Title Page

Recommendation System Using Content Based Filtering For E-commerce

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

in

Information Technology

Under the Supervision of

Dr. POOJA JAIN

By

TARANG GUPTA

Enrollment no. 111453

To

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled “Recommendation System Using Content Based Filtering For E-commerce”, submitted by Tarang Gupta in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Dr. Pooja Jain**

Assistant Professor

Acknowledgement

I would like to express my deepest appreciation to all those who have been helping me throughout the project and without whom this project would have been a very difficult task. I would like to extend my sincere thanks to all of them.

I am highly indebted to Dr. Pooja Jain for her guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in doing my project. I would like to express my gratitude towards my parents & members of JUIT for their kind co-operation and encouragement which helped me in doing this project. My thanks and appreciations also go to my colleagues who have helped me out with their abilities in developing the project.

Date: 08-05-2015 Tarang Gupta

Contents

Abbreviation and symbols

HTML5	Hyper Text Markup Language 5
tf idf	Term frequency-inverse document frequency
pdf	Portable Document Format
txt	Text File

List of Figures

List of Tables

Abstract

Recommender Systems are new generation internet tool that help user in navigating through information on the internet and receive information related to their preferences.

The main goal is to propose a framework for recommendation system using content based filtering for e-commerce. Predictive models will use the products information and user information as sources of data which will be compared based on various parameters and the most appropriate product will be recommended. It is a user specific model. These models, in turn, will allow the system the decision of ordering a set of items according to their predicted usefulness. To show the validity and feasibility of our approach, a prototype application has been built, that implements solutions to the recommendation problem from different standpoints: identification of the data sources, inference based on similarity of items and user, recommending the most valued product and also updating the model from time to time.

CHAPTER 1

Recommendation System

1.1 What is recommendation system?

With the popularization of the Internet and the development of E-commerce, the Ecommerce sites offer millions of products for sale. Choosing among so many options is challenging for consumers. So users usually get lost in the vast space of commodity information and cannot find the goods they really want. Recommender systems have emerged in response to this problem. A recommender system for an E-commerce site recommends products that are likely to fit user's needs. Recommendation systems changed the way inanimate websites communicate with their users. Rather than providing a static experience in which users search for and potentially buy products, recommender systems increase interaction to provide a richer experience. Recommender systems identify recommendations autonomously for individual users based on past purchases and searches, and on other users' behavior.

Under the increasingly intense competitive circumstance the E-commerce recommender system can effectively reserve users, keep them from losing and increase the cross selling ability. According to the research, with the personalized recommender system used in Ecommerce marketing industries, the sales improved by 2%-8%, especially in those industries such as books, movies, CD audio-video products, and articles of daily use, which are cheap and various in kinds, and great in extent of using personalized recommender system. The recommender system can greatly boost sales. Recommender systems have become extremely common in recent years, and are applied in a variety of applications. The most popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general. However, there are also recommender systems for experts, jokes, restaurants, financial services, life insurance, persons (online dating), and Twitter followers. Suggestions for books on Amazon, or movies on Netflix, ebay, Levis, Moviefinder.com are real world examples of the operation of industry-strength recommender systems. Two main technologies are usually adopted in personalized recommendation systems: content-based filtering and collaborative filtering, and hybrid filtering.

1.2 Standard Data Sources

The data that we need to process to recommend product comes from a variety of sources. The process of obtaining the data is different for different types of data.

1.2.1 Web Server, Cookies & Database Logs

The logs & cookies, collected and used primarily for technical reasons, can also be of assist to marketing strategies. From these logs we can extract information like which pages a certain customer visited, how much time he spend on each one, if he reached a page following a link from another page or if he clicked any links on this page. From our database we have access to things like which products a customer has bought, his orders' values, when he placed his orders (time and date), what items he bought together. A smart system could learn from these data a customer's shopping habits and use it for suggestions.

1.2.2 Customer's Profile – Personalization

In order to use an e-shop, someone usually needs an account. This alone enables us to use many features. It gives us large amount of data to process. Typical registration data are name, address, age and e-mail. Besides these typical data, there are more information we can ask from a customer upon his registration, like to choose product categories that interest him most and what he thinks as an acceptable amount of e-mail. Most people are ok and even expect to give this information since we live in the age of social or they can skip the process and proceed.

1.2.3 Products Speaking for Themselves

Products for an e-commerce platform are entries to a database. By using separate standardized fields for each of the characteristics we can easily perform content-based suggestions. A classification system is also of use, to assign products to categories and subcategories.

1.2.4 User’s Feedback

It includes every action a user may do that isn’t directly related to the shopping process. The most common feedback forms are ratings and comments. These days likes on Facebook, tweets on twitter and +1s on Google Plus are also common. Ratings and review are considered standard these days. Ratings is the classic star system which usually takes values from 1 to 5 stars and no stars equal to no rating. Reviews can be scanned using automatic text analysis methods to extract useful information. We may also omit some criteria from ratings and try to extract them from comments. The great example of user feedback is Amazon.

1.3 Types of Technologies for Recommender Systems

1.3.1 Collaborative Filtering

One approach to the design of recommender systems that has seen wide use is collaborative filtering. Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. Collaborative filtering methods are based on collecting and analyzing a large amount of information on users’ behaviors, activities or preferences and predicting what users will like based on their similarity to other users. It tries to find persons with similar tastes as the current visitor and then proceeds to suggest items that they like. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. Collaborative Filtering are based on the assumption that people who agree in past will agree in future too and that they will like the similar kinds of items they like in the past. A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see.

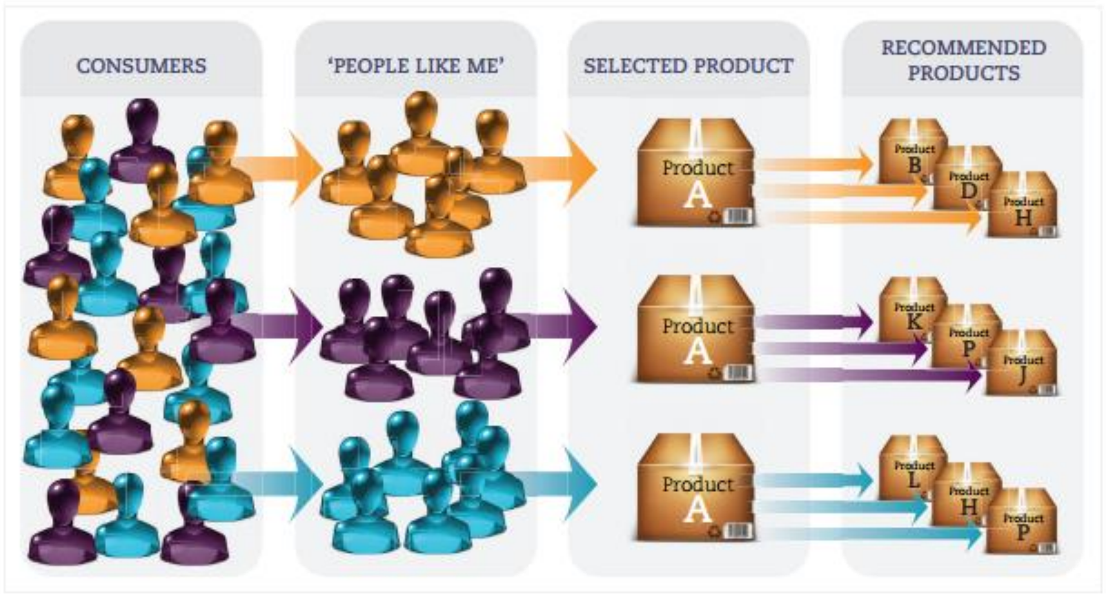


Figure 1Collaborative

One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's recommender system. Other examples include:

- As previously detailed, Last.fm recommends music based on a comparison of the listening habits of similar users.
- Facebook, Myspace, LinkedIn, and other social networks use collaborative filtering to recommend new friends, groups, and other social connections (by examining the network of connections between a user and their friends). Twitter uses many signals and in-memory computations for recommending who to follow to its users.

Collaborative filtering algorithms often require (1) users' active participation, (2) an easy way to represent users' interests to the system, and (3) algorithms that are able to match people with similar interests.

Typically, the workflow of a collaborative filtering system is:

1. A user expresses his or her preferences by rating items (e.g. books, movies or CDs) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user’s ratings against other users’ and finds the people with most “similar” tastes.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

A key problem of collaborative filtering is how to combine and weight the preferences of user neighbors. Sometimes, users can immediately rate the recommended items. As a result, the system gains an increasingly accurate representation of user preferences over time.

1.3.1.1 Challenges of Collaborative Filtering

Data sparsity (Start)

Many commercial recommender systems are based on large datasets. As a result, the user-item matrix used for collaborative filtering could be extremely large and sparse, which brings about the challenges in the performances of the recommendation. One typical problem caused by the data sparsity is the cold start problem. As collaborative filtering methods recommend items based on users’ past preferences, new users will need to rate sufficient number of items to enable the system to capture their preferences accurately and thus provides reliable recommendations. Similarly, new items also have the same problem. When new items are added to system, they need to be rated by substantial number of users before they could be recommended.

Scalability

As the numbers of users and items grow, traditional CF algorithms will suffer serious scalability problems. For example, with tens of millions of customers $O(M)$ and millions of items $O(N)$, a CF algorithm with the complexity of n is already too large. As well, many systems need to react immediately to online requirements and make recommendations for all users regardless of their purchases and ratings history, which demands a higher scalability of a CF system.

