الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFICRESEARCH

YAHIA FARES UNIVERSITY OF MEDEA FACULTY OFSCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTERSCIENCE



Graduation Project

Submitted for graduation in Computer science master
Specialty: Systems Engineering and Web Technology

TITLE

Movies recommendation based on user interaction with social network

Proposed and directed by:
Mrs.FATIHA SADOUKI

Prepared and presented by:
Mr. AYOUB AGADI
Mr. FEKHAR MEHDI

Academicyear 2019 - 2020

ACKNOWLEDGMENT

We are infinitely thankful to our God for giving us power and faith toedit this work.

Firstly, we would like to express our sincere gratitude to our advisorand supervisor (Mrs. F. SADOUKI) for the continuous support thathe gave us, for his patience, motivation, and immense knowledge. Hisguidance helped us during this project and writing of this report. Wecould not have imagined having a better advisor and supervisor for ourthesis.

Besides our advisor, we would like to thank our juries (Mr.K. BOUDJEBOUR and Mr. T.RAFAA).

Finally, we must express our profound gratitude to our parents forproviding us with unfailing support and continuous encouragementthroughout our years of study and through the process of researchingand writing this project. This accomplishment would not have beenpossible without them. Thank you.

DEDICATION

I dedicate this work:

To The almighty God, who's without guidance,

love and protection, It would be impossible.

To My mother« Bahiya », my friend so dear throughout my life,

you're always near, a tender smile, to guide my way, you're the sunshine to light my days. To my wonderful Father « Boudjemaa».

To my sisters « Amira and Manel», to my great brother « Riadh »,

for being friendly and helpful.

To all my family (Agadi - Choukri) and people around us,

to my friends (Malak – Lhadi – Tarak – Badiss – Zinou – Chamssou – Maxi – Islem – Bouyahia – Fanicho - Aniss).

To la clique Hamo and la clique L7ouma (Rahim ra3da - Atchi wtrig larab3a and Colombo Byna w Adel Bensalem w Hamada Khouya)

To (Khetkhet and sia breezy, …),

To my partner during this project Mehdi Fekhar, who have been unselfishly extending their efforts and support.

Hope this work gets the summet of your wishes about me.

(Ayoub Agadi) ***thanks to all of you***

DEDICATION

I dedicate this work:

To The almighty God, who's without his guidance,

love and protection,

It would be impossible .

To My mother that was my main insperation,

you are always near to me, always nice,

you are my sunshine.

To my wonderful Father that, without him a wouldnt be here,

to both of my brotheers and my sweet sister.

To all my familly and people around us,

who inspired and gave me the perseverance,

and determination in order to accomplish this work.

To my soulmate Madina Benamara

To my partner during this project Ayoub Agadi,

To my friends,

who have been unselfishly extending their efforts and support.

Hope this work gets the summet of your wishes about me.

Mehdi Fekhar

ABSTRACT

Recommendation systems (RS) play an important role in artificial intelligence and help make a system capable of choosing what is right for users. The personalization of information is the key that has made recommendation systems at the highest level, for this we have realized a system that uses the user's personal information to process the case of a new user and make a personalized recommendation.

In this project a new idea for recommending movies has been created, this system is based on user interaction with tweeter, it is designed based on the similarity between users tweets and movies summaries, by finding this type of resemblance we can discover personal interests for each user and we can determine whether the user like a particular item or not only by computing the similarity. After collecting those information and decide which movies a new user like, the recommendation part comes a head by implementing matrix factorization approach for recommendation systems, basically when a new user who have no interaction with any item, the quality of recommendation will be at its worst. But by discovering the movies that user like based on the similarity calculation the system will be having information’s about user personal interests and then the quality of the recommendation could be better by using user interaction with tweeter.

Key words : recomandation system , personalization information , enrishment , item

RESUME

Les systèmes de recommandation (RS) jouent un rôle important dans l'intelligence artificielle et contribuent à rendre un système capable de choisir ce qui convient aux utilisateurs. La personnalisation des informations est la clé qui a fait des systèmes de recommandation au plus haut niveau, pour cela nous avons réalisé un système qui utilise les informations personnelles de l'utilisateur pour traiter le cas d'un nouvel utilisateur et faire une recommandation personnalisée. Dans ce projet, une nouvelle idée de recommandation de films a été créée, ce système est basé sur l'interaction de l'utilisateur avec le tweeter, il est conçu sur la base de la similitude entre les tweets des utilisateurs et les résumés de films, en trouvant ce type de ressemblance, nous pouvons découvrir des intérêts personnels pour chaque l'utilisateur et nous pouvons déterminer si l'utilisateur aime un élément particulier ou pas uniquement en calculant la similitude. Après avoir collecté ces informations et décidé quels films un nouvel utilisateur aime, la partie recommandation vient à la tête en mettant en œuvre une approche de factorisation matricielle pour les systèmes de recommandation, essentiellement lorsqu'un nouvel utilisateur qui n'a aucune interaction avec un élément, la qualité de la recommandation sera à son pire. Mais en découvrant les films que l'utilisateur aime sur la base du calcul de similarité, le système disposera d'informations sur les intérêts personnels de l'utilisateur et la qualité de la recommandation pourrait être meilleure en utilisant l'interaction de l'utilisateur avec le tweeter.

Mots clés : recommandation système , les informations personnels , enrichissements , les items

ملخص

تلعب أنظمة التوصيات (RS) دورًا مهمًا في الذكاء الاصطناعي وتساعد في جعل النظام قادرًا على اختيار ما هو مناسب للمستخدمين. يعد تخصيص المعلومات هو المفتاح الذي جعل أنظمة التوصية على أعلى مستوى، ولهذا حققنا نظامًا يستخدم المعلومات الشخصية للمستخدم لمعالجة حالة مستخدم جديد وتقديم توصية مخصصة.

في هذا المشروع، تم إنشاء فكرة جديدة للتوصية بالأفلام، وهذا النظام يعتمد على تفاعل المستخدم مع tweeter، وهو مصمم على أساس التشابه بين تغريدات المستخدمين وملخصات الأفلام.من خلال إيجاد هذا النوع من التشابه يمكننا اكتشاف الاهتمامات الشخصية لكل مستخدم ويمكننا تحديد ما إذا كان المستخدم يحب عنصرًا معينًا أم لا فقط عن طريق حساب التشابه. وبعد جمع تلك المعلومات وتحديد الأفلام التي يرغب فيها مستخدم جديد، يأتي جزء التوصية من خلال تنفيذ نهج عامل المصفوفة لأنظمة التوصية، بشكل أساسي عندما مستخدم جديد ليس له أي تفاعل مع أي عنصر، ستكون جودة التوصية في أسوأ حالاتها، ولكن من خلال اكتشاف الأفلام التي يحبها المستخدم بناءً على حساب التشابه، سيحصل النظام على معلومات حول اهتمامات المستخدم الشخصية ومن ثم يمكن أن تكون جودة التوصية أفضل باستخدام تفاعل المستخدم معtweeter

الكلمات المفتاحية

أنظمة التوصيات , المعلومات الشخصية للمستخدم , عنصر

CONTENTS TABLE

ABSTRACT .....
RESUME .....
ملخص .....
CONTENTS TABLE .....
LIST OF FIGURES .....
LIST OF TABLES .....
GENERAL INTRODUCTION 1
CHAPTER I: RECOMMENDATION SYSTEM 3
I.1 Introduction..... 3
I.2 Definition Recommender Systems: 3
I.3 Basic Models of Recommender Systems: 4
I.3.1 Content-based filtering: 5
I.3.2 Collaborative filtering:..... 6
➢ I.3.2.1 Memory based techniques:..... 7
A.Matrix Factorization: 9
➢ I.3.2.2 Model-based techniques: 10
A. Association rule: 11
B.Clustering: 11
C. Decision tree: 12
D. Artificial Neural network: 12
E. Link analysis 12
F. Regression:..... 12
G. Bayesian Classifiers: 13
➢ I.3.2.3 Pros and Cons of collaborative filtering techniques:..... 13
A. Cold-start problem: 14
B. Data sparsity problem: 14
C. Scalability: 14
D. Synonymy:..... 14
➢ I.3.2.4 Example of collaborative systems: 15
I.3.3 Hybrid filtering:..... 15
➢ I.3.3.1 Weighted hybridization:..... 16
➢ I.3.3.2switching hybridization:..... 16
➢ I.3.3.3 Cascade hybridization:..... 17
➢ I.3.3.4 Mixed hybridization: 17
➢ I.3.3.5 Feature-combination: 17
➢ I.3.3.6 Feature-augmentation: 17

- I.3.3.7 Meta-level: 18
- I.3.4 Demographic Recommender Systems: 18
- I.4 Evaluation metrics for recommendation algorithms: 19
- I.5Conclusion: 21

CHAPTER II: User profiling and modeling 22

- II.1 Introduction: 22
- II.2 Collection of information: 22
  - II.2.1 Explicit feedback: 22
  - II.2.2 Implicit feedback: 23
  - II.2.3 Hybrid User Profiling: 23
- II.3 Techniques: 23
  - II.3.1 Profile extraction: 23
  - II.3.2 Profile Integration: 25
  - II.3.3 Interest Discovery: 25
- II.4 Representation of the user model: 25
  - II.4.1 Representation as vectors: 25
  - II.4.2 Representation in the form of hierarchical concepts: 26
  - II.4.3 Representation in the form of ontologies: 26
  - II.4.4 Representation in the form of graphs: 26
- II.5 Recommender system using social networks: 27
  - II.5.1 Educational recommender system using social networks interactions and linked data - BROAD-RSI: 27
  - II.5.2 Efficient User Profiling in Twitter Social Network Using Traditional Classifiers: 27
  - II.5.3 User Profile Extraction from Twitter for Personalized News Recommendation: .. 28
  - II.5.4 TRUPI: Twitter Recommendation Based on Users' Personal Interests: 28
  - II.5.5 Hashtag Recommendation Based on User Tweet and Hashtag Classification on Twitter: 29
  - II.5.6 Twitter User Modeling and Tweets Recommendation Based on Wikipedia Concept Graph: 30
  - II.5.7 A Personalized Tweet Recommendation Approach Based on Concept Graphs: ... 30
  - II.5.8 AMovie Recommendation based on Users' Tweets: 31
- II.6.Conclusion: 31

CHAPTER III:The proposed approach based on user interaction with tweeter 32

- III.1 Introduction: 32
- III.2 General Architecture: 32
  - III.2.1Data Collection: 33
  - III.2.2 Data preprocessing: 35
    - A. Keywords extraction: 36

➢ B. Similarity between movies and tweets: 37
III.2.3 Recommendation: 38
III.3 Implementation: 41
III.3.1 Extracting users' tweets: 41
III.3.2 Keyword Extraction: 41
III.4 Conclusion: 42
CHAPTER IV:Experimental evaluation of the proposed approach 43
IV.1 Introduction 43
IV.2 Evaluation framework 43
IV.2.1 Database used 43
IV.2.2The results of the prediction algorithms SVD and SGD: 44
IV.2.3 The impact of the enrichment based on user interaction with tweeter: 48
IV.3 Conclusion: 53
GENERAL CONCLUSION 54
REFERENCES 55
Annex A 63
Annex B 64
Annex C 66

LIST OF FIGURES:

Figure I.1:Image summarizes the content based filtering 06
Figure I.2: Exploring Collaborative Filtering on Implicit Data for Jo Recommendations.....07
Figure I.3: Collaborative filtering process.....07
Figure I.4: Amason book recommender interface. Sourced from: www.amason.com 15
Figure III.1: General Architecture 33
Figure III.2: Data Collection.....34
Figure III.3:Obtaining final dataset.....34
Figure III.4: Data preprocessing 36
Figure III.5: Recommendation.....38
Figure IV.1: RMSE against epoch with enrichment 44
Figure IV.2: RMSE against epoch without enrichment 45
Figure IV.3: MSE against epoch with enrichment.....45
Figure IV.4: MSE against epoch with enrichment.....45
Figure IV.5: RMSE without enrichment 46
Figure IV.6: RMSE with enrichment 46
Figure IV.7: MSE without enrichment.....47
Figure IV.8: MSE withenrichment.....47
Figure IV.9: RMSE,MSE with enrichment SVD and SGD 48
Figure IV.10: SGD with and without enrichment.....50
Figure IV.11: SVD with enrichment andwithout.....51
Figure IV.12:SVD and SGD with enrichment.....52
Figure IV.13:SVD at top K=10 users' ratings 53

LIST OF TABLES:

TableIV.1: A comparison of RMSE values of svd and sgd with and without enrichment.....48

TableIV.2 : Compare the precision recall f-measure of the SGD with enrichment and without enrichment 49

TableIV.3: Compare the precision recall f-measure of the SVD with enrichment and without enrichmen 50

Table IV.4:Compare the precision recall f-measure of the SVD with enrichment and without enrichmentin k=10 highest estimated ratings 52

GENERAL INTRODUCTION

Recommender systems provide users with personalized suggestions for products or services. They are becoming more and more important in the success of electronic commerce, and being utilized in various applications such as Amazon, YouTube, Netflix and Google news. Generally speaking, a recommendation system builds up items' profiles, and users' profiles based on their previous behavior recorded, then it makes a prediction on the rating given by certain user on certain item which he/she has not yet evaluated. Based on the prediction, the system makes recommendations. Various techniques for recommendation generation have been proposed and successfully deployed in commercial environments, among which collaborative filtering (CF) and content-based methods are most commonly used [1, 2].

Recommender systems collect information based on preferences of users on items (movies, news, videos, products etc.). There are two types of category of preferences as expressed by users in literature 1) Explicit feedback, 2) implicit feedback. Explicit feedbacks are those which users directly report their interest on product. For example, Netflix collects star ratings for movies and TiVo users indicate their preferences for TV shows by hitting thumbs-up/down buttons. Because explicit feedback is not always available, some recommenders infer user preferences from the more abundant implicit feedback, which indirectly reflects opinion through observing user behavior. Types of implicit feedback include purchase history, browsing history, search patterns, or even mouse clicks. For example, a user who purchased many books by the same author probably likes that author [3].

So there is a big problem we can find , when a new user came how we can recommend an item to him and this called a cold and state problem and our work will fix this problem

In this work we will be presenting a new model that use user interaction with tweeter in order to collect personal information and discover preference of a user on an item, we will be talking about recommendation system and different techniques and works made in this field of study, after that we will Detaille every part of the proposed model and to finish we will present the results in order to see the effect of the proposed model.

This project is divided in four chapters, the first chapter present the recommendation systems, the used techniques and the different evaluation metrics for recommendation systems; in the second chapter we will explain what is user profiling and the different techniques used to make user

profiling, as well we did clarify user modeling and the diverse aspects of user modeling, next we did mention works that use recommendation systems based on user interaction with social networks, in chapter three we will describe and detail how to get personalized data about every user and how to find the relation between user data(tweets) and the items that should be recommended only using user interaction with tweeter, after finding the relation between user tweets and the items a matrix factorization recommendation system is applied; in chapter four we will be evaluating the proposed approach and testing the effectiveness of the system by comparing the results of recommendation system and the prediction before using the collected information's for users and after collecting the information's.

CHAPTER I: RECOMMENDATION SYSTEM

I.1 Introduction:

The explosive growth in the amount of available digital informationand the number of visitors to the Internet have created apotential challenge of information overload which hinderstimely access to items of interest on the Internet. Informationretrieval systems, such as Google, DevilFinder and Altavistahave partially solved this problem but prioritization and personalization(where a system maps available content to user's interestsand preferences) of information were absent. This hasincreased the demand for recommender systems more than everbefore. Recommender systems are information filtering systemsthat deal with the problem of information overload [4] by filteringvital information fragment out of large amount of dynamicallygenerated information according to user′s preferences,interest, or observed behavior about item [5]. Recommender systemhas the ability to predict whether a particular user wouldprefer an item or not based on the user's profile.

I.2 Definition Recommender Systems:

Recommender Systems (RSs) are software tools and techniques providing suggestions for items to be of use to a user [6, 7, 8]. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read.

"Item" is the general term used to denote what the system recommends to users.

A RS normally focuses on a specific type of item (e.g., CDs, or news) and accordingly its design, its graphical user interface, and the core recommendation technique used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item[9].

RSs are primarily directed towards individuals who lack sufficient personal experience or competence to evaluate the potentially overwhelming number of alternative items that a Web site, for example, may offer [10]. A case in point is a book recommender system that assists users to select a book to read. In the popular Web site, Amazon.com, the site employs a RS to personalize the online store for each customer [11]. Since recommendations are usually personalized, different users or user groups receive diverse suggestions. In addition there are also

non-personalized recommendations. These are much simpler to generate and are normally featured in magazines or newspapers. Typical examples include the top ten selections of books, CDs etc. While they may be useful and effective in certain situations, these types of non-personalized recommendations are not typically addressed by RS research[9].

In their simplest form, personalized recommendations are offered as ranked lists of items. In performing this ranking, RSs try to predict what the most suitable products or services are, based on the user's preferences and constraints. In order to complete such a computational task, RSs collect from users their preferences, which are either explicitly expressed, e.g., as ratings for products, or are inferred by interpreting user actions. For instance, a RS may consider the navigation to a particular product page as an implicit sign of preference for the items shown on that page. RSs development initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine, daily decisions [12, 13]. For example it is common to rely on what one's peers recommend when selecting a book to read; employers count on recommendation letters in their recruiting decisions; and when selecting a movie to watch, individuals tend to read and rely on the movie reviews that a film critic has written and which appear in the newspaper they read[13].

