because models based on keywords may not address individual needs. They also do not, usually, in cases of collaborative filtering, association rules, and decision trees, use historical information about the courses. Content-based filtering models also fall short as they are usually based on specific recommendations only rather than more generalized recommendations. They also do not provide comprehensive information about the courses that are most relevant to students.

The proficiency of an algorithm to predict a student's performance in a particular course can be instrumental in aiding students' course selection choices and data mining and machine learning techniques have been seen to be useful in this domain [3]. Singular Value Decomposition (SVD) based algorithms have proved to yield very good results in course recommendation systems but they are limited by the density of the data matrix [4]. CGPA has also been found to be an important measure of course readiness [3]; thus, making it ideal for making predictions. Baseline predictors, Student/Course k-NNs collaborative filtering, Latent factor models - Matrix factorization (MF), Latent factor models - Biased MF (BMF) are other methods that have, so far, been used to build course recommendation systems [5]. Artificial Neural Network models have also been used in an attempt to solve this problem [6].

Recommender systems aim at providing their users with relevant information. A model in a recommender system evaluates the personal information of the user, and a model estimating scores for items not yet seen by the user is also developed [7]. The personalized guidance is based on prior grades of other students, collected in a historical database. Similarities between the elements are evaluated, to find the most suitable elements. The focus of this study is to develop a course recommendation system by training a Logistic Regression model, alongside a Wide and Deep Recommender on collected data of past students' grades to solve this problem. The paper intends to show that the logistic regression model could be effectively used to predict course outcomes for students while the Wide and Deep Recommender could be effectively used to make good recommendations about courses to choose for students after being trained with historical grade data from previous students. A huge amount of grade data can be gathered from school records of students that have passed through the school system previously and more data is known to beat better algorithms [8].

The Python programming language will be used to implement this research due to its available and extensive machine learning, mathematical and statistical packages such as NumPy, Sckit-Learn, MatPlotLib, etc. The course recommendation system to be developed will contribute to making course selection choices easier for undergraduate students. It will also help level advisers to better advice students on what courses they would do better in. Conventional measures of recommendation quality will also be studied in this paper. While some recommender systems have been put in place to solve this problem, they generally struggle due to a lack of sufficient training data or imbalances in data [9]. Also, many recommender systems are based on students' learning styles but these have been proven to change over time [10]. Furthermore, the use of the logistic regression model in course recommendation are still scanty while the effects of the wide and deep recommendation in this area, that is, course recommendation, have not been empirically tested.

In this paper, balanced historical grade data in course recommendations are used due to the huge amount of immutable data that can be gotten from school records of students who have gone through the school. Also, the paper recommends the use of the logistic regression model and the wide and deep recommender in an attempt to solve this problem. Each of these models will be trained and evaluated with historical grade data and implemented using Flask, a minimal web-based framework.

This paper aims to create a course recommendation system for students in undergraduate degree programmes based on collected historical records of past grades. The main contributions of this paper are:

(i) to select the variables used for each course to be predicted;
(ii) to design, implement and train the recommendation models;
(iii) to evaluate the trained models.

2 Related Work

One method that has been used in this domain is the use of an objective function to distinguish between courses that are expected to increase or decrease a student's GPA [11]. Morsy and Karypis [11] tried to tackle this problem by combining the grades predicted by grade prediction methods with the classifications generated by course recommendations to improve the final course rankings. In both methods, authors adjusted two commonly-used representation methods to find out the ideal chronological sequence in which courses are to be taken. The representation methods used in this paper were Singular Value Decomposition (SVD) and Course2vec. SVD factorizes a known matrix through getting a solution to $X = U \sum V^T$, where the columns of U and V are the left and right singular vectors respectively, and $\sum$ is a diagonal matrix containing the singular values of X. It was applied on a previous-subsequent co-occurrence frequency matrix F, with Fij being the number of students that had taken course i before taking course j. Having used SVD to estimate the previous and subsequent course sets, and computed each student's inherent profile by averaging over the sets of the courses taken by the said student in all preceding terms, after which, they computed the dot product between said student's profile and the SVD estimates of each course, ranking the courses in non-increasing order corresponding to the dot products and then selecting the top courses for final recommendation. Course2vec, on the other hand, was modelled using a many-to-one, log-linear model, which was inspired by the word2vec Continuous Bag-Of-Word (CBOW) model. While word2vec works on sequences of individual words in a given document where a set of words within a pre-defined window size are used to predict the intended word, course2vec works on sequences of ordered terms taken by each student where each term contains a set of courses and the previous set of courses would be used to predict future courses for each student. The results of the methods showed that course recommendation methods that consider grade information perform better than course recommendation methods which do not consider grade information.

