Introduction

Recommender systems are an important part of the informative websites, commercial websites and e-commerce system. Recommender system plays an important role in the social media like Facebook, Twitter, YouTube etc. Movie recommendation system aims to provide the best movie recommendation using the past data where in algorithms are used to evaluate the results .They represent a powerful method for enabling users to filter through large information and product spaces. Nearly two decades of research on collaborative filtering have led to a varied set of algorithms and a rich collection of tools for evaluating their performance.

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access and more recently generated technologies that allow users to navigate through their data in real time. Data mining takes this evolutionary process beyond retrospective data access and navigation to prospective and proactive information delivery. Data mining tools predict future trends and behaviors & allowing businesses to make proactive & knowledge-driven decisions. The automated, prospective analyzes offered by data mining move beyond the analyzes of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations

Effective deployments began with careful analysis of prospective users and their goals. Based on this analysis, system designers have a host of options for the choice of algorithm and for its embedding in these surrounding user experience. Hence to overcome these problems our aimed hybrid algorithm takes the general movie dataset, metadata and the particular data of single movie to recommend the movie which may come to future.

Problem Life Cycle

2.1. Problem Identification:

The outbreak of information in 21st century has led to overgrowth of data and possible choices that one can have. Which movie should I watch next? Which book should I read? Which link should I click next? We all are inundated with such questions all the times. This decision making process has especially become serious in today's world as people can find everything on Internet.

Recommendation systems are used to suggest information products and services to the regular customers based on the history, transactions and feedback [3]. Here the similarity between user or item Is used. In the growing field of Big-data, the number of products, customers and providers are increasing tremendously. Hence recommendation system become need and a growing challenge to produce results.

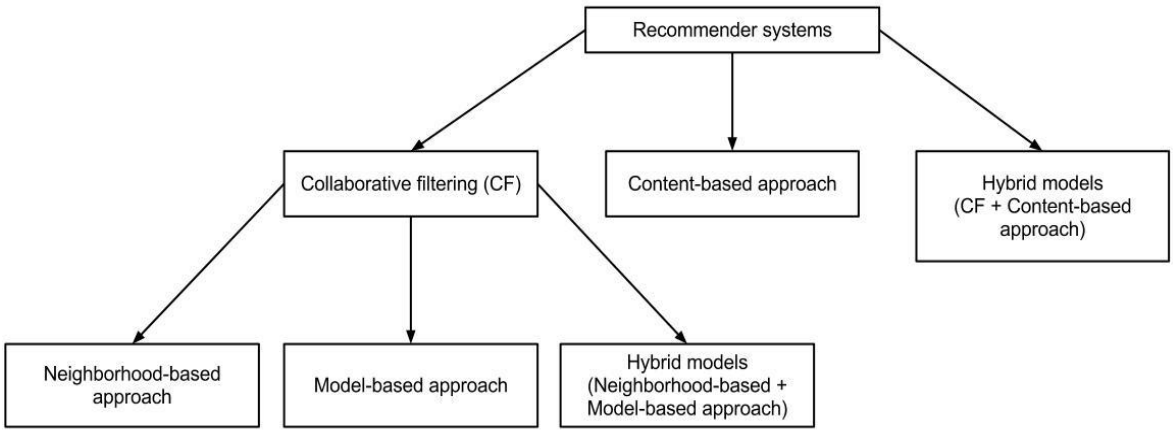


Fig.2.1. Types of recommender system

Description: The above figure shows the classification of recommender systems. They are classified mainly into collaborative filtering and content based approach and the combination of both is called as hybrid models.

2.2. Problem Selection:

Many recommender systems recommend items to user are CF technologies [5].Major problems of CF are scalability, cold start, sparsity which can be reduced with the help of hybrid systems using combination of different algorithms.

2.3. Problem Definition:

2.3.1. Problem Formulation

Movie recommendation can be given a Movie under consideration & presents recommend to similar entities. Given the large number of related entities in the knowledge base, we need to select the most relevant ones to show based on the current query of the user. Movie ranking helps to retrieve the entities, which are most relevant, based on popularity, authority, relevance etc. Following data states about input, output, data, Movie and problem that this project addresses.

Input: A Movie

Output: Recommended entities given the input Movie

Data: semi structured or unstructured data

Movie: represent an object, structured data

The problem is to provide a solution to build a recommendation engine using Big Data processing technology, Apache Spark. The goal of a Movie recommendation system over big data is to design a system that is scalable, efficient, and provides the best possible results for a large variety of queries which can solve scalability, cold start, sparsity problems. Challenges that a Movie recommendation system over big data faces are stated as below:

1. Unstructured Data: Movie recommendation over big data involves processing and storing the vast amount of unstructured data[5].

2. Movie Resolution and Movie Disambiguation: “Brad pitt” may refer to the actor Movie “Brad Pitt” or the boxer Movie “Brad Pitt (boxer)”. Moreover, there may be cases with a common meaning for the string (e.g., the Movie “XXX (movie)” is not the most likely intent for query string “xxx”). Hence, the problem here is to identify the most likely intent for a given Movie string.[1]

3. Ranking: For a given Movie, not all results might interest the user. We need to rank the results. There are many ranking mechanisms based on various features such as click frequency, PageRank etc[5].

2.3.2 Objectives:

The following are the list of objectives:

- To present a new hybrid solution for research purpose
- To alleviate the scalability, cold start, sparsity problem of Collaborative Filtering by correlating the users to products through features
- To provide additional and probably unique personalized service for the customer
- To develop the hybrid system which helps to user to get best recommendation and faster

2.3.3. Terminology

The following terms are widely used in the report:

● *Movie*: a concept or abstract that has a complete meaning by itself. In this project, Movie represents an object with unique id and properties. Movie may include but not limited to persons, subjects, records, concepts

● *Similarity*: denotes the relevancy between a Movie and a query, as a numerical value computed by a similarity functions. The higher the value, the more closely a Movie relates to the query

●*Popularity*: the concept that measures the credential of the Movie, how popular is a particular Movie compared to the common ground of all other entities

●*Field/Property*: an attribute of a Movie

2.4Problem Analysis:

Why analysis

- Why Recommender Systems?
    1. Given the increasing amount and growing variety of products, services and information, which are daily made available on the Web, and the rapid introduction of new e-business services, making a choice from such wide range of options can be somewhat complex and difficult to manage.
    2. It can be argued that, while being able to choose is good, having so many options to choose from is not always more rewarding.
    3. To prevent overwhelmed users from making poor decisions, recommender systems came into focus, facilitating users' access to information about the items they are most likely to

be interested, whether such items are books, movies, music, videos, Web pages, news or services, among others.

4. Recommender systems provide users an efficient way for reviewing their options and selecting the most suitable items, since attempting to infer their preference and recommend new items that maximizes the probability of fitting each of the users' preferences.

- Why Hybrid Systems?
    1. The hybrid approach combines the aforementioned approaches to increase the efficiency of a recommender system.
    2. Hybrid approach also has the potential to make recommender system more accurate.
    3. The collaborative filtering and content based filtering face the challenge of cold start. Hybrid approach can address this to some extent.
    4. In case of hybrid approach, the system starts with content based filtering and gradually switches the focus towards collaborative filtering as the database for user information matures.
- Why Apache Spark?
    1. Apache spark is an Open Source data analytics cluster computing framework .Spark belongs to Hadoop Open Source community.
    2. It is built on top of Hadoop Distributed File System. Although such similarities exist, Spark performs better than Hadoop in case certain specific applications.
    3. Spark is not restricted by two stage Map reduce paradigm. It delivers 100 times better performance than Hadoop for certain applications. Spark provides the capabilities for in memory cluster computing which allows user programs to load data into cluster's memory.
- Why Anaconda?
    1. Anaconda is a freemium open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment.

2. Control and Manage Data Science Assets: Data Science Governance means enterprises can rest assured that their policies and controls can be complied with for their data science assets.