In seeking to mimic this behavior, the first RSs applied algorithms to leverage recommendations produced by a community of users to deliver recommendations to an active user, i.e., a user looking for suggestions .The recommendations were for items that similar users (those with similar tastes) had liked. This approach is termed collaborative-filtering and its rationale is that if the active user agreed in the past with some users, then the other recommendations coming from these similar users should be relevant as well and of interest to the active user [9].

As e-commerce Web sites began to develop, a pressing need emerged for providing recommendations derived from filtering the whole range of available alternatives. Users were finding it very difficult to arrive at the most appropriate choices from the immense variety of items (products and services) that these Web sites were offering[9].

I.3 Basic Models of Recommender Systems:

The basic models for recommender systems work with two kinds of data, which are (i) the user-item interactions, such as ratings or buying behavior, and (ii) the attribute information about the

users and items such as textual profiles or relevant keywords. Methods that use the former are referred to as *collaborative filtering* methods, whereas methods that use the latter are referred to as *content-based recommender* methods. Note that content-based systems also use the ratings matrices in most cases, although the model is usually focused on the ratings of a single user rather than those of all users. In *knowledge-based* recommender systems, the recommendations are based on explicitly specified *user requirements*. Instead of using historical rating or buying data, external knowledge bases and constraints are used to create the recommendation. Some recommender systems combine these different aspects to create *hybrid* systems. Hybrid systems can combine the strengths of various types of recommender systems to create techniques that can perform more robustly in a wide variety of settings. In the following, we will discuss these basic models[9].

I.3.1 Content-based filtering:

Content-based technique is a domain-dependent algorithm and it emphasizes more on the analysis of the attributes of items in order to generate predictions. When documents such as web pages, publications and news are to be recommended, content-based filtering technique is the most successful.

In content-based filtering technique, recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past [14,15]. Items that are mostly related to the positively rated items are recommended to the user. CBF uses different types of models to find similarity between documents in order to generate meaningful recommendations. It could use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier [16], Decision Trees [17] or Neural Networks to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques. Content-based filtering technique does not need the profile of other users since they do not influence recommendation. Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time. The major disadvantage of this technique is the need to have an in depth knowledge and description of the features of the items in the profile[18].

CB filtering techniques overcome the challenges of CF. They have the ability to recommend new items even if there are no ratings provided by users. So even if the database does not contain user

preferences, recommendation accuracy is not affected. Also, if the user preferences change, it has the capacity to adjust its recommendations in a short span of time. They can manage situations where different users do not share the same items, but only identical items according to their intrinsic features. Users can get recommendations without sharing their Profile, and this ensures privacy [19]. CBF technique can also provide explanations on how recommendations are generated to users. However, the techniques suffer from various problems as discussed in the literature [20]. Content based filtering techniques are dependent on items' metadata. That is, they require rich description of items and very well organized user profile before recommendation can be made to users. This is called limited content analysis. So, the effectiveness of CBF depends on the availability of descriptive data. Content overspecialization [21] is another serious problem of CBF technique. Users are restricted to getting recommendations similar to items already defined in their profiles.

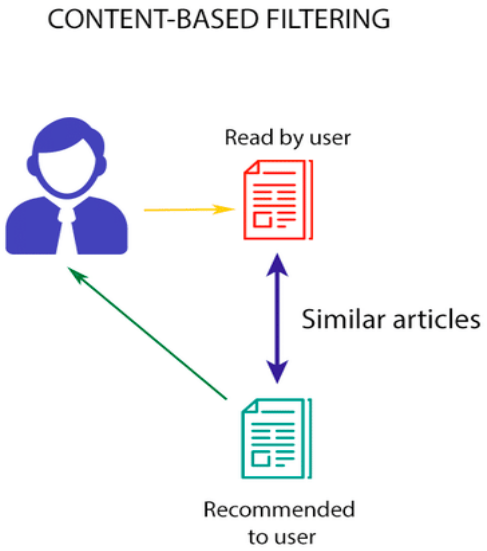


Figure I.1: Image summarizes the content based filtering

I.3.2 Collaborative filtering:

Collaborative filtering is a domain-independent prediction technique for content that cannot easily and adequately be described by metadata such as movies and music .Collaborative filtering technique works by building a database (user-item matrix) of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations [22]. Such users build a group called neighborhood. A user gets recommendations to those items that he has not rated before but that were already positively rated by users in his neighborhood. Recommendations that are produced

by CF can be of either prediction or recommendation. Prediction is a numerical value, Rij, expressing the predicted score of item j for the user i, while Recommendation is a list of top N items that the user will like the most as shown in Figure 1.3. The technique of collaborative filtering can be divided into two categories: memory-based and model-based [15,23].

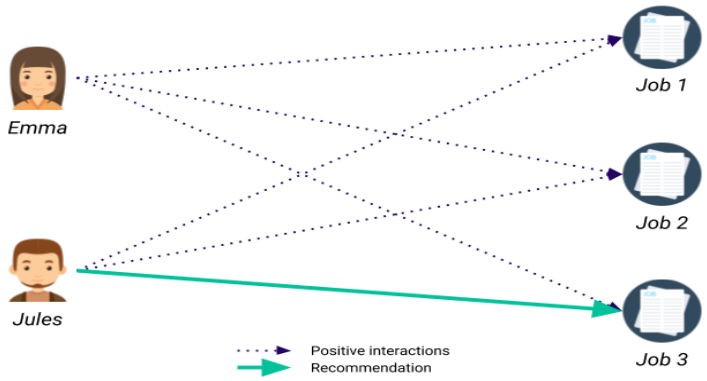


Figure I.2:Exploring Collaborative Filtering on Implicit Data for Jo Recommendations

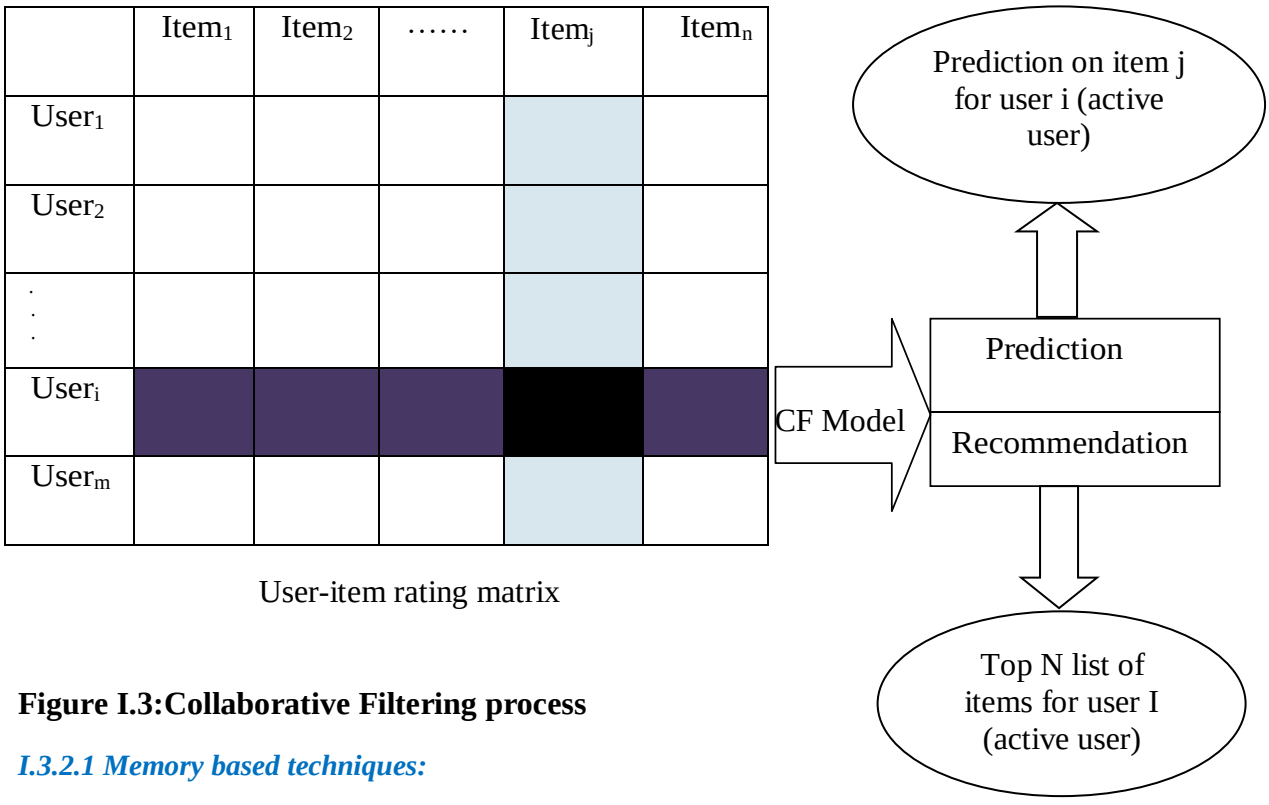


Figure I.3:Collaborative Filtering process

I.3.2.1 Memory based techniques:

The items that were already rated by the user before play a relevant role in searching for a neighbor that shares appreciation with him [24,25]. Once a neighbor of a user is found, different

algorithms can be used to combine the preferences of neighbors to generate recommendations. Due to the effectiveness of these techniques, they have achieved widespread success in real life applications. Memory-based CF can be achieved in two ways through user-based and item-based techniques. User based collaborative filtering technique calculates similarity between users by comparing their ratings on the same item, and it then computes the predicted rating for an item by the active user as a weighted average of the ratings of the item by users similar to the active user where weights are the similarities of these users with the target item. Item-based filtering techniques compute predictions using the similarity between items and not the similarity between users. It builds a model of item similarities by retrieving all items rated by an active user from the user-item matrix, it determines how similar the retrieved items are to the target item, then it selects the *k* most similar items and their corresponding similarities are also determined. Prediction is made by taking a weighted average of the active users rating on the similar items *k*. Several types of similarity measures are used to compute similarity between item/user. The two most popular similarity measures are correlation-based and cosine-based. Pearson correlation coefficient is used to measure the extent to which two variables linearly relate with each other and is defined as [26,27]

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i}-\overline{r_a})(r_{u,i}-\overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i}-\overline{r_a})^2}\sqrt{\sum_{i=1}^{n}(r_{u,i}-\overline{r_u})^2}} \quad \text{(I.1)}$$

Fromthe above equation, $s(a,u)$denotes the similarity between two users *a*and*u*, $r_{a,i}$is the rating given to item *I* by user *a*,$\overline{r_a}$ is the mean rating given by user a while *n* is the total number of items in the user-item space. Also, prediction for an item is made from the weighted combination ofthe selected neighbors' ratings, which is computed as the weighted deviation from the neighbors' mean. The general prediction formula is:

$$p(a,i) = \overline{r_a} + \frac{\sum_{i=1}^{n}(r_{a,i}-\overline{r_u})\times s(a,u)}{\sum_{i=1}^{n} s(a,u)} \quad \text{(I.2)}$$

Cosine similarity is different from Pearson-based measure in that it is a vector-space model which is based on linear algebra rather that statistical approach. It measures the similarity between two n-dimensional vectors based on the angle between them. Cosine-based measure is widely used in the fields of information retrieval and texts mining to compare two text documents, in this case, documents are represented as vectors of terms. The similarity between two items u and v can be defined as [20,22,27] follows:

$$s(\vec{u},\vec{v})=\frac{\overrightarrow{u}.\overrightarrow{v}}{|\vec{u}|*|\vec{v}|}=\frac{\sum_i r_{u,i}r_{v,i}}{\sqrt{\sum_i r^2{}_{u,i}}\times\sqrt{\sum_i r^2{}_{v,i}}} \quad \text{(I.3)}$$

Similarity measure is also referred to as similarity metric, and they are methods used to calculate the scores that express how similar users or items are to each other. These scores can then be used as the foundation of user- or item-based recommendation generation. Depending on the context of use, similarity metrics can also be referred to as correlation metrics or distance metrics [20].

A.Matrix Factorization:

Matrix factorization can be explained on customer data where *m* customers buy *n* products. The authors explain collaborative filtering in a comprehensive language. For demonstrative purposes, the author of this article [28]demonstrates the concept on a specific case.

Let matrix $\boldsymbol{R_{m*n}}$ represent the ratings on movies assigned by each user, also called the utility matrix. Specifically, the value $r_{i,j}$represents the rating of user *i* assigned to movie *j*. However, the individual's preference is determined by *k* factors. For example, the user's age, sex, income, education, etc. are likely to affect the user's behavior. Accordingly, the individual's rating of a movie ($r_{i,j}$) is determined by some weighted combinations of the hidden factors[28].

In practice, customer's behavior can be characterized by a k-dimensional vector with much lower dimensions than the original matrix RR with m∗nm∗n dimensions.

The vector's components, also called the latent factors, represent the weight of each factor. For example, given a vector v2=[0.2,0.8] it can be hypothesized that there are only two (unknown) latent factors with subsequent weights describing the rating (behavior).

Matrix factorization is an effective CF technique because it benefits from the properties of linear algebra. Specifically, consider matrix RR as a record of various elements. As it is possible to decompose any integer into the product of its prime factor, matrix factorization also enables humans to explore information about matrices and their functional properties an array of elements [28].

- Singular Value Decomposition (SVD):

SVD decomposes any matrix into **singular vectors** and **singular values**. If the reader has previous experience with machine learning, particularly with dimensionality reduction, they

would find traditional use of SVD in Principal Component Analysis (PCA). Simply put, SVD is equivalent to PCA after mean centering, i.e. shifting all data points so that their mean is on the origin[28].

Formally, SVD is decomposition of a matrix R into the product of three matrices:

$$\boldsymbol{R_{m*n}=U_{m*n}D_{m*n}V^{t}{}_{n*n}}$$

- Stochastic Gradient Descent(SGD)

Although the traditional SGD algorithm is a sequential approach when updating the factor matrices and other parameters, for the independent blocks in the rating matrix, the correspondingfactor vectors of these independent blocks are independent of each other too. DSGD just uses this feature to achieve the parallelization of SGD in DCE. It grids rating matrix into d×d blocks, and two blocks are mutually independent if they are both not in the same line and column[29]. The following equation explains how to get the prediction matrix

Relation between original matrix and the vector p,q [25]:

$$r_{ij}=p^{t}{}_{i}q_{j}=\sum_{k=1}^{k}p_{ik}\,q_{kj} \quad \text{(I.4)}$$

The squared error:

$$e_{ij}^{2}=(r_{ij}-\sum_{k=1}^{k}p_{ik}\,q_{kj})^{2}+\frac{3}{2}\sum_{k=1}^{k}(\|p\|^{2}+\|Q\|^{2}) \quad \text{(I.5)}$$

The update role equation is:

$$p'_{ik}=p_{ik}+\alpha\frac{\delta}{\delta_{pk}}e_{ij}^{2}=p_{ik}+\alpha(2e_{ij}q_{kj}-\beta p_{ik}) \quad \text{(I.6)}$$

$$q'_{ik}=q_{ik}+\alpha\frac{\delta}{\delta_{qk}}e_{ij}^{2}=q_{ik}+\alpha(2e_{ij}p_{kj}-\beta q_{ik}) \quad \text{(I.7)}$$

The error converges equation is:

$$E=\sum_{(u_i,d_{j,r_{ij}})}\epsilon T^{e_{ij}}=\sum_{(u_i,d_{j,r_{ij}})}\epsilon T^{(r_{ij}-\sum_{k=1}^{k}p_{ik}\,q_{kj})^{2}} \quad \text{(I.8)}$$

I.3.2.2 Model-based techniques:

This technique employs the previous ratings to learn a model in order to improve the performance of Collaborative filtering Technique. The model building process can be done using

machine learning or data mining techniques. These techniques can quickly recommend a set of items for the fact that they use pre-computed model and they have proved to produce recommendation results that are similar to neighborhood-based recommender techniques. Examples of these techniques include Dimensionality Reduction technique such as Singular Value Decomposition (SVD), Matrix Completion Technique, Latent Semantic methods, and Regression and Clustering. Model-based techniques analyze the user-item matrix to identify relations between items; they use these relations to compare the list of top-N recommendations. Model based techniques resolve the sparsity problems associated with recommendation systems. The use of learning algorithms has also changed the manner of recommendations from recommending what to consume by users to recommending when to actually consume a product. It is therefore very important to examine other learning algorithms used in model-based recommender systems[30]:

A. Association rule: Association rules mining algorithms [31] extract rules that predict the occurrence of an item based on the presence of other items in a transaction. For instance, given a set of transactions, where each transaction is a set of items, an association rule applies the form A →B, where A and B are two sets of items [32]. Association rules can form a very compact representation of preference data that may improve efficiency of storage as well as performance. Also, the effectiveness of association rule for uncovering patterns and driving personalized marketing decisions has been known for sometimes [5]. However, there is a clear relation between this method and the goal of a Recommendation System but they have not become mainstream.