Zabriskie et al. [3] used random forest and logistic regression models to construct early warning models of student success in two physics courses at a university. Combining in-class variables such as homework grades with institutional variables such as CGPA, they were able to predict if students would receive grades less than "B" in the two courses that were considered with 73% accuracy in one and 81% accuracy in the other.

Researchers in [12] approached the course recommendation problem by trying to find relationships between students' activities through the association rules method to enable students to select the best learning materials. The focus of their research was on the analysis of past historical data of course registration or log data. Their article essentially examined the frequent item sets concept to discover the note-worthy rules in the transaction database. Using those rules, the study established a list of more appropriate courses based on the student's behaviours and preferences. Their model was essentially based on the parallel FP-growth algorithm made available by Spark Framework and the Hadoop ecosystem.

The proposed recommendation system involved three major stages: firstly, data collection; secondly, the discovery of connections between user behaviours; thirdly, the recommendation of more appropriate courses for users. FP-growth (frequent pattern growth), being a proficient, scalable, and fast algorithm for extracting items that seem more closely associated, was used as the method of implementing the association rules method to determine the more appealing relationships between items in the database. FP-growth itself can be implemented in several ways. They used a parallel version of the algorithm called parallel FP-growth (PFP) which is based on a new computation distribution scheme, that is, it allocates tasks around a collection of nodes using the MapReduce model, making PFP faster and more scalable than the conventional FP-growth algorithm which is based on single-machine.

Another method that has been considered effective in course recommendation is the k-nearest neighbour algorithm [13]. The system was based on the notion that students who did well in the same previous courses would do well in the same future electives. It comprised of a knowledge base that had gathered prediction proficiency and a collection of rules for applying the knowledge base to individual circumstances. Using a neighbourhood-based approach, the developed system recommended electives as a function of their comparison to the students' grades and predictions for each course were attained by computing a weighted average of the ratings of the chosen electives. The system developed had an accuracy score of 95.65% for correctly classified instances and 4.35% for incorrectly classified instances while it had a mean absolute error of 0.0146 and a root-mean-square error of 0.1178.

It was reported on the examination of a web-based decision assistance application that assists student advising [14]. The tool was evaluated and found to be successful and efficient for academic advising by a substantial percentage of respondents; however, the details of its application were not included in the research.

Techniques for detecting responses [15–17] use a data model based on missing data theory to model NMAR data. To represent NMAR data, the method suggested in [15] modified probabilistic matrix factorization by introducing two modifications. The first variation assumes that the likelihood of detecting a rating is solely determined by its value. The second variation considers that the likelihood is likewise affected by the user and the latent components of the item. None of these strategies take into account the user and item characteristics that influence reaction patterns.

Hana [18] proposed a mechanism-based technique for recommending courses to students by investigating the student's academic record and comparing it to the records of others to determine similarities. The system then determines and advises the course he is good at or interested in taking so that he can pass the course.

The acronym PEL-IRT refers to "Personalized E-Learning System Using Item Response Theory" [19]. It suggests relevant course content to students, taking into account both the difficulty of the course material and the student's competence. Students can utilize PEL-IRT to search for interesting course material by selecting course categories and units and using appropriate keywords.

In [20], educational data was connected to a user/item. The recommendation was generated using the matrix factorization technique, and the approach was validated using logistic regression.

3 Methodology

The study addresses the practical problem of course recommendation concerning historical grade data using logistic regression and the wide and deep recommender. This study will be experimental in nature as it would involve running experiments on collected data. The work will be applied in nature to develop a usable technology for course recommendation as a practical problem. It will also be exploratory in nature as the researchers intend on studying the usage of the logistic regression model and the wide and deep recommender in this domain, a position that has not been taken so far in explored literature.