- Why Scala & Python?
    1. Scala is an acronym for “Scalable Language”. This means that Scala grows with you. You can play with it by typing one-line expressions and observing the results.
    2. But you can also rely on it for large mission critical systems, as many companies, including Twitter, LinkedIn, or Intel do.
    3. It has a great package manager. It comes with a lot of useful things preinstalled and ready to run for a typical machine learning project.
    4. You can do nearly everything in anaconda vs standard python, but be prepared to waste a lot of time installing packages and making them run.
- Why Movie lens Dataset?
    1. Due to copyright, Netflix data is not available for download. So, to perform the recommendation evaluation on the movies domain, the Movie Lens data is used.
    2. The Movie Lens dataset consists of anonymous ratings of movies collected by the Group Lens Research that currently operates a movie recommender based on collaborative filtering .
    3. The Movie Lens data set contains approximately 10 million ratings from 71,567 users on 10,681 movies. Ratings are made on a 5-star scale (whole-star ratings only) and each user has at least 20 ratings.
    4. The data set was collected and made available by Group Lens Research at their webpage.

2.5. End Users

- Movie Feedback Providing Agencies
- Website Developers
- Users surfing internet
- Third party service providers
- Researchers
- Every person using suggestion based websites.

Literature Survey:

3.1. Basic Approaches for Recommender Systems

Recommender Systems generally take one of the two approaches: Collaborative Filtering or Content Based Filtering. Some recommender systems also take the hybrid approach of combination of these two approaches.

3.1.1. Collaborative Filtering

In case of this approach, the recommendation is based on model of previous user behavior. Recommendations given by collaborative filtering are based on automatic collaboration of multiple users and filtered by users with similar tastes. The model can be built based on behavior of single user or it can be based on behavior of group of users who have similar taste. When the model is based on group knowledge, it takes into consideration the preferences put out by a group of users who have similar taste as you and based on these preferences, makes a new recommendation [1].

For example, suppose a recommendation engine for videos on video providing service like YouTube or Netflix is being built. To do so, information from all users who subscribe and use these services can be used. Users with similar preferences can be grouped together. Using this information, most popular video for the group can be decided and can be recommended to other members in that group who have not watched the video.

Following table explains how collaborative filtering is used for video recommendation. The entry in each cell represents how many videos of a particular genre have been watched by particular user.

Here, this group of users can be clustered together as they have similar interest in many of the video gener. Based on this information, we can recommend Cristina to watch videos under comedy gener, Preston to watch videos under drama genre and Ellen, the Sci-Fi genre. Collaborative filtering can also be defined using similarity-difference approach. Users with similar taste are grouped together and differences in their tastes are potential areas for recommendation.

Video Genre/ Users	**Cristina**	**Preston**	**Ellen**
Comedy	--	9	12
Drama	10	--	15
Sci-Fi	8	11	--

Table No. 1: Simple example of Collaborative Filtering

3.1.2. Content Based Filtering

In this model, recommendations are given based on user's behavior. User's browsing information is taken into account while recommendations are made. For example, if user visits the videos in Comedy category more, it is more likely that he or she would watch the next video under comedy genre. Content based filtering recommends similar content to the user for which he has expressed interest in the past. For example in the above example of collaborative filtering, it is clear that Preston has interest in watching videos in Sci-Fi category. So the content based filtering would recommend him similar movies/videos. The recommendations are based on behavior of the user under consideration and is independent of behavior of other users of the system.

3.1.3. Hybrid Approaches

The hybrid approach combines the aforementioned approaches to increase the efficiency of a recommender system. Hybrid approach also has the potential to make recommender system more accurate. The collaborative filtering and content based filtering face the challenge of cold start. Hybrid approach can address this to some extent. In case of hybrid approach, the system starts with content based filtering and gradually switches the focus towards collaborative filtering as the database for user information matures.

3.2. Basic Algorithms for Recommender Systems

3.2.1. Memory based Algorithms

These algorithms try to user that is similar to active user. This algorithm uses the preferences by similar users in order to recommend something to active user.[2] In order to find the similarity between two users we need to find their correlation.

3.2.1.1. Pearson Correlation

Pearson Correlation coefficient gives an idea about how two entities are correlated with each other. This algorithm measures the linear dependence between two variables (or users) as a function of their attributes [1]. However, this dependence is not calculated on the entire dataset. Instead, it is calculated on sub-groups or neighbor hoods of dataset which are similar to each other on higher level. For example, group of users who have interest in comedy genre video.

The Pearson correlation coefficient is calculated as:

$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}}$$

Where:

N	=	number of pairs of scores
Σxy	=	sum of the products of paired scores
Σx	=	sum of x scores
Σy	=	sum of y scores
Σx^2	=	sum of squared x scores
Σy^2	=	sum of squared y scores

Figure 3.2: Pearson Correlation Coefficient

The following graphs explain as what does the positive, no or negative correlation means.

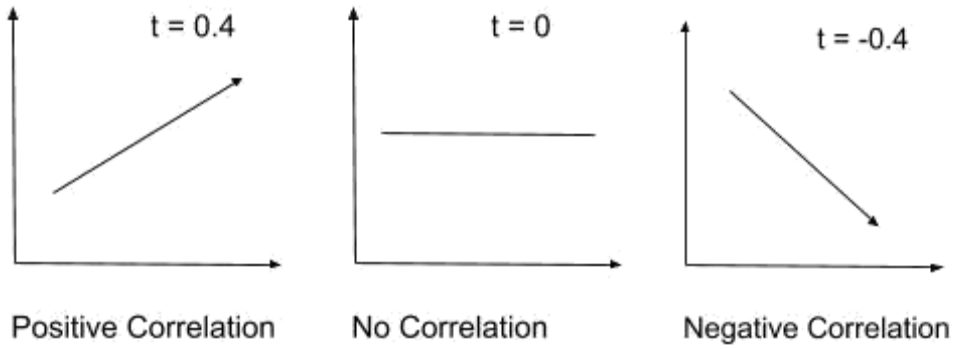


Figure 3.3: Pearson Correlation

3.2.1.2. SVD Algorithm

Singular Value Decomposition (SVD) and the closely-related Principal Component Analysis (PCA) are well established feature extraction methods that have a wide range of applications. Oracle Data Mining implements SVD as a feature extraction algorithm and PCA as a special scoring method for SVD models.

1)Matrix Manipulation

$$\mathbf{X = USV'}$$

-SVD is a factorization method that decomposes a rectangular matrix X into the product of three matrices.

-The U matrix consists of a set of 'left' orthonormal bases

-The S matrix is a diagonal matrix

-The V matrix consists of set of 'right' orthonormal bases

SVD essentially performs a coordinate rotation that aligns the transformed axes with the

directions of maximum variance in the data. This is a useful procedure under the assumption that the observed data has a high signal-to-noise ratio and that a large variance corresponds to interesting data content while a lower variance corresponds to noise

2)Scalability by SVD

In Oracle Data Mining, SVD can process data sets with millions of rows and thousands of attributes. Oracle Data Mining automatically recommends an appropriate number of features, based on the data, for dimensionality reduction.

SVD has linear scalability with the number of rows and cubic scalability with the number of attributes when a full decomposition is computed. A low-rank decomposition is typically linear with the number of rows and linear with the number of columns. The scalability with the reduced rank depends on how the rank compares to the number of rows and columns. It can be linear when the rank is significantly smaller or cubic when it is on the same scale.

3.2.1.3. Slope-one Algorithm

Slope One is a family of algorithms used for collaborative filtering, introduced in a 2005 paper by Daniel Lemire and Anna Maclachlan. Arguably, it is the simplest form of non-trivial item-based collaborative filtering based on ratings. Their simplicity makes it especially easy to implement them efficiently while their accuracy is often on par with more complicated and computationally expensive algorithms. They have also been used as building blocks to improve other algorithms. They are part of major open-source libraries such as Apache Mahout and Easyrec.

When ratings of items are available, such as is the case when people are given the option of ratings resources (between 1 and 5, for example), collaborative filtering aims to predict the ratings of one individual based on his past ratings and on a (large) database of ratings contributed by other users.

Example: Can we predict the rating an individual would give to the new Celine Dion album given that he gave the Beatles 5 out of 5?

In this context, item-based collaborative filtering [10][11] predicts the ratings on one item based on the ratings on another item, typically using linear regression (f (x) = a x + b {\displaystyle f(x)=ax+b}). Hence, if there are 1,000 items, there could be up to 1,000,000 linear regressions to be learned, and so, up to 2,000,000 regressors. This approach may suffer from severe overfitting unless we select only the pairs of items for which several users have rated both items.