B.Clustering: Clustering techniques have been applied in different domains such as, pattern recognition, image processing,statistical data analysis and knowledge discovery [33]. Clustering algorithm tries to partition a set of data into a set of sub-clusters in order to discover meaningful groups that exist within them [34]. Once clusters have been formed, the opinions of other users in a cluster can be averaged and used to make recommendations for individual users. A good clustering method will produce high quality clusters in which the intra-cluster similarity is high, while the inter-cluster similarity is low. In some clustering approaches, a user can have partial participation in different clusters, and recommendations are then based on the average across the clusters of participation which is weighted by degree of participation [35]. K-means and Self-Organizing Map (SOM) are the most commonly used among the different clustering methods. K-

means takes an input parameter, and then partitions a set of n items into K clusters [36]. The Self-Organizing Map (SOM) is a method for an unsupervised learning, based on artificial neurons clustering technique [37]. Clustering techniques can be used to reduce the candidate set in collaborative based algorithms.

C. Decision tree: Decision tree is based on the methodology of tree graphs which is constructed by analyzing a set of training examples for which the class labels are known. They are then applied to classify previously unseen examples. If trained on very high quality data, they have the ability to make very accurate predictions [38]. Decision trees are more interpretable than other classifier such as Support Vector machine (SVM) and Neural Networks because they combine simple questions about data in an understandable manner. Decision trees are also flexible in handling items with mixture of real-valued and categorical features as well as items that have some specific missing features.

D. Artificial Neural network: ANN is a structure of many connected neurons (nodes) which are arranged in layers in systematic ways. The connections between neurons have weights associated with them depending on the amount of influence one neuron has on another. There are some advantages in using neural networks in some special problem situations. For example, due to the fact that it contains many neurons and also assigned weight to each connection, an artificial neural network is quite robust with respect to noisy and erroneous data sets [39]. ANN has the ability of estimating nonlinear functions and capturing complex relationships in data sets also, they can be efficient and even operate if part of the network fails. The major disadvantage is that it is hard to come up with the ideal network topology for a given problem and once the topology is decided this will act as a lower bound for the classification error.

E. Link analysis: Link Analysis is the process of building up networks of interconnected objects in order to explore pattern and trends [40]. It has presented great potentials in improving the accomplishment of web search. Link analysis consists of Page Rank and HITS algorithms. Most link analysis algorithms handle a web page as a single node in the web graph [41].

F. Regression: Regression analysis is used when two or more variables are thought to be systematically connected by a linear relationship. It is a powerful and diversity process for analyzing associative relationships between dependent variable and one or more independent variables. Uses of regression contain curve fitting, prediction, and testing systematic hypotheses

about relationships between variables. The curve can be useful to identify a trend within dataset, whether it is linear, parabolic, or of some other forms[9].

G. Bayesian Classifiers: They are probabilistic framework for solving classification problems which is based on the definition of conditional probability and Bayes theorem. Bayesian classifiers [12] consider each attribute and class label as random variables. Given a record of N features (A1, A2, . . ., AN), the goal of the classifier is to predict class Ck by finding the value of Ck that maximizes the posterior probability of the class given the data P(Ck|A1, A2, . . ., AN) by applying Bayes' theorem, P(Ck|A1, A2, . . ., AN) ∞P(A1, A2, . . ., AN|Ck)P(Ck). The most commonly used Bayesian classifier is known as the Naive Bayes Classifier. In order to estimate the conditional probability, P(A1, A2, . . ., AN|Ck), a Naive Bayes Classifier assumes the probabilistic independence of the attributes that is, the presence or absence of a particular attribute is unrelated to the presence or absence of any other. This assumption leads to P(A1, A2, . . ., AN|Ck) = P(A1|Ck)P(A2|Ck). . . P(AN|Ck). The main benefits of Naive Bayes classifiers are that they are robust to isolated noise points and irrelevant attributes, and they handle missing values by ignoring the instance during probability estimate calculations. However, the independence assumption may not hold for some attributes as they might be correlated. In this case, the usual approach is to use Bayesian Networks. Bayesian classifiers may prove practical for environments in which knowledge of user preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which users preference models must be updated rapidly or frequently. It is also successful in model-based recommendation systems because it is often used to derive a model for content-based recommendation systems[9].

I.3.2.3 Pros and Cons of collaborative filtering techniques:

Collaborative Filtering has some major advantages over CBF in that it can perform in domains where there is not much content associated with items and where content is difficult for a computer system to analyze (such as opinions and ideal). Also, CF technique has the ability to provide serendipitous recommendations, which means that it can recommend items that are relevant to the user even without the content being in the user's profile [42]. Despite the success of CF techniques, their widespread use as revealed some potential problems such as follows.

A. Cold-start problem: This refers to a situation where a recommender does not have adequate information about a user or an item in order to make relevant predictions [43]. This is one of the major problems that reduce the performance of recommendation system. The profile of such new user or item will be empty since he has not rated any item; hence, his taste is not known to the system.

B. Data sparsity problem: This is the problem that occurs as a result of lack of enough information, that is, when only a few of the total number of items available in a database are rated by users [10,44]. This always leads to a sparse user-item matrix, inability to locate successful neighbors and finally, the generation of weak recommendations. Also, data sparsity always leads to coverage problems, which is the percentage ofitems in the system that recommendations can be made for [45]

C. Scalability: This is another problem associated with recommendation algorithms because computation normally grows linearly with the number of users and items [44]. A recommendation technique that is efficient when the number of dataset is limited may be unable to generate satisfactory number of recommendations when the volume of dataset is increased. Thus, it is crucial to apply recommendation techniques which are capable of scaling up in a successful manner as the number of dataset in a database increases. Methods used for solving scalability problem and speeding up recommendation generation are based on Dimensionality reduction techniques, such as Singular Value Decomposition (SVD) method, which has the ability to produce reliable and efficient recommendations.

D. Synonymy: Synonymy is the tendency of very similar items to have different names or entries. Most recommender systems find it difficult to make distinction between closely related items such as the difference between e.g. baby wear and baby cloth. Collaborative Filtering systems usually find no match between the two terms to be able to compute their similarity. Different methods, such as automatic term expansion, the construction of a thesaurus, and Singular Value Decomposition (SVD), especially Latent Semantic Indexing are capable of solving the synonymy problem. The shortcoming of these methods is that some added terms may have different meanings from what is intended, which sometimes leads to rapid degradation of recommendation performance[46].

I.3.2.4 Example of collaborative systems:

Ringo [47] is a user-based CF system which makes recommendations of music albums and artists. In Ringo, when a user initially enters the system, a list of 125 artists is given to the user to rate according to how much he likes listening to them. The list is made up of two different sections. The first session consists of the most often rated artists, and this affords the active user opportunity to rate artists which others have equally rated, so that there is a level of similarities between different users' profiles. The second session is generated upon a random selection of items from the entire user-item matrix, so that all artists and albums are eventually rated at some point in the initial rating phases.

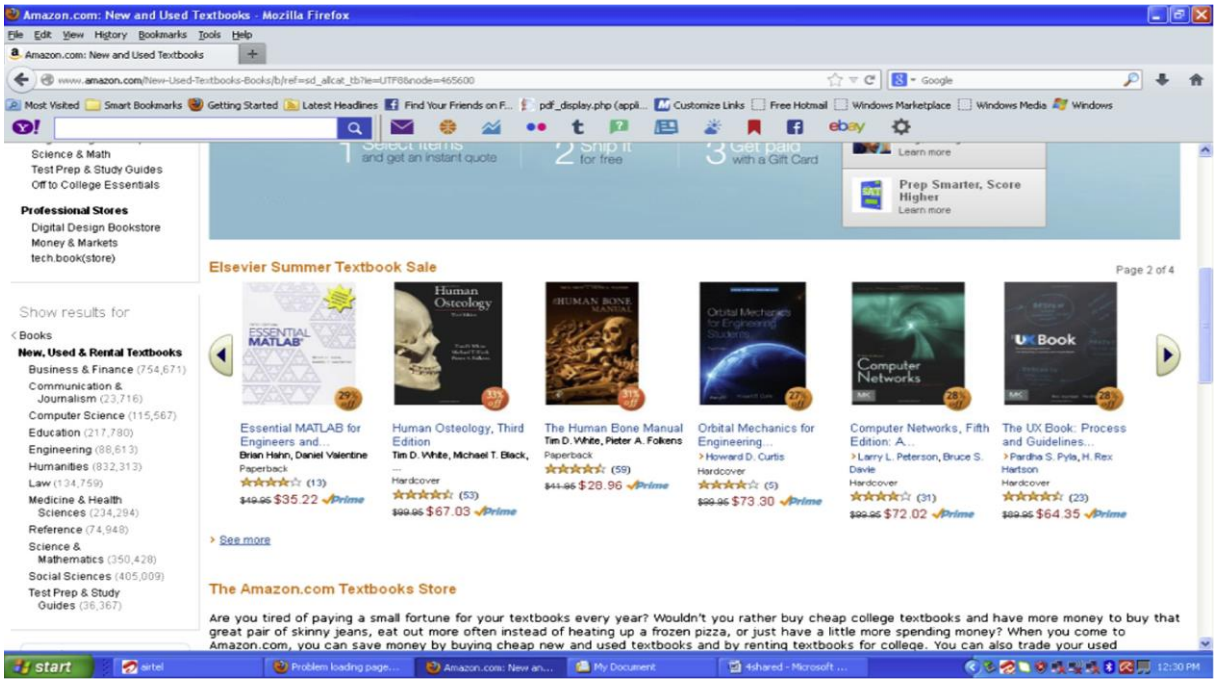


Figure I 4: Amason book recommender interface.[1]

I.3.3 Hybrid filtering:

Hybrid filtering technique combines different recommendation techniques in order to gain better system optimization to avoid some limitations and problems of pure recommendation systems [48,49]. The idea behind hybrid techniques is that a combination of algorithms will provide more

[1] www.amason.com.

accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm [42]. Using multiple recommendation techniques can suppress the weaknesses of an individual technique in a combined model. The combination of approaches can be done in any of the following ways: separate implementation of algorithms and combining the result, utilizing some content-based filtering incollaborative approach, utilizing some collaborative filtering in content-based approach, creating a unified recommendation system that brings together both approaches.

I.3.3.1 Weighted hybridization:

Weighted hybridization combines the results of different recommenders to generate a recommendation list or prediction by integrating the scores from each of the techniques in use by a linear formula. An example of a weighted hybridized recommendation system is P-tango [50]. The system consists of a content-based and collaborative recommender. They are given equal weights at first, but weights are adjusted as predictions are confirmed or otherwise. The benefit of a weighted hybrid is that all the recommender system's strengths are utilized during the recommendation process in a straightforward way.

I.3.3.2switching hybridization:

The system swaps to one of the recommendation techniques according to a heuristic reflecting the recommender ability to produce a good rating. The switching hybrid has the ability to avoid problems specific to one method e.g. the new user problem of content-based recommender, by switching to a collaborative recommendation system. The benefit of this strategy is that the system is sensitive to the strengths and weaknesses of its constituent recommenders. The main disadvantage of switching hybrids is that it usually introduces more complexity to recommendation process because the switching criterion, which normally increases the number of parameters to the recommendationsystem, has to be determined [14]. Example of a switching hybrid recommender is the DailyLearner [51] that uses both content-based and collaborative hybrid where a content-based recommendation is employed first before collaborative recommendation in a situation where the content-based system cannot make recommendations with enough evidence.

I.3.3.3 Cascade hybridization:

The cascade hybridization technique applies an iterative refinement process in constructing an order of preference among different items. The recommendations of one technique are refined by another recommendation technique. The first recommendation technique outputs a coarse list of recommendations which is in turn refined by the next recommendation technique. The hybridization technique is very efficient and tolerant to noise due to the coarse-to-finer nature of the iteration. EntreeC [14] is an example of cascade hybridization method that used a cascade knowledge-based and collaborative recommender.

I.3.3.4 Mixed hybridization:

Mixed hybrids combine recommendation results of different recommendation techniques at the same time instead of having just one recommendation per item. Each item has multiple recommendations associated with it from different recommendation techniques. In mixed hybridization, the individual performances do not always affect the general performance of a local region. Example of recommender system in this category that uses the mixed hybridization is the PTV system [52] which recommends a TV viewing schedule for a user by combining recommendations from content-based and collaborative systems to form a schedule. Profinder [53] and PickAFlick [54] are also examples of mixed hybrid systems.

I.3.3.5 Feature-combination:

The features produced by a specific recommendation technique are fed into another recommendation technique. For example, the rating of similar users which is a feature of collaborative filtering is used in a case-based reasoning recommendation technique as one of the features to determine the similarity between items. Pipper is an example of feature combination technique that used the collaborative filter's ratings in a content-based system as a feature for recommending movies [55]. The benefit of this technique is that, it does not always exclusively rely on the collaborativedata.

I.3.3.6 Feature-augmentation:

The technique makes use of the ratings and other information produced by the previous recommender and it also requiresadditional functionality from the recommender systems. Forexample, the Libra system [56] makes content-based recommendationof books on data found

in Amazon.com by employing a naıve Bayes text classifier. Feature augmentationhybrids are superior to feature-combinationmethods in that they add a small number of features to the primaryrecommender.

I.3.3.7 Meta-level:

The internal model generated by one recommendation technique is used as input for another. The model generated is always richer in information when compared to a single rating. Meta-level [57] hybrids are able to solve the sparsity problem of collaborative filtering techniques by using the entire model learned by the first technique as input for the second technique. Example of meta-level technique is LaboUr [58] which uses instant-based learning to create content-based user profile that is then compared in a collaborative manner.

I.3.4 Demographic Recommender Systems:

In demographic recommender systems, the demographic information about the user is leveraged to learn classifiers that can map specific demographics to ratings or buying propensities. An early recommender system, referred to as *Grundy* [59], recommended books based on the library of manually assembled stereotypes. The characteristics of the user were collected with the use of an interactive dialogue. The work in [60] observed that the demographic groups from marketing research can be used to recommend items. Another work [61] makes Web page recommendations on the basis of the demographic characteristics of users that have rated a particular page highly. In many cases, demographic information can be combined with additional *context* to guide the recommendation process. This approach is relatedto the methodology of *context-sensitive recommender systems*.

More recent techniques have focused on using classifiers for making recommendations. One of the interesting systems in this respect was a technique that extracted features from user home pages in order to predict their likelihood of liking certain restaurants. Rule-based classifiers [62, 63] are often used to relate the demographic profile to buying behavior in an interactive way. While the approach in [62, 63] was not specifically used to recommend specific items, it can easily be paired with a recommender system. Such recommender systems are not very different from the vanilla classification and regression modeling problem, in which feature variables correspond to the demographic profiles and the dependent variables correspond to the ratings or

to the buying behavior. Although demographic recommender systems do not usually provide the best results on a stand-alone basis, they add significantly to the power of other recommender systems as a component of hybrid or ensemble models. Demographic techniques are sometimes combined with knowledge-based recommender systems to increase their robustness.

I.4 Evaluation metrics for recommendation algorithms:

The quality of a recommendation algorithm can be evaluated using different types of measurement which can be accuracy or coverage. The type of metrics used depends on the type of filtering technique. Accuracy is the fraction of correct recommendations out of total possible recommendations while coverage measures the fraction of objects in the search space the system is able to provide recommendations for. Metrics for measuring the accuracy of recommendation filtering systems are divided into statistical and decision support accuracy metrics [64]. The suitability of each metric depends on the features of the dataset and the type of tasks that the recommender system will do [16].

Statistical accuracy metrics evaluate accuracy of a filtering technique by comparing the predicted ratings directly with the actual user rating. Mean Absolute Error (MAE) [65], Root Mean Square Error (RMSE) and Correlation are usually used as statistical accuracy metrics. MAE is the most popular and commonly used; it is a measure of deviation of recommendation from user's specific value. It is computed as follows [49]:

$$MAE = \frac{1}{N}\sum_{u,i} |p_{u,i} - r_{u,i}| \quad \text{(I.9)}$$

where $p_{u,i}$ is the predicted rating for user u on item i, $r_{u,i}$ is the actual rating and N is the total number of ratings on the item set. The lower the MAE, the more accurately the recommendation engine predicts user ratings. Also, the Root Mean Square Error (RMSE) is given by Cotter et al. [66] as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}\left(p_{u,i} - r_{u,i}\right)^2} \quad \text{(I.10)}$$

Root Mean Square Error (RMSE) puts more emphasis on larger absolute error and the lower the RMSE is, the better the recommendation accuracy.

Decision support accuracy metrics that are popularly used are Reversal rate, weighted errors, Receiver Operating Characteristics (ROC) and Precision Recall Curve (PRC), Precision, Recall and F-measure. These metrics help users in selecting items that are of very high quality out of the available set of items [67]. The metrics view prediction procedure as a binary operation which distinguishes good items from those items that are not good. ROC curves are very successful when performing comprehensive assessments of the performance of some specific algorithms. Precision is the fraction of recommended items that is actually relevant to the user, while recall can be defined as the fraction of relevant items that are also part of the set of recommended items [68]. They are computed as:

$$Precision = \frac{\text{Correctly recommended items}}{\text{Total recommended items}} \quad \text{(I.11)}$$

$$Recall = \frac{\text{Correctly recommended items}}{\text{Total useful recommended items}} \quad \text{(I.12)}$$

F-measure defined below helps to simplify precision and recall into a single metric. The resulting value makes comparison between algorithms and across data sets very simple and straight forward [64].

$$F - mesure = \frac{2PR}{P+R} \quad \text{(I.13)}$$

Coverage has to do with the percentage of items and users that a recommender system can provide predictions. Prediction may be practically impossible to make if no users or few users rated an item. Coverage can be reduced by defining small neighborhood sizes for user or items [69].