Synonyms

Synonyms refers to the tendency of a number of the same or very similar items to have different names or entries. Most recommender systems are unable to discover this latent association and thus treat these products differently.

For example, the seemingly different items “children movie” and “children film” are actually referring to the same item. Indeed, the degree of variability in descriptive term usage is greater than commonly suspected. The prevalence of synonyms decreases the recommendation performance of CF systems. Topic modeling (like the Latent Dirichlet Allocation technique) could solve this by grouping different words belonging to the same topic.

Grey sheep

Grey sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering. Black sheep are the opposite group whose idiosyncratic tastes make recommendations nearly impossible. Although this is a failure of the recommender system, non-electronic recommenders also have great problems in these cases, so black sheep is an acceptable failure

Shilling attacks

In a recommendation system where everyone can give the ratings, people may give lots of positive ratings for their own items and negative ratings for their competitors. It is often necessary for the collaborative filtering systems to introduce precautions to discourage such kind of manipulations.

Diversity and the Long Tail

Collaborative filters are expected to increase diversity because they help us discover new products. Some algorithms, however, may unintentionally do the opposite. Because collaborative filters recommend products based on past sales or ratings, they cannot usually recommend products with limited historical data. This can create a rich-get-richer effect for popular products, akin to positive feedback. This bias toward popularity can prevent what are otherwise better consumer-product matches. A Wharton study details this phenomenon along with several ideas that may promote diversity and the "long tail.

1.3.1.2 Types

Memory-Based

This mechanism uses user rating data to compute similarity between users or items. This is used for making recommendations. This was the earlier mechanism and is used in many commercial systems. Typical examples of this mechanism are neighborhood based CF and item-based/user-based top-N recommendations.

The neighborhood-based algorithm calculates the similarity between two users or items, produces a prediction for the user taking the weighted average of all the ratings. Similarity computation between items or users is an important part of this approach. Multiple mechanisms such as Pearson correlation and vector cosine based similarity are used for this.

The Pearson correlation similarity of two users x, y is defined as

$$\mathrm{simil}(x,y) = \frac{\sum\limits_{i\in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum\limits_{i\in I_{xy}} (r_{x,i} - \bar{r_x})^2 \sum\limits_{i\in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

Where I_{xy} is the set of items rated by both user x and user y.

The cosine-based approach defines the cosine-similarity between two users x and y as:

$$\mathrm{simil}(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}\cdot\vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum\limits_{i\in I_{xy}} r_{x,i}r_{y,i}}{\sqrt{\sum\limits_{i\in I_x} r_{x,i}^2}\sqrt{\sum\limits_{i\in I_y} r_{y,i}^2}}$$

The user based top-N recommendation algorithm identifies the k most similar users to an active user using similarity based vector model. After the k most similar users are found, their corresponding user-item matrices are aggregated to identify the set of items to be recommended.

Model-Based

Models are developed using data mining, machine learning algorithms to find patterns based on training data. These are used to make predictions for real data. There are many model-based CF algorithms. These include Bayesian networks, clustering models, latent semantic models such as singular value decomposition, probabilistic latent semantic analysis, Multiple Multiplicative Factor, Latent Dirichlet allocation and markov decision process based models. This approach has a more holistic goal to uncover latent factors that explain observed ratings. Most of the models are based on creating a classification or clustering technique.

Hybrid

A number of applications combines the memory-based and the model-based CF algorithms. These overcome the limitations of native CF approaches. It improves the prediction performance. Importantly, it overcomes the CF problems such as sparsity and loss of information. However, they have increased complexity and are expensive to implement. Usually most of the commercial recommender systems are hybrid, for example, Google news recommender system.

1.3.2 Content-Based Filtering

Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research.

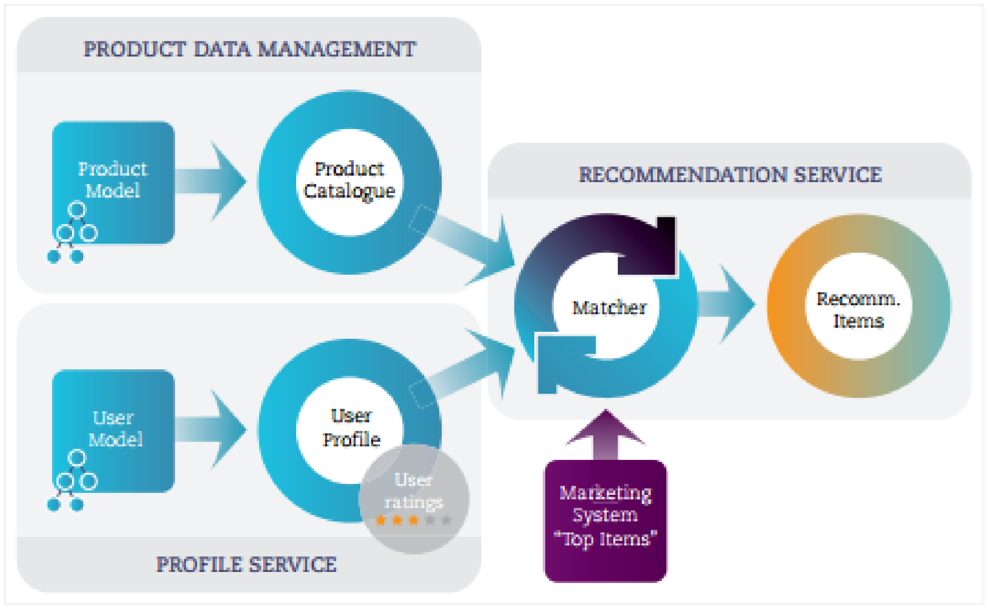


Figure 2Content-based

To abstract the features of the items in the system, an item presentation algorithm is applied. A widely used algorithm is the tf–idf representation (also called vector space representation). To create user profile, the system mostly focuses on two types of information:

1. A model of the user's preference.

2. A history of the user's interaction with the recommender system.

Basically, these methods use an item profile characterizing the item within the system. The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item. Direct feedback from a user, usually in the form of a like or dislike button, can also be used.

A key issue with content-based filtering is whether the system is able to learn user preferences from user's actions regarding one content source and use them across other content types.

As previously detailed, Pandora Radio is a popular example of a content-based recommender system that plays music with similar characteristics to that of a song provided by the user as an initial seed. Also systems using this approach are Rotten Tomatoes, Internet Movie Database, Jinni, Rovi Corporation, Jaman and See This Next.

1.3.2.1 Challenges of Collaborative Filtering

LIMITED CONTENT ANALYSIS

Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend. Domain knowledge is often needed, e.g., for movie recommendations the system needs to know the actors and directors, and sometimes, domain ontologies are also needed. No content-based recommendation system can provide suitable suggestions if the analyzed content does not contain enough information to discriminate items the user likes from items the user does not like. Some representations capture only certain aspects of the content, but there are many others that would influence a user’s experience. Both automatic and manually assignment of features to items could not be sufficient to define distinguishing aspects of items that turn out to be necessary for the elicitation of user interests. The terms have to be represented such that both the user profile and the items can be compared in a meaningful way.

OVER- SPECIALIZATION

Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile, hence the user is going to be recommended items similar to those already rated. This drawback is also called serendipity problem to highlight the tendency of the content-based systems to produce recommendations with a limited degree of novelty. To give an example, when a user has only rated movies directed by Stanley Kubrick, she will be recommended just that kind of movies. A “perfect” content-based technique would rarely find anything novel, limiting the range of applications for which it would be useful.

NEW USER

Enough ratings have to be collected before a content-based recommender system can really understand user preferences and provide accurate recommendations. Therefore, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations. A learning algorithm has to be chosen that is able to learn the user profile based on seen items and can make recommendations based on this user profile.

1.3.2.2 Algorithms

Item Representation Model

Vector space model

The vector space model is a text representation model. It has the text and all the functional items constitute the basic unit of the terms set project. Each item can be expressed as a vector and the dimension of the vector is the number of item sets. General is not fixed and we can also specify a fixed size. Because the characteristic frequency of the word document to a certain extent reflects the theme of the file, so each component is the number of items in the feature vector document. This concentration of resources in the resource can be expressed as a term sets of vectors.

Probability model

The probability model is firstly established in the field of the classification model and then calculates the classification probability distribution of all the files and users interested in the model. Used to denote the probability distribution of documents and users' interests can better reflect the diversity of user interest, and easy to implement. The classification model is using the Bayesian method of training. The expression of the interests of users and files are the same.

Improved probabilistic model

Vector space model method can only express user interest keywords. It cannot distinguish the difference between the user interests. Despite the differences can be distinguished on the probability model approach is based on the user's interests, the diversity of the user's interest, but cannot express the love of the user of the level of interest rates. Therefore, in order to improve the method, the improved probability model can express user interest keywords and express the level of user interest.

User Interest Model

Interest to the user and the candidate documents match the calculation, first need to define the user's computer interest and candidate documents said. We use the classic VSM model document, said the candidate, that candidate document D can be expressed as follows:

D = (w ,w , ...,w)

Where w_i is the first document D_i a feature term weight. We select the word as the feature item, and use the relative term frequency as the characteristics of term weight. Relative Frequency Words can tf-idf formula is as follows:

$$w_i = tf_i \cdot idf_i = \frac{freq_i}{\max_1 freq_1} \cdot \log \frac{N}{n_i}$$

That words which appear in document D the number of documents which indicated that the number of times the word occurs, N the total number of that document.