3.3. Literature Survey of Papers:

Data mining techniques area unit the results of a protracted method of analysis and products development. This evolution began once business information was initial keep on computers, continuing with enhancements in information access , and a lot of recently, generated technologies that permit users to navigate through their information in real time. Data mining takes this biological process on the far side retrospective information access and navigation to prospective and proactive data delivery. Data processing tools predict future trends and behaviors, permitting businesses to create proactive, knowledge-driven choices. The machine-driven, prospective analyzes offered by data processing move on the far side the analyzes of past events provided by retrospective tools typical of call support systems. Data processing tools will answer business queries that historically were too time intense to resolve. They scour databases for hidden patterns, finding prophetic data that specialists could miss as a result of it lies outside their expectations.

The paper is on "Movie Recommendation Engine using Collaborative Filtering" by Sadanand Howal, Vrushali Desai, Rohan Neralekar, Avadhut Mote, Rushikesh Vanjari, Harshada Rananaware. In this paper the authors use a different collaborative algorithms to achieve efficient outputs. The main aspects of paper is to managing large amount of data and information and testing both trained data and tested data to give best recommendations. The Apache spark is the massive framework used in this project. They implemented basic algorithms in java using mahout libraries & Eclipse. In this paper authors implemented algorithms in Python language because it processes efficient results than java language.

The major literature survey is done by data science community. The book –DATA MINING – Concepts and Techniques by Jiewei Han has mentioned various collaborative filtering techniques. [5] By reading and analyzing IEEE conference paper by Sasmita Panigrahi, Rakesh Ku.Lenka and Anaya Stitipragyan "A Hybrid Distributed Collaborative Filtering Recommender Engine using Apache Spark" we got an idea of recommendation system using collaborative filtering.[10] We are going to execute the idea on Apache Spark. The paper "Hybrid web recommender systems" written by Robin Burke talks about variety of algorithms have been for getting recommendation, including content based collaborative and other algorithms. This paper surveys the hierarchical pattern of actual and possible hybrid recommenders, and introduces a great hybrid system that combines the knowledge based recommendation and collaborative filtering to recommend restaurant. It purely relies on the physical characteristics of the user. [3]

The paper "Clustering Methods for Collaborative filtering" by authors Lyle H. Ungar and Dean P.

Foster discusses about a new method of grouping and clustering of items. The methods used in this paper to optimize the model estimations give the best base rates and link probabilities. Hence, for better results of algorithm it is necessary to use appropriate input methods. [6] The review article "A New Parallel Item-Based Collaborative Filtering Algorithm Based on Hadoop" by Qun Liu, Xiaobing Li talks about various features extraction methods that can be used for filtering using parallelization design for Item-Based Collaborative Filtering. They describe various features and feature extraction methods. The overall article gives the idea about constructing user's preference vectors and computing co-occurrence matrix. [11]

We aim to study these methods and use the best suited one among them and combine the best suited techniques to achieve the better results. The authors Reena Pagare and Shalmali A. Patil in their paper titled as "Study of Collaborative Filtering Recommendation Algorithm – Scalability Issue" talk about statistical approach of problems in filtering techniques of datasets. The paper discusses about challenges in recommendation system. So from this paper we get an idea to analyze the scalability of the algorithms. They talk about challenging problem of collaborative filtering. [9]

The Paper "Recommendation System Based on Collaborative Filtering" by Zheng Wen gives the proper idea to connect the choice of characteristics based on "matching" of user's profile specific characteristics of an item .Collaborative filtering algorithm such as sparse matrix SVD approach model both user's and movies by giving them coordinates in a low dimensional feature space.[8]

The authors A.H.M Ragab, A.F.S. Mashat and A.M.Khedra in their paper entitled"HRSPCA: Hybrid Recommender System For Predicting College Admission" talsk about hybrid recommender based on data mining techniques and knowledge discovery rules for tracking college admission problems. This paper gives an absolute idea about high prediction accuracy rate, flexibility in advantage as the clustred hybrid algorithms to perform attributes task faster and fairly. [12]

Proposed System, Requirement Specification & Modules

4.1. Proposed System & Modules:

1) Module 1: Implementation of basic algorithms in java, using eclipse & mahout.

2) Module 2: Installation of software which are needed for Recommendation System.

3) Module 3: Study and research of Collaborative filtering

4) Module 4: Study and research regarding Content based recommendation.

5) Module 5: Study of Apache Spark, Scala, Hadoop.

6) Module 6: Study & implementation of effective new recommendation algorithms combination such as:

- Tanimoto Algorithm
- Pearson Algorithm
- Slope Algorithm
- SVD Algorithm

7) Module 7: Implementation of above hybrid algorithm using Anaconda framework in Python language.

8) Module 8: Testing the sparsity using Root Mean Square Error in percentage and creating Test cases.

4.2. Software/Hardware Requirements Specification:

Softwares:

Ubuntu 14.04 operating system running on 2.50GHz processors with 4 processing cores. The master node of a cluster was allocated 4GB RAM while each slave node was allocated 2GBRAM

1. Apache Hadoop-2.7.2
2. Apache Hive 2.0
3. Spark-1.6.0
4. Scala -2.11.7
5. SBT-0.13.9
6. Anaconda2-4.3
7. Python 2.7

Hardware:

Hardware Name	Quantity
Laptop(Master)	1
PCs(Slaves)	2(min)

Table No. 2 .Hardware Requirements

4.3. Significance of the project:

4.3.1. Innovation & Usefulness

This project implements a recommendation engine using Apache Spark & Anaconda. Results from this project indicates the advantages of using Apache Spark for distributed, big-data processing especially for the algorithms which are iterative in nature. The recommendation engine can handle large amount of data as Apache Spark is a big-data handling technology that has inbuilt properties for processing iterative and interactive algorithms like List-Net. It is faster than traditional Map-reduce paradigm.

4.4. Scope of Project:

This project and its result can be helpful for future research work by modifying the algorithm. It is used by customers to get recommendation on various fields like ecommerce, dating sites, ratings sites, informative sites and other purposes where recommendation is needed. The providers like amazon, flip kart, Netflix, informative websites can also use the recommendation systems.

4.5. Deployment Requirements:

The deployment requirements of the project are as above mentioned in 4.2. Software/Hardware Requirements Specification.

4.6. Project Cost Estimation:

The project uses the open-source software for it implementation and research work. Hence the cost of project is nil. But if we take into consideration the hours/person criteria work in the project, the estimation can vary from person to person depending on the efficiency of that person.

4.7. Project Deliverables:

- Report hard & soft copy
- Progress report/ Diary
- Software’s and tools required

- Source code files
- Datasets used Queries
- Published paper & certificates of competition

4.8. Project Success:

- **Deployment:** Successfully completed the project 100% with deliverables of two research papers, one for each phase.
- **Internship:** Successfully completed the internship under Mr. Sumant Ghadge in ETech Media, Pune.
- **Product development:** This research project can be implemented anywhere in e-commerce product development.
- **Achievements:**
    - **Publications**: Presented paper on "Movie Recommendation using Collaborative Filtering" in Springer sponsored 'Smart Computing & informatics (SCI) 2017' Organized by ANITS on 3rd& 4th March 2017 in Vishakhapatnam. It is in the proceeding of getting published in August edition of Springer 2017.
    - **Prize**: Won Runner up prize for Paper presentation- RIVISTA in "INFINITY'17" organised by WCE-ACM chapter at Walchand College of Engineering.
    - **Participation**: Our project and paper were highly appreciated at :
        - "INNOVATION 2k17" at KIT's College Of Engineering Kolhapur
        - "Girls Project Competition 2017" at RIT, Islampur
        - "Quantum 2017" at RIT, Islampur