I.5Conclusion:

Recommender systems open new opportunities of retrieving personalized information on the Internet. It also helps to alleviate the problem of information overload which is a very common phenomenon with information retrieval systems and enables users to have access to products and services which are not readily available to users on the system. This paper discussed the two traditional recommendation techniques and highlighted their strengths and challenges with diverse kind of hybridization strategies used to improve their performances. Various learning algorithms used in generating recommendation models and evaluation metrics used in measuring the quality and performance of recommendation algorithms were discussed. This knowledge will empower researchers and serve as a road map to improve the state of the art recommendation techniques.

CHAPTER II: User profiling and modeling

II.1 Introduction:

Personalization has got a lot importance in the field of computer science, being specific in the applications of Recommendation Systems. There are many users which a recommender system needs to deal with, and each user has their own preference requirements. Recommender System needs to fulfill the requirements of every user by recommending user specific items to them or by modifying itselfas per the needs of the user [70].

So user profiling helps recommender system to know user requirements and behave in accordance of them. User Profiling can be defined as the process of identifying the data about a user interest domain. This information can be used by the system to understand more about user and this knowledge can be further used for enhancing the retrieval for providing satisfaction to the user. User profiling has two important aspects as efficiently knowing user and based on those recommending items of his interest System [70].

II.2 Collection of information:

To be relevant, a recommendation system must be able to make predictions on the interests of users. It is therefore necessary to be able to collect a certain amount of data on them in order to be able to build a profile for each user. A distinction can be made between 2 forms of data collection [71]:

II.2.1 Explicit feedback:

The Collection is based on the fact that the user explicitly indicates their interests to the system.

- ***Example*:** Ask a user to comment, tag, rate, like or add as favorite's content (objects, articles ...) that interest him. We often use a rating scale ranging from 1 star (I don't like it at all) to 5 stars (I like a lot) which are then transformed into numerical values so that they can be used by recommendation algorithms [71].
- ***Advantage*:** Ability to reconstruct an individual's history and ability to avoid to aggregate information that does not correspond to this single user (several people on the same workstation) [71]

II.2.2 Implicit feedback:

It is based on an observation and an analysis of user behavior carried out implicitly in the application that embeds the recommendation system, everything is done in the "background" (basically without asking the user anything) [71].

- ***Examples*:**
  - Get the list of items that the user has listened to, watched or purchased in line.
  - Analyze the frequency of consultation of content by a user, the time passed on a page.
  - Monitor the online behavior of the user.
- ***Advantage*:** No information is requested from users, all information is collected automatically [71].
- ***Inconvenient*:** The recovered data is more difficult to attribute to a user and can therefore contain attribution biases (common use of the same account by several users). A user may not like certain books he bought, or he may have bought it for someone else [71].

II.2.3 Hybrid User Profiling:

This type of user profiling combines the advantages of both Implicit as well as explicit user profiling. It takes into consideration both the static characteristics about a user and also goes retrieving the behavioral information regarding a user. This approach helps profiling more efficient and maintains the accuracy of temporal information as information gets updated temporally [70].

II.3 Techniques:

By looking at different research literature we can be inferred that a user profiling system can be subdivided into subtasks like profile extraction, profile integration and user interest discovery. Now we will look for different techniques used for each of these tasks[70].

II.3.1 Profile extraction:

Profile extraction is nothing extracting the useful information about a user from different sources. For this cause many methodologies and models have been used by different researchers. Some of which can be mentioned as extracting data from web, extracting data from social media

websites as well as some user behavior based techniques which help user profiling system gather interesting information about users [70].

Jie Tang [72] has explained the task of extracting profiles involves finding relevant pages from web and then finding relevancy using support vector machine algorithm. In this technique, Jie tang used preprocessing on the relevant information using linear conditional random fields and tree structured conditional random fields. It has been also found that the Tree Structured Random fields perform well as compared to linear random fields.

Inanother approach by Jiwei Li [73] has tried to extract user information from social networking websites like Twitter, Google plus and Facebook. In this they used present a weakly supervised approach to user profile extraction from Twitter [73]. Data from these websites are used as a distant source of supervision for extraction of their attributes from user-generated text. In addition to traditional linguistic features used in distant supervision for information extraction, this approach also takes into account network information, a unique opportunity offered by social media.

Yoshinori Hijikata [74] has done a very interesting study in which user's mouse operation are traced to extract text parts which the user might be interested in from the whole text of the Web page. An implicit relevance feedback [75] is given to the system by which system can generate relevant recommendations.

Some of the main operations that are beingpointed out specifically are [70]:

- Text tracing: Moving the mouse pointer along a sentence while reading.
- Link pointing: Positioning the mouse pointer on a link, but not clicking the link.
- Link clicking: Clicking on a link to move to another page.
- Text selection: Selecting text by dragging the mouse pointer.
- Scrolling: Scrolling a window at a certain speed.
- Bookmark registration: Registering a page as a bookmark.
- Saving: Saving an HTML document.
- Printing: Printing a page.
- Window movement: Moving a window of the Web browser.
- Window resizing: Changing the window size of the Web browser.

II.3.2 Profile Integration:

After the extraction of relevant information is done there might be sometimes a problem of data cleaning. Its might happen that some of the data that has been collected might be a duplicate data or it may look like duplicate data but that might be a unique data. So to resolve this issue unique and duplicate data must be identified which will help profiling system in the next subtask[72].
One of such problem is identified by Jie Tang et .al. [72] Which they referred it as Name Disambiguation Problem. A very efficient technique is being used to resolve the name disambiguation problem viz. Hidden Markov Model. Hidden Markov Random Field (HMRF) is a member of the family of MRF(Markov random field)and its concept is derived from Hidden Markov Models (HMMs)

II.3.3 Interest Discovery:

After getting some information has been collected about the users it is necessary to chop down the users into different classes, as this will help giving feedback to the system. This can be done by grouping the users based on their behaviors into some classes. This technique is also called as filtering [70].

II.4 Representation of the user model:

Once the data about the user has been collected and the characteristics of the user model established, the question of the representation of this data and this model arises. The representation of the user model characterizes the schema of data collected on the user. there is several types of user modeling like a simple representation by keywords or vector representation and semantic representation and multidimensional to be more precise user modeling is a sub-discipline of information filtering, which is a sub-discipline of information retrieval, and a "research area that offers tools for discriminating between relevant and irrelevant information by providing personalized assistance for continuous retrieval of information"[75].

II.4.1 Representation as vectors:

It is one of the most used profile representations, especially for its simplicity. Usually, the profile vector corresponds to a set of characteristics by giving for each one its weight (the value of the

coordinate for a dimension). The way to calculate the weight varies depending on the application. For example, a profile based on the keywords associated with the user is based on a TF-IDF weighting from the field of information retrieval [76].

II.4.2 Representation in the form of hierarchical concepts:

Compared to the representation in vector form, the use of hierarchical concepts allows the system to generalize the profile. For example, a user interested in the Olympic Games is probably also interested in sport. The level of the hierarchy of concepts can be fixed [77] or be dynamic [78]. Here the concept can replace the characteristics (interests, emotions skills ...) more often this technique is used to represent interests.

II.4.3 Representation in the form of ontologies:

Ontology is a specification of concepts and relations between them. It defines “content specific agreements” on vocabulary usage, sharing, and reuse of knowledge. It is used to alleviate the communication problems between systems due to ambiguous usage of different terms. When building user profiles, ontologies are used to address the so-called “cold-start problem”. When initially learning user interests, systems usually perform poorly until they collect enough relevant information. Since initial behavior information is matched with existing concepts and relations between them, using ontologies as the basis of the profile helps to avoid or ease this problem[79].

II.4.4 Representation in the form of graphs:

This method is widely used to have an analysis of the relationships between the pairs of characteristics (interests, emotions …) where a node represents an instance of characteristics and an arc the relationship between two nodes. This representation makes it possible to analyze semantic relationships [80, 81] and is close to the representation in the form of ontologies when the graph is a tree. Most of the time, this technique is used to analyze the relationships between users in a social network where a node represents a user but this type of representation can be used for a profile of a user, considering only the ego network.

There are other types of representation of a network such as Bayesian networks, Oriented graph andnon-oriented graph.

II.5 Recommender system using social networks:

II.5.1 Educational recommender system using social networks interactions and linked data - BROAD-RSI:

The continuous and ever growing usage of social networks facilitates the recognition through random contact of specific details regarding identity, desires, expectations, style and behavior. This paper provides a framework capable of extracting from the social network of Facebook users' profile and educational background and proposing educational tools.Data analysis Techniques and Semantic Web Technology helpthe proposal to capture enrich and describe user identity and interests. The suggestion methodology is focused on databases of learning objects, related data and video databases;it optimizes the user's time spent on the site. The assessment of the proposal was made from a prototype creation, three design proofs and one case study. The evaluation demonstrated the acceptance by users of significant interaction about their educational interests, created automatically from social networks and enriched to find implicit interests.It also validated the feasibility of recommending individuals, allowing the creation of a network of interest, focused on a particular topic, showing good collaborators for study and analysis[82].

BROAD-RSI (Recommender Program Based on Social Interactions) was designed to satisfy the following requirements : (1) define user profile and context features based on randomly generated knowledge from social network interactions; (2) expand the user profile and context to assist the collection and suggestion of non-educational resources; (3) provide users with educational recommendations;(4) indication of people with similar educational interests, seeking to create a network of common interests[82].

BROAD-RSI has been established to process information extracted from a social network, set user profile and context characteristics, and generate educational recommendations that are compatible with those functions [82].

II.5.2 Efficient User Profiling in Twitter Social Network Using Traditional Classifiers:

Any social media conversation can be beneficial if the people participating in the conversation are connected to a specific area. This is useful to identify consumers who are interested in the content of the event in a similar way to advertising an event. This paper introduces an effective supervised machine learning approach that categorizes Twitter users in six interest groups- Politics, Media, Entrepreneurship, Journalism, Science & Technology and Healthcare-based on

three essential features (Tweet-based, User-based, and Time series).They contrast the proposed feature set with different standard classifiers such as Support Vector Machines, Naive-Bayes, k-Nearest Neighbors, Decision Tree and Logistic Regression, and achieve a classification precision of up to 89.82 per cent. They also offer a proposal for a real-time user identification program for Twitter, along with a prototype implementation[83].

Because they use Tweet-based functionality to define the user's complex interest, they believe the user is tweeting about topics he / she has an interest in. They also use the Weka machine learning library to perform various feature selection & machine learning tasks as well as to implement the Java real-time application[83].

II.5.3 User Profile Extraction from Twitter for Personalized News Recommendation:

This paper aims at exploring a particular way of creating user profiles using the details gathered from Twitter to offer customized news advisory service. For a Twitter user, his method uses tweets, re-tweets and hashtags which extract essential keywords to create personal profiles. This method's usefulness is validated by implementing a prototype news recommendation service and performing a user survey. Using a simple measure of cosine similarity, we compare the variations between the user accounts, and also between the suggested news lists, to test the discriminative capacity of the proposed process. The accuracy of the news recommendation's prediction is measured against a small group of users[87].

This paper presents a version of the content-based news review study. Existing customized news review methods often use information obtainable from real user-read news content such as names, messages, and click-through data[84],[85],[86] while few strategies try to do the job without using news content information[87].

II.5.4 TRUPI: Twitter Recommendation Based on Users' Personal Interests:

This article proposes TRUPI, a customized timeline recommendation system for users of the twitter where tweets are ranked by the user's personal preferences. The suggested framework incorporates the psychological features and experiences of the individual as well as the context of the tweet content to fulfill their interests. The system captures dynamically the users' interests by modeling them in different topics as a time variant to accommodate the changing of those interests over time. To do so, TRUPI recognizes over time the changing interests of the users, and then ranks the received tweets accordingly. More precisely, TRUPI utilizes an interest

classifier ensemble that shows the most possible interest mark on the user's timeline for each tweet. Posts are then loaded into a rating algorithm to order the tweets according to the preferences of the user based. The preferences of the user are based on various topics as a time-varying degree of concern. Their comprehensive performance evaluation on a publicly accessible database illustrates TRUPI's efficacy and reveals that it surpasses the competitive state of the art by 25% on nDCG@25, and 14% on MAP[88].

II.5.5 Hashtag Recommendation Based on User Tweet and Hashtag Classification on Twitter:

In Twitter, self-defined metadata called hashtags are added to the tweets. But, because of the large variety of hashtags, people are having trouble identifying acceptable hashtags for their tweets. In this paper they propose a new hashtag recommendation scheme for user tweets based on analysis of user tweets and classification of hashtags. More precisely, they collect phrases from user tweets using TF-IDF and use the naïve bayes algorithm to sort the hashtags into predefined groups. First, they pick a category of user interest based on user tweet keywords to represent the user interest. They measure the ranks of candidate hashtags by analyzing matching tweets, user engagement, and hashtag usage to propose acceptable hashtags to the users. To show their scheme's performance, they developed an Android application called "TWITH" and assessed its accuracy of recommendations. Via numerous experiments they show that the hashtag suggestion makes their scheme very successful[89].

This paper suggests that lets people use acceptable hashtags for their messages by evaluating current messages and hashtags. Below are the core components of the planned system[89].

- Extracting a collection of keywords from each tweet
- Translating the set of keywords into one of the predefined groups
- Determining the ranks of the nominee hashtags
- Suggesting the most suitable hashtags to the user

As a consequence, when a user finishes a tweet with no hashtags, their code will automatically suggest acceptable hashtags related to the tweet. Hashtag is a fantastic resource on Twitter to collect knowledge.When the people who follow a certain hashtag becomeinvolved in a message so the hashtag will be added to the post [89].

II.5.6 Twitter User Modeling and Tweets Recommendation Based on Wikipedia Concept Graph:

Twitter messages are only shown by time recency, which means user, can miss any important or interesting tweets if he cannot read all of the messages. In this article, they recommend re-ranking tweets in the history of the user by creating a user profile based on past tweets from the user and calculating the correlation between a tweet and user interest. The user experience profile is defined as Wikipedia definitions, which is a relatively broad and interlinked source of online information. They use the Clear Semantic Analysis algorithm to extract similar concepts from tweets and then extend the user profile by random walk on the topic graph of Wikipedia, using the inter-links between articles on Wikipedia. Their tests demonstrate how effective and powerful their model is in suggesting tweets to users.

Many users have generated large amounts of messages in social networks , especially on Twitter, that imply their interest, which are a precise and readily accessible source for building user interest profiles. This paper makes use of this information to build user profiles. Having a user profile can help reorganize a user's history by being important to the interest of the users. They suggest a model for mapping Wikipedia related definitions to a tweet and a person. After they have developed Wikipedia-based representation for user interests and tweets, they can calculate each tweet's significance and then rate all tweets according to the user interest[90].

II.5.7 A Personalized Tweet Recommendation Approach Based on Concept Graphs:

Twitters get their timeline with the latest tweets from their followers. The concern is that not all of these tweets are really of value to a person.However, because of the time needed this tweet overload complicates the manual tweet retrieval. A solution to this issue is the development of effective recommender mechanisms that enable users sort the interesting tweets to save precious time from browsing through uninteresting tweets and forwarding some very interesting tweets along. This work proposes a new content-based method for custom tweet recommendation, which takes advantage of the semantic connectedness of user-interested concepts. They could leverage their approach to create an effective, online tweet recommender. They construct a concept graph (CG), incorporating range of principles and utilizing graph theory algorithms not yet implemented in the study of social networks to generate graded recommendations[91].

Using the Concept Graph helps one to eliminate concerns such as over-recommendation and over-specialization, as their approach takes into consideration the real and unbiased relationship between the Terms of Interest (ToIs) of a customer and the Concept Graph itself. The idea behind the approach is that the ToIs of a user are linked objectively in a manner that creates a profile of the user. By applying this to a dataset, they validate their process and compare it by contrasting the findings with various state-of-the-art methods[91].

II.5.8 AMovie Recommendation based on Users' Tweets:

In this paper new idea for recommending movies has been designed. This system is based on machine learning algorithm which calculates similarity between user's tweets and scenario of the movies. This method could recommend movies which are more similar to users' interest. For this purpose we need history or some data of user. To overcome this cold start, system will connect to user's twits and retrieve posts. After collecting user post we will extract keywords and remove those unwanted words. Then after doing some analysis system compare these extracted keywords with movies' scenario to calculate which movies are similar to users' posts. Therefore among those list algorithm can select top N first of the list as a recommendation to the user.In this work a personal research scholar database for movies and Python API for collecting users' post in twitter have been used .Now there is database of users' posts and scenario of each user is available. Using methods to compare similarity between users's post and scenario is the concern of this work[92].