3.1 Data Collection

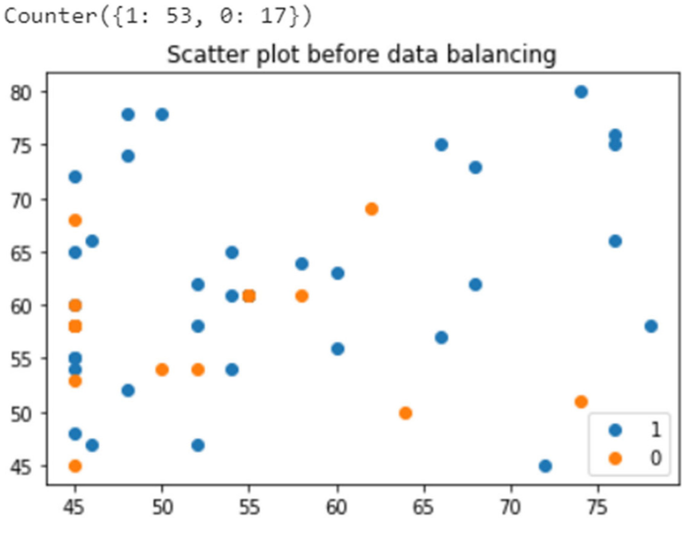
The data that was used for this study was acquired from University of Ilorin. It contained student scores in both core and elective courses for the 2018/2019 session.

3.2 Data Pre-processing

After the data had been collected and organized, the classifier undertook data pre-processing operations such as data cleaning, data balancing, etc.

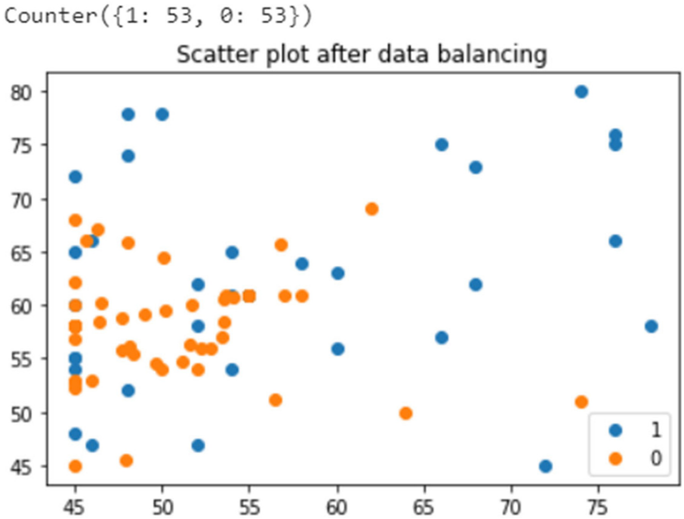
Data cleaning consisted mainly of cleaning missing data from the dataset. Missing data mainly took the form of courses that were not taken by students who came into the school through direct entry rather than the Unified Tertiary Matriculation Examination (UTME) and courses that were taken in some sessions and not taken in others. Machine learning algorithms generally struggle to work with missing data and we tried to avoid this difficulty by filling in the average score of all the students who took such courses.

Usually, when dealing with real-world classification problems, the data tends to be skewed towards a particular class. In elective courses, pass rates are generally positively skewed and this will lead to inaccurate predictions as the model is likely to predict more passes than expected due to finding more passes in the data. To fix this problem, the model will be using the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset. This will be done by creating "new" instances of failed outcomes using the current failed outcomes in the dataset until the dataset is more equally balanced. This is expected to yield more accurate predictions. Scatterplots showing the observations in the dataset before and after balancing are shown in Figs. 1 and 2.



```
Counter({1: 53, 0: 17})
```

Fig. 1. Scatter plot before data balancing



```
Counter({1: 53, 0: 53})
```

Fig. 2. Scatter plot after data balancing

3.3 Feature Selection

While the implemented algorithm may be the main focus of any machine learning task, it is also important to concentrate on the data that is being passed into the algorithm when trying to solve real-world problems. Like all other computerized tasks, machine learning algorithms work on the GIGO rule: Garbage In, Garbage Out, and in this case, "Garbage In" refers to noisy, uncorrelated data while "Garbage Out" refers to poor model performance. The goal of feature selection is to pick out only features (independent variables) that show a strong correlation to the output (dependent variable). When feature selection is properly done, it leads to faster model optimization time, reduced model convolution, better model accuracy and performance, and above all, less overfitting.