Design

5.1 Class Diagram

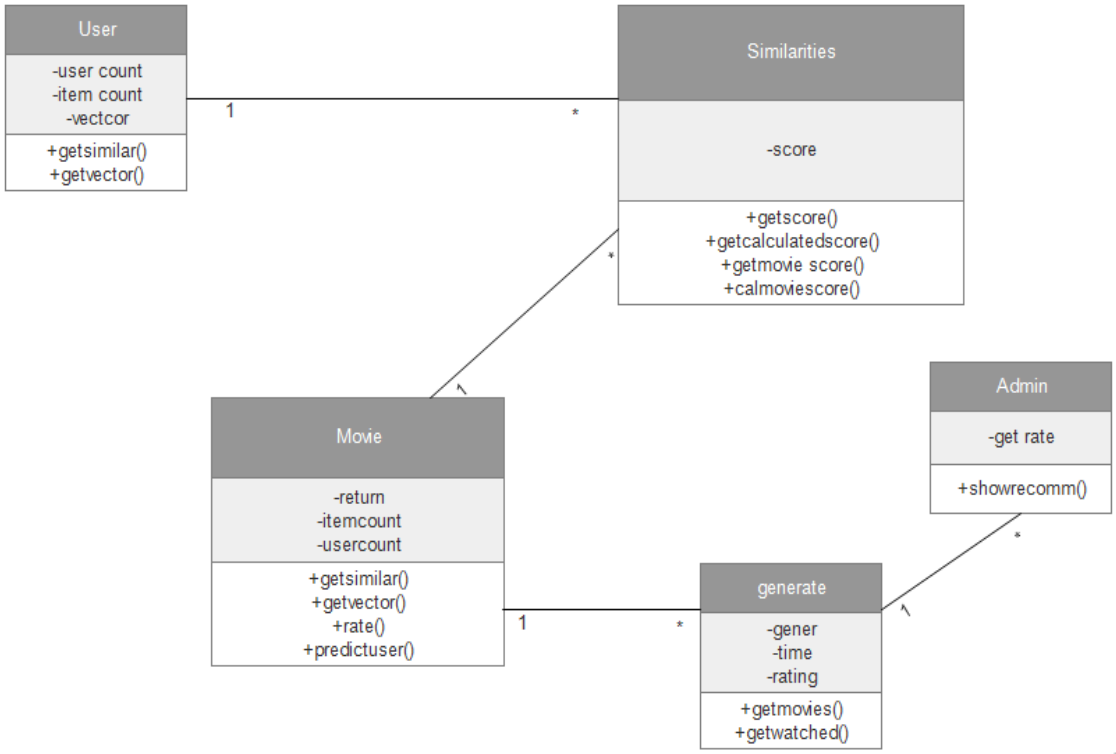


Fig.5.1 Class Diagram

Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. From above fig5.1 user is main class it describes the no of users and count of items and it can perform operations like get similarities in ratings

Admin generate movie similarities these are different classes with its attributes like score rating time genre and performs operations like watched movies score of movie.

5.2 Block Diagram

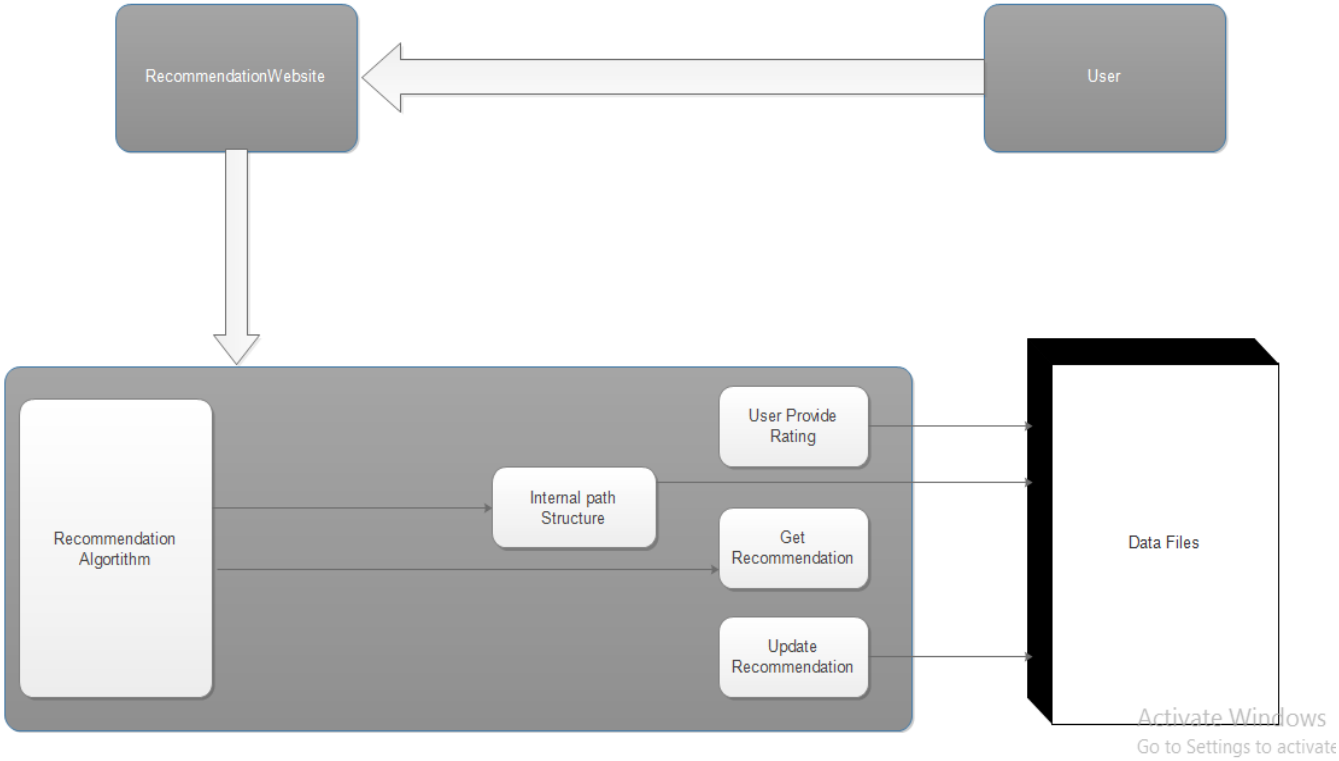


Fig.5.2 Block Diagram

Block diagram is contained in a "What is a Diagram" area of ConceptDraw Solution Park. Use the libraries from the Block Diagrams solution to draw block diagrams for your business documents. From fig 5.2 describes the connection between user to website of recommendation which contain recommendation algorithms which provides user rating internal path updating of recommendation which is connected with data sets.