II.6.Conclusion:

In this chapter, we have presented the different types of content in a user profile using data from social networks as well as the techniques representation of these profiles. These profiles can be used either on the sites social networks themselves or in external applications. We will also be interested in the use of user profiles in our research, and we have mentioned other works that it’s done on the field of recommendation system using social network information and users.

CHAPTER III:The proposed approach based on user interaction with tweeter

III.1 Introduction:

Due to the fact that recommendation systems using matrix factorization did gave a very promising and good results wish lead some company to use matrix factorization approach in their recommendation system, but it have some disadvantage like the could start problem we cannot recommend items for users who have no interaction with the items and a low rank matrix is another disadvantage means no pertinent results. In this approach a method is proposed to make an auto ranking phase based on user interaction with tweeter. The objective of this work is to fulfill the ratings matrix of new users based on their interaction with tweeter, the approach is divided in three parts, the first part is about Data acquisition where a dataset is created based on combining two other datasets, the first contains movies summaries with users rating the same movies, and a second contains users tweets. The second part is about data preprocessing for both extracted datasets, the realized work in this part is making a keyword extraction for user tweets and movies summaries and then calculate the similarity with every user tweets and every movies. The third part comes a head the recommendation, where an auto ranking phase is realized based on the results of the similarity and predict ratings for all users and then make recommendation.

III.2 General Architecture:

The work in the below figure is divided on five steps, the first one is to combine two datasets, then the second part is to extract keyword from both datasets, third step is calculate the similarity between user tweets and movies summaries, forth make an auto ranking phase to have the final ratings matrix, then fifth make a recommendation system using matrix factorization based on user interaction with tweeter. Each part will be detailed.

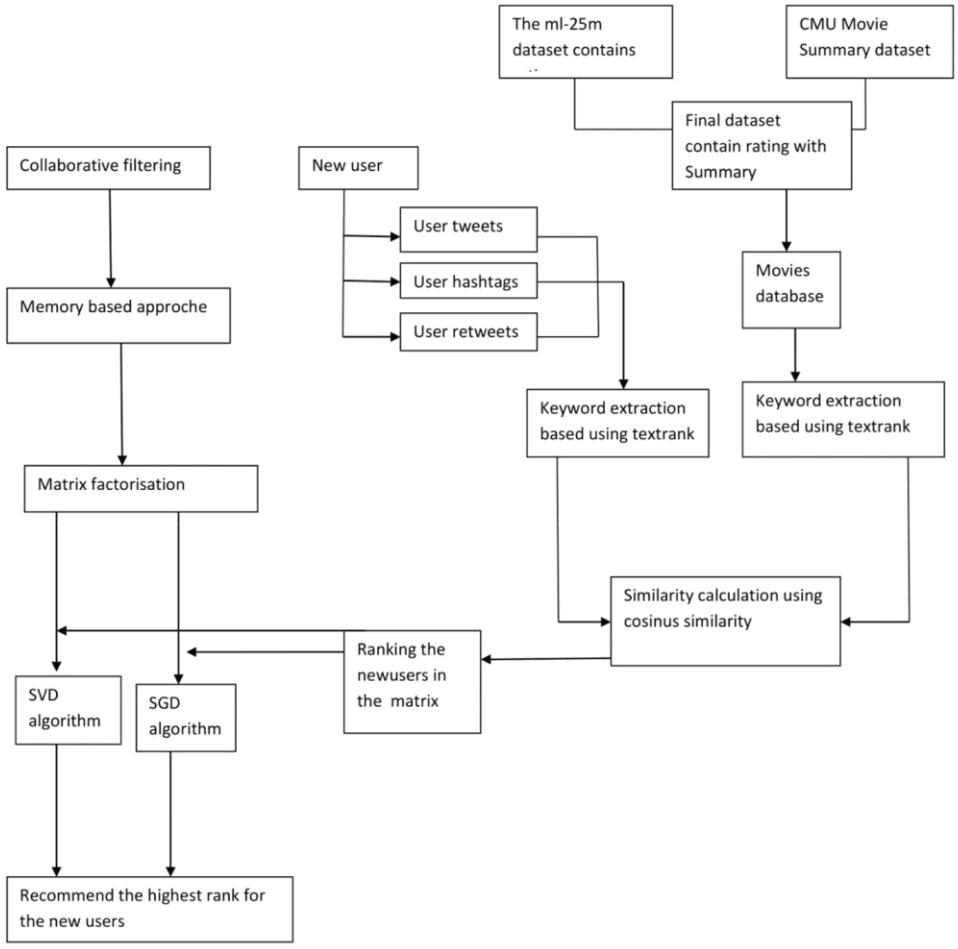


Figure III.1: General Architecture

III.2.1Data Collection:

This part contains two major sub-parts the aim of the first sub-part is to create dataset that contains movies with summaries and users rating the same movies(movies database) the second part is to extract user's tweets, hashtags and retweets.

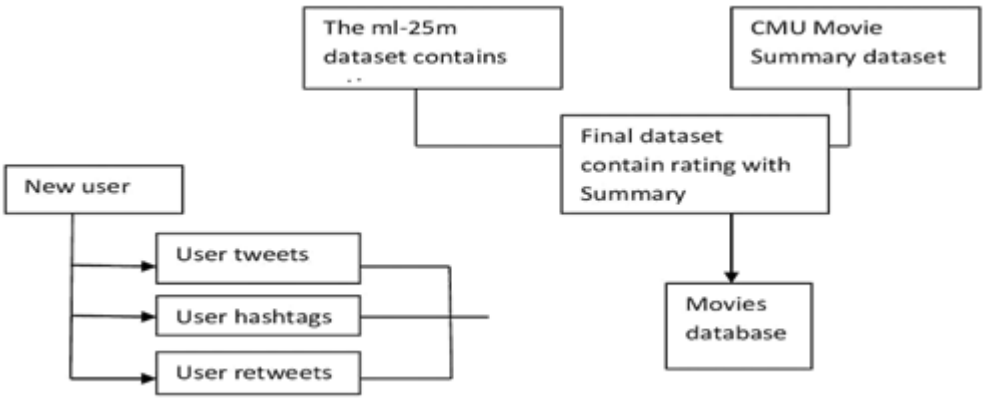


Figure III.2: Data Collection

A. Create movies dataset contains movies, ratings, summaries:

Figure III.3 will explain how we will create the data set after checking the movies that have the same name and year and then combine the both.

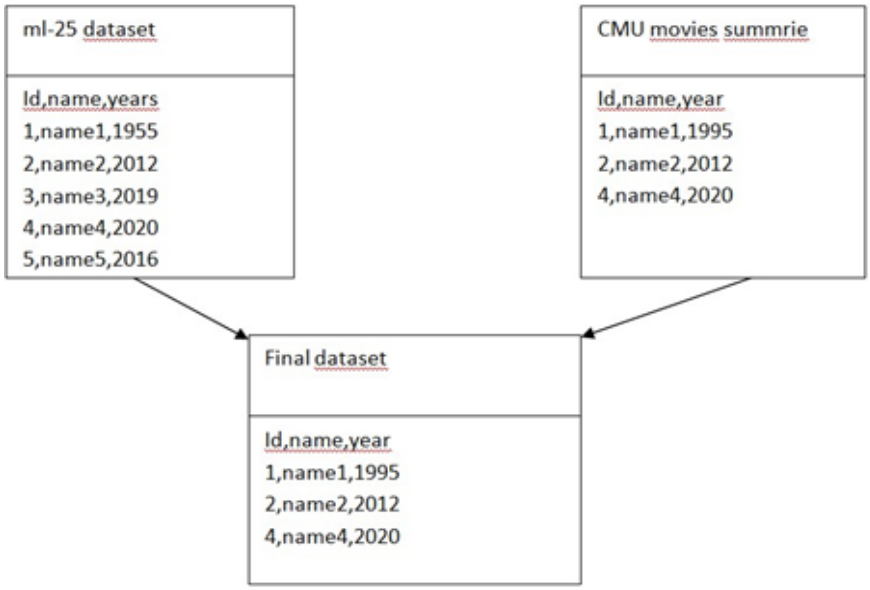


Figure III.3:Obtaining final dataset

The aim behind this part is to create a dataset contains movies summaries and user who did rate the same movies in order to make a recommendation based on user tweets and movies summaries and movies rating, the verification of the name with year of the movie is necessary cause some movies have the same name but different year (just to avoid this conflict).

Algorithm Data acquisition

Input ml−25 data, Movies summaries data;

Output Cos-similarity between movies and tweets;

Begin

- Normalization of both datasets
- Remove all special characters
- Turn all to lower case
- Remove additional species
- Find movies that have same name and year
- Create new dataset contain movies with similar year and name
- Save them as a file

End.

B. User tweets extraction:

Algorithm Users' tweets Extraction

Input tweeter access key, tweeter consumer key;

Output users, tweets, hashtags, retweets, time;

Begin

- Connect to tweeter
- Download random users screen names
- Use tweeter function to download tweets, hashtags, retweets, time
- Save user name, tweets, hashtags and retweets
- Save data as corpus

End.

In this sub-part an extraction of user- tweets is realized using the provided tool by tweeter in order to allow individuals to download tweets, hashtags and retweets.

III.2.2 Data preprocessing:

This part is divided into two parts the first one is keywords extraction for movies summaries and user tweets and the second one is computing similarity between movies summaries and user tweets

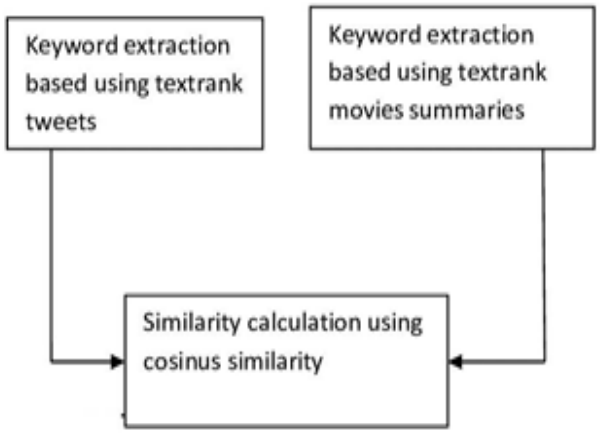


Figure III.4: Data preprocessing

A. Keywords extraction:

Algorithm Keyword Extraction

Input final-dataset, users tweets;

Output movies summaries keywords, user's tweets keywords;

Begin

- Normalization of both datasets
- Remove all special characters
- Turn all to lower case
- Remove additional spaces
- Segmentation
- Lemmatization
- Stop words
- Save NOUN, PROPN,VERB
- Divide text to sentences
- Specify the window size K
- Specify damping coefficient
- Specify iteration steps
- Get token pairs (schema)
- Save general graph in the matrix based on token pairs
- Calculate the weight of the word (equation below)
- Save them as a file

End.

The following equation is used to calculate the weight of words.

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad \text{(III.1)}$$

- $S(V_i)$ the weight of word i
- d damping factor, in case of no outgoing links
- $In(V_i)$ inbound links of i,which is a set
- $Out(V_j)$ outgoing links of j, which is a set
- $|Out(V_j)|$ the number of outbound links

An example to demonstrate exactly how the TextRank extracts the keywords is in annexes A.

B. Similarity between movies and tweets:

Algorithm Make similarity between movies and tweets;

Input User tweets, Movies summaries;

Output Cos_similarity between movies and tweets;

Begin

- Append user tweets and movies summaries in a list;
- Make train set and test set;
- Transform text to vector using TF/IDF;
- Transfer data to numeric form;
- Calculate similarity using cosine similarity;

End.

the steps needed to calculate the similarity between user tweets and movies summaries are appending user tweets keyword and movies summaries keyword in a list not that it necessary to calculate the similarity for every user with all the movies and calculate the similarity using the below equation:

$$similarity(A,B) = \frac{A.B}{\|A\| * \|B\|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i}^{n} B_i^2}} \quad \text{(III.2)}$$

III.2.3 Recommendation:

The recommendation part is also divided into two sub-parts the first is to enrich the original matrix with ratings based on similarity results and the second one is predict ratings using factorization matrix

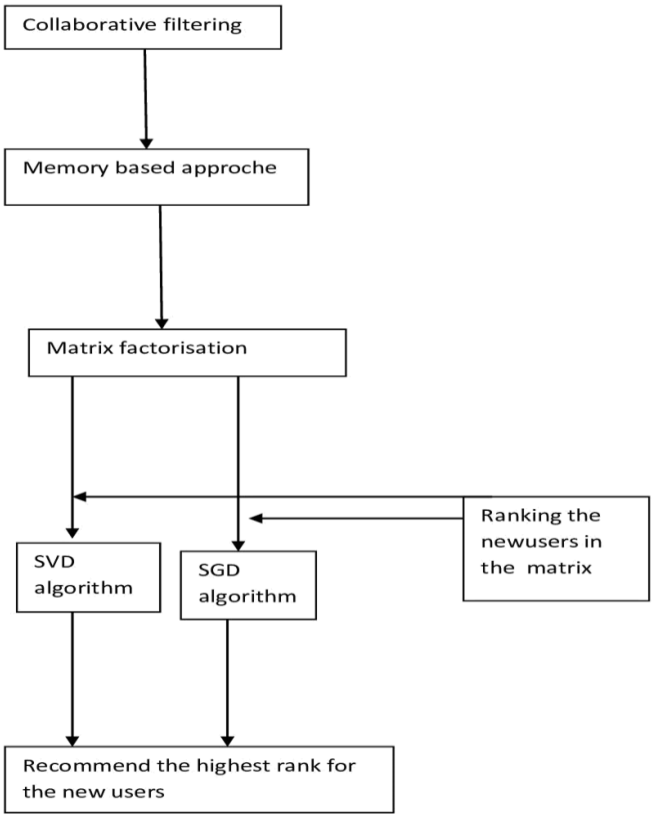
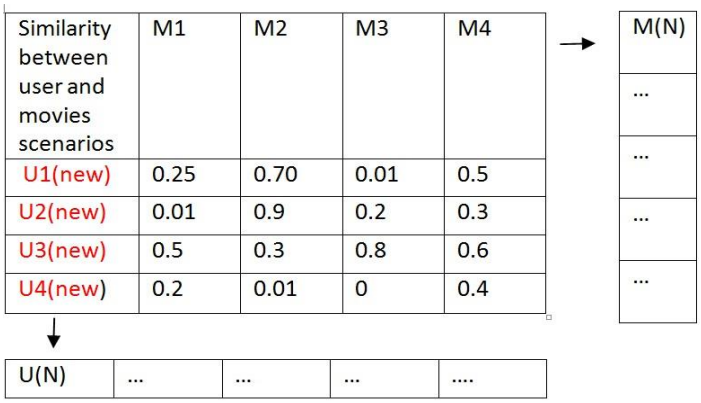


Figure III.5: Recommendation

A. Ranking the matrix based on the similarity:

After the similarity is calculated, a matrix is obtained that contains new users-id and the similarity between users and each movies. A ranking phase is necessary to rank the original matrix wish contains the movies with users rating the first and the second matrix will be demonstrated in Example 1and Example 2:

Example 1:

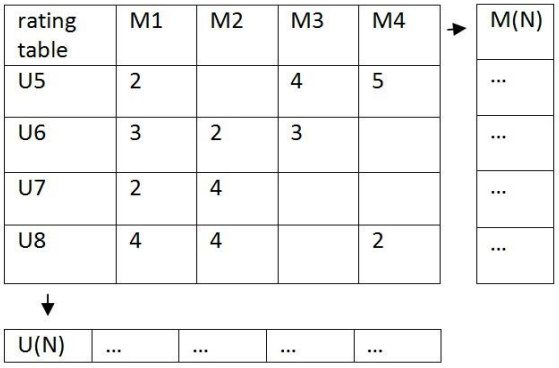


Similarity between user and movies scenarios	M1	M2	M3	M4
U1(new)	0.25	0.70	0.01	0.5
U2(new)	0.01	0.9	0.2	0.3
U3(new)	0.5	0.3	0.8	0.6
U4(new)	0.2	0.01	0	0.4

M(N)
...
...
...
...

U(N)	...	...	...	....

Example 2:

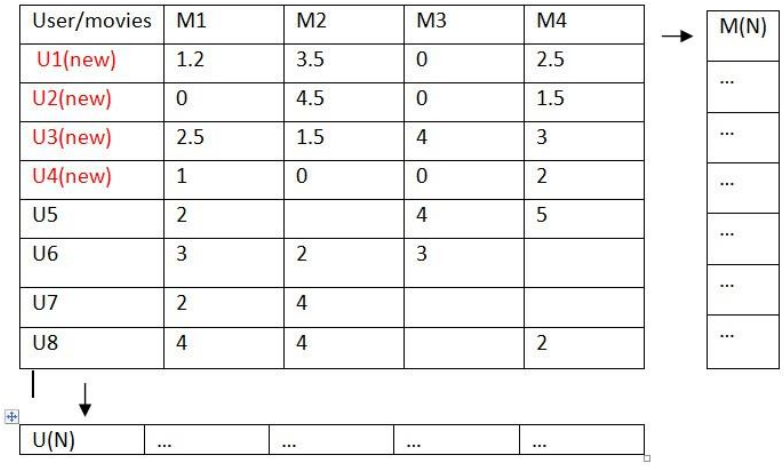


rating table	M1	M2	M3	M4
U5	2		4	5
U6	3	2	3	
U7	2	4		
U8	4	4		2

M(N)
...
...
...
...