Several feature selection methods exist including Forward Selection, Recursive Feature Elimination, Bi-directional elimination, and Backward Elimination. For this study, Backward elimination will be used because according to Kleinbaum et al. [15], in models that focus on prediction, backward elimination is one of the more appropriate methods to be used. The algorithm for backward elimination is shown in Fig. 3. Also, the columns in the dataset before feature selection are shown in Fig. 4 while the columns after feature selection are shown in Fig. 5.

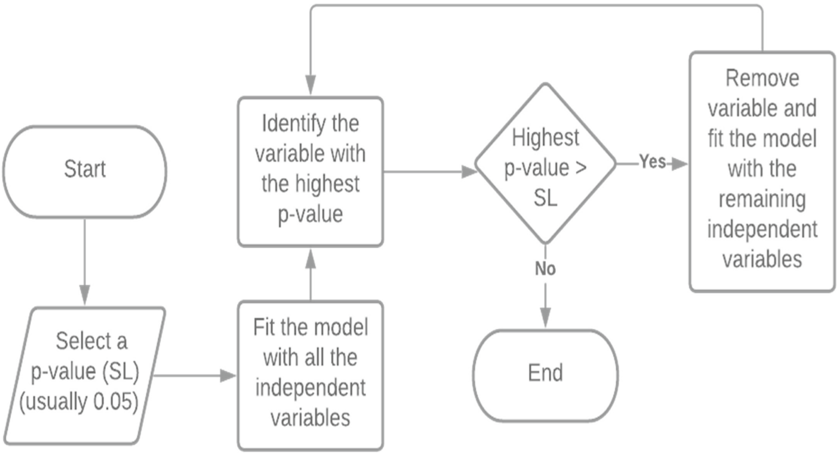
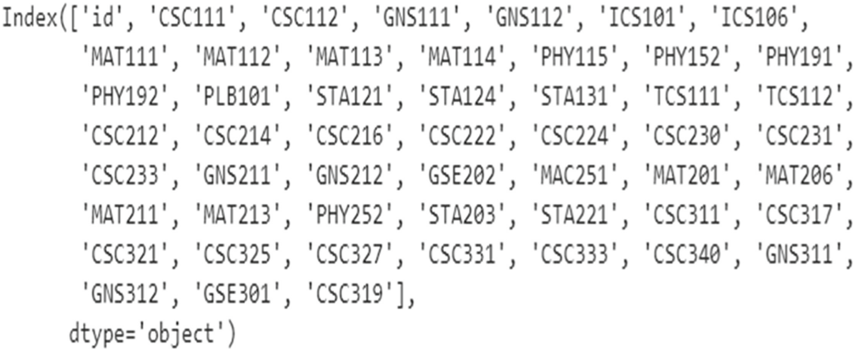
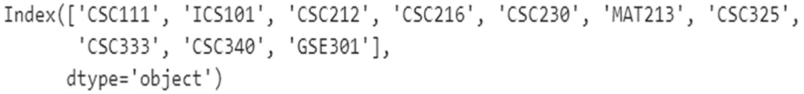

Start
Select a p-value (SL) (usually 0.05)
Fit the model with all the independent variables
Identify the variable with the highest p-value
Highest p-value > SL
Yes
Remove variable and fit the model with the remaining independent variables
No
End

Fig. 3. Backward elimination



```
Index(['id', 'CSC111', 'CSC112', 'GNS111', 'GNS112', 'ICS101', 'ICS106',
       'MAT111', 'MAT112', 'MAT113', 'MAT114', 'PHY115', 'PHY152', 'PHY191',
       'PHY192', 'PLB101', 'STA121', 'STA124', 'STA131', 'TCS111', 'TCS112',
       'CSC212', 'CSC214', 'CSC216', 'CSC222', 'CSC224', 'CSC230', 'CSC231',
       'CSC233', 'GNS211', 'GNS212', 'GSE202', 'MAC251', 'MAT201', 'MAT206',
       'MAT211', 'MAT213', 'PHY252', 'STA203', 'STA221', 'CSC311', 'CSC317',
       'CSC321', 'CSC325', 'CSC327', 'CSC331', 'CSC333', 'CSC340', 'GNS311',
       'GNS312', 'GSE301', 'CSC319'],
      dtype='object')
```

Fig. 4. Columns before feature selection



```
Index(['CSC111', 'ICS101', 'CSC212', 'CSC216', 'CSC230', 'MAT213', 'CSC325',
       'CSC333', 'CSC340', 'GSE301'],
      dtype='object')
```

Fig. 5. Columns after feature selection

3.4 Division of Data

The data was then divided into a training dataset and a test dataset. This was done by randomly sampling the whole dataset without replacement. The training dataset (which contained 70% of the original data) was used to construct the model while the test dataset (which contained 30% of the original data) was reserved for model appraisal with "new" data.