User Interest Model can be based on Interest Document Vector, Interest Vector, Multi-Interest Vector or based on Role.

Latent Semantic Indexing

Describing documents and profiles with single terms has several drawbacks. The same concept can usually be described with several words (synonymy) which means that profiles and document will need to contain exactly the same terms in order to be matched. The words “car” and “vehicle” for example can be used to refer to the same type of information. Conversely, many words have more than one distinct meaning (polysemy). A profile containing the term “mouse” for example will produce a match with documents that also contain this term, regardless of whether it was used in the context of computers or animals. Latent semantic indexing (LSI) is an extension of the vector space model that tries to overcome these deficiencies by incorporating semantic information. The major advantage of LSI is the fact that terms in documents and profiles can be very different but still produce a match based on the semantic relation. A disadvantage is the computational complexity of the matrix computations which could reduce run-time performance to an unacceptable level.

Similarity Algorithms

Cosine Similarity

In this case, two users are regarded as two vectors in the n dimensional item space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Formally, similarity between users i and j is given by

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Here A and B are the two vectors.

Pearson’s Collection Similarity

In this case, similarity between users i and j is measured by computing the Pearson correlation. To make the correlation computation accurate we isolate the co-rated cases. The correlation similarity is given by

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

Here Pearson's correlation coefficient p , cov is the covariance, σ_X is the standard deviation of X, μ_X is the mean of X, and E is the expectation.

Learning Method

The efficiency of a learning method does play an important role in the decision of which method to choose. The most important aspect of efficiency is the computational complexity of the algorithm, although storage requirements can also become an issue as many user profiles have to be maintained. The ability of a learning method to adapt to changes in the user's preferences also plays an important role. The learning method has to be able to evaluate the training data as instances do not last forever but become obsolete as the user's interests change. Another criteria is the number of training instances needed. A learning method that requires many training instances before it is able to make accurate predictions is only useful when the user's interests remain constant for a long period of time. Learning methods also differ in their ability to modulate the training data as instances age. Relevance feedback, genetic algorithms, neural networks, and the Bayesian classifier are among the learning techniques for learning a user profile. The vector space model and latent semantic indexing can both be used by these learning methods to represent documents.

1.3.3 Hybrid Recommender Systems

Combining collaborative filtering and content-based filtering could be more effective in various cases. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model. Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommender systems. A hybrid recommender system is one that combines multiple techniques together to achieve some synergy between them.

- Collaborative: The system generates recommendations using only information about rating profiles for different users. Collaborative systems locate peer users with a rating history similar to the current user and generate recommendations using this neighborhood.
- Content-based: The system generates recommendations from two sources: the features associated with products and the ratings that a user has given them. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on product features.
- Demographic: A demographic recommender provides recommendations based on a demographic profile of the user. Recommended products can be produced for different demographic niches, by combining the ratings of users in those niches.
- Knowledge-based: A knowledge-based recommender suggests products based on inferences about a user’s needs and preferences. This knowledge will sometimes contain explicit functional knowledge about how certain product features meet user needs.

Seven hybridization techniques are:

- Weighted: The score of different recommendation components are combined numerically.
- Switching: The system chooses among recommendation components and applies the selected one.
- Mixed: Recommendations from different recommenders are presented together.
- Feature Combination: Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.

- Feature Augmentation: One recommendation technique is used to compute a feature or set of features, which is then part of the input to the next technique.
- Cascade: Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.
- Meta-level: One recommendation technique is applied and produces some sort of model, which is then the input used by the next technique.

CHAPTER 2

Literature Review

According to the paper on Learning User Interest Model for Content-based Filtering in Personalized Recommendation System, with the emergence and evolution of Networks, the information on the Internet has increased greatly. Retrieving useful information from a large amount of information has become a key technology in the information area. The application of personalized recommendation in the Internet effectively improved its service, especially the service of E-commerce. Traditional search engine do not take different user’s interest into consideration, so the result they retrieved cannot satisfy user’s specified needs. In order to effectively solve the problem, this paper presented a personalized recommendation system employing user interest model for content-based filtering. This paper analyzes the system of five different components: document information extraction, document vectors representation, user interest model representation; matching algorithms, user feedback update. This personalized recommendation system can describe user’s interest type and interest degree well, and can enhance the personalized information service efficiency.

Also in the journal, An Ontology- Content-based Filtering Method, traditional content-based filtering methods usually utilize text extraction and classification techniques for building user profiles as well as for representations of contents, i.e. item profiles. These methods have some disadvantages e.g. mismatch between user profile terms and item profile terms, leading to low performance. Some of the disadvantages can be overcome by incorporating a common ontology which enables representing both the users' and the items' profiles with concepts taken from the same vocabulary. It proposes a new content-based method for filtering and ranking the relevancy of items for users, which utilizes a hierarchical ontology. The method measures the similarity of the user's profile to the items' profiles, considering the existing of mutual concepts in the two profiles, as well as the existence of "related" concepts, according to their position in the ontology. The proposed filtering algorithm computes the similarity between the users' profiles and the items' profiles, and rank-orders the relevant items according to their relevancy to each user. The method is being implemented in ePaper, a personalized electronic newspaper project, utilizing a hierarchical ontology designed specifically for classification of News items. It can, however, be utilized in other domains and extended to other ontologies.

HE Weihong, CAO Yi from Wuhan University also talks about the increase in sales if ecommerce incorporates good recommender system. The architecture and the workflow of the proposed system is explained where users' unique features are explored by means of vector space model firstly. Then based on the qualitative value of products information, the recommender lists were obtained. This system can adapt to the users feedback automatically, therefore its performance is enhanced comprehensively. Finally the evaluation of the system was done by calculating the precision and recall and the experimental results were presented.

Recommender System are new generation internet tool that help user in navigating through information on the internet and receive information related to their preferences. Although most of the time recommender systems are applied in the area of online shopping and entertainment domains like movie and music, yet their applicability is being researched upon in other area as well. The Recommender Systems which are currently working in the domain of online book shopping can combine user choices with not only similar users but other users as well to give diverse recommendation that change over time. The overall architecture of the proposed system is presented and its implementation with a prototype design is described. An empirical evaluation of the system based on a survey reflect the impact of such diverse recommendations on the user choices.

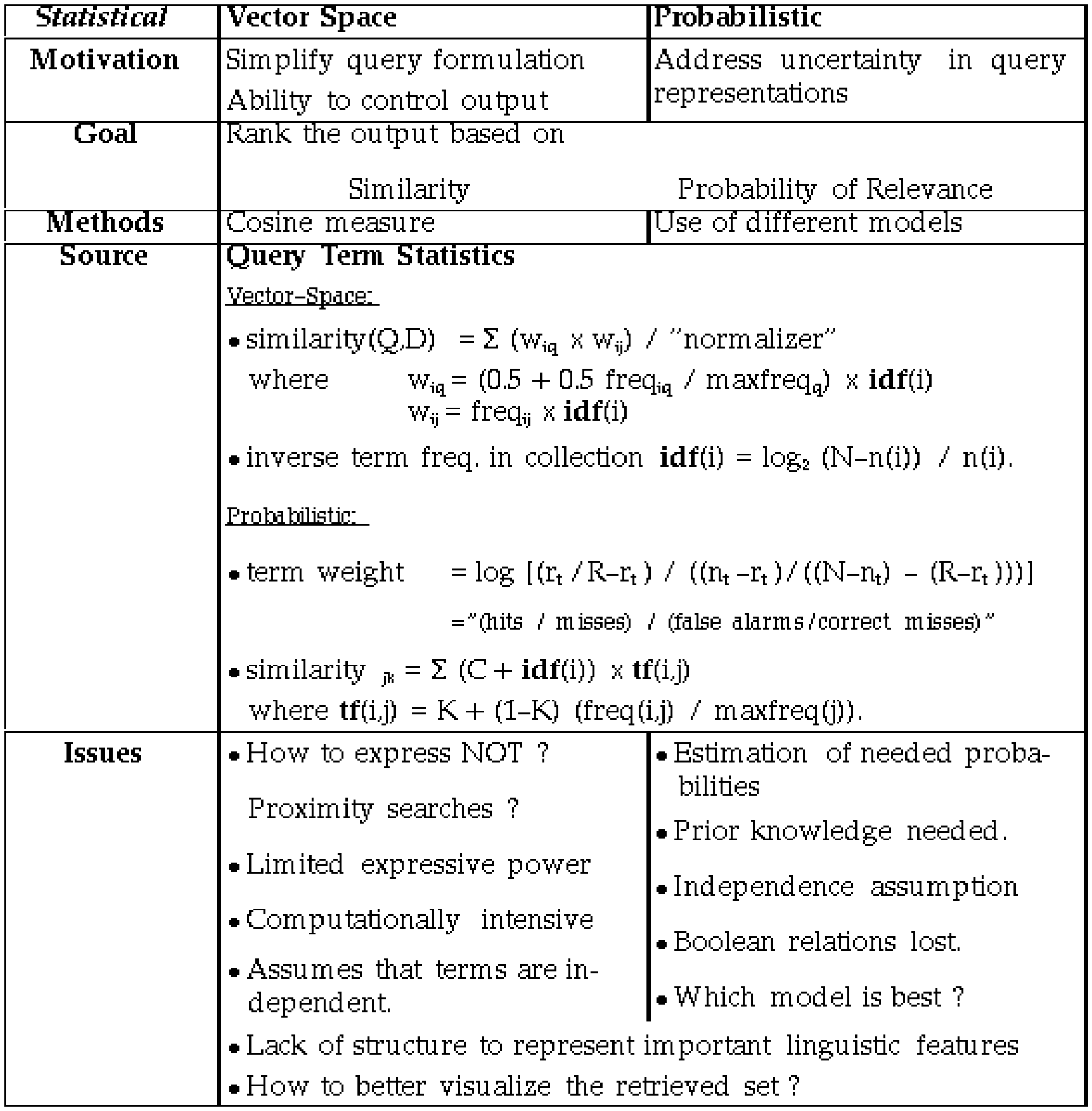
“Methods of Determining the similarity of documents“ paper gave details description of the terms document and similarity. This paper provides an overview on what is involved when one wants to index digital text documents using the open source search engine library Apache Lucene. A document is as a piece of written, printed, or electronic matter that provides information or evidence or that serves as an official record. This definition gives us at least two clues. First, documents can exist in many different forms both analog and digital and they contain information. The following list are a few examples of different document kinds. • Book • Online article • Newspaper article • Photography • Letter • Movie Going further in describing documents we can find a whole bunch of attributes like title, author, type, creation date or date of last change just to name a few. But when we need to decide if we want to read a document, mostly we want to know what the content of a document is all about. Similarity is the state or fact of being similar while similar is referring to a resemblance in appearance, character, or quantity, without being identical.

