Choosing a course from a large list of courses can prove to be a challenging job for a student. In order to solve this problem the proposed approach uses a recommendation system which plays a significant role in suggesting the best suitable course to the student according to his/ her personal learning ability. The ideology is that the system recommends the courses based on the similarities of the students who want the recommendation with the students who have already taken and rated the courses earlier, rather than basing the recommendations purely on a knowledge base. In the case of new students prior suggesting a course the approach uses some questionnaire and a set of parameters for developing a knowledge base of these students so that the recommendation can be optimized. The initial approach used is collaborative based, where the user item matrix is considered for getting the score of the user and his similar users. The proposed recommender system performs better by mitigating the weakness of basic individual recommender systems along with improving the classifiers used in real-time. This is primarily done by analyzing each user currently using the system. **Keywords: Course recommendation system, Collaborative, User-Item Matrix.**

The course recommender system would suggest best suitable courses to the new students after making a knowledge base and finding a similar user to them. This recommendation should be based on the attributes given by the student and should give the best suitable course. The system should also be able to do semantic analysis of the user’s reviews [4].The system considers a set of courses from the dataset as the domain of recommendation. Once a class of similar users is found, their recommendation should be shown to the new user .This approach is unique as for considering the similar users the feature set [4] contains the user attributes along with course reviews given by the existing users. The reviews help in determining if a user liked or disliked the course. Once the review is analyzed using the proposed sentiment analysis algorithm, a rating indicating this like/dislike factor for the existing user can be determined .The courses for which the similar users give a like rating will be chosen for recommendation. The proposed system should have a User Interface for the new user, where he or she can enter his or her details and search for the course needed and should also contain a User Interface which would show the courses recommended. The proposed system has difficulty in finding similar users as the existing user’s data needs to be updated, and the clusters or the classes have to be measured with less abstraction than current implementation.

The purpose of the proposed approach is to recommend the user the best course from a large number of courses available on different sites on the same field of interest. It helps the user to save their time by avoiding to search about each course over the internet.. The proposed system maintains a dataset of users and their reviews on different courses on various sites.

The proposed system analyzes the best course taken by a similar group of users on the basis of their reviews on the courses [1].

The new users are dealt with by making a knowledge base about them.The proposed approach then suggests the course based on reviews by the similar group of users.

In order to understand the structure of the proposed recommendation a dataflow diagram is presented in Figure 1. It depicts the contextual level DFD which shows recommendation system flow of data and emphasizes the way it interacts with external entities. It only contains one process that generalizes the function of the entire system in relationship to external entities.

The rest of the paper is organized as follows: section II presents the motivation of the proposed approach. Section III and IV details related work and the proposed framework of the recommendation system respectively. Section V shows experimental result and analysis. Section VI and Section VII presents the analytical discussion and threats to validity. Section VIII presents the conclusion and future work.

The proposed approach suggests an optimum course for each student separately. This would give each user (student) the flexibility to try new courses, which he/she would be able to complete within the stipulated time and would help the user to gain knowledge using these courses. The user will be able to access the course with confidence without conceiving whether the suggested course would be too easy or too difficult for him/her to understand. This system would also save the time of user which could otherwise have been lost due to sheer number of online courses available on the internet.

Recommendations in the field of education has been a phenomena in the last decade . The volume of e learning content has grown exponentially and so has the parallel research for efficiently learning new concepts and subjects. In the last years a number of works have focused on the use of data mining and machine learning techniques in the context of educational environment for instance. Educationally Adaptive HyperMedia (EAH)[9]. These techniques are used by online courses to discover patterns which can be helpful in making the courses better for professors and students.

In the literature, research has also been done on recommendation models based on the feedback data gathered from student’s assessment of courses [9] relevant to their career goals. Along with this many authors have also researched in the field of encouraging users’ contribution [10] in getting recommendation for the choice they might take. Many authors have also analyzed the application of data mining techniques to recommender System. Some of which have been dealing with a reward based system so as to increase a user’s contribution[10] .Others have been based on collaborative and hybrid techniques of classification .

In the proposed approach the task of finding similar users is done using cosine similarity distance and applying the hierarchal clustering model [5]. This approach has shown to be of greater accuracy than a knowledge base approach where mean of values are taken without considering the context of courses the users have done earlier. The dataset of users and the courses they have taken has led to the requirement of sentiment analysis and hence the proposed approach is specific for the collaborative recommender system use-case.

Course recommendation is a special project taking into consideration the problems a student goes through in his/her daily life ,it can be used by any student with a keen interest to learn and grow.