5.3 ER Diagram

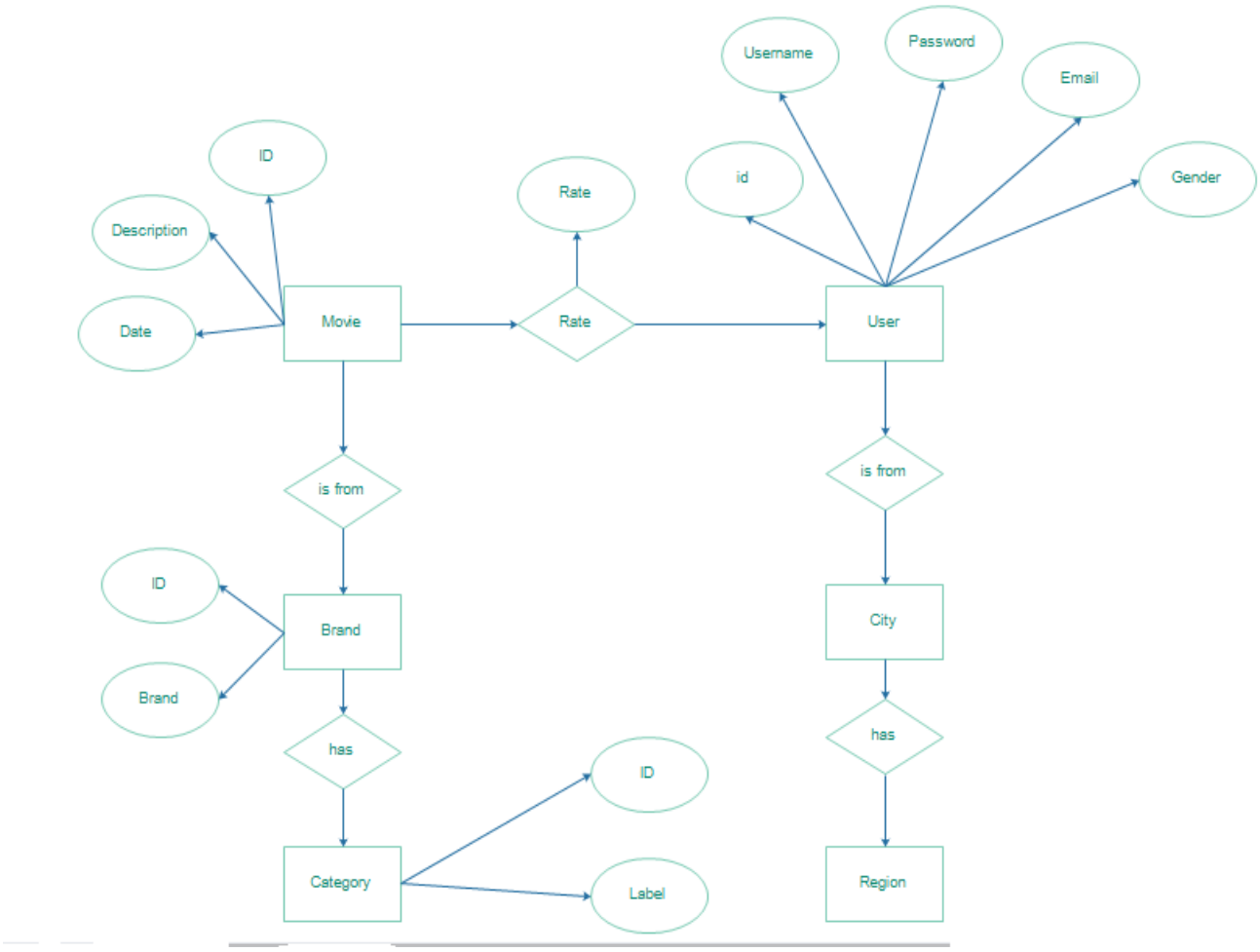


Fig.5.3 ER Diagram

ER diagram is the collection of controls with its modules .from diagram 5.3 movie rate user are main modules and Brand ,category ,city, region, are sub modules.

5.4 Flow Chart

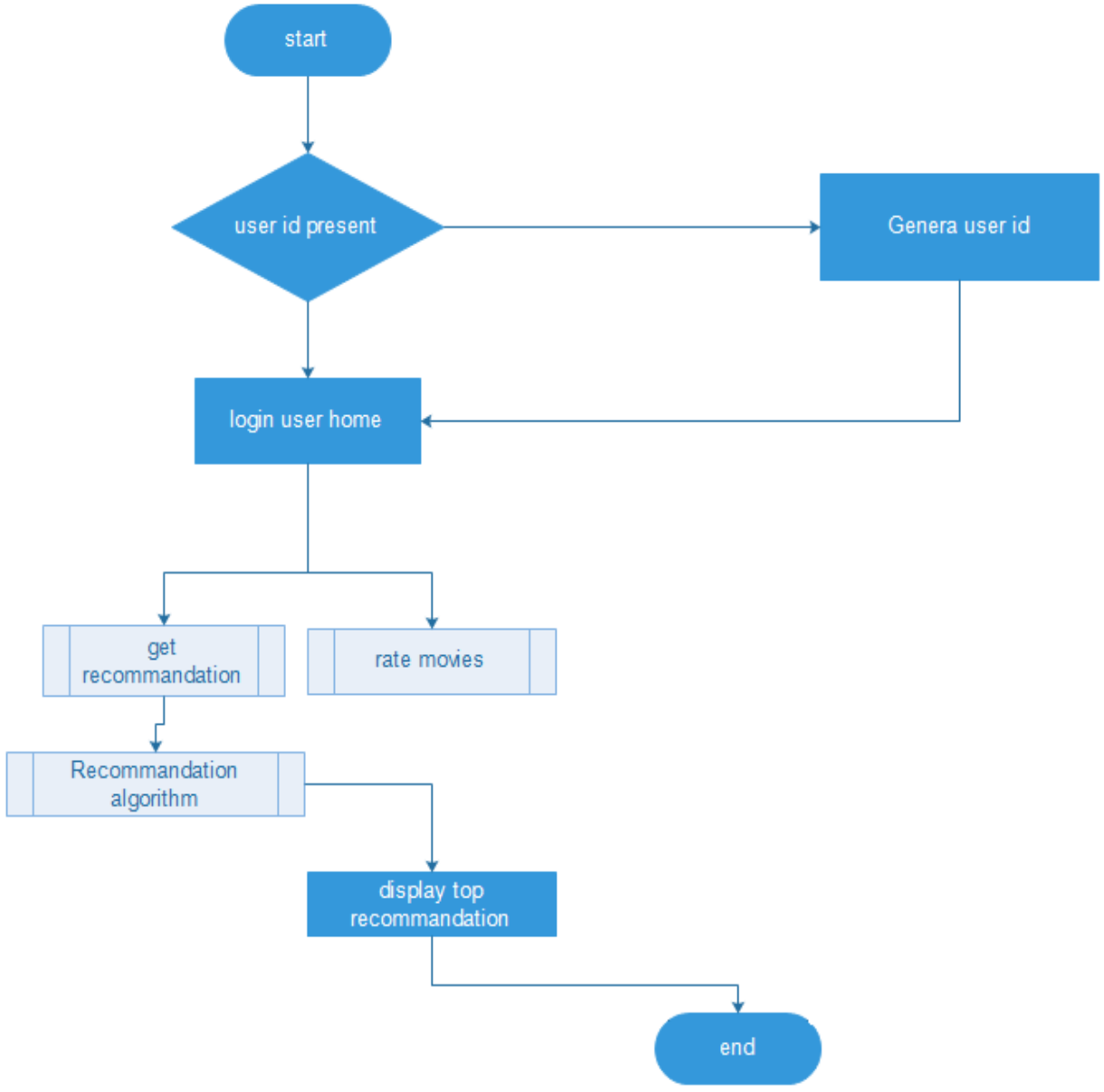


Fig.5.4 Flow Chart Diagram

Flow chart diagram is flow of data set used in project. from fig 5.4 user those are register for login creation That user is validate user if id is present then it will be proceed for recommendation which includes ratings of movie and performing algorithm display recommendation.

5.5. Sequence Diagram

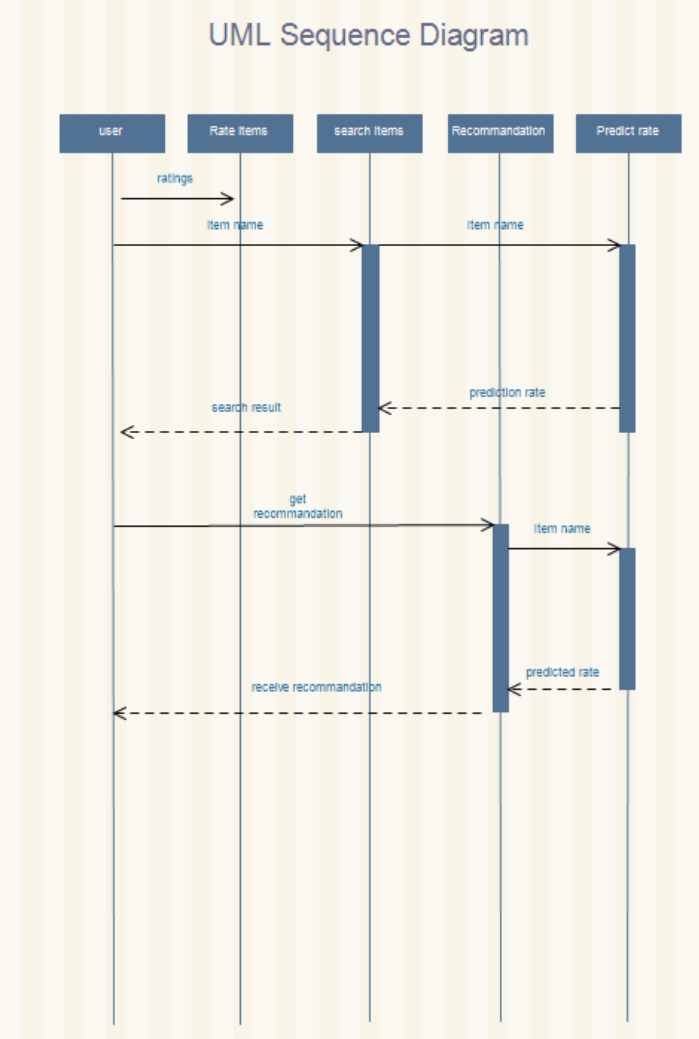


Fig.5.4 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. "User firstly rates the items which surveyed by the system. If another user search another items at that time system provides the recommended item based on different basis. After receiving recommended items user gets simple way to watch particular movie.