Learning materials recommendation using good learners' ratings and content-based filtering. The enormity of the amount of learning materials in e-learning has led to the difficulty of locating suitable learning materials for a particular learning topic, creating the need for recommendation tools within a learning context. This paper address this need by proposing a novel e-learning recommender system framework that is based on two conceptual foundations—peer learning and social learning theories that encourage students to cooperate and learn among themselves. The framework works on the idea of recommending learning materials with a similar content and indicating the quality of learning materials based on good learners' ratings. A comprehensive set of experiments were conducted to measure the system accuracy and its impact on learner's performance. The obtained results show that the proposed e-learning recommender system has a significant improvement in the post-test of about 12.16% with the effect size of 0.6 and 13.11% with the effect size of 0.53 when compared to the e-learning with a content-based recommender system and the e-learning without a recommender system, respectively.

2.1 Relevance to my topic

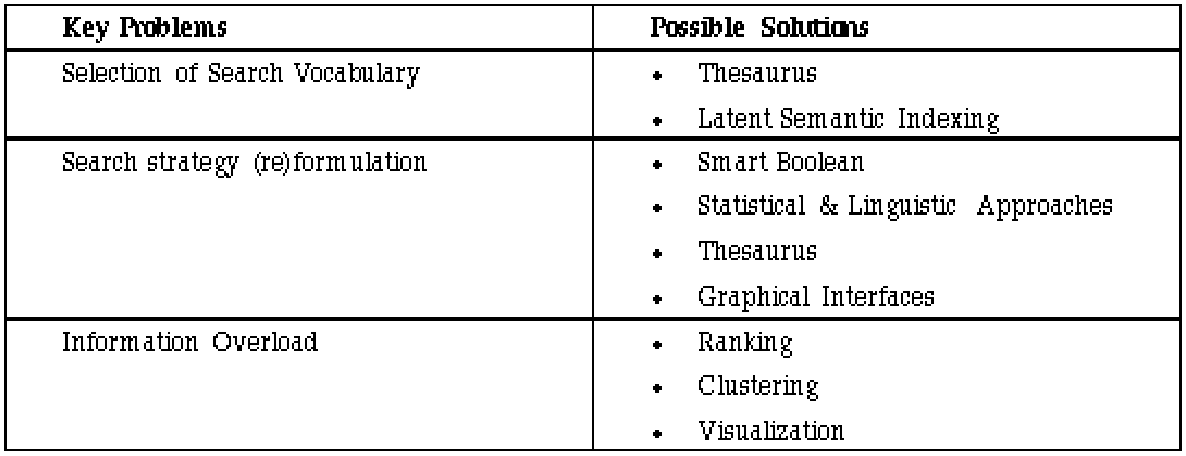
The above papers are centered on my project topic that is recommendation system using content based filtering in ecommerce. These paper propose various algorithms to implement it. Also it tells about the shortcomings present in today's systems along with the methods to improve them. Studying these paper I am able to make a comparative analysis of various algorithms and approaches and chose the appropriate one.

Table 1:Comparison between vector space and probabilistic model



Statistical	**Vector Space**	**Probabilistic**
Motivation	Simplify query formulation Ability to control output	Address uncertainty in query representations
Goal	Rank the output based on Similarity	Probability of Relevance
Methods	Cosine measure	Use of different models
Source	**Query Term Statistics** <u>Vector-Space:</u> • similarity(Q,D) $= \Sigma\ (w_{iq} \times w_{ij})$ / "normalizer" where $w_{iq} = (0.5 + 0.5\ freq_{iq} / maxfreq_q) \times$ **idf**(i) $w_{ij} = freq_{ij} \times$ **idf**(i) • inverse term freq. in collection **idf**(i) $= \log_2 (N-n(i)) / n(i)$. <u>Probabilistic:</u> • term weight $= \log [(r_t / R-r_t) / ((n_t - r_t) / ((N-n_t) - (R-r_t)))]$ ="(hits / misses) / (false alarms/correct misses)" • similarity $_{jk} = \Sigma\ (C +$ **idf**(i)) x **tf**(i,j) where **tf**(i,j) = K + (1-K) (freq(i,j) / maxfreq(j)).	
Issues	• How to express NOT ? Proximity searches ? • Limited expressive power • Computationally intensive • Assumes that terms are in-dependent.	• Estimation of needed proba-bilities • Prior knowledge needed. • Independence assumption • Boolean relations lost. • Which model is best ?
	• Lack of structure to represent important linguistic features • How to better visualize the retrieved set ?	

Table 2:showing key problems and its solution:



Key Problems	Possible Solutions
Selection of Search Vocabulary	• Thesaurus • Latent Semantic Indexing
Search strategy (re)formulation	• Smart Boolean • Statistical & Linguistic Approaches • Thesaurus • Graphical Interfaces
Information Overload	• Ranking • Clustering • Visualization

CHAPTER 3

Methodology

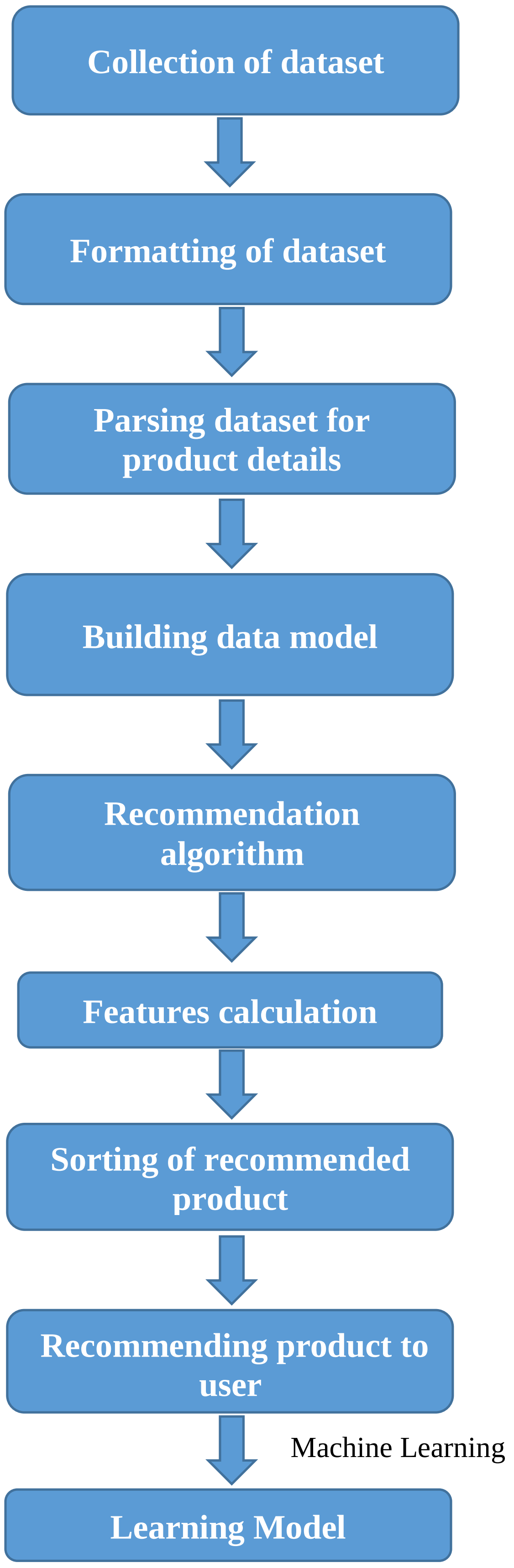
3.1 Objective

- Study
- Identify the resources
- Extract data set
- Identify features as a base for recommendation
- Apply algorithm for recommendation
- Build GUI to present the project
- Integrate

3.2 Materials Used

JAVA	:parsing xml, algorithm
Swings	:Search engine :development
Mysql 5.6.17	:database collection
Dataset	:XML/CSV/Txt
Apache Lucene	

3.3 Method


Collection of dataset
Formatting of dataset
Parsing dataset for product details
Building data model
Recommendation algorithm
Features calculation
Sorting of recommended product
Recommending product to user
Machine Learning
Learning Model

Steps in detail:

1. For building a recommender system first we need to collect data set. The dataset is of MovieLens. MovieLens data sets were collected by the GroupLens Research Project. The data set consists of:
    a. 100,000 ratings (1-5) from 943 users on 1682 movies.
    b. Each user has rated at least 20 movies.
2. The dataset has files :
    a. MovieLens5MRatings.csv : This file contains the ratings for movies. The full set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of user id | item id | rating.
    b. tmp: This folder has 963 text files. Each filed is named after the movie id. These file has been made by parsing the u.item file which has information about the items (movies). Each file has details such as moive title, release date, genres, and the summary for that movie.
    c. Similarity: This is a text file that has been prepared using java programs to calculate the cosine similarity between each pair of the movie file. The file contains the movie id of both the files along with its similarity. This has been computed previously so that on run time the system does not take much time in parsing all the file to calculate the similarity. This has been done before hand in order to have faster calculation when user is using the system.
3. In order to compare a huge amount of documents we need an index where we can look up documents by certain keywords. Lucene is a library written in Java that allows the programmer to build and maintain an index of documents and allows to query this index to find those documents one is interested in. It also allows to build clusters, which is the process of grouping similar documents.