Model Optimization

Logistic Regression

Logistic regression is a mathematical modelling approach that can be used to describe the relationship of several X's to a dichotomous dependent variable, such as D [15] where X represents the independent variable and D represents the dependent variable. It is a very widely used machine learning technique because the logistic function never extends below 0 and above 1 and it provides an attractive S-shaped depiction of the collective influences of the various dependant variables being used in the prediction [15]. The logistic regression model is based on the logistic function, which is given in Eq. 1.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

To get the logistic model from the logistic function, we use the formula $z = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$, where the X's are independent variables and α and the βi are constant terms representing unknown parameters. Thus, z becomes an index that pools the X's together, giving us a new logistic function shown in Eq. 2.

$$f(z) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i X_i)}} \tag{2}$$

The logistic model that was employed in this work considers the following framework: Independent variables X1, X2, and so on, up to Xk have been observed on many students for whom elective grades have been determined as either 1 (if "A" or "B") or 0 (if less than "B"). Using this information, the proposed model will attempt to describe the probability that other students will do well (score an "A" or a "B") or not (score less than a "B") in a particular elective course at a particular time having measured the independent variable values X1, X2, up to Xk. The probability being modelled can be denoted by the conditional probability statement in Eq. 3.

$$P(X) = P(D = 1 \vee X_1, X_2, \ldots, X_k) \tag{3}$$

To illustrate how the logistic model was used in this work, we will assume that D is the elective course to be predicted and it is coded 1 if the grade is an "A" or a "B" and it is coded 0 if the grade is less than a "B". For this example, we will consider just three independent variables C1, C2 and C3, each holding continuous values between 0 and 100 that is the student's score in that particular course. Here, the logistic model will be defined as in Eq. 4.

$$P(X) = \frac{1}{1 + e^{-(\alpha + \beta_1 C1 + \beta_2 C2 + \beta_3 C3)}} \tag{4}$$

The proposed system attempts to use the data gathered to approximately guess the unknown factors α, β_1, β_2, and β_3.

Logistic regression is very suitable for predicting course outcomes as it is a problem that satisfies some assumptions. One is that the variable to be predicted must follow a binomial distribution [21] and that is the case with a two-class classification problem like this. Also, each outcome must be statistically independent and that is the case with student grades as one student's grade in a course does not affect another student's grade in the same course. Likewise, in educational data mining, logistic regression has been proven to provide more accurate results than some other algorithms [22].

Furthermore, logistic regression is known to perform well even when there is not so much data being used in training [23] and this is important for this study because there are many elective course options for students, thus limiting the number of students who could apply for each course. Besides, logistic regression models tend to suffer in situations where the decision boundary between classes is multiple or non-linear but this isn't such a problem in predicting course outcomes as the decision boundary tends to be quite clear. Lastly, but probably most importantly, logistic regression models can easily be brought up-to-date with fresh data as it becomes available [24].

3.5 Wide and Deep Recommender

It has been noted that while generalized linear models can learn associations between wide independent and dependent variables, they generally struggle to take a broad view of features without adequate feature engineering; this is not a problem for deep neural networks as they can generalize more easily but sometimes, too easily when there isn't much interaction between those making the recommendations and the items being recommended, and this leads to poor recommendations [23]. To solve this problem, Cheng et al. [25] proposed Wide and Deep Learning, that is, a recommender that combines wide linear models (beneficial for remembering associations) with deep neural networks (high generalization ability).

The wide and deep recommender is a combination of two separate approaches. The wide element of the recommender is a summarized linear model of the form $y = w^T x + b$ where y is the dependent variable, $x = [x_1, x_2, \ldots, x_d]$ is a vector with d independent variables, $w = [w_1, w_2, \ldots, w_d]$ are the model constraints and b is the bias. On the other hand, the deep element of the recommender is a feed-forward artificial neural network whose categorical features are initially transformed into low-dimensional and dense real-valued vectors (known as embedding vectors) with a dimensionality between $O(10)$ and $O(100)$. These embedding vectors are initially primed arbitrarily after which they are trained to diminish the final loss function while the model is being trained. Subsequently, the embedding vectors are passed into the hidden layers of a neural network in the forward pass, with each hidden layer of the neural network performing the computation in Eq. 5.

$$a^{(l+1)} = f\left(W^{(l)} a^{(l)} + b^{(l)}\right) \tag{5}$$

with l indicating the layer number and f being the activation function (usually rectified linear units – ReLUs); also, a(l), b(l), and W(l) will be the activation, bias and, weights at the l-th layer.