U(N)	...	...	...	...

Know it is necessary to combine the two matrix into one matrix the first matrix contains the dataset of the similarity should be normalized and rated on scale of 1 of 5 the result after the combining the two matrix's will look like the Example 3



User/movies	M1	M2	M3	M4
U1(new)	1.2	3.5	0	2.5
U2(new)	0	4.5	0	1.5
U3(new)	2.5	1.5	4	3
U4(new)	1	0	0	2
U5	2		4	5
U6	3	2	3	
U7	2	4		
U8	4	4		2

M(N)
...
...
...
...
...
...

U(N)	...	...	...	...

B. Matrix factorization for recommendation:

Know the recommendation system comes a head as it is explain in chapter one matrix factorization is simply, factorize a matrix, i.e.to find out two (or more) matrices such that when you multiply them you will get back the original matrix.

Algorithm Matrix factorization

Input matrix combined;

Output rated matrix;

Begin

- Initialize number of steps
- Initialize number of Alpha
- Initialize number of Beta
- Define PandQ

#P,Q are the vectors such when we multiply them we get the original matrix

- initialize the values of P and Q randomly
- find the values of P and Q using SGD
- calculate the squared error
- if squared error is less than the old

  then update rule and values of P and Q

  endif

- if the error converges to its minimum

  then break

endif

End.

Initializing steps is for the grand descent to find the most adequate results; the values of alpha and beta are for introducing regularization to avoid over-fitting defining P and Q as this equation:

$$r_{ij}=p^{t}{}_{i}q_{j}=\sum_{k=1}^{k}p_{ik}\,q_{kj} \quad \text{(III.3)}$$

To get the prediction of a rating of an item Dj by Ui, we can calculate the dot product of the two vectors corresponding to Ui and Dj the used way to find the values of P and Q is first initialize the two matrices with some values, calculate how different their product is to M, and then try to minimize this difference iteratively. Such a method is called gradient descent (SGD),aiming at finding a local minimum of the difference.

The matrix factorization algorithm approach using SVD is as the following

Algorithm Matrix factorization SVD

Input matrix combined

Output rated matrix

Begin

- Initialize the number of factors
- Initialize number of alpha
- Initialize number of beta
- Divide data to train set and test set
- Call the SDV surprise library
- Save results

End

III.3 Implementation:

III.3.1 Extracting users' tweets:

The data acquisition of tweets is realized with Tweepy, a library made by tweeter, tweeter delver an access key and a consumer key to allow the individuals to download data, to start downloading the data a connection is required not that tweeter could limit your download if he found any suspicious action, the Tweepy library is the only one that allows individuals todownload data and extract them from tweeter (annex A).

III.3.2 Keyword Extraction:

In this part Textrank is the used tool,Textrank is responsible to calculate the weight of word the main work done in this part is to calculate the weight of words and extract the top k of them wich will allow us to compute the similarity.An example of how Textrank work is in annexB.

III.4 Conclusion:

The used approach shows a method to deal with new users who have no interaction with items.This method used user tweets and movies summaries to start a recommendation based on the user interaction with tweeter and to know more about the user only depending on his tweets, hashtags, retweets finding similar movies to a user help to rank the factorization matrix by ranking this matrix.

CHAPTER IV:Experimental evaluation of the proposed approach

IV.1 Introduction

After detailing the work mentioned above, it is know the time to evaluate the performance of the system. The test is made base on the three data sets, user tweets, the movies dataset ml-25 and the movies summaries. The way used to evaluate this system is to check how the system is reacting before using user tweets and after using user tweets meaning before enrichment and after enrichment; the prediction comparison is made between two major algorithms SVD and SGD.

IV.2 Evaluation framework

IV.2.1 Database used

In this approach three of data sets have been used:

1. The first one called ml-25 describes 5-star rating and free-text tagging activity from [MovieLens][2], a movie recommendation service. It contains 25000095 ratings and 1093360 tag applications across 62423 movies. These data were created by 162541 users between January 09, 1995 and November 21, 2019. This dataset was generated on November 21, 2019. Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.
2. The second used data set the CMU Movie Summary Corpus, a collection of 42,306 movie plot summaries and metadata at both the movie level (including box o ce revenues, genre and date of release) and character level (including gender and estimated age).
   (a) plot summaries.txt.gz [29 M] Plot summaries of 42,306 movies extracted from the November 2, 2012 dump of English-language Wikipedia. Each line contains the Wikipedia movie ID (which indexes into movie. metadata.tsv) fol-lowed by the summary.

[2](http://movielens.org)

(b) movie.metadata.tsv.gz [3.4 M]

- Wikipedia movie ID
- Movie name
- Movie release date
- Movie runtime

3. The third used dataset is user tweets is a corpus contain 217 tweeter user each user have at least 300 tweets and at max 3000 tweets including hash-tags and retweets.

Note that after combining the ml-25 dataset and the CMU movies summaries the number of movies became 10554 movies with summary.

IV.2.2The results of the prediction algorithms SVD and SGD:

As it is mentioned above a test before enrichment and after enrichment is required. The RMSE, MSE are the evaluations measures of these two systems the evaluation will be made as the following:

A. Compare the RMSE and MSE of the SGD before enrichment and after:

Note that the number of iteration for the SGD algorithm equals 100 and the number k-features = 20 and the Regularization strength beta=0.05 and the Learning rate alpha=0.01.

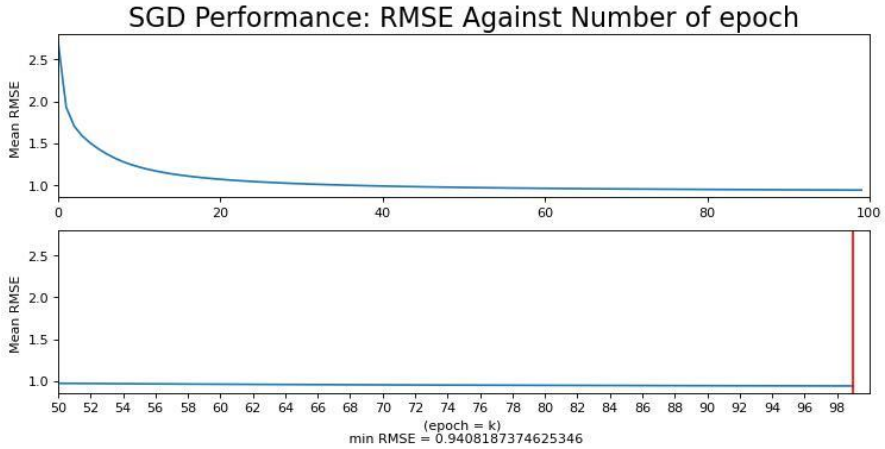


Figure IV.1: RMSE against epoch with enrichment

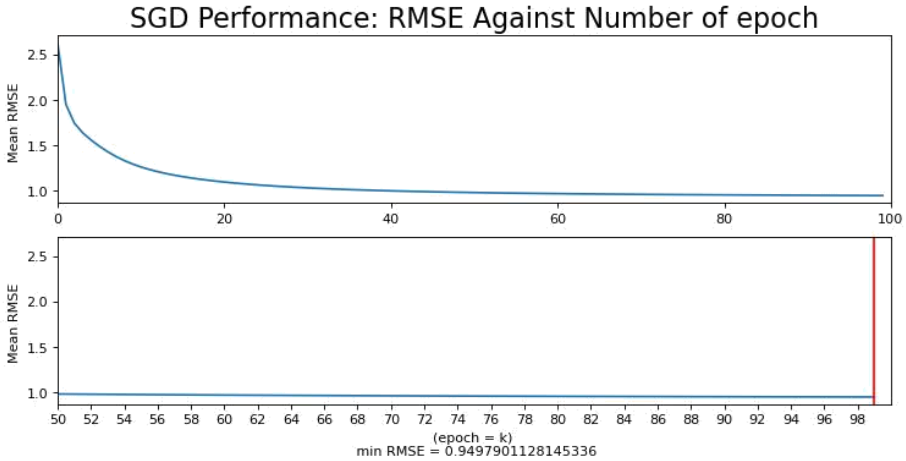


Figure IV.2: RMSE against epoch without enrichment

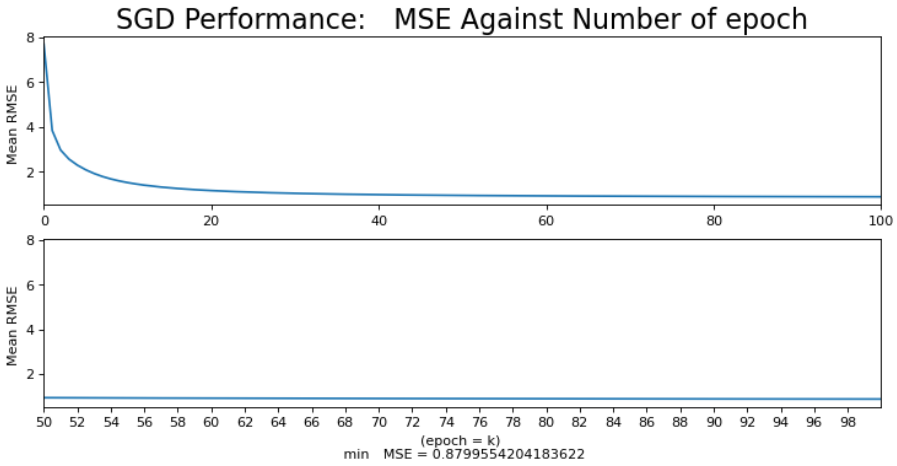


Figure IV.3: MSE against epoch with enrichment

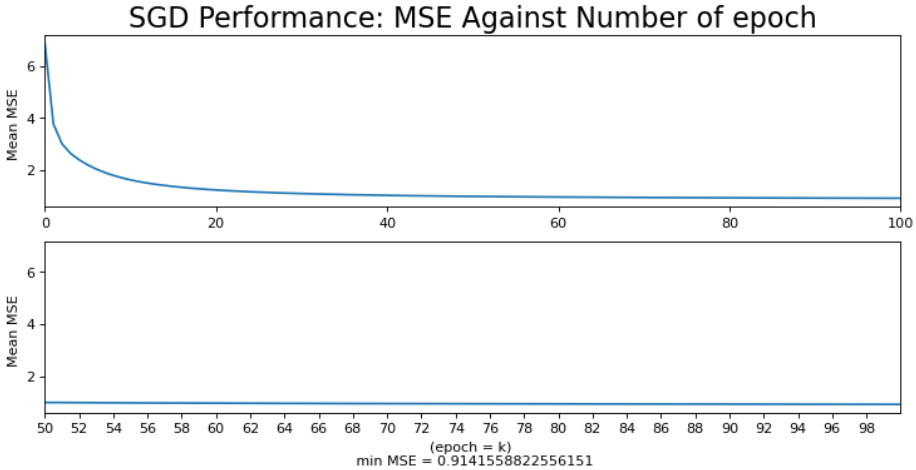


Figure IV.4: MSE against epoch without enrichment

As it is noticed in figure IV.1,figure IV.2, figure IV.3and figure IV.4 the value of the RMSE and MSE are better when using the SGD algorithm with enrichment the best RMSE value detected without enrichment equals 0.94979011 and the RMSE value detected with enrichment equals 0.940818.The MSE value detected without enrichment equals 0.91415588 and the MSE value detected with enrichment equals 0.87995542.It is clear that the enrichment method using SGD algorithm did gave better results.

B. Compare the RMSE and MSE of the SVD before enrichment and after:

Note that the number of factors for the SVD algorithm equals 100 and the number folds in every iteration = 5 (cv=5) and the best number of factors found equals 9.

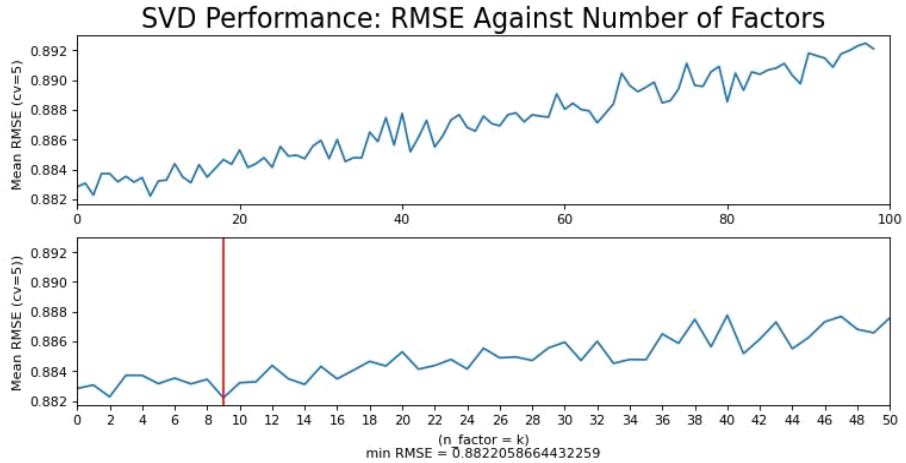


Figure IV.5: RMSE without enrichment

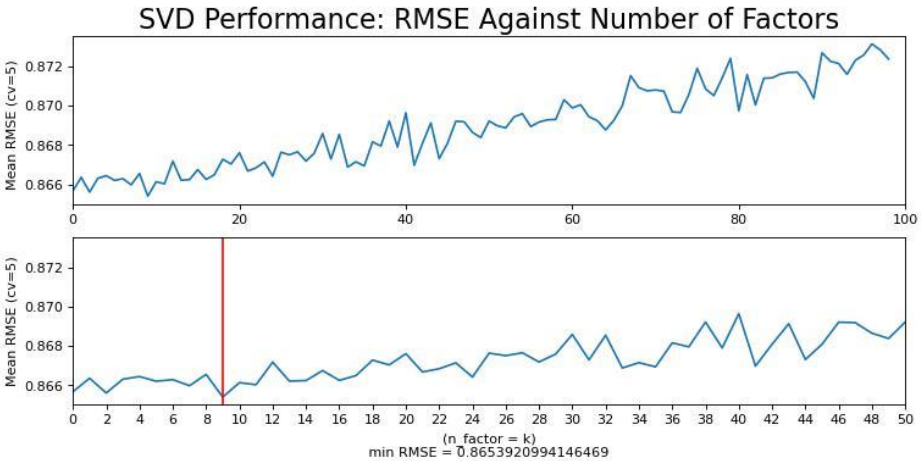


Figure IV.6: RMSE with enrichment

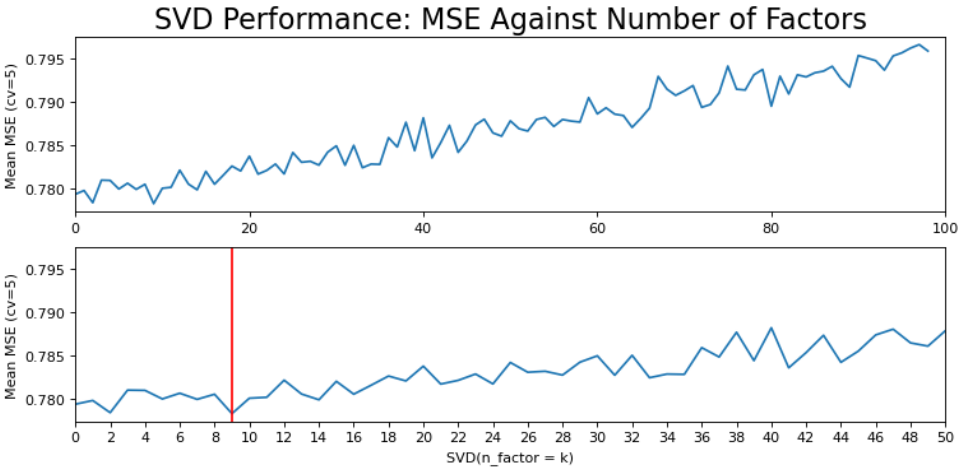


Figure IV.7: MSE without enrichment

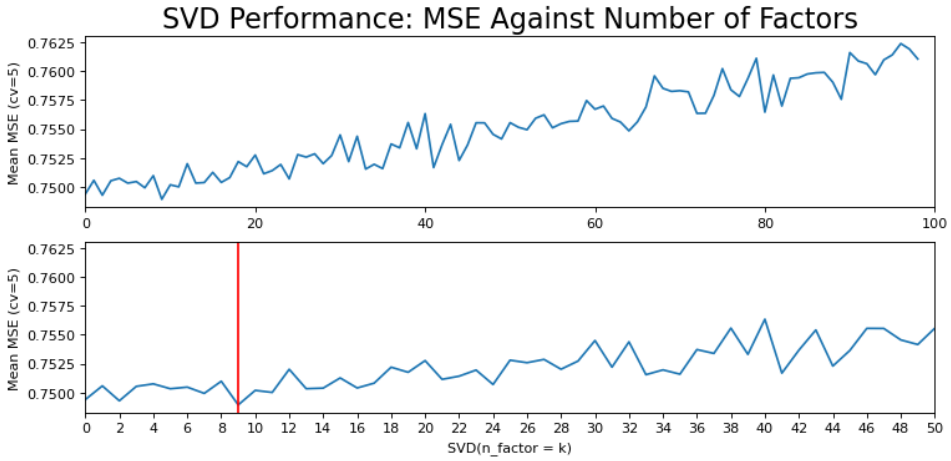


Figure IV.8: MSE with enrichment

As it is noticed in figure IV.5, figure IV.6,figure IV.7 and figure IV.8 the value of the RMSE and MSE are better when using the SVD algorithm with enrichment. The best RMSE value detected without enrichment equals 0.88220586 and the RMSE value detected with enrichment equals 0.86539209; the best MSE value detected without enrichment equals 0.775, the MSE value detected with enrichment equals 0.745. There is a quiet big difference between the two RMSE and MSE when using the enrichment.