5.6 Use Case Diagram

5.6 .1 Use case Diagram (Level 0)

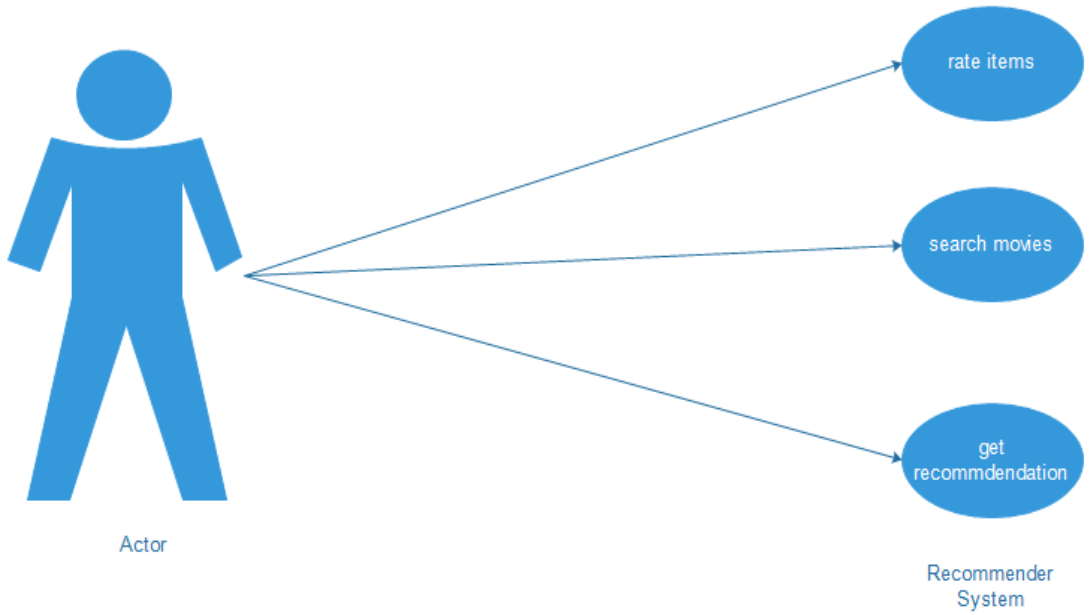


Fig.5.6.1.Use Case Diagram (level 0)

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

First user rate some movies on the basis of his/her favour then user search movies. At the time of user searching system will provide the recommended movies based on another users rates.it is visible to the user by this way user gets recommended movies.

5.6.2 Use case Diagram (Level 1)

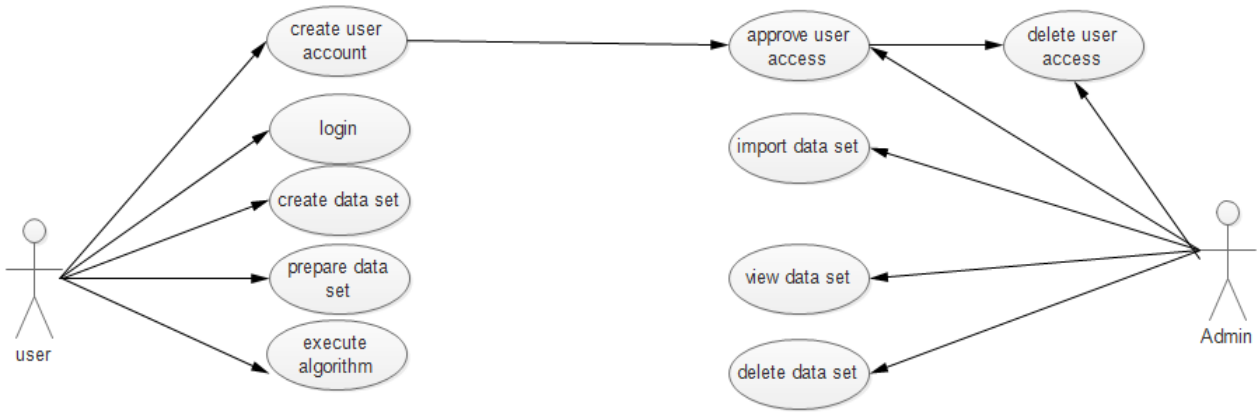


Fig.5.6.2Use Case Diagram (level 1)

Use Case Level 1:

Use case diagrams are drawn to capture the functional requirements of a system.to draw an use case diagram we should have the following items identified.

1) Functionalities 2) Actors 3) Relationships among the use cases and actors.

Firstly user create their own account. Then user login into account using their user-id and password. Then system create and prepare their data sets and execute algorithm. Then admin of system check-out the user access to that particular user. Then system perform functions like approve user access, import data sets, view data sets, delete data sets, and delete user access.

5.7 Activity Diagram

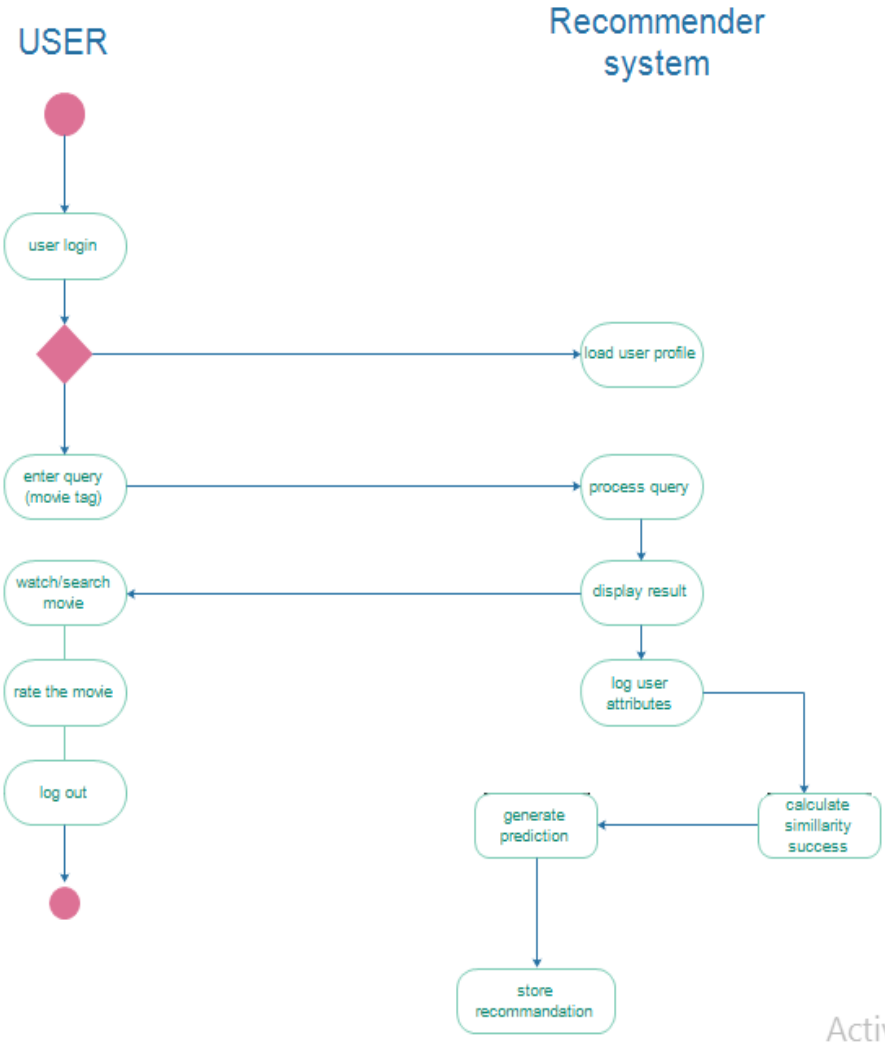


Fig.5.7 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Firstly user log in to system .System will loads user's profile. After this user will ready to enter his/her query about the movie. System will process the query and display the result of that query .User watch that movie and finally rates the movie and get logout from the system. At the time of displaying the query result, system store the recommendation on the basis of user log attributes. And at another time it will display the result from the stored recommendation.