When training a wide and deep recommender, each element, that is, the wide and deep elements, are merged using a weighted sum of their output log odds as the model projection and this is then passed into a shared logistic loss function for both elements to be trained simultaneously.

Three sets of data can be passed into the wide and deep model and each of them is explained below:

User-Item-Ratings: This dataset contains three columns only which are identifiers for those making recommendations (in this case, students), the item which is recommended (in this case, elective courses), and a numerical rating is given to the recommended item, with a higher rating indicating that an item comes highly recommended while a lower rating indicates that the item comes unfavourably recommended. In this case, ratings were given to courses based on the grade scored by the student in that course; that is, students who scored an "A" in a particular course gave that course a rating of 5, students who scored a "B" in a particular course gave that course a rating of 4, students who scored a "C" gave the course a rating of 3, and so on. This dataset is the only dataset required for the wide and deep recommender to run.

User Features: This dataset contains information about individual users who have provided recommendations. The first column of this dataset (student identifier) must match the identifiers provided in the user-item-ratings dataset. In this study, scores in the required and core courses taken in the student's first three years of study were used to populate the user features dataset. This dataset is not required for the model to provide recommendations but it would lead to better recommendations when provided.

Item Features: This dataset contains information about the potential courses that could be recommended to users. It is also not required for the model to provided recommendations and was not used in the study as this could not gather adequate information to necessitate the use of this particular dataset.

The wide and deep model has been proven to be more effective in making recommendations than wide-only and deep-only models [25], thus making it a good choice for this system. Users would supply their grades in required courses as input to the model while the model would produce a set of recommendations as output.

3.6 Performance Evaluation Metrics

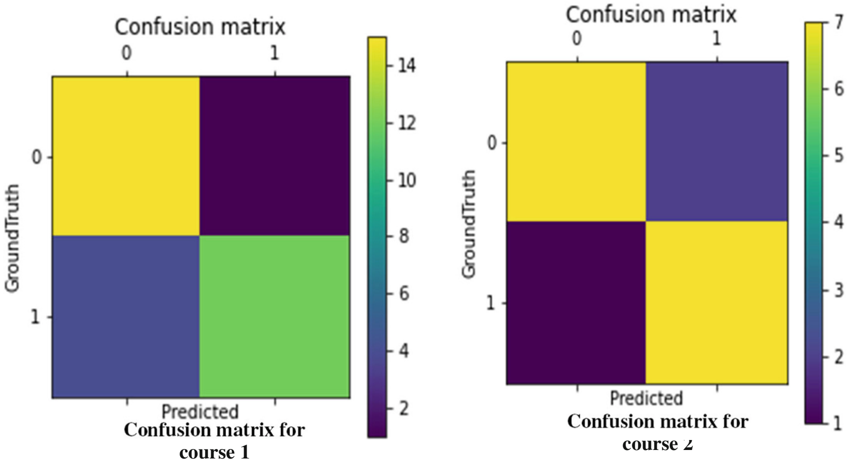
A number of metrics were used to determine the quality of the trained models. These metrics are described in the table below (Table 1):

Table 1. Performance evaluation metrics

Metric	Description	Formula
Accuracy	Accuracy measures the correctness of the classification	$\frac{TN+TP}{The\ total\ number\ of\ test\ items}$
Precision	Precision measures the exactness of the classification. It describes the ability of a classification model to return only relevant instances	$\frac{TP}{TP+FP}$
Recall	Recall measures the completeness of classification. It describes the ability of a classification model to identify all relevant instances	$\frac{TP}{TP+FN}$
F1-Measure	F1-measure is a combination of precision and recall. The quality of the model in terms of both precision and recall is combined into a single score, calculated as the standard harmonic mean of precision and recall	$\frac{2*precision*recall}{precision+recall}$
Receiver operating characteristic curve – Area under curve (ROC-AUC)	ROC curve is a graph showing the performance of a classification model at all classification thresholds. AUC measures the entire two-dimensional area underneath the entire ROC curve	
Mean squared error (MSE)	It measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value	$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i-\widehat{Y}_i\right)^2$
Root-Mean-Square error (RMSE)	It represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences	$\sqrt{\frac{\sum_{i=1}^{n}\left(\widehat{Y}_i-Y_i\right)^2}{n}}$