The course recommender is one of the most powerful tools a student can get his hands on, it is one of the tools that is a game changer in terms of being effective and providing suggestions to the student. Course Recommendation System is necessary as many e learning courses are available online and the proposed system would be beneficial for the students looking to gain knowledge in optimised manner. A student can learn new things from multiple websites like Udemy, Coursera etc. Where one can search the course they want that’s fine but how to know which one is the best they don’t know every website that is there they have not gone through all the courses present at that website. Ratings can be deceiving and so many courses have so many ratings [5] which among them is for you so the proposed project has been created. This project provides the one with the best course for them in the whole of the internet which will give you the best result there is.

The proposed algorithm also considers how well versed the user is with the subject and also keeps in consideration their older courses taken to recommend the best course there because being a smart worker is always better than being a hard worker. The recommended courses can be paid or free. The courses considered for the proposed approach are mostly free to audit courses.

With reference to Figure 1, in Figure 2, shows the break-up of the single process into many sub processes which are part of the recommender system and how they transform and move through these sub processes. Figure 2 is a conceptual view of the proposed approach and it includes a user database where the existing user attributes and the courses they have taken along with the reviews for the course have been kept.

The second database is course database which will be used for training the sentiment analysis model for getting the sentiment of each reviewed course. The ‘API’ is an interface where the routes for getting recommendations and for updating course reviews will be stored and the front end application will be able to make call to the backend. The admin works on the training of the sentiment model and optimizing it and deploys the code again for recommendation and review classification.

In Figure 3, the design of the sentiment training classifier is elaborated with the help of a flowchart. The flow of process for training the classifier to rate the courses is based on the user’s intent and reviews. Firstly, the dataset for course review with name of each course and their review is included into the program [8]. Once the set of review has been added the next step is to pre-process the dataset and to remove any anomaly present in the data that could cause the system to give inaccurate ratings to the courses. Such as the language of text, null values etc.

The next step is to make feature set from given dataset to get the best features so that each review can be correlated to certain words that are important words and that define the intent of the user. Once the feature set of the data has been made next step is to select the classifier to get the optimum result so here in the intended project Naive Bayes binary classifier has been used for initial purpose so as to rate the data as 1 for bad and 5 for good. NLP library has the Naive Bayes function both multinomial and binary function [6] included so the project used the inbuilt Naïve Bayes function of *‘nltk’* as the classifier. Once the classifier has been chosen the next step is to divide the data into testing data and training data which can be done with the help of cross validations train test split [5]. For the proposed approach the test set taken is divided based on the 4:1 ratio.

Once the training of the data has been done the classifier was adjusted to achieve and optimum accuracy with Naive Bayes and approximately 70% was the result the algorithm of Naive Bayes was able to achieve in the specific case of this project. The intention of the above process is to include as many reviews so that if a user gives a new review it can be labeled and the classifier could be adjusted further to give adjusted score to each course for getting the best courses and the worst courses based on the selected features of the course review given by the current user [3].

1. **Working Methodology**

Firstly as the label for each course is available any user needs to login to the system to access his/her data. Once a user registers by answering certain important questions which help the system form a knowledge base for initial recommendation to the new user he/ she can access the recommended courses [1]. Now coming to the backend part it functions based on the collaborative model of recommendation system .Where similar users receive similar recommendations. To achieve the proposed model a user item matrix is made using the data from the previously acquired user data where each user has given review for courses they have done or taken up earlier.

Certain key factors which influence the recommendation such as the length of course needed and the year of study of the student have been taken as the crucial point for making the recommendation [2]. Now the next step is to give a structure or score to each of such user who has given review which has been done by employing the method of average score[2] by giving each feature a score and taking average of all such features.

Once the score is calculated a group or cluster of user has been made by help of K-means clustering [2] where ‘k’ has been taken as the number of courses considered for the purpose of training the score classifier of linear regression. A new user can be made to login and his/her score can be calculated based on the points or features question and recommendations can be considered based on the closeness to the mean points. K means is as iterative algorithm which intends to group datasets based on their closeness to one other or similarity to one another. The recommendation user similarity so they are in true nature with the collaborative model and have been optimized by suggesting the top 5 courses or recommendations to ease the search.

1. **Functional Requirements and Settings in Training and Testing Phase**

In this section, the functional requirements and settings in training and testing phase have been elaborated. The official definition of a functional requirement is that it essentially specifies something the system should do. Typically, functional requirements will specify a behavior or function of the system under the consideration.

* It should ask relevant questions from the user.
* It should be able to take input as the answers provided by the user.
* It should display the list of recommended courses.