5.8 Data Flow Diagram

5.8.1 DFD (Level 0)

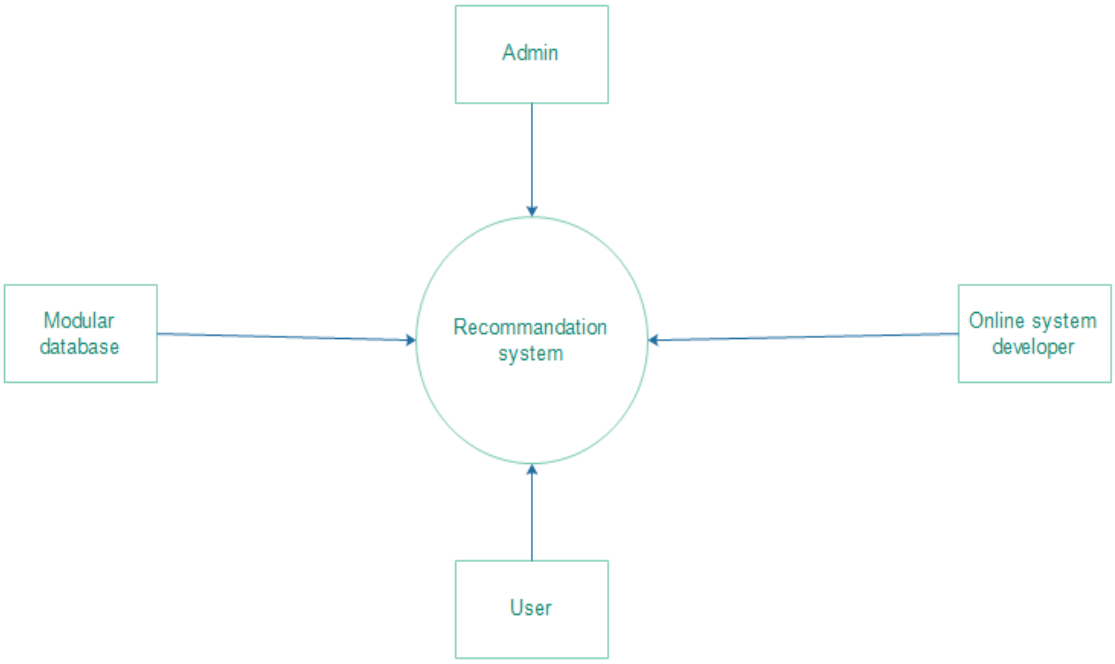


Fig.5.8.1Data Flow Diagram (level 0)

5.8.2 DFD (Level 1)

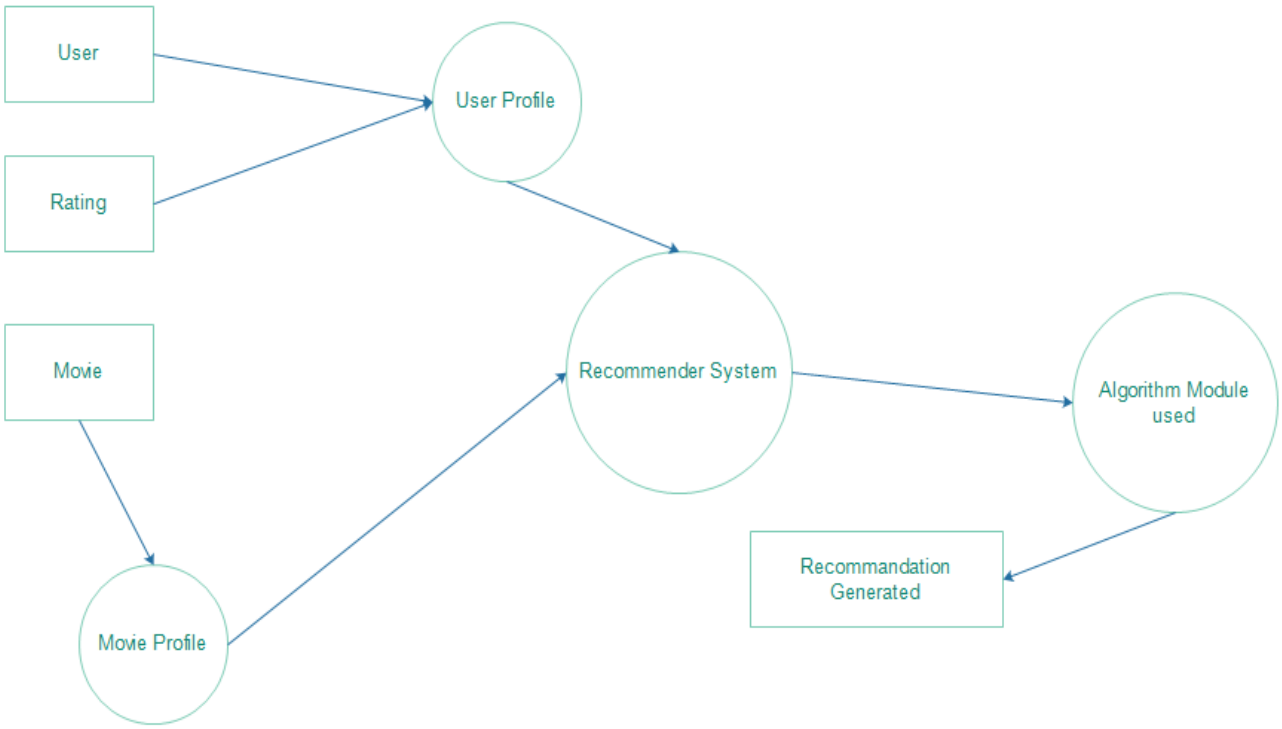


Fig.5.8.2. Data Flow Diagram (level 1)

A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities.

In this case, Recommendation system is surrounded by User, Admin, and Modular Database and Online system developer. In this case, User's process will generalise the functions of entire system in relationships to external entities like functions of Modular Database, Online System Developer and Admin of the system.

Development & Implementation Details

6.1. Phase I Installation Steps & Screenshots of Hadoop, Scala, Apache spark:

- Requirements:

  -OS-Ubuntu 15.04/ LINUXMINT

  -Scala-2.11.7

  -spark-spark-1.4.0-bin-hadoop2.6

  -hadoop-2.7.2.tar.gz
- CHECK IF YOU HAVE JDK INSTALLED :java –version
- STEP 1. Install Scala :

  -sudo apt-get remove scala-library scala

  -sudo wget http://www.scala-lang.org/files/archive/scala-2.11.7.deb

  -sudo dpkg -i scala-2.11.7.deb

  -sudo apt-get update

  -sudo apt-get install scala
- STEP 2.Install Spark

  -wget http://apache.mirrors.ionfish.org/spark/spark-1.4.0/spark-1.4.0-bin-hadoop2.6.tgz

  -tar -zxvf spark-1.4.0-bin-hadoop2.6.tgz

  -mv spark-1.4.0-bin-hadoop2.6 /usr/local/spark
- STEP 3. Get hadoop version

  -hadoop version

  -It should show 2.7.0
- STEP 4. Add spark home

  -sudo vi ~/.bashrc

  -add

  -export SPARK_HOME=/usr/local/spark

  -source ~/.bashrc
- STEP 5. Spark Version

  Since spark-1.4.0-bin-hadoop2.6.tgz is an built version for hadoop 2.6.0 and later, it is also usable for hadoop 2.7.0. Thus, we don't bother to re-build by sbt or maven tools, which are indeed complicated. If you download the source code from Apache spark org, and build with command build/mvn -Pyarn -Phadoop-2.7 -Dhadoop.version=2.7.0 -

DskipTests clean package. There are lots of build tool dependency crashed. So, no bother about building spark.