4 Results and Discussion

Based on the metrics in Table 2, it can be noted that the models performed effectively in predicting passes in the electives considered and in recommending courses to students. The models also performed better than other models evaluated in Sect. 2 other than the k-nearest neighbour algorithm used in Ogunde and Ajibade [13]. Figure 6 displays the confusion matrix of the proposed classification model.



Confusion matrix for course 1

Confusion matrix for course 2

Fig. 6. Confusion matrix of the proposed classification model

Table 2. The performance metrics for the proposed system

	Accuracy	Precision	Recall	F_1-Measure	ROC AUC	MSE	RMSE
Course 1	0.84	0.92	0.75	0.83	0.84	0.48	1.63
Course 2	0.76	0.75	0.75	0.75	0.83		

The perfect precision score is 1.0. The first course predicted was found to have a precision score of 0.92 while the second course had a precision score of 0.75. The first course predicted was found to have a recall score of 0.75 and the second course had a recall score of 0.75. The first course predicted was found to have an F_1 score of 0.83 and the second course had an F_1 score of 0.75. The constructed model was seen to have a Mean Squared Error value of 0.48. RMSE is always non-negative and a value of zero shows that the model perfectly suits the data; thus, the lower the value, the better the model. The constructed model was seen to have a Root-Mean-Square Error value of 1.63. The first course predicted was found to have an accuracy score of 0.84 while the second course had an accuracy score of 0.76.

5 Conclusion

Elective course selection tends to be a problem for many undergraduate students in tertiary institutions. Some reasons for this are the large number of possible electives to be selected, inadequate information about elective courses before selection, insufficient time for level advisers to gather course information, etc. While some recommender systems have been put in place to solve such problems, they tend to struggle because of inadequate training data, imbalances in training data, changes in students' learning styles over time, etc. This study is meant to provide a better method of creating a course recommendation system for students in undergraduate degree programmes based on collected historical records of past grades by collecting and pre-processing data of past students from the school system, selecting good features to be used in the machine learning models, implementing and training the recommendation models, and evaluating the trained models. This study was carried out using the logistic regression statistical model for elective course pass prediction and the wide and deep recommendation model for elective course recommendation. This, being a supervised machine learning problem, was done by first collecting the data of past students from the Department of Computer Science, University of Ilorin, performing data pre-processing operations on the collected data (mainly data cleaning and data balancing), feature selection (which involves selecting the best independent variables to be used in the models), division of the data into training datasets and test datasets, training and optimizing the models, and finally, designing the web application to be used to access the models.

References

1. Abiodun, M.K., et al.: Cloud and Big Data: A Mutual Benefit for Organization Development. J. Phys. Conf. Ser. **1767**(1), 012020 (2021)
2. Crestani, F.: Application of spreading activation techniques in information retrieval. Artif. Intell. Rev. **11**(6), 453–482 (1997)
3. Zabriskie, C., Yang, J., DeVore, S., Stewart, J.: Using machine learning to predict physics course outcomes. Phys. Rev. Phys. Educ. Res. **15**(2), 020120 (2019). https://doi.org/10.1103/PhysRevPhysEducRes.15.020120
4. Praserttitipong, D., Srisujjalertwaja, W.: Elective course recommendation model for higher education program. Songklanakarin J. Sci. Technol. **40**(6), 1232–1239 (2018). https://doi.org/10.14456/sjst-psu.2018.151
5. Thanh-Nhan, H.L., Nguyen, H.H., Thai-Nghe, N.:. Methods for building course recommendation systems. In: 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE), November 2017, pp. 163–168 (2016). https://doi.org/10.1109/KSE.2016.7758047
6. Jiang, W., Pardos, Z.A., Wei, Q.: Goal-based course recommendation. In: Proceedings of the 9th International Conference on Learning Analytics & Knowledge - LAK19, March, pp. 36–45 (2019). https://doi.org/10.1145/3303772.3303814
7. Kunaver, M., Požrl, T.: Diversity in recommender systems–a survey. Knowl.-Based Syst. **123**, 154–162 (2017)
8. Awotunde, J.B., Adeniyi, A.E., Ogundokun, R.O., Ajamu, G.J., Adebayo, P.O.: MIoT-based big data analytics architecture, opportunities and challenges for enhanced telemedicine systems. Stud. Fuzziness Soft Comput. **2021**(410), 199–220 (2021)