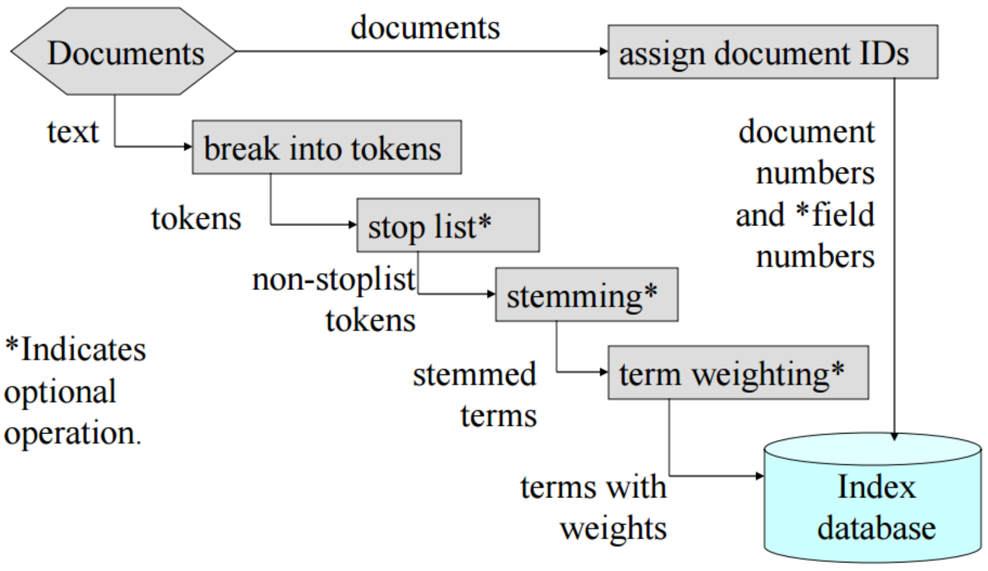


Figure 3Method of DataPrep

4. Building an index : , Lucene stores a list of keywords for each indexed documents. But instead of using a fixed vocabulary, Lucene extracts this keywords out of the documents text. This process is called tokenization. Here it filters out the rubbish and only keep what really matters for the given document. In order to do so Lucene allow to apply filters both before and after the tokenization. Lucene brings along a whole bunch of analyzers both optimized for a specific language or make use of a specific tokenization algorithm. Those analyzers are build with general purpose in mind, so one may want to bring in optimizations specific to the given documents.
5. Parsing: Extracting content from document. For Apache Lucene can only deal with plain text, the contents from the document need to be extracted which might contain also formating and oder non textual elements. To have document attributes indexed in their own fields we extracted them too allowing us to search by them separately. This process is called information extraction which is the process of filling the fields and records of a database from unstructured or loosely formatted text. Depending on the diversity of the document collection and the degree of structuring we need to take more or less effort in order to extract whatever information we are interested in. Given a online article that makes use of Hyper Text Markup Language 5 (HTML5) in combination with Resource Description Framework (RDF) we can consider ourself lucky. The same holds true for offline documents like Portable

Document Format (PDF) or the file formats from word processors like Word or OpenOffice if and only if the meta data fields are set. But in most of the cases we are confronted with documents that does not provide any structured information and we are challenged with the need of parsing native language.

6. Analyzing: Tolenization and Filtering. Tokenization refer to the process of splitting up a text in to small pieces (terms) which then will be used to match search query and documents against each other. The most common techniques is stemming and stop word filtering. In native language one term can appear in different form e.g. singular and plural. Stemming uses an algorithmic approach to reduce such words to their stem. Stemming is highly language specific and needs to be fine tuned. Lucene has implementations of the Porter, Hunspell and Snowflake stemmer. Each of the language specific analyzers provided by Lucene make use of language specific stemming.
7. Stop words are small words that appears often inside a text and therefore does not have any document specific relevance. Filtering out those words will lead to a term list that contains proportionally more relevant terms. Depending on the language other words should be removed got achieve good results. If a document collection belongs to a common domain improvements can be made by using a domain specific list of stop words. Apache Lucene contains a list of stop words withing each of its language specific analyzer. While this lists are relatively short (below hundred), it is highly recommended to apply custom stop word lists.
8. Similarity Measures: In order to compute the similarity of documents we need some mathematical expression or an algorithm the computer can work with. This is called a similarity or distance measure witch maps down the similarity or difference to one single numeric value.

   Let x and y be any two objects in a set and d(x, y) be the distance between x and y.

   1. The distance between any two points must be non- negative, that is, $d(x, y) \geq 0$.

   2. The distance between two objects must be zero if and only if the two objects are identical, that is, $d(x, y) = 0$ if and only if $x = y$.

3. Distance must be symmetric, that is, distance from x to y is the same as the distance from y to x, ie. d(x, y) = d(y, x).

4. The measure must satisfy the triangle inequality, which is d(x, z)≤d(x, y) + d(y, z).

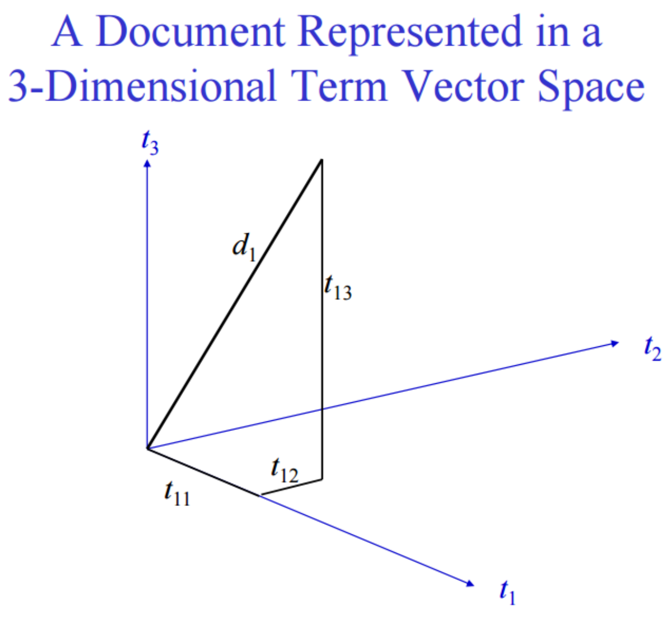


Figure 4TermVectorSpace

9. Cosine similarity: By default Lucene uses a similarity measure based on the so called vector space model as documented in Class TFIDFSimilarity. This model describes a document as a multi dimensional vector where each occurring therm in the whole document collection represents one dimension. The cosine similarity is then simply the angle between two of this document vectors (Huang, 2008). The cosine similarity is combined with the term frequency–inverse document frequency (tf idf) weighting factor. This factor reduces the relevance of common words so they do get dominant. Lucene uses the following formula to calculate the tf idf factor.

$$tf(t \text{ in } d) \cdot idf(t) = f \wedge (1/2 \cdot (1 + \log(D /(df + 1)))$$

Where f is the frequency the term appears in the given document, D is the number of documents and df the frequency the term appears in all documents.

10. After calculating cosine similarity it is stored in the seperate file. Similarly the vector of user profile is also made.
11. According to the similarity of user profile vector and movie vector the list of recommendation is made. The list contains the movie having similarity more than a particular threshold.
12. User will get the result set that will be sorted in descending order which can be further either downloaded or purchased.
13. A learning model is made that will update the model based on the users' activity.