- STEP 6. Verify installation

  cd $SPARK_HOME

- STEP 7. Launch spark shell (refer to this)

  -./bin/spark-shell

  -It means spark shell is running

- STEP 8. Test spark shell

  -scala:> sc.parallelize(1 to 100).count()

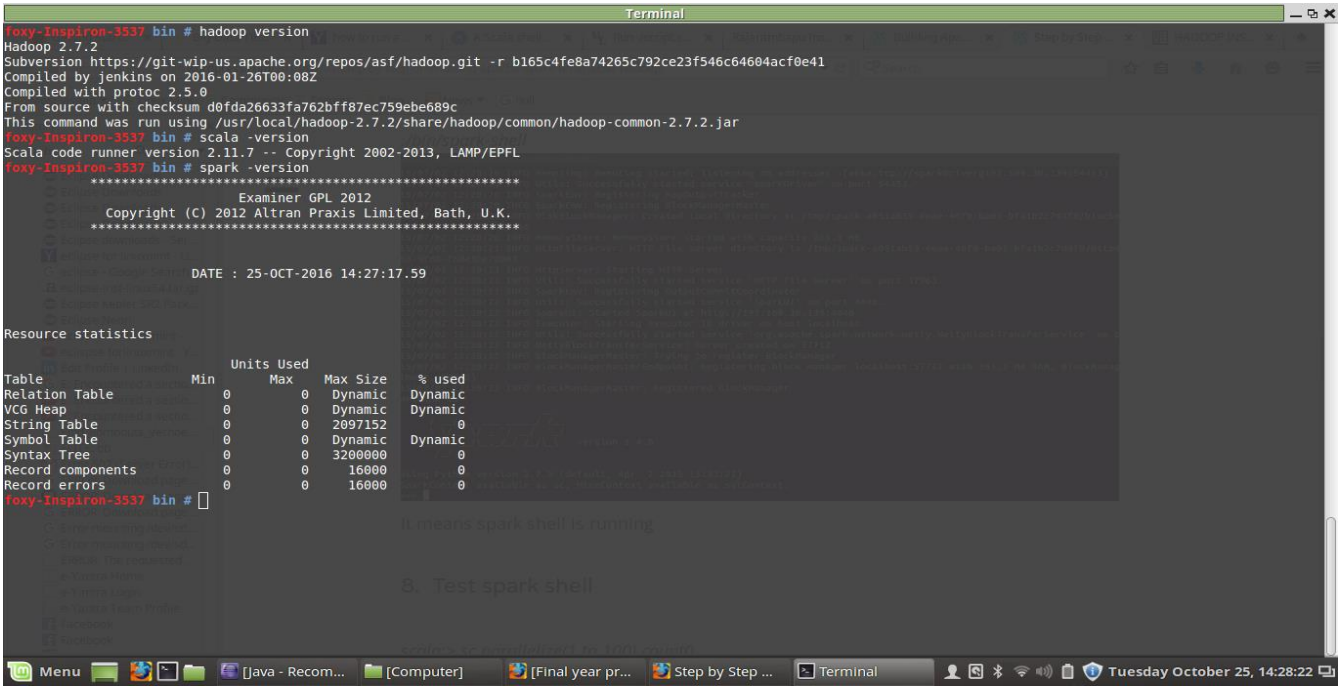
  -background info

  -scala:> exit

- STEP 9. Try typical example

  bin/spark-submit –class org.apache.spark.examples.SparkPi –master local[*] lib/spark-example* 10. The last variable 10 s the argument for the main of the application. For here is the slice number used for calculation Pi 4

We have finished Spark installation and next we can start using this powerfull tool to perform data analysis.



Screenshot 6.1.Installation of Hadoop, Scala, Apache spark.

6.2. Phase II Installation Steps & Screenshots for Anaconda for Windows:

- Download the Anaconda installer from www.continuum.io/downloads.html
- Click Run to launch the installer.

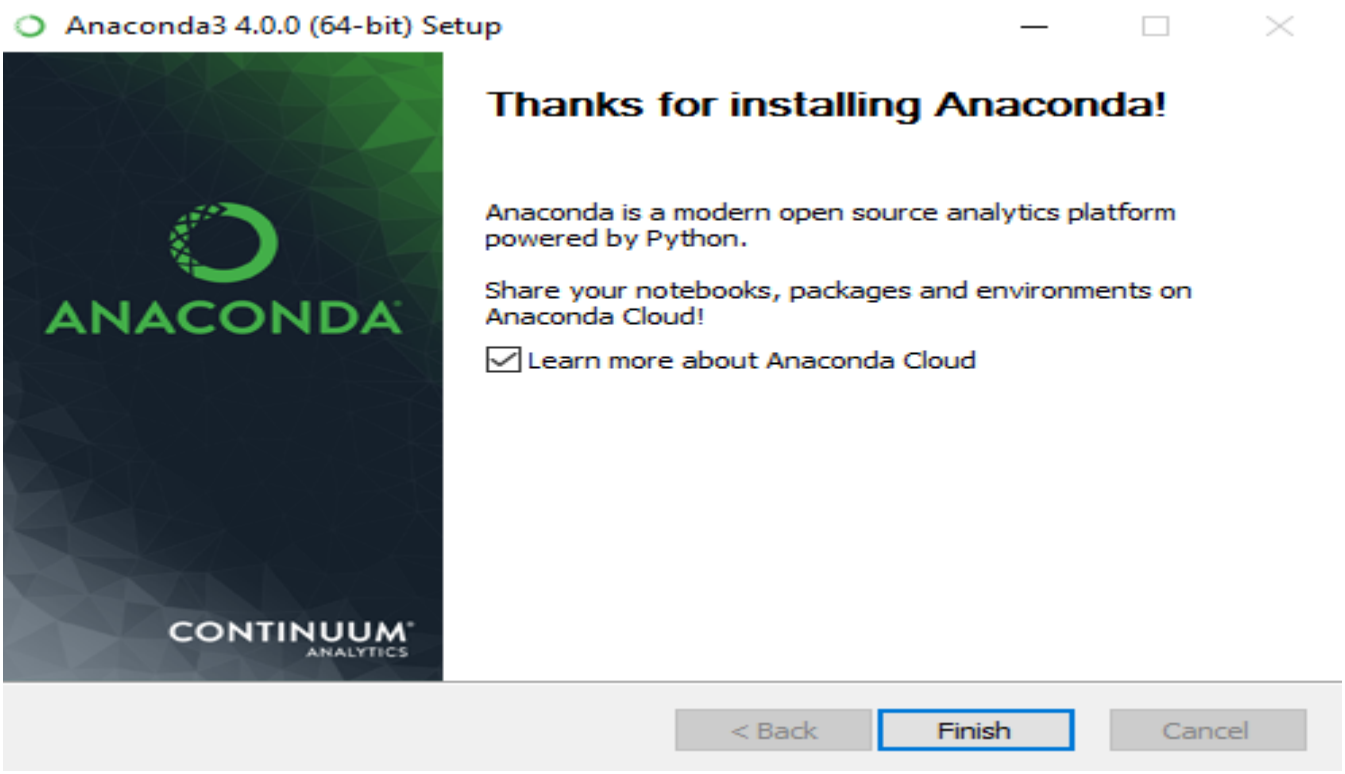
NOTE: If you encounter any issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you have installed for all users, uninstall Anaconda and re-install it for your user only and try again.

- Click Next.
- Read the licensing terms and click I Agree.
- Select an install for "Just Me" unless you're installing for all users (which requires Windows Administrator privileges).
- Select a destination folder to install Anaconda and click Next.

NOTE: Install Anaconda to a directory path that does not contain spaces or Unicode characters.

NOTE: Do not install as Administrator unless admin privileges are required.

- Choose whether to add Anaconda to your PATH environment variable. Unless you plan on installing and running multiple versions of Anaconda, or if you want to limit the length of your PATH variable, you should accept the default and leave this box checked.
- Choose whether to register Anaconda as your default Python 3.6. Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, you should accept the default and leave this box checked.
- Click Install. You can click Show Details if you want to see all the packages Anaconda is installing.
- Click Next.
- After a successful installation you will see the "Thanks for installing Anaconda" image:



Screenshot 6.2. Installation of Anaconda.

- You can leave the "Learn more about Anaconda Cloud" box checked