C. Compare the RMSE and MSE of the SVD vs SGD

The SVD algorithm gave better RMSE and MSE value then the SGD algorithm in both cases, with enrichment and without enrichment. The values of the RMSE, MSE for the SGD algorithm with enrichment equals 0.940818 and 0.87995542, the values of the RMSE, MSE for the SVD

algorithm with enrichment equals 0.86539209 and 0.740. And the same reaction happens using SGD and SVD without enrichment meaning the SVD is better than the SGD algorithm. In this case, based on the values of the RMSE and MSE and the enrichment using the SVD is also way better than the SGD.

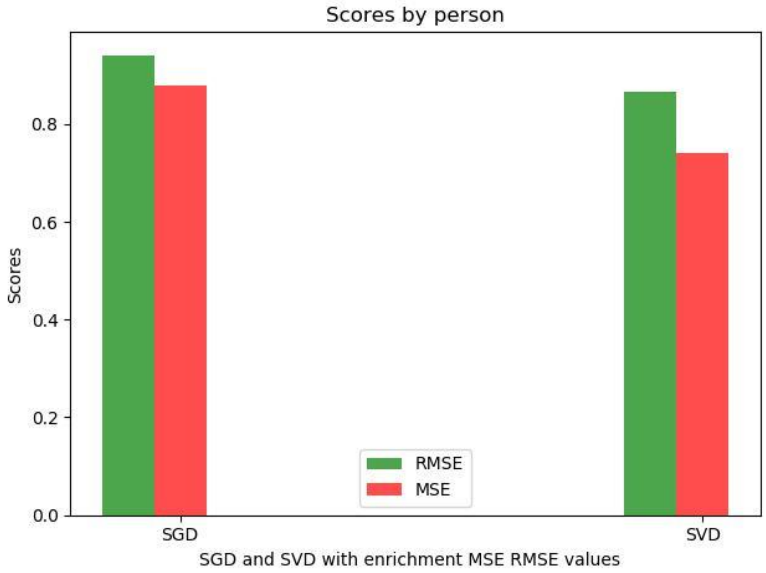


Figure IV.9: RMSE,MSE with enrichment SVD and SGD

D. Compare the RMSE SVD and SGD with other works:

Enrichment = True		Enrichment = False			
SVD	SGD	SVD	SGD	NSVD	NSGD
0.8665	0.9408	08822	0.9497	0.889	0.9425

Table V.1: A comparison of RMSE values of SVD and SGD with and without enrichment and the following works in [93, 94].

IV.2.3 The impact of the enrichment based on user interaction with tweeter:

Thesetests are made based on the following parameters of the SGD and the SVD algorithms:

SVD:

Number of factors = 100

Folds in every iteration = 5 (cv= 5)

SGD:

Number of latent factor =20

Number of iterations = 100

Regularization strength = 0.5

Learning rate =0.01

A comparison of each algorithm is required to make a decision on whether the enrichment system is effective or not in the followingfigures;a comparison of Recall,Precision and F-measure is made.

A. Compare the Precision , Recall,F-measure of the SGD with enrichment and without enrichment:

Precision with enrichment	0.94384944902
Precision without enrichment	0.942703323
Recall with enrichment	0.5493230022
Recall without enrichment	0.534045651
F-measure with enrichment	0.69446528100
F-measure without enrichment	0.68183099

Table IV.2 : Compare the precision recall f-measure of the SGD with enrichment and without enrichment

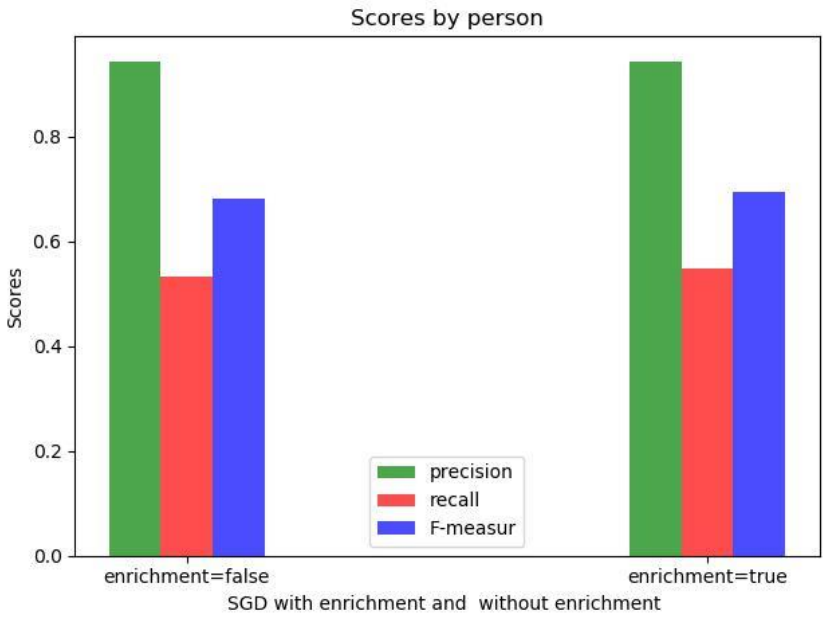


Figure IV.10: SGD with and without enrichment

The values of Precision,Recall and F-measure are all better with enrichment when using the SGD algorithm.

B. Compare the Precision,Recall and F-measure of the SVD with enrichment and without enrichment:

Precision with enrichment	0.9885807504078303
Precision without enrichment	0.8360522022838499
Recall with enrichment	0.7423330965185715
Recall without enrichment	0.6983328944794146
F-measure with enrichment	0.84794076
F-measure without enrichment	0.76101202

Table IV.3: Compare the Precision, Recall and f-measure of the SVD with enrichment and without enrichment

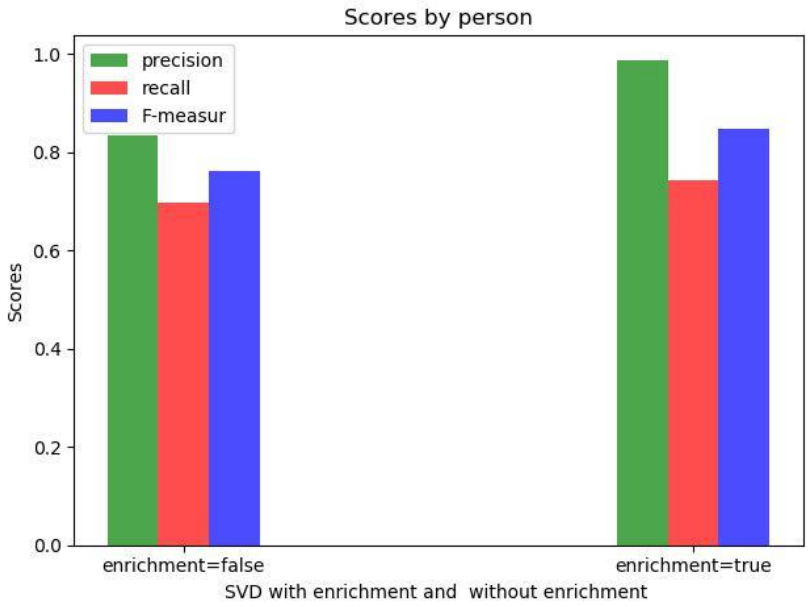


Figure IV.11: SVD with enrichment and without

The values of Precision,Recall and F-measure are all better with enrichment when using the SVD algorithm.

C. Compare the Precision,Recall and F-measure of the SVD and SGD with enrichment:

In every measurement formula (Recall, Precision, F-measure, RMSE), the SDV algorithm is always better than the SGD algorithm, meaning it is better to count on matrix factorization using SVD than matrix factorization using SGD in this particular case.

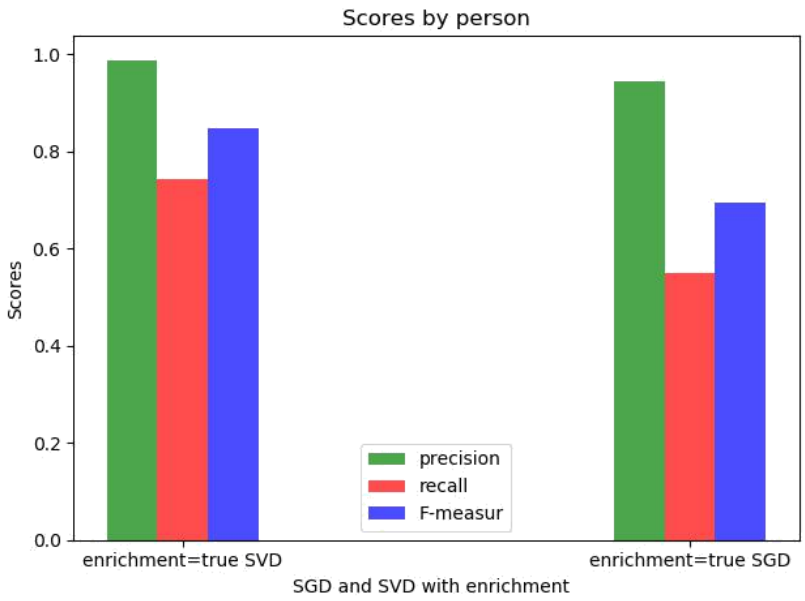


Figure IV.12:SVD and SGD with enrichment

D. Compare the precision recall f-measure of the SVD with enrichment and without enrichmentin k=10 highest estimated ratings :

Precision with enrichment	0.9893
Precision without enrichment	0.8360
Recall with enrichment	0.9740
Recall without enrichment	0.8327
F-measure with enrichment	0.9815
F-measure without enrichment	0.8343

Table IV.4: Compare the precision recall f-measure of the SVD with enrichment and without enrichmentin k=10 highest estimated ratings

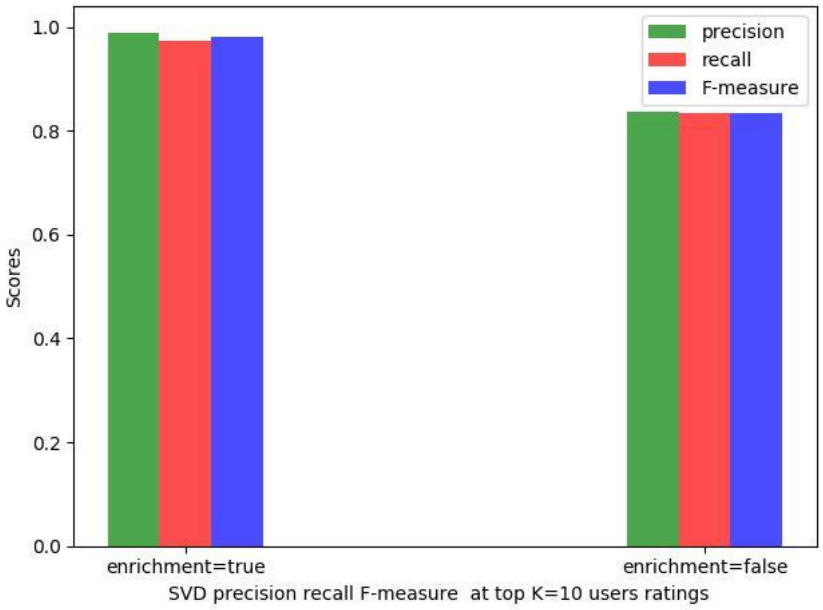


Figure IV.13:SVD at top K=10 users' ratings

IV.3 Conclusion:

The enrichment method used in this approach has positive effects on the recommendation system; the quality of the approach proved his effectiveness on the system. By comparing the different measurement formula and by providing the system with more pertinent data lead us to a recommendation with more pertinent results.

GENERAL CONCLUSION

To conclude the results of using personalized information about new users helped the system to find the relation between a user interest and items which lead to recommend the items for new users with total efficiency and confidence. The proposed approach did prove it major effectiveness on the prediction and recommendation as it is explained during the evaluation section.A simple act of computing and the similarity of users tweets are used to rate the original rating matrix which is the aim of this work .the following work is considered as a new beginning for different researches and will definitely make the system more personalized and we'll end up by finding the main relation between user interest and items ,of course by focusing on the methods previously explained.

REFERENCES

[1] J. Breese, D. Heckerman and C. Kadie.Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Technical Report of Microsoft Research, 1998.

[2] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor,Recommender Systems Handbook, Springer-Science+Business Media, LLC,Springer US 2011.

[3]Koren, Y., & Bell, R. (2011). Advances in Collaborative Filtering.In RecommenderSystems Handbook (pp. 145–186). https://doi.org/10.1007/978-0-387-85820-3

[4] Konstan JA, Riedl J. Recommender systems: from algorithms to user experience. User Model User-Adapt Interact 2012;22:101–23.

[5] Pan C, Li W. Research paper recommendation with topic analysis. In Computer Design and Applications IEEE 2010;4, pp. V4-264.

[6] Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM (2009)

[7]Burke, R.: Hybrid web recommender systems. In: The AdaptiveWeb, pp. 377–408. Springer Berlin / Heidelberg (2007).

[8] Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997).

[9] F. Ricci et al. (eds.), Recommender Systems Handbook, 1 DOI 10.1007/978-0-387-85820-3_1, © Springer Science+Business Media, LLC 2011.

[10]Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40(3), 56–58 (1997).

[11]Jannach, D.: Finding preferred query relaxations in content-based recommenders. In: 3rd International IEEE Conference on Intelligent Systems, pp. 355–360 (2006).

[12] Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM (2009)

[13]McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: KDD ’09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 627–636. ACM, New York, NY, USA (2009)

[14] Burke R. Hybrid recommender systems: survey and experiments. User Model User-adapted Interact 2002;12(4):331–70.

[15] Bobadilla J, Ortega F, Hernando A, Gutierrez A. Recommender systems survey. Knowl-Based Syst 2013;46:109–32.

[16] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Mach Learn 1997;29(2–3):131–63.

[17] Duda RO, Hart PE, Stork DG. Pattern classification.John Wiley & Sons; 2012.

[18] Bishop CM. Pattern recognition and machine learning, vol. 4, no. 4.Springer, New York; 2006.

[19] Shyong K, Frankowski D, Riedl J. Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. In: Emerging trends in information and communication security. Berlin, Heidelberg: Springer; 2006. p. 14–29. [40] Zhang

[20] Adomavicius G, Tuzhilin A. Toward the next generation of recommender system. A survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 2005;17(6):734–49.

[21] Zhang T, Vijay SI. Recommender systems using linear classifiers. J Mach Learn Res 2002;2:313–34.

[22] Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Trans Inform Syst 2004;22(1):5–53.

[23] Breese J, Heckerma D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th conference on uncertainty in artificial intelligence (UAI-98); 1998. p. 43–52.

[24] Zhao ZD, Shang MS. User-based collaborative filtering recommendation algorithms on Hadoop. In: Proceedings of 3rd international conference on knowledge discovering and data mining, (WKDD 2010), IEEE Computer Society, Washington DC, USA; 2010. p. 478–81. doi: 10.1109/WKDD.2010.54.

[25] Zhu X, Ye HW, Gong S. A personalized recommendation system combining case-based reasoning and user-based collaborative filtering. In: Control and decision conference (CCDC 2009), Chinese; 2009. p. 4026–28.

[26] Melville P, Mooney-Raymond J, Nagarajan R. Content-boosted collaborative filtering for improved recommendation. In: Proceedings of the eighteenth national conference on artificial intelligence (AAAI), Edmonton, Canada; 2002. p. 187–92.

[27] Jannach D, Zanker M, Felfernig A, Friedrich G. Recommender systems – an introduction. Cambridge University Press; 2010.

[28] a Department of Mathematical Science, Ekiti State University, Ado Ekiti, Nigeria b Department of Computer Science, University of Ibadan, Ibadan, Nigeria c Department of Computer Science, Federal University of Technology, Akure, Nigeria Received 13 March 2015; revised 8 June 2015; accepted 30 June 2015 Available online 20 August 2015

[29]Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 69–77. ACM, 2011.

[30]] Keshavan RH, Montanari A, Sewoong O. Matrix completion from a few entries. IEEE Trans Inform Theor 2010;56(6):2980–98

[31] Mobasher B, Jin X, Zhou Y. Semantically enhanced collaborative filtering on the web. In: Web mining: from web to semantic web. Berlin Heidelberg: Springer; 2004. p. 57–76.

[32] Yoon HC, Jae KK, Soung HK. A personalized recommender system based on web usage mining and decision tree induction. Expert Syst Appl 2002;23:329–42.

[33] Ku_zelewska U. Advantages of information granulation in clustering al-gorithms. In: Agents and artificial intelligence. NY: Springer; 2013. p. 131–45.