Hardware requirements are the requirements needed by the external system using its hardware components so that it is able to run the desired applications within proper response time and with better performance. The implementation is conducted on a Pentium IV processor with 2GB RAM and 8 GB storage. The software requirements are the different software’s needed by the system to run the application properly. The application is made using Python IDE (PyCharm) NLTK toolkit with Numpy.

The data set was extracted through a web scrapper based on node which is a JavaScript based runtime environment build on chromes v8 JavaScript engine using packages like express which is fast, un-opinionated, minimalist web framework for Node.js.

The objective was to make sure that test set met the following criteria it was supposed to be large enough to yield statistically meaningful results and is representative of the data set as a whole getting a dataset with similar characteristics to that of the training set. As our set meets the preceding conditions our goal was to create a model that generalizes well to the new data. The results were not getting high accuracy that might have indicated that the test data leaked into the training set. It was inadvertently trained on some test data as a result we measured accurately about how well our model generalizes to new data.

Here in the considered project model cross validation train/test split uses a 1:4 ratio for **test : train** data where 20 % data is used for testing and to avoid over-fitting the training data is taken to be as random as possible by shuffling the dataset.

1. **Experimental Result and Analysis**

In this section, the detailed experimental results with analysis are presented. The experimental results are validated using python notebook functions (as given in the screenshots). The design of this section is further divided into five subsections (each of which are explained in detail with the results). For the readers, Figure 5 clearly depicts these five subsections.

The five subsections depicted in Figure 5, are the five steps for constructing recommendation system. The proposed system uses *jupyter notebook* for implementation of each module and the pre-requisites mentioned above require *nltk.download()* function to download the various libraries. The loading of dataset into python is helped by the use of *pandas* [5]. *Numpy library* is used for ease of processing. The loaded datasets has been shown in Figure 6. Where the reviews have a serial no and the text, and the course id . ‘*nltk’* library has been used to deal with the pre-processing.

Once the data has been loaded the next step involves the process of preprocessing the data. This step includes selecting only the English reviews which has been done with the help of enchant library where the first word of a set of 10000 reviews has been selected and checked using the enchant if it was not in English then its index was stored in a array and the set was subtracted from 10000 to get new indexes which contained only English reviews [5]. This preprocessing of data can be seen in the Figure 7.

The motive behind the use of *enchant function* rather than using the *‘nltk’* inbuilt library function is to make a separate list of English words only and to create a new table of such reviews instead of removing them from current table. This helps in effective computation of the text. Code snippet for the same is shown in the Figure 8.

Words for reference is the array which has all the first words which are checked using enchant for getting the index of non-English words and to remove them from the total sets. This has been shown in Figure 9 and 10.

In this phase, the non-English indexed reviews have been removed from all reviews and new array of indices to keep have been constructed [7]. For easy reference, this is shown in figure 11.

Once the pre-processing of data is done, the next step involves the selection of features. In order to select the features from reviews the most frequent 3000 words form the set of review words were selected as features. Further, to select most common words *‘nltk’* inbuilt library function most\_common() is used and pandas inbuilt function unique() [3] is used. This can be seen in Figure 12.

The above explained function gets the names of the courses for which reviews have already been considered after pre-processing (please refer Figure 13).

Tokenizing each sentence and to get words from each sentence, the above function *get\_words\_from\_sentence* have been used which gets a review as an argument and divides it into words and then stores them into words python list (please refer Figure 14).

Once the words have been tokenized they have to be related to a label for the purpose of training to make a classifier which understands to relate each word to a good or bad review context (please refer Figure 15).

In order to optimize the process of training, all the unnecessary words are removed by using the *nltk.stopwords* which includes a list of all unnecessary words that can be removed from the text (please refer Figure 16 and 17).

This gives the list of most common words which occur in the selected review and also gives an insight as to which words to remove further (please refer Figure 18). Selecting 3000 words as the feature set where if any word one of these if present or not present can help identify the sentiment (please refer Figure 19).

This function makes a 2D list or a key value dictionary where, if a word is present in the current review it gets true for the corresponding word from the feature set of words, else false (please refer Figure 20). The **complexity** of proposed algorithm is **O(n\*m)** where ‘*n’* is the size of review and ‘*m’* is the number of feature set words (which in the considered case is 3000).

Once the feature section is done, the training process is initiated where Naïve Bayes have been used from the *‘nltk library’* so that the reviews get classified within the category of 1 or 5. The cross validation is used to divide the reviews into two sets where, 80 % data is assigned for the training and 20 % for testing. This has been particularly done so as to avoid overfitting of data.