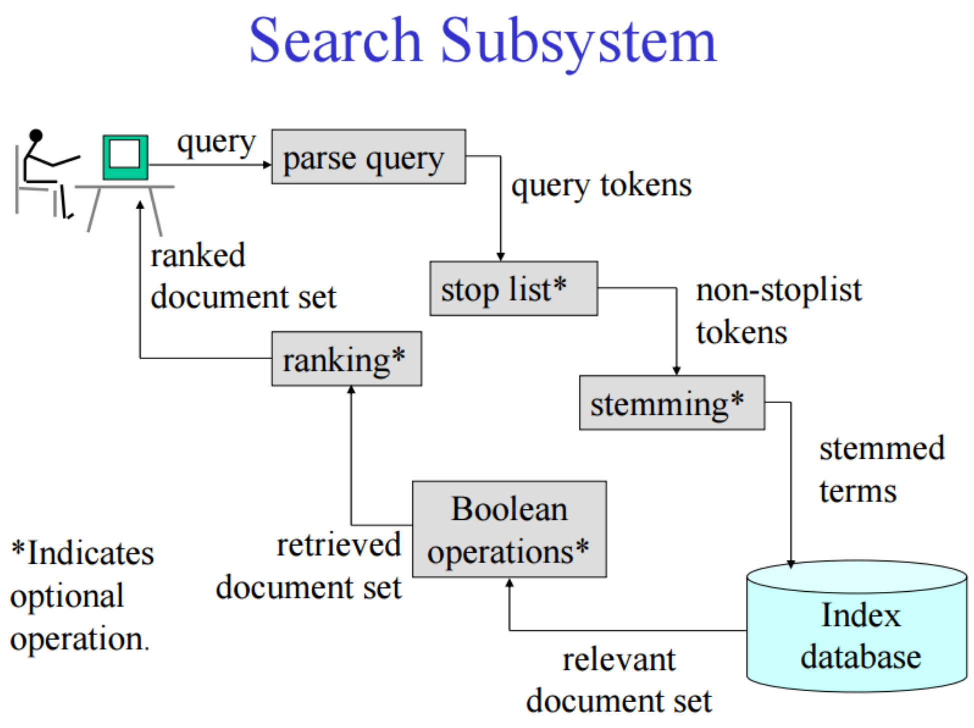


Figure 5Search Subsystem

3.4 Use case diagram

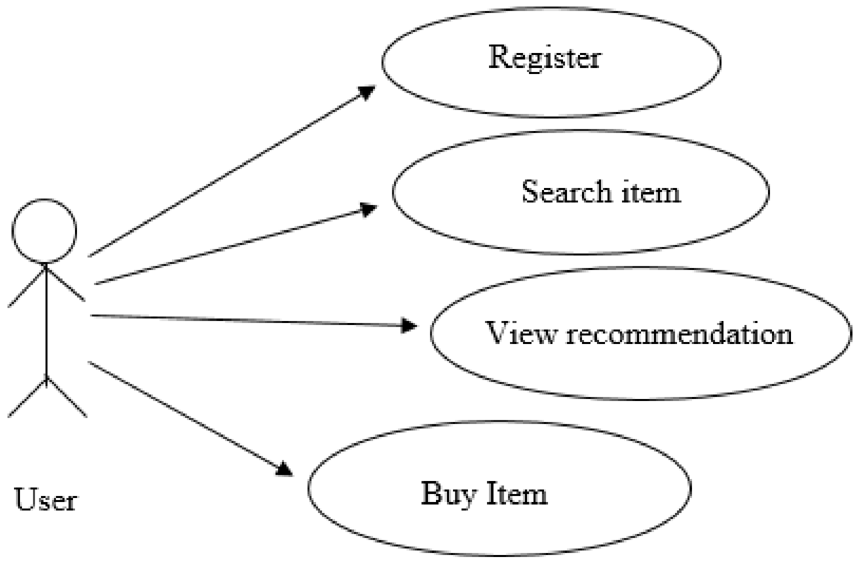


Figure 6use case diagram:user

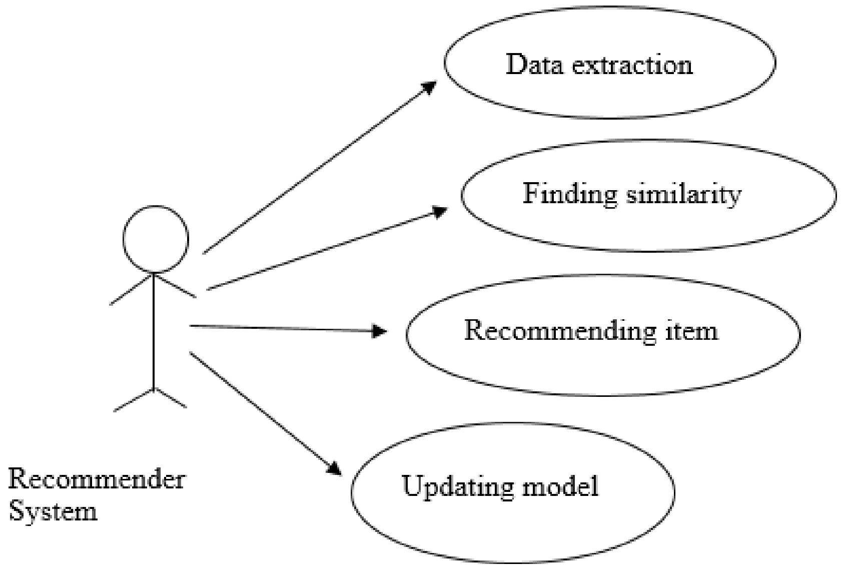


Figure 7Use case diagram : recommender system

3.5 DFD

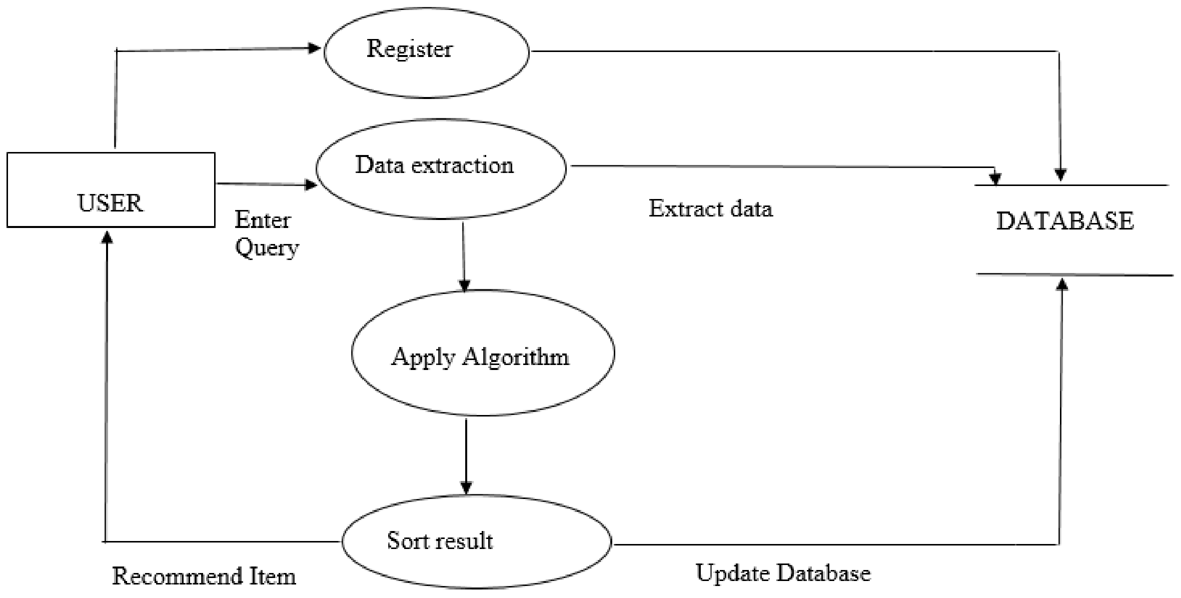


Figure 8Data flow diagram

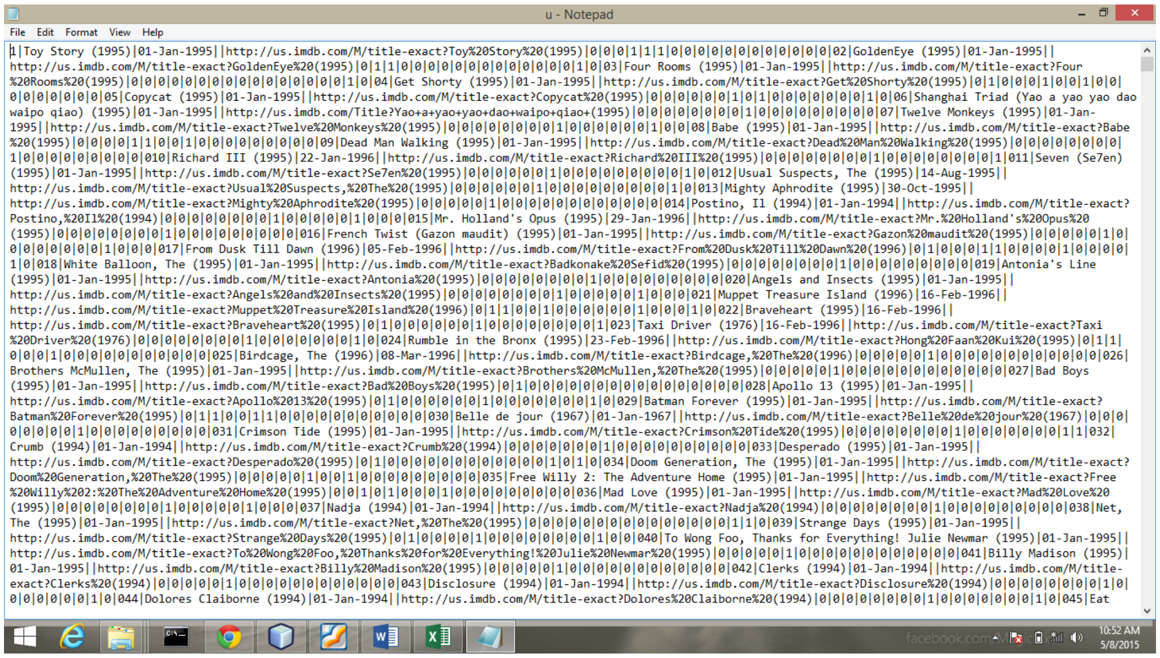
3.6 Database

Itemprofile
Id (int) Title (varchar) Author (varchar) Publisher (varchar) Genre (varchar) Abstract (varchar) Year (varchar) Filename (varchar) Price (varchar)

