# **The development of a recommender system for students for course registration on Course registration portal**

**Research Question**

For the university students will an across departments course recommender system based on user query on course registration portal help them taking informed decision in selection of subjects, considering the large scale of choices available.

**Introduction:**

The area that deals with providing recommendations to a user using predictive modeling has been widely used in a lot of applications from online retail stores to restaurant menu suggestions. The machine learning techniques like topic modelling, clustering provide effective ways which can be used by the researchers to work on the probable choice that a user might opt in future and hence make recommendations. In this way it is beneficial for the user by selecting the options directly from the recommendations rather than wasting unnecessary time and energy in browsing through all the choices and then reaching on to the desired one.

The same logic can be used to benefit students as well when they are struggling online to register for the courses on their university portals. There are hundreds of courses in different departments and for a student to know all the courses by browsing and spending time over each course description can be time consuming and frustrating. The registration portals typically give the flexibility to choose a particular course on the basis of semester(fall/spring/summer), course name, course number (course code) and instructor name. The portals generally lack the flexibility to equip the user with a support of providing potential courses that a student might want to register which can take off some of the burden from student’s perspective to remember different courses to some extent. The situation can be handled in a better way if much attention is given to the online registration portals to perform research that can provide pertinent suggestion to students using appropriate predictive modeling for course registration. The proposed system will equip the student to insert the course name as a query and get top 10 potential courses closely matching the query by applying appropriate natural language processing techniques in the background. Thus helping the student in making an informed decision while registering of the courses.

**Literature Review:**

***Recommender systems using different techniques in their methodology***

Different techniques like clustering, topic modelling etc. can be employed for the purpose of finding similar items or the items which might fall in same category, due to some of their characteristics. These techniques can serve as an initial step for the similarity functions which provide the final output of interest i.e. the relevant items that can be used for recommender systems. Al-Badarenah et al. [1] proposed the recommendation system using K means clustering algorithm [2] to find the group of students with same kind of track record of grades. The system used *n-nearest neighbor* as their similarity function to decide to which group a new student in question may fall. Consequently, the association rules were developed to predict the grades of the students in the recommended subject. Association rules are way to find the correlations between the items based on their occurrence together in a transactional dataset [1] However, the proposed system tend to promote the students to select only those subjects where chances of getting better grades are higher. The dataset used for the study consisted the records of sample students out of which one part was used as training set and rest were used as testing set. To evaluate the performance precision and recall curves were used.

Similarly, Grewal et al. [4] presented the recommender system employing the K-means clustering algorithm to cluster students based on the grades of five courses and their interests. The system used student’s interests as an added characteristic as opposed to Al-Badarenah et al. who used only the grades as the basis for clustering. Grewal et al. used the association rules in similar fashion to find the links between the subgroups and to reach to a conclusive suggestion [7, 8]. The dataset was created efficiently with the help of students providing their information on a form containing a questionnaire, however the evaluation of the system was based on the student’s feedback and the rating of satisfaction given by students. However, the evaluation entirely based on student’s feedback may not be able to provide a robust way of assessment because the bias in sample population was not considered in the study and the experiment was not controlled as the response from the population without recommender system was not compared with the population experiencing recommender system.

The common techniques discussed above like clustering, association rules used in recommender system present a quick way to get the recommendations for users. There have been attempts to improve the quality of recommendation system by deploying added filtration before recommendation. Sunita et al. [11] proposed a system where the ADTree Classification algorithm was applied on the clusters after clustering was performed. Sunita et al. assert that applying association rule algorithms like apriori on the classified dataset helps getting rules with increased strength of association rather than obtaining the rules directly through the association rule algorithms on the entire dataset.