The input for prediction is a feature set where the tuple is true for each word of the features present in the current review and false otherwise. The classifier gives an accuracy score of 71% and the most informative features have been shown in Figure 21.

In the proposed algorithm, the classifier used is the *nltk’s* inbuilt classifier Naive Bayes . This is a probabilistic classifier which works on the above made feature set (please refer Figure 22).

Figure 23, shows words that are most informative words which help differentiate a review from a good or bad based on the trained classifier.

The above code in Figure 24 can be used to save the trained classifier for future purposes and to save time for further use. This includes saving the pickle file in write mode and then to open it in the read mode to get the saved classifier and use it.

1. **User item matrix**

After the reviews have been classified the next step is to get the user item matrix so as to create a novel model of collaborative recommendation. The user item matrix is a 2D matrix where the users are the reviewers and item, in the proposed algorithm, are the courses which the users have taken. This matrix helps classify a similar user to them and the process of recommendation gets much simpler. This can be seen in the Figure 25, 26 and 27.

In order to get the train the classifier to accept numerical data, the string data values are converted into numerical values.

The score for each user is calculated so as to get a basis for clustering for those users who have given the reviews.

Once the user item matrix had been made the next step proposed was to make a set of scoring mechanism for the users but the purpose involved in this case was solved temporarily using the method of averages. The proposed system intended to use k-means clustering for users to make a group and the use K-NN for classifying new users [3]. The knowledge base for the new users (cold start) is developed by asking a set of question from the new users as shown in Figure 28.

The result of the recommendation model is the list of suggested courses for the aforementioned schema of the things as shown in Figure 29.

The end result of recommendation is summarized in Figure 29 which depicts the input parameters taken from the user and the suggested course as recommendation in array format .

The sentiment analysis model proposed for this approach uses Gaussian Naive Bayes classifier which is initially a binary classifier and in the study of the proposed approach a multinomial Naive Bayes is seen to have greater impact as binary model just classifies the review as good/bad but a multinomial classifier will be able to classify the review on the scale of 1- 5. The classifier accuracy is 70 percent based on testing data taken from the course dataset. This approach justifies the use of Naive Bayes classifier as it has been seen to have a high accuracy for text based classification and sentiment analysis [5].

The task of finding similar users, defining features which had greater impact than others include difficulty level of the course, length of course etc. The initial algorithm using the mentioned features was trained on Linear Regression model. The further study based on the distance parameters led to cosine similarity with threshold of 0.

The hierarchal clustering model applied uses similarity method as cosine and for the different values of threshold the number of clusters also varied. The optimum value floats in the range of 0.005 to 0.001, as shown in Table 1, which show the comparison of threshold versus classes formed. The Cosine Similarity distance and Jaccard similarity score were considered appropriate for the proposed approach as these methods show less ambiguity while dealing with the features, especially Cosine similarity at 0.004 threshold, shows an inflexion point while applying it at clustering metric.

The extent to which the proposed approach can expand define its limits. It might expand its reach by even suggesting YouTube channels that people should subscribe to or books to read or newsletters or blogs to look forward to and subscribe to. It can even suggest independent teachers who are teaching on web that are not yet included to the database and can even add local education centers.

1. **Maturation**

The scope of the proposed approach is to extend its reach to people through website and an application to support. This will help the people to have an easy access to most important and relevant programs.

1. **System Failure**

The system can't provide accurate results for the courses for which user have given no input about, making the system ineffective and bias against systems it is trained on. Users need to input their recommendations that might not be correct as they may not have knowledge of better courses present. Some course owners can try to make their course better by creating multiple users who recommend their course .These system failure scenarios may cause the system to give inaccurate results and hence must be dealt with.

**VIII. Conclusion and Future Work**

The main use of the intended project has been achieved to a certain initial degree and this will be helpful in a very strategic manner for students to get the best courses suitable for them and also it will save the time of the users. It is working with the review classifier having accuracy with Naïve Bayes of 71% and the user classifier has accuracy with the basic linear model of 60%. This here shows that many recommendation are being considered by the end users while the probability of a new course being added to their recommendation is also high the system will get better as the database of the users and reviews gains a certain depth in it. This intended project has been developed as a collective effort of the team involved in the research and its first users have also been able to give the recommendation to make the classifier better such that the end product works within certain accuracy aforementioned in the article.

The classification is performed using regression and K-means algorithm. It can be further optimized by using string matching algorithms and giving priority based scores to each feature of the user. Further the accuracy of recommendations can be improved by including content based filtering, which in turns leads to a hybrid recommendation system. The data need to be extracted from various MOOC websites and will compare courses from various courses available on other websites too.