[34] McSherry D. Explaining the pros and cons of conclusions in CBR. In: Calero PAG, Funk P, editors. Proceedings of the European conference on case-based reasoning (ECCBR-04). Madrid (Spain): Springer; 2004. p. 317–30.

[35] Linden G, Smith B, York J. Amazon.com recommendation: itemto- item collaborative filtering. IEEE Internet Comput 2003;7(1): 76–80.

[36] Michael JA, Berry A, Gordon S, Linoff L. Data mining techniques. 2nd ed. Wiley Publishing Inc.; 2004.

[37] Hosseini-Pozveh M, Nemartbakhsh M, Movahhedinia N. A multimedia approach for context-aware recommendation in mobile commerce. Int J Comput Sci Inform Secur 2009;3(1).

[38] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Cohen W. Moore AW, editors. Machine Learning, Proceedings of the twenty-third international conference, ACM, New York; 2003. p. 161–8.

[39] Discovering Knowledge in Data: An Introduction to Data Mining, By Daniel T. Larose ISBN 0-471-66657-2 Copyright C 2005 John Wiley & Sons, Inc.

[40] Berry MJA, Linoff G. Data mining techniques: for marketing, sales, and customer support. New York: Wiley Computer Publishing; 1997.

[41] Deng C, Xiaofe H, Ji-Rong W, Wei-Ying M. Block-level link analysis. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval; 2004. p. 440–7.

[42] Schafer JB, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W, editors. The Adaptive Web, LNCS 4321. Berlin Heidelberg (Germany): Springer; 2007. p. 291–324. http://dx.doi.org/10.1007/ 978-3-540-72079-9_9.

[43] Burke R. Web recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W, editors. The Adaptive Web, LNCS 4321. Berlin Heidelberg (Germany): Springer; 2007. p. 377–408. http://dx.doi.org/10.1007/978-3-540-72079-9_12.

[44] Park DH, Kim HK, Choi IY, Kim JK. A literature review and classification of recommender systems research. Expert Syst Appl 2012;39(11):10059–72.

[45] Su X, Khoshgoftaar TM. A survey of collaborative filtering techniques. Adv Artif Intell 2009;4:19.

[46] Shardanand U, Maes P. Social information filtering: algorithms for automating ‘‘word of mouth’’. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM Press/ Addison-Wesley Publishing Co.; 1995. p. 210–7.

[47] Shardanand U, Maes P. Social information filtering: algorithms for automating ‘‘word of mouth’’. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM Press/ Addison-Wesley Publishing Co.; 1995. p. 210–7.

[48] Adomavicius G, Zhang J. Impact of data characteristics on recommender systems performance. ACM Trans Manage Inform Syst 2012;3(1).

[49] Stern DH, Herbrich R, Graepel T. Matchbox: large scale online bayesian recommendations. In: Proceedings of the 18th international conference on World Wide Web. ACM, New York, NY, USA; 2009. p. 111–20.

[50] Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M. Combining content-based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR workshop on recommender systems: algorithms and evaluation, Berkeley, California; 1999.

[51] Billsus D, Pazzani MJ. A hybrid user model for news story classification. In: Kay J, editor. In: Proceedings of the seventh international conference on user modeling, Banff, Canada. Springer-Verlag, New York; 1999. p. 99–108.

[52] Smyth B, Cotter P. A personalized TV listings service for the digital TV age. J Knowl-Based Syst 2000;13(2–3):53–9.

[53] Wasfi AM. Collecting user access patterns for building user profiles and collaborative filtering. In: Proceedings of the 1999 international conference on intelligent user, Redondo Beach, CA; 1999. p. 57–64.

[54] Burke R, Hammond K, Young B. The FindMe approach to assisted browsing. IEEE Expert 1997;12(4):32–40.

[55] Basu C, Hirsh H, Cohen W. Recommendation as classification: using social and content-based information in recommendation. In: Proceedings of the 15th national conference on artificial intelligence, Madison, WI; 1998. p. 714–20.

[56] Mooney RJ, Roy L. Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on digital libraries. ACM; 2000. p. 195–204.

[57] Pazzani MJ. A framework for collaborative, content-based and demographic filtering. Artific Intell Rev 1999;13:393–408, No. 5(6).

[58] Schwab I, Kobsa A, Koychev I. Learning user interests through positive examples using content analysis and collaborative filtering. Draft from Fraunhofer Institute for Applied Information Technology, Germany; 2001.

[59] E. Rich. User modeling via stereotypes.*Cognitive Science*, 3(4), pp. 329–354, 1979.

[60] B. Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2), pp. 37–45, 1995.

[61] M. Pazzani. A framework for collaborative, content-based and demographic filtering.*Artificial Intelligence Review*, 13, (5–6), 1999.

[62] C. Aggarwal, Z. Sun, and P. Yu.Online generation of profile association rules.*ACM KDD Conference*, pp. 129–133, 1998.

[63] C. Aggarwal, Z. Sun, and P. Yu. Online algorithms for finding profile association rules, *CIKM Conference*, pp. 86–95, 1998.

[64] Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, ACM, Hong Kong, China; 2001. p. 295.

[65] Goldberg K, Roeder T, Gupta D, Perkins C. Eigentaste: a constant time collaborative filtering algorithm. Inform Retrieval J 2001;4(2):133–51.

[66] Cotter P, Smyth B. PTV: Intelligent personalized TV guides. In: Twelfth conference on innovative applications of artificial intelligence; 2000. p. 957–64.

[67] Sarwar BM, Konstan JA, Herlocker JL, Miller B, Riedl JT. Using filtering agents to improve prediction quality in the grouplens research, collaborative filtering system. In: Proceedings of the ACM conference on computer supported cooperative work. New York (NY, USA): ACM; 1998. p. 345–54.

[68] Drosou M, Pitoura E. Search result diversification. SIGMOD Rec 2010;39(1):41–7.

[69] Papagelis M, Plexousakis D. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. Int J Eng Appl Artif Intell 2005;18(4):781–9.

[70] User Profiling Trends, Techniques and Applications Sumitkumar Kanoje, Sheetal Girase, Debajyoti Mukhopadhyay Dept. of Information Techonlogy, Maharashtra Institute of Technology, Pune Jan 2014.

[71] Mathieu, 25.04.2012, in Dossiers, Mathieu Les algorithmes de recommandation,http://www.podcastscience.fm/dossiers/2012/04/25/ les-algorithmes-de-recommandation/

[72] Jie Tang, Limin Yao, Duo Zhang, Jing Zhang “A Combination Approach to Web User Profiling” in ACM Transactions on Knowledge Discovery from Data, Volume 5, No 1, Article 2, New York USA, Dec 2010

[72] Li, Jiwei, Alan Ritter, and Eduard Hovy. "Weakly Supervised User Profile Extraction from Twitter." ACL, Baltimore, 2014.

[73] Hijikata, Yoshinori. "Implicit user profiling for on demand relevance feedback." In Proceedings of the 9th international conference on Intelligent user interfaces, pp. 198-205, ACM, Island of Madeira, Portugal 2004

[74] Kelly, Diane, and Jaime Teevan. "Implicit feedback for inferring user preference: a bibliography." In ACM SIGIR Forum, vol. 37, no. 2, pp. 18-28. ACM, 2003

[75]. International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 1, Issue 1, Jan 2014.

[76] Robertson S. E., Jones K. S., « Relevance weighting of search terms », Journal of the American Society for Information science, vol. 27, no 3, p. 129-146, 1976

[77] Trajkova J., Gauch S., « Improving ontology-based user profiles », Coupling approaches, coupling media and coupling languages for information retrieval, p. 380-390, 2004.

[78] Tchuente D., Canut M.-F., Jessel N., Peninou A., Sèdes F., « A community-based algorithm for deriving users' profiles from egocentrics networks : experiment on Facebook and DBLP », Social Network Analysis and Mining, vol. 3, no 3, p. 667-683, 2013.

[79] Joana Trajkova, Susan Gauch Electrical Engineering and Computer Science University of Kansas {trajkova, sgauch} @eecs.ku.edu

[80] Valafar M., Rejaie R., Willinger W., « Beyond friendship graphs : a study of user interactions in Flickr », Proc. of the 2nd ACM workshop on Online social networks, p. 25-30, 2009

[81] Jin L., Chen Y., Wang T., Hui P., Vasilakos A. V., « Understanding user behavior in online social networks : A survey », IEEE Communications Magazine, vol. 51, no 9, p. 144-150, 2013.

[82] Crystiam Kelle Pereira, Fernanda Campos, Victor Ströele, José Maria N. David &Regina Braga *Journal of Internet Services and Applications* volume 9, Article number: 7 (2018)

[83]Raghuram M.A., Akshay K., Chandrasekaran K. (2016) Efficient User Profiling in Twitter Social Network Using Traditional Classifiers. In: Berretti S., Thampi S., Dasgupta S. (eds) Intelligent Systems Technologies and Applications. Advances in Intelligent Systems and Computing, vol 385.Springer, Cham.https://doi.org/10.1007/978-3-319-23258-4_35

[84] O Phelan, K McCarthy, and B Smyth, "Using twitter to recommend real-time topical news," in Proc. ACM Recsys'09, pp. 385-388, 2009.

[85] J Wang, Z Li, J Yao, Z Sun, M Li, and W Ma, "Adaptive user profile model and collaborative filtering for personalized news," Frontiers of WWW Research and Development-APWeb'06. Springer Berlin Heidelberg, pp. 474-485, 2006.

[86] R Carreira, JM Crato, D Gonçalves JA Jorge, "Evaluating adaptive user profiles for news classification," in Proc. ACM IUI'04, pp. 206-212, 2004

[87] F Abel, Q Gao, GJ Houben, K Tao, "Analyzing user modeling on twitter for personalized news recommendations," User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, pp. 1-12, 2011.

[88] Alexandria University, Computer and Systems Engineering, Alexandria 21544 GIS Technology Innovation Center, Umm Al-Qura University, Makkah, Saudi Arabia Microsoft Research Advanced Technology Lab, Cairo, Alexandria University, SmartCI Research Center, Alexandria 21544, Virginia Tech, Computer Science Department, Blacksburg, VA 24061, USA

[89] This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF2013R1A1A2012627) and the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1001) supervised by the NIPA(National IT Industry Promotion Agency).

[90] Lu, C. & Lam, W. & Zhang, Y..(2012). Twitter user modeling and tweets recommendation based on wikipedia concept graph. AAAI Workshop - Technical Report.33-38.

[91]2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)

[92] Jae-jin Lee College of Information and Communication Engineering Sungkyunkwan University Suwon 440-746, Republic of Korea

[93] Zhouxiao Bao, Haiying Xia; Published 2012. In this paper, we build a movie rating prediction system based on selected training sets provided by MovieLens.

[94] Improving regularized singular value decomposition for collaborative filtering. Arkadiusz Paterek. Institute of Informatics, Warsaw University ul.de A Paterek - 2007

Annex A

What is Tweepy?

The Twitter API exposes dozens of HTTP endpoints that can be used to retrieve, create and delete tweets, retweets and likes. It provides direct access to rich and real-time tweet data, but requires having to deal with a lot of low level details (and very not fun debugging).

Tweepy is an open source package that allows you to bypass a lot of those low level details. Twitter's developer website has great documentation I recommend exploring to get example responses and see the type of data you are able to access. Tweepy's documentation will furthermore provide code snippets and some basic documentation for the Tweepy module.

Getting Twitter API Authentication

Twitter API uses OAuth, which is an open authorization protocol to authenticate requests. You will need to create and configure your authentication credentials to access Twitter API. As promised, this is a step-by-step guide so follow along!

Step 1: Open a Twitter account. If you already have a Twitter account, skip this step

Step 2: Apply for a developer account. Go to their developer site and go to apply for access and select "Apply for a developer account". You will be prompted to log in to your Twitter account.

Select your choice path and fill the details in the next page and fill in some personal details. When you get to intended use, there are a couple of fields with minimum character limit.

Step3: Create your app.

Step 4: Generate your access token and access token secret.

And the Tweepy is ready to use.

Annex B

The following example will demonstrate exactly how to the textrank will extract the keywords.

- Text example: I Like Flipkart. He likes Amazone. She likes Snap deal. Flipkart and amazon are on top of Google search.
- So dictionary of unique word, whose Parts-of-Speech (POS) starts with ’verb’, ’noun’ and ’PNON’.

The content of the figure B is explained here:

- keep: If POS starts with ’v’, ’n’, ’j’ then keep = 1 else 0.
- word id: After filtering out data based on Parts-of-Speech (starting from v, n, j).Word id will be id of those unique words based on appearance in the text.
- idx: It will auto increase word by word.
- POS tag: Parts of speech like Verb, Adjective etc.
- Raw word: Raw word is input word like: “Flipkart”, “Amazone” etc.

Know the graph drawing part comes a head the relations between words comes with the textrank algorithm by choosing pairs and know using the mentioned equation III.1

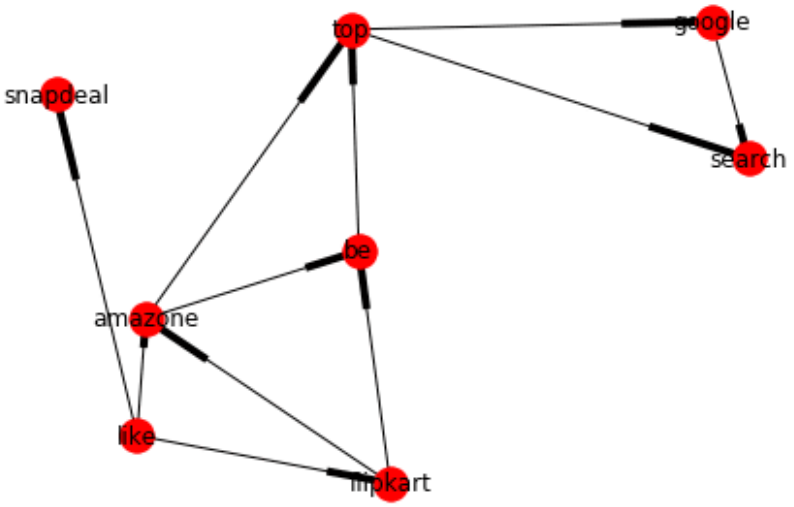


Figure B: graph of words relation

Word_Id	Raw_Word	Lemma_Word	POS	Is_Stop
0	"I"	"i"	"PRP"	False
1	"Like"	"like"	"VBP"	True
2	"Flipkart"	"flipkart"	"NNP"	False
0	"."	"."	"."	False
0	"He"	"he"	"PRP"	False
1	"likes"	"like"	"VBZ"	True
3	"Amazone"	"amazone"	"NNP"	False
0	"."	"."	"."	False
0	"she"	"she"	"PRP"	True
1	"likes"	"like"	"VBZ"	True
4	"Snapdeal"	"snapdeal"	"NNP"	False
4	"Snapdeal"	"snapdeal"	"NNP"	False
0	"."	"."	"."	False
2	"Flipkart"	"flipkart"	"NN"	False
0	"and"	"and"	"CC"	True
3	"amazone"	"amazone"	"JJ"	False
5	"is"	"be"	"VBZ"	True
0	"on"	"on"	"IN"	True
6	"top"	"top"	"NN"	True
0	"of"	"of"	"IN"	True
7	"google"	"google"	"NNP"	False
8	"search"	"search"	"NN"	False
0	"."	"."	"."	False

Table B: Data and divided words

Annex C

The first one called ml-25 describes 5-star rating and free-text tagging activity from [MovieLens](http://movielens.org), a movie recommendation service. It contains 25000095 ratings and 1093360 tag applications across 62423 movies. These data were created by 162541 users between January 09, 1995 and November 21, 2019. This dataset was generated on November 21, 2019.

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

Movie Summary Corpus: the second used data set the CMU Movie Summary Corpus, a collection of 42,306 movie plot summaries and metadata at both the movie level (including box, revenues, genre and date of release) and character level (including gender and estimated age).

a. plot summaries.txt.gz [29 M]

   Plot summaries of 42,306 movies extracted from the November 2, 2012 dump of English-language Wikipedia. Each line contains the Wikipedia movie ID (which indexes into movie.metadata.tsv) fol-lowed by the summary.

b. corenlp plot summaries.tar.gz [628 M, separate download]

The plot summaries from above, run through the Stanford CoreNLP pipeline (tagging, parsing, NER and coref). Each lename begins with the Wikipedia movie ID (which indexes into movie.metadata.tsv).

c. movie.metadata.tsv.gz [3.4 M]

- Wikipedia movie ID
- Freebase movie I
- Movie name
- Movie release date
- Movie box o ce revenue
- Movie runtime
- Movie languages (Freebase ID:name tuples)
- Movie countries (Freebase ID:name tuples)
- Movie genres (Freebase ID:name tuples)

d. character.metadata.tsv.gz [14 M]

Metadata for 450,669 characters aligned to the movies above, extracted from the November 4, 2012 dump of Freebase. Tab-separated; columns:

- Wikipedia movie ID
- Freebase movie ID
- Movie release date
- Character name
- Actor date of birth
- Actor gender
- Actor height (in meters)
- Actor ethnicity (Freebase ID)
- Actor name
- Actor age at movie release
- Freebase character/actor map ID
- Freebase character ID
- Freebase actor ID