***Recommender systems following different concepts to apply above mentioned techniques***

***Content based recommender system***

Different recommendation algorithms are used by recommender systems to provide the personalized recommendations. Memory based collaborative recommendation technique where the entire dataset is used to make the prediction fails to meet the current need of sparsity of the data and the scalability. In order to meet the requirement of tremendous growth in the data and the user, Badrul S. et al. [3] state different item - based recommendation generation algorithms. The item - based recommendation algorithm prove to be better approach since it focuses on computing similarity between the items and then selecting the most similar items. Different techniques like item -item correlation, cosine similarities between item vectors are used in order to find the similarity. Techniques like weighted sum and regression model are used for the recommendation. The author experimentally compared the result to the basic k-nearest neighbor approach which is the most widely accepted technique used in the recommender system. The results suggest that the item - based algorithm provide better performance than the traditional user based algorithms and also provide better quality.

***Context aware recommender system***

But the collaborative filtering and item/content based recommender system uses simple models. These approaches considers user ratings as a vector and considers the user’s preference for the recommendation and hence ignores the importance of the context which plays an important role in user’s preference. A user’s choice will differ in different context and hence recommending choices without considering the context may not prove beneficial for both the users and the business. Context - independent representation loses its prediction ability. Thus Adomavicius G. et al. [2] proposes context based recommender system which is build on the basis of the knowledge of partial contextual user preferences. The author suggests the use of contextual prefiltering where the current context selects only relevant set of the data and then the traditional recommendation system is utilized for the prediction. While the contextual post filtering the first the 2D recommendation is applied to predict the rating on the full set of data and ignoring the context. Then the recommendations are finally adjusted by applying the contextual information.These contextual filtering approach could be applied separately or in combination for efficient recommendation. Thus contextual based filtering not only considers the criteria that which item was liked by the user but also the contextual information in which the particular item was used by the user.

***Context aware recommender system in multi-dimensional model***

Adomavicius G. [19] suggests a multiple dimension model to provide recommendation by considering the different dimension including the context. Traditional recommender system takes only two dimensions in account , users and items.Initially the ratings are implicitly referred by the system or explicitly provided by the users. Once the initial ratings are specified the rating function R is calculated for the new incoming item which has not been yet rated by the users. The rating function is mentioned as below :-

R : Users × Items → Ratings

Now the different approaches namely content-based, collaborative, and hybrid are employed to recommend the rating.These are discussed below :-

In Content Based recommender system the rating for an item i by the user u i.e. R(u,i) is estimated by the rating R by same user u on similar item i’ in terms of context.

In Collaborative Recommender Systems, for a particular customer rating of an item is predicted by many collaborative recommender systems on the basis of previous rating on the same item by other customers.

Hybrid Recommender Systems, collaborative and Content methods can be combined into the hybrid approach in several different ways to produce recommendation.

In contrast the Multi Dimensional Model provides recommendation over several other dimensions like Time, Place, and so on along with user and item. The different dimensions can be thought of the different sides of the cube in 3D space. One important question here is to select which dimensions are relevant and should be kept in the multi-dimensional model. The question is related to the feature selection which has been addressed using the machine learning and data mining algorithms.Thus a traditional recommender system can provide recommendation of a particular type like “ top N items to a user” on the other hand Multi

Dimensional recommender system provides suggestions like “best N items to recommend for each user/time combination” or “best N times to recommend for each user/item combination,” and so on.

***Knowledge based recommender system***

Another approach of finding the recommendations is employing knowledge-based recommender system. Burke, Robin [12] proposed knowledge-based recommender system over content - based, collaborative- or social filtering type. Burke states that knowledge-based recommender system can avoid a lot of drawbacks in other recommender systems as it does not need a large dataset, also there is no dependence on an initial base user set. However there has not been much development in the area of knowledge-based recommender systems but such systems can be of great help where availability of dataset is not there.

***Recommendation system using topic modelling and Association rule mining***

***Topic modelling (latent dirichlet allocation LDA)***

With growing importance of recommendation in E-Commerce it has also find its place in the checking Application behaviour against its description.It has been a concern for developers to check whether an App behaves as it has been advertised or not. There are a large number of malware which does not provide functions what they claim to. Till date, the malicious behavior of any application was detected by comparing the application code with respect to a predefined malicious code. But in today's era, this technique will not be able to handle the new attacks as in one App the behavior can be beneficial while in another App it might be considered as malicious. Thus, Gorla, A. et al. [5], suggest the technique to check the behavior of the App against its implementation. The author has developed CHABADA approach which can be explained as below: -

First Android Apps has been collected from Google Play Store. Then Latent Dirichlet Allocation has been used in the App description for the topic modeling of each application and clustering to group similar topics followed by the identification of static API called in each App. Finally, unsupervised One - Class SVM has been utilized to detect any anomaly in terms of API usage signaling mismatch in the implementation and the description of the App. It was able to identify 56 percent of novel malware without requiring any prior training of existing malware pattern. This approach exceeds previous approaches in that it completely automatically learns association of topics with their corresponding API and does not require manual annotation like WHYPER framework of Pandita et al. [6]. The approach of Gorla A. et al. of performing LDA gives the topics like “weather”, “map” etc. and then putting the apps in a group that share several topic in their description gives a way to implement similar approach in using the description of courses to get different topics and then putting them together on the basis of percentage of topics they share.

Similar strategic approach has also been used in detecting the insider threats in hospitals that covers illegitimate access to patients' records either by masquerading or by synthetically prepared moves of a synthetic user. Siddharth Gupta et al. [9] proposes the use of Random Topic Access (RTA) model to get the anomalous behavior detection.

This technique mainly focuses on categorizing a set of topics under a particular article paper attempts to leverage this characteristic to signify users as articles and the content in their records as the set of topics associated with them. Here it can be noted that the authors emphasize that there approach is novel in a way from Random Object Access anomaly detection where illegitimate users are modeled as ones who access less randomly [9], that it applies RTA which enables them to identify the outlier even if a masquerader logs in imitating as a particular user and tries to access a record which is not categorized under the logged in user. Random Object Access would not be able to differentiate such intrusions. To denote topics or features the Random Topic Access Detection (RTAD) uses Latent Dirichlet Allocation (LDA) and clustering is used to separate out the anomalous behavior which is accomplished by the k-nearest neighbor (K-NN). The LDA is used over the hospital logs for all of the patients' access to the records and hence they are divided under the class of topics which is related to the diagnosis, medications, procedures, locations/services. The LDA results further enabled to create a class of derived categories into five class that ranges between two extreme categories i.e. strongly to weakly which consequently help in detecting the random topic access user by the calculation of Area Under curve from Receiver Operating Curve for each of the ve cases. The more class specific the user is the better performance was and the more class agnostic the user was the weaker the performance was.

***Association rule mining***

The collaborative filtering is employed as one of the major techniques in recommender systems, but the underlying instrument i.e. association rule mining that enables finding the similar item for recommendation is mostly used conventionally. Although Weiyang et al. [13] have tried to work on association rule mining specifically and instead of using conventional algorithms like Apriori they have proposed for a target user recommendation the association rule mining require modified version of rule mining techniques. Shardanand and Maes [14] and Resnick et al., [15] have also put forth variants that use the pearson correlation between the ratings of two users. The conventional rule mining uses the entire dataset which can lead to a lot of time consumption and can lead to either with a lot of rules or with a few rules. Weiyang et al. have proposed to mine rule only for one target user at a time and a range of desired number of rules specific to that user is provided before the mining process as opposed to using a minimum support. That is to say the for each user the algorithm select a certain number of rules on the basis of adjusted minimum support that the algorithm decides. This plays a huge role in saving a significant amount of processing time when the rule mining is applied to a subset of transaction in the dataset.

Several different types of Data Mining techniques are there for the implementation of Recommender Systems. The data needs to be first preprocessed using different methods like sampling or dimensionality reduction.Principal Component Analysis (PCA) is a statistical mechanism which helps in finding the patterns in high dimensionality data sets. It forms an ordered list of components on the basis of largest amount of the variance from the data in terms of least square errors: i. e. the first component accounts for the larger variance than the second component and so on. Hence we can neglect the components with small contributions in the variance in order to reduce the dimensionality. Classification techniques such as K-nearest neighbours, Rule - based classifiers, bayesian, decision trees, Support Vector Machine and clustering techniques like k means clustering have been widely used in the execution of Recommender System as suggested by Amatriain X. [24] et al.

***Recommender systems using additional controls***

Course recommender systems in majority of the literature lack an important quality i.e. taking the constraints like prerequisites, taking 2 out of 5 Math courses etc. into account for the students. Parameswaran et al. [22] proposed the recommender system that takes constraints into account. The system is a complementary system that presents a best possible match for a set like the traditional recommender system along with constraints considered while recommending the items. The max flow algorithm was applied like Ford-Fulkerson max-flow algorithm to obtain the minimum number of additional courses which required to be taken to satisfy all the conditions. All those courses have a score calculated and the score represents the utility or usefulness of of taking a course. The score can be used to rank the sets and the flow algorithm can assure the constraints to be met.

Feng et al. [23] have proposed a recommender system that considers the hidden features of the items while recommending to the users. The system not only used the rating given by the user which does not reflect any of the specific characteristics of the movie like the adventure picturized in the movie or some other characteristic of the movie, and these might be the hidden aspects that could have made the user loved the movie. The system was built to consider the explicit and implicit interactions of the user with the recommender system to get the hidden aspects of the users’ preference. Topic model was used to find the latent association of users’ preferences with respect to an item.

***Evaluation metrics of recommender systems***

Rashid et al. [16] have worked on different but an important aspect in the construction of a recommender system. The collection of new user’s preferences for movies is done in a precise manner that can serve as data for analysis used for recommendations. Important aspects required to build a recommender system like User Effort in signing up, user satisfaction with sign up process, accuracy of the recommender system as in the appropriateness of the recommended items and system utility which enables to find whether other systems can learn from the developed system. They have used popularity and entropy as a means of data collection. The entropy is calculated by using the relative frequency of five possible ratings which gives the information of all kinds of ratings, and the information that can help in decision making in recommendation system is the variance and entropy both used in combination [17]. Popularity was collected for the movie with the help of already available ratings for the movie and presenting them in a descending order i.e. popularity gives the probability of user rating the movie. In popularity based data enough amount of data was not collected for all the ratings as compared to entropy based data. Rashid et al. found out that entropy and popularity do not have much correlation.

It has been a very challenging task to identify the identifying the best algorithm for a given purpose, since the researchers do not agree on the attributes to be measured and the metrics to be considered for each attribute. As mentioned by Herlocker, J. L [21] evaluating recommender system and their algorithms is very tough task due to the different reasons mentioned below:- The performance of different algorithms depend on the datasets, it can perform better on one dataset while worse on the other. Many collaborative filtering algorithms have been designed to handle the dataset where number of users are more than the number of items and hence such algorithms will prove inefficient on the dataset where the number of items exceeds the number of users. Similar differences exist for ratings scale, ratings density, and other properties of data sets.

The second reason why evaluation is difficult is that the goals for the evaluation performed may differ. Earlier the evaluation mainly focused on “accuracy” of collaborative filtering in prediction but in today’s era the evaluation relies on other properties which have much more effect on the user satisfaction. Some of the properties are the serendipity of the recommendation plays a great role in this system. Also, the system should take the recommendations into account if they have been explored by the user in the past or not, these are few aspects which do not depend on accuracy of the system

Finally, another challenge in evaluation is that decision of combination of measures to be used in a comparative evaluation. In recent trend the researchers have succeeded in finding newest algorithm which yields relatively very less mean absolute error. Though these algorithms may be better in performance as compared to the older algorithms, but the author find out that when all these algorithms are tuned to their optimum then they produce almost similar results and hence algorithm can be improved in terms of several other factors rather than just improving the mean absolute error.

All the recommender systems get evaluated on rather common evaluation techniques like accuracy, precision and recall etc. McNee, S. M et al. [18] discussed about the common metric of evaluation of recommender systems as accuracy which can hurt the quality of recommender system even if the accuracy of the system is very high. Since the suggestion given by the recommender system take all the option into account there are high chances that a user sees the same recommendations again and again. Such recommender system will achieve higher accuracy even though practically they are not a quality recommender system. Rashid et al. [16] also expressed concerns about the same suggestion popping up over and over again in a recommender system with the example of star trek (once a user rated one Star Trek movie she would only receive recommendations for more Star Trek movies). Hence Sean et al. proposed some aspects which can be considered while looking for the quality of a recommender system. Serendipity is proposed as one aspect which can cause for the user to experience an unexpected and fortuitous item recommendation. From the user's perspective such item suggestion can make a recommendation system highly useful but may not score higher on the scale of accuracy. Similarly, user’s experience and expectations are another aspect which play a great role in the system’s quality. Many recommender systems with high accuracy are employed in diverse setting without taking into account as to which background and culture the recommender system was originally developed. Performance of such recommender system tend to disappoint users in a different background [20]. Finally the similarity aspect of it which is considered to be the key ingredient of the recommender system may actually spoil the recipe as the higher the similarity of the suggested items with the searched item higher the accuracy of the system but this ignores the fact that user may not be interested in exactly same type of other item with little or no differences hence a balance in similarity level is required. Hence if we look at the course recommender system then since we have a finite number of courses available it is possible that the most of the times the recommender system picks up the same recommendations again and again which can make the system futile.

**Methodology**

The recommendations given to a student for course selection requires data about all the subjects which can be recommended. The course description is one of the key aspects of the subjects which can help segregate subjects for recommendations.

**Dataset and cleaning**

For the dataset we will scrape the data from the course catalog of University of North Carolina at Chapel Hill (URL: <http://catalog.unc.edu/courses/>). We will use BeautifulSoup, a Python library which can be used to read HTML file and search relevant content in the file using find\_all functionality. BeautifulSoup helps create a dataset in a csv format out of a web page. The catalog provides the following information:

* Department abbreviation
* Course Number
* Course Name
* Number of credits
* Short course description
* Requisites
* Grading Status

All the information scraped does not prove to be a discriminating factor i.e. there are some fields which have the identical values for majority of the courses and therefore need to be removed while developing the model.

Among the fields mentioned above the Grading Status field will be dropped from scraping because Grading Status is not a distinctive feature for the courses. All the courses have same Grading Status with the exception of a few, for instance almost all the graduate courses in the department of School of information and Library Science have Grading Status as Letter Grade. Similarly, the number of credits ranges from a range of 0 to 4 for all the courses and cannot play a role in distinguishing what a course is about, so number of credits will be dropped. Furthermore, Course Number and Requisites also do not play a role in discriminating one course from another, on the contrary they can create ambiguity while topic modelling since different departments tend to have same course numbers for different courses. Hence the modelling will be performed with the help of three fields described above i.e. Department abbreviation, Course Name and Short Course Description. These three fields play major role in course distinction. The Course Description is most crucial as it elucidates different characteristics of the course, the Department abbreviation helps in distinguishing the department of the course and the course name separates itself from other courses as all the courses have distinct names.

Using above three fields we will create three datasets.

Dataset 1 (Training dataset):

The first dataset will be created using three departments namely, School of Information and Library Sciences, Statistics and Operations Research and Computer Science. These three departments are chosen to create the first dataset which will be used for training the model because students in these departments are more likely to take the courses among each other’s departments. Also, these departments have subjects which are of common interest with respect to the career paths. The content of the courses overlaps to some extent in these departments and many of the courses take same references as well. So, students searching courses in one of these departments could potentially find the subjects useful in other two departments. Hence, these departments will serve as a good dataset for the topics development which is the foundation of the model. The dataset will contain … (needs to be filled) courses scraped from all three departments.

Dataset 2 (Testing Dataset):

The second dataset will be developed using the three departments different than training dataset namely, (to be filled along with the justification why they were chosen...). The same information i.e. Department abbreviation, Course Name and Short Course Description will be scraped for the departments. This dataset will be used for testing purpose and will contain … (needs to be filled) courses

Dataset 3 (Combined Dataset):

The third dataset will be an amalgamation of the two datasets mentioned above i.e. the training and testing datasets and will contain … (needs to be filled) courses. (Need to discuss about this set)

Once the three fields are scraped in a csv format for all the datasets, we will perform the preprocessing of the document. The preprocessing will involve following steps:

Tokenization is the initial and necessary step in Natural Language Processing like parsing, word counting, spell checking and other statistical analysis of text. There are different Python packages available for tokenization like NLTK, tokenizer and spaCy. spaCy as opposed to NLTK is newer library which keeps the space tokens as well, so the spaces must be removed as a separate task, but spaCy has a lot of other new features which are useful. I will evaluate both these packages and use one of them.

Then we will perform lemmatization or stemming to convert the inflected words like “walking”, “walked”, “walking” to the base form as “walk”. However, lemmatization and stemming differ in various ways. Since the morphological analysis performed in stemming is less accurate than lemmatization as the root of the word resulted from stemming does not necessarily have to be a meaningful expression [25]. Therefore, for recommender systems lemmatization is a good choice as compared to stemming. The resulting corpus can be put forth for stop words removal.

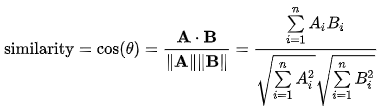
The preprocessing steps provide a corpus which can be used in our model building process. One of the processes which needs to be performed in most of the cases for recommender system model development for convenient data handling is vectorization. We will use Genism, an efficient open source python toolkit used for vector space modelling and topic modelling. Gensim can be used to convert the corpus into vector space representation which essentially is number of occurrences of the feature terms in each document or in other words a bag of words representation. So, in bag of words representation all the words are given the same weight and only the frequency is used to create the tuple of term-frequency for each document. We will also use Gensim to transform the vector representation created using bag of words to another vector representation that uses TF-IDF (term frequency–inverse document frequency). The TF-IDF representation gives a numerical statistic which shows how important a word is in a document in the corpus. The importance of a word (numerical weight) is calculated by multiplying the term frequency with Inverse document frequency. Hence the representation here consists of the tuple of term-weight for each document. The two approaches have different way representing a document and I want to analyze the effects of each approach when used in LDA for generating topics.

**Proposed system**

We propose to develop a model that uses LDA (Latent Dirichlet Allocation). The implementation of LDA model will help extracting different topics which form initial step to put items in different category on the basis of similarity of one document to another. In our model, description of one course along with its name and department abbreviation refers to one document. With the help of the topics provided by LDA each document can be represented in a vector form which tells the percentage of the topics constituting a document. This is a powerful information as it gives an incentive to find distances between the document in the same vector notation. Therefore, the similarity can be calculated using different distance metrics like Manhattan distance, Euclidean distance or Cosine Similarity. We will use Cosine Similarity in our model. To apply the LDA model we will use frequency of the words (features) in each of the documents present in our dataset with the help of bag of words and word-document matrix. In word-document matrix we have used TF-IDF which helps to find the weight (importance) of a word for a document in our dataset.

The LDA algorithm creates a particular number of topics which are given initially to the algorithm as an input while it fits the corpus of the system. In result of the algorithm we get the specified number of topics (fed as input) and each topic contains a set of tuples containing words and frequencies. The number of topics for the model is selected by finding the coherence score using Gensim for different number of topics and the number which gives the highest coherence score is taken as the number of topics for the model. This helps in getting the best possible topics that can be extracted from LDA. The frequency of the words in every topic can be considered as the finger print of that topic and hence each topic becomes distinct topic with respect to a corpus. Therefore, the topics resulted from the LDA algorithm are a set of words and frequencies.

For similarity function we will define a function that uses cosine similarity. The range of the similarity is between -1 to 1 where larger number represents more similarity. The function yields a value of 0 if the vectors are orthogonal and are independent of each other and a 1 if they are maximally correlated. This function can be very useful since it has a low complexity when it comes to sparse vectors as it considers only the non-zero dimensions. Below is the function which will yield the cosine similarity:



Where A and B are the vectors and Ai, Bi are the components of the vector.

Above system gives us a model of finding similarities between documents which can be quantified and hence can be utilized in finding the closest documents of a given document.

**Deployment**

The above developed model serves as the backend for our system which can be fed with an input given by the user. The user inputs a string which is as per their academic interest like “Analytics”, “Information Retrieval”, “Big Data” etc., to the system. The model will take the input string and perform the task of finding the most similar courses using LDA. Once the similar courses are fetched by the algorithm, it will present a list of 10 courses from highest similarity to lowest similarity in front of the user irrespective of the department or the course already taken by the student. We have chosen 10 courses in the list to maintain a balance in the recommendation and not make the list too crowded. We will use Python UI packages like Tkinter or other open source web technologies to present a user interface that uses the above developed model in the backend.

**Evaluation:**

The evaluation of the system will involve a lot of aspects to be considered both for the performance of the algorithm and the overall model. Since we are using the LDA in our model some of the ways through which we can evaluate the topics are as follows:

The developed model trained on the training dataset will be first analyzed on the basis of a few queries fired against the model, around 15 queries can yield results which can reflect the goodness of developed model. The results (recommendations) will of the queries will examined with respect to the query and will be stored for future use to compare the results when the same queries are fired on the test set and the combined dataset. We will also look at the list of the recommendations and examine at which ranking the recommendations starts getting less relevant, for instance, if for all the queries top 3 results are very related to the course in query but then the courses may be non-relevant as well.

The testing set will be used to test the developed model on an unseen set of documents. Here we will use the same 15 queries used to analyze the model on training set. The reason behind using the same queries as training set is that it would be easier to compare the results obtained from both the models. We will analyze the relevance of the recommendations in training set as compared to the testing set. The comparison will be helpful in finding why did a query do badly in test set and not in training set or vice-versa.

The evaluation on the combined set using the same queries as in test and training set (need to discuss)

We will consider improving our feature set by eliminating stop words and provide less weightage to the words which do not give useful information in differentiating one topic from another using TF-IDF. The other important aspect of LDA is deciding the number of topics to be generated. We will run the model for different number of topics and check the terms in each topic whether they make sense or not. This may be checked manually by seeing the words constituting top share in the topic and see whether they are coherent or not i.e. the words are closely related or not.

As discussed in the dataset creation section we will have each dataset with at least … (needs to be filled) documents which is good data size to train the algorithm, so that LDA runs for a sufficient time such that there is no Topic drift or topic difference. This will help ensuring that the topics are well defined and cannot undergo major differences further.

One of the most basic and simplest way to judge the topics generated through LDA is to analyze the top 10 or more terms in each topic and try to come up with a representative name, if it is intuitive and we are able to do it manually then it is an indication of well-formed topics.

Another way is to use the already developed visual tools that can help understand the terms distribution even if the dataset is huge and the number of words is large in the topics. We will use Termite: Visualization Techniques for Assessing Textual Topic Models [26], one of the efficient tools that can help present the distribution of all the terms under different topics. We will use the results of Termite in our evaluation to get a broader perspective of the terms and their distribution in the topics.

**References**

[1] Al-Badarenah Amer, Alsakran Jamal An Automated Recommender System for Course Selection, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 3, 2016

[2] Adomavicius G., Mobasher B., Ricci F., and Tuzhilin A., (2011). Context-Aware Recommender Systems. AI Magazine. 32. 67-80.

[3] Badrul S, George K., Joseph K., and John R., Item-Based Collaborative Filtering Recommendation Algorithms, *Files.grouplens.org.* (2018). [online] Available at: http://files.grouplens.org/papers/www10\_sarwar.pdf

[4] DS Grewal, K Kaur, Developing an Intelligent Recommendation System for Course Selection by Students for Graduate Courses, Bus Eco J 7:209. doi:10.4172/2151-6219.1000209

[5] Gorla, A., Tavecchia, I., Gross, F., & Zeller, A. (2014). Checking app behavior against app descriptions. Proceedings of the 36th International Conference on Software Engineering - ICSE 2014. doi:10.1145/2568225.2568276

[6] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications. In USENIX Security Symposium, pages527–542, 2013.

[7] Koren Y, Bell R, Volinsky C (2009) Matrix Factorization Techniques for Recommender Systems. IEEE Computer Society Press 42: 30-37.

[8] Ricci F, Rokach L, Shapira B, Kantor PB (2011) 1st Recommender Systems Handbook, Springer-Berlin, 1-29.

[9] Gupta, S., Hanson, C., Gunter, C. A., Frank, M., Liebovitz, D., & Malin, B. (2013). Modeling and detecting anomalous topic access. 2013 *IEEE International Conference on Intelligence and Security Informatics.* doi:10.1109/isi.2013.6578795

[11] Dol Aher, Sunita & L. M. R. J., Lobo. (2012). Combination of Clustering, Classification & Association Rule based Approach for Course Recommender System in E-learning. International Journal of Computer Applications. 39. 8-15. 10.5120/4830-7087.

[12] Burke, Robin. (2000). Knowledge-Based Recommender Systems. Encyclopedia of library and information systems. 69.

[13] Lin Weiyang, Alvarez Sergio, Ruiz Carolina, Ecient Adaptive{Support Association Rule Mining for Recommender Systems. Data Mining and Knowledge Discovery · January 2002, DOI: 10.1023/A:1013284820704

[14] Shardanand, U. and P. Maes: 1995, `Social Information Filtering: Algorithms for Automating \Word of Mouth"'. In: Proceedings of the Conference on Human Factors in Computing Systems (CHI95). pp. 210{217

[15] Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl: 1994, `GroupLens: an Open Architecture for Collaborative Filtering of Netnews'. In: Proceedings of the Conference on Computer Supported Cooperative Work (CSCW94). pp. 175-186

[16] Rashid, Al & Albert, Istvan & Cosley, Dan & Lam, Shyong & McNee, Sean & Konstan, Joseph & Riedl, John. (2002). Getting to Know You: Learning New User Preferences in Recommender Systems. International Conference on Intelligent User Interfaces, Proceedings IUI. 10.1145/502716.502737.

[17] Kohrs, A., and Merialdo, B. Improving Collaborative Filtering for New Users by Smart Object Selection, Proceedings of International Conference on Media Features (ICMF) 2001 (oral presentation).

[18] McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. In CHI'06 Extended Abstracts on Human Factors in Computing Systems, CHI EA'06 (pp. 1097-1101) https://doi.org/10.1145/1125451.1125659

[19] Adomavicius, G., Sankaranarayanan, R. Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems,* 23(1), 103-145. doi:10.1145/1055709.1055714

[20] Torres, R., McNee, S.M., Abel, M., Konstan, J.A., and Riedl, J. Enhancing digital libraries with TechLens+. In Proc. of ACM/IEEE JCDL 2004, ACM Press (2004) 228-236.

[21] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5-53. doi:10.1145/963770.963772

[22] Parameswaran Aditya, Venetis Petros, Garcia-Molina Hector Recommendation systems with complex constraints: A course recommendation perspective, Journal ACM Transactions on Information Systems (TOIS), Volume 29 Issue 4, December 2011, Article No. 20

[23] Feng Shanshan, Cao Jian, Wang Jie, Qian Shiyou Recommendations Based on Comprehensively Exploiting the Latent Factors Hidden in Items’ Ratings and Content, journal, ACM Transactions on Knowledge Discovery from Data (TKDD), Volume 11 Issue 3, April 2017, Article No. 35

[24] Amatriain, Xavier, and Josep M. Pujol. “Data Mining Methods for Recommender Systems.” *Recommender Systems Handbook*, 2015, pp. 227–262., doi:10.1007/978-1-4899-7637-6\_7.

[25] C. D. Manning, P. Raghavan,H. Schütze,“Introduction to Information Retrieval”, Cambridge University Press. 2008

[26] Knights, Dan & Mozer, Michael & Nicolov, Nicolas. (2009). Detecting Topic Drift with Compound Topic Models.