9. Chau, V.T.N., Phung, N.H.: Imbalanced educational data classification: An effective approach with resampling and random forest. In: Proceedings - 2013 RIVF International Conference on Computing and Communication Technologies: Research, Innovation, and Vision for Future, RIVF 2013, November 2013, pp. 135–140 (2013). https://doi.org/10.1109/RIVF.2013.6719882
10. Gurpinar, E., Bati, H., Tetik, C.: Learning styles of medical students change in relation to time. Am. J. Physiol. Adv. Phys. Educ. **35**(3), 307–311 (2011). https://doi.org/10.1152/advan.00047.2011
11. Morsy, S., Karypis, G.: Will this course increase or decrease your gpa? towards grade-aware course recommendation. J. Educ. Data Min. **11**(2), (2019). http://arxiv.org/abs/1904.11798
12. Dahdouh, K., Dakkak, A., Oughdir, L., Ibriz, A.: Large-scale e-learning recommender system based on spark and hadoop. J. Big Data **6**(1), 1–23 (2019). https://doi.org/10.1186/s40537-019-0169-4
13. Ogunde, A.O., Ajibade, E.: A K-nearest neighbour algorithm-based recommender system for the dynamic selection of elective undergraduate courses. Int. J. Data Sci. Anal. **5**(6), 128–135 (2019). https://doi.org/10.11648/j.ijdsa.20190506.14
14. Feghali, T., Zbib, I., Hallal, S.: A web-based decision support tool for academic advising. Educ. Technol. Soc. **14**(1), 82–94 (2011)
15. Ling, G., Yang, H., Lyu, M.R., King, I.: Response aware model-based collaborative filtering, uncertain. In: Artificial Intelligence - Proceedings of 28th Conference UAI 2012, pp. 501–510 (2012)
16. Marlin, B.M., Zemel, R.S.: Collaborative prediction and ranking with non-random missing data. In: Proceedings of 3rd ACM Conference Recommender Systems, RecSys 2009, pp. 5–12 (2009). https://doi.org/10.1145/1639714.1639717
17. J. M. Hernández-Lobato, N. Houlsby, and Z. Ghahramani, “Probabilistic matrix factorization with non-random missing data,” 31st Int. Conf. Mach. Learn. ICML 2014, vol. 4, pp. 3394–3436, 2014.
18. Bydžovská, H.: Course enrollment recommender system. In: Proceedings of 9th International Conference Educational Data Mining, EDM 2016, no.10, pp. 312–317 (2016)
19. Chen, C.M., Lee, H.M., Chen, Y.H.: Personalized e-learning system using Item response theory. Comput. Educ. **44**(3), 237–255 (2005). https://doi.org/10.1016/j.compedu.2004.01.006
20. Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L.: Recommender system for predicting student performance. Procedia Comput. Sci. **1**(2), 2811–2819 (2010). https://doi.org/10.1016/j.procs.2010.08.006
21. Kleinbaum, D.G., Klein, M.: Logistic Regression A Self-Learning Text. In: Survival (3rd ed.). Springer, New York (2010)
22. Hosmer, D., Lemeshow, S.: Applied Logistic Regression (Issue October). Wiley, Hoboken (2013). https://doi.org/10.1080/00401706.1992.10485291
23. Shaun, R., Baker, J. De, J. E.B., (eds.) Educational Data Mining 2008 The 1st International Conference on Educational Data Mining. Network, January, 187 (2008). http://gdac.uqam.ca/NEWGDAC/proceedingEDM2008.pdf#page=87
24. Sawarkar, N., Raghuwanshi, M.M., Singh, K.R.: Intelligent recommendation system for higher education. Int. J. Future Revolution Comput. Sci. Commun. Eng. **4**(4), 311–320 (2018)
25. Cheng, H.-T., et al.: Wide & Deep Learning for Recommender Systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS 2016, pp. 7–10 (2016)