CHAPTER 4

Work Done Till Now

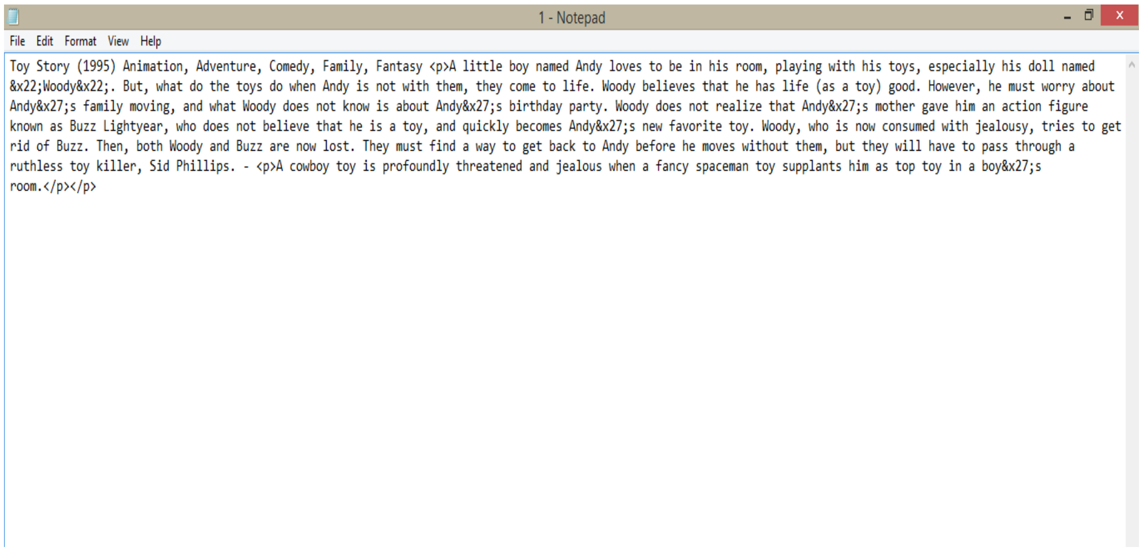
- The data set has been collected. The collection of data is done from movie lens.it contains rating of the user on atleast 20 items(movies) and set of file each represented by their movie id having description aboutthem like name, year of release, genre,links,description etc. The raw data having movie details is :



```
1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|02|GoldenEye (1995)|01-Jan-1995||
http://us.imdb.com/M/title-exact?GoldenEye%20(1995)|0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|03|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four
%20Rooms%20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|04|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)|0|1|0|0|0|1|0|0|1|0|0|
0|0|0|0|0|0|0|05|Copycat (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Copycat%20(1995)|0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|0|1|0|06|Shanghai Triad (Yao a yao yao dao
waipo qiao) (1995)|01-Jan-1995||http://us.imdb.com/Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|07|Twelve Monkeys (1995)|01-Jan-
1995||http://us.imdb.com/M/title-exact?Twelve%20Monkeys%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|1|0|0|08|Babe (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Babe
%20(1995)|0|0|0|0|1|1|0|0|1|0|0|0|0|0|0|0|0|0|09|Dead Man Walking (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Dead%20Man%20Walking%20(1995)|0|0|0|0|0|0|0|0|
1|0|0|0|0|0|0|0|0|0|010|Richard III (1995)|22-Jan-1996||http://us.imdb.com/M/title-exact?Richard%20III%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|1|011|Seven (Se7en)
(1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Se7en%20(1995)|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|1|0|012|Usual Suspects, The (1995)|14-Aug-1995||
http://us.imdb.com/M/title-exact?Usual%20Suspects,%20The%20(1995)|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|1|0|013|Mighty Aphrodite (1995)|30-Oct-1995||
http://us.imdb.com/M/title-exact?Mighty%20Aphrodite%20(1995)|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|014|Postino, Il (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?
Postino,%20Il%20(1994)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0|015|Mr. Holland's Opus (1995)|29-Jan-1996||http://us.imdb.com/M/title-exact?Mr.%20Holland's%20Opus%20
(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|016|French Twist (Gazon maudit) (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Gazon%20maudit%20(1995)|0|0|0|0|0|1|0|
0|0|0|0|0|0|0|1|0|0|0|017|From Dusk Till Dawn (1996)|05-Feb-1996||http://us.imdb.com/M/title-exact?From%20Dusk%20Till%20Dawn%20(1996)|0|1|0|0|0|1|1|0|0|0|0|1|0|0|0|0|
1|0|018|White Balloon, The (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Badkonake%20Sefid%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|019|Antonia's Line
(1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Antonia%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|020|Angels and Insects (1995)|01-Jan-1995||
http://us.imdb.com/M/title-exact?Angels%20and%20Insects%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0|021|Muppet Treasure Island (1996)|16-Feb-1996||
http://us.imdb.com/M/title-exact?Muppet%20Treasure%20Island%20(1996)|0|1|1|0|0|1|0|0|0|0|0|0|1|0|0|0|1|0|022|Braveheart (1995)|16-Feb-1996||
http://us.imdb.com/M/title-exact?Braveheart%20(1995)|0|1|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|1|023|Taxi Driver (1976)|16-Feb-1996||http://us.imdb.com/M/title-exact?Taxi
%20Driver%20(1976)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|1|0|024|Rumble in the Bronx (1995)|23-Feb-1996||http://us.imdb.com/M/title-exact?Hong%20Faan%20Kui%20(1995)|0|1|1|
0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|025|Birdcage, The (1996)|08-Mar-1996||http://us.imdb.com/M/title-exact?Birdcage,%20The%20(1996)|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|026|
Brothers McMullen, The (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Brothers%20McMullen,%20The%20(1995)|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|027|Bad Boys
(1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Bad%20Boys%20(1995)|0|1|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|028|Apollo 13 (1995)|01-Jan-1995||
http://us.imdb.com/M/title-exact?Apollo%2013%20(1995)|0|1|0|0|0|0|0|0|1|0|0|0|0|0|0|0|1|0|029|Batman Forever (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?
Batman%20Forever%20(1995)|0|1|1|0|0|1|1|0|0|0|0|0|0|0|0|0|0|0|030|Belle de jour (1967)|01-Jan-1967||http://us.imdb.com/M/title-exact?Belle%20de%20jour%20(1967)|0|0|0|
0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|031|Crimson Tide (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Crimson%20Tide%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|1|1|032|
Crumb (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?Crumb%20(1994)|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|033|Desperado (1995)|01-Jan-1995||
http://us.imdb.com/M/title-exact?Desperado%20(1995)|0|1|0|0|0|0|0|0|0|0|0|0|0|0|1|0|1|0|034|Doom Generation, The (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?
Doom%20Generation,%20The%20(1995)|0|0|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|035|Free Willy 2: The Adventure Home (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Free
%20Willy%202:%20The%20Adventure%20Home%20(1995)|0|0|1|0|1|0|0|0|1|0|0|0|0|0|0|0|0|0|036|Mad Love (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Mad%20Love%20
(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0|037|Nadja (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?Nadja%20(1994)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|038|Net,
The (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Net,%20The%20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|1|0|039|Strange Days (1995)|01-Jan-1995||
http://us.imdb.com/M/title-exact?Strange%20Days%20(1995)|0|1|0|0|0|1|0|0|0|0|0|0|0|0|0|1|0|0|040|To Wong Foo, Thanks for Everything! Julie Newmar (1995)|01-Jan-1995||
http://us.imdb.com/M/title-exact?To%20Wong%20Foo,%20Thanks%20for%20Everything!%20Julie%20Newmar%20(1995)|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|041|Billy Madison (1995)|
01-Jan-1995||http://us.imdb.com/M/title-exact?Billy%20Madison%20(1995)|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|042|Clerks (1994)|01-Jan-1994||http://us.imdb.com/M/title-
exact?Clerks%20(1994)|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|043|Disclosure (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?Disclosure%20(1994)|0|0|0|0|0|0|0|0|1|0|
0|0|0|0|0|0|1|0|044|Dolores Claiborne (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?Dolores%20Claiborne%20(1994)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|1|0|045|Eat
```

Figure movie details in one file

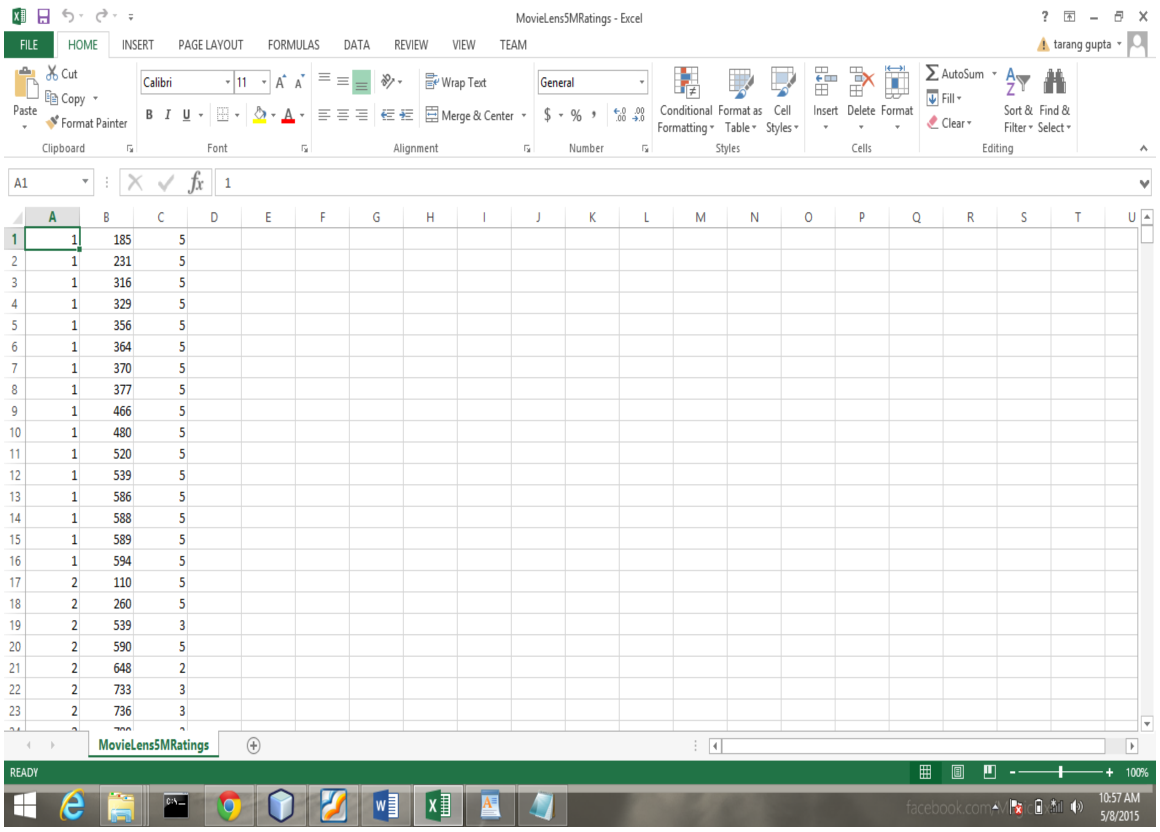
Sample of individual file with movie details:



```
Toy Story (1995) Animation, Adventure, Comedy, Family, Fantasy <p>A little boy named Andy loves to be in his room, playing with his toys, especially his doll named
&x22;Woody&x22;. But, what do the toys do when Andy is not with them, they come to life. Woody believes that he has life (as a toy) good. However, he must worry about
Andy&x27;s family moving, and what Woody does not know is about Andy&x27;s birthday party. Woody does not realize that Andy&x27;s mother gave him an action figure
known as Buzz Lightyear, who does not believe that he is a toy, and quickly becomes Andy&x27;s new favorite toy. Woody, who is now consumed with jealousy, tries to get
rid of Buzz. Then, both Woody and Buzz are now lost. They must find a way to get back to Andy before he moves without them, but they will have to pass through a
ruthless toy killer, Sid Phillips. - <p>A cowboy toy is profoundly threatened and jealous when a fancy spaceman toy supplants him as top toy in a boy&x27;s
room.</p></p>
```

Figure 9Sample movie file

Movielens5MRating file: have user id, item id and rating

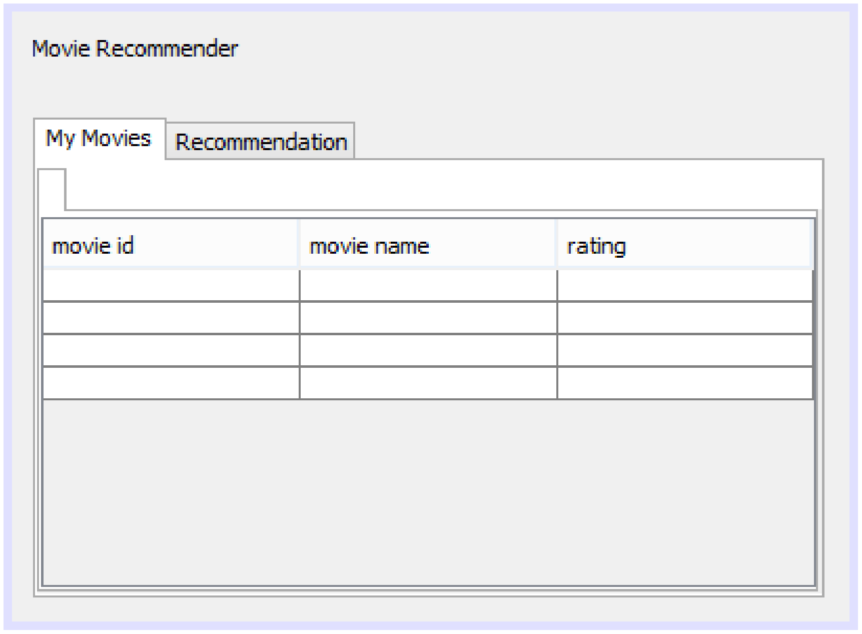


	A	B	C
1	1	185	5
2	1	231	5
3	1	316	5
4	1	329	5
5	1	356	5
6	1	364	5
7	1	370	5
8	1	377	5
9	1	466	5
10	1	480	5
11	1	520	5
12	1	539	5
13	1	586	5
14	1	588	5
15	1	589	5
16	1	594	5
17	2	110	5
18	2	260	5
19	2	539	3
20	2	590	5
21	2	648	2
22	2	733	3
23	2	736	3

Figure 10Sample user rating

- The steps to convert data set in document vector which then calculates the tfidf and finally the cosine similarity are:
    - An object of the class index is made and its function index is called through it. The class index creates lucene index from files. It generates tokens, terms and their frequencies and store them in the Lucene Index.
    - Next a Vector Generator class object is called which generate Document Vectors from Lucene Index and get the document vectors which is stored in class doc vector and all the terms in the index.
    - Index opener class gets the lucene index Reader and returns the total number of documents in the index.
    - Finally the cosine similarity class is called which calculate the similatiy between two DocVector and displays the result.
    - A mysql database is made in which the top 5 recommendation for a particular item is saved that has been calculated in previous steps so that the retrieval is easy.
    - The same is done to prepare the profile vector of user and the the recommendation are calculated according to the user preferenes in sorted order showing the id of the product recommended.
    - The Lucene libraries used in this are lucene-core-4.10.2.jar , lucene-ananlyzer-common-4.10.2.jar, common-math-2.0.jar.

- The search engine is made in which the user can enter the query string the simultaneously user can see the results also in the dropdown menu. User can view all the recommended items based on the entered string and can buy it. This has been done using Ajax. Ajax helps in implementing auto-search feature and hence enhances the user interface. The search engine is as shown:



movie id	movie name	rating

Figure 11interface

CHAPTER 5

Conclusion and Future Work

While confronted with more and more documents we need tools to efficiently find information we are interested in. Apache Lucene is such a tool. It provides us with a foundation to build a information retrieval system for our documents. While Lucene is capable of creating indexes our document collections and allow us to search this index it does not help us with the task of information extraction and therefore can only be one building block of an retrieval system. In the process of building a search index, I consider the task of information extraction the most complex, divers and important.

Future work for this can be to: Take preference from user at rum time and show the updated preferences at run time. The limitation of this method is that no content-based recommendation system can give good recommendations if the content does not contain enough information to distinguish items the user likes from items the user doesn’t like. In recommending some items, e.g., jokes or poems, there often isn’t enough information in the word frequency to model the user’s interests. While it would be possible to tell a lawyer joke from a chicken joke based upon word frequencies, it would be difficult to distinguish a funny lawyer joke from other lawyer jokes. As a consequence, other recommendation technologies, such as collaborative recommenders, should be used in such situations. In this way the shortcomings can be overcome.

References

[1] Songjie Gong, Learning User Interest Model for Content-based Filtering in Personalized Recommendation System, International Journal of Digital Content Technology and its Applications (JDCTA), Volume 6, Number 11, June 2012

[2] Peretz Shoval, Veronica Maidel, Bracha Shapira, AN ONTOLOGY- CONTENT-BASED FILTERING METHOD, International Journal "Information Theories & Applications" Vol.15 , 2008

[3] HE Weihong, CAO Yi, An E-Commerce Recommender System Based on Content-Based Filtering, Wuhan University Journal of Natural Sciences, Volume 11 No.5, 2006.

[4] CHHAVI RANA, SANJAY KUMAR JAIN, Building a Book Recommender system using time based content filtering, Issue 2, Volume 11, February 2012

[5] Christian Häusler, Topics in Business Information Technology , Methods for Determining the Similarity of Documents.

[6] Khairil Imran Ghauth, Nor Aniza Abdullah, Learning materials recommendation using good learners' ratings and content-based filtering, Published online on 9 March 2010

[7] en.wikipedia.org/wiki/Recommender_system

[8] www.ibm.com/developerworks/library/os-recommender1/

[9] http://hcil2.cs.umd.edu/trs/96-10/node7.html

[10] http://recommender-systems.org

[11] http://scholar.google.co.in/

[12] http://nlp.stanford.edu/IR-book/html/htmledition/xml-retrieval-1.html

[13] http://infolab.stanford.edu/~ullman/mmds/ch9.pdf

[14] http://comminfo.rutgers.edu/~aspoerri/InfoCrystal/Ch_2.html

[15] https://lucene.apache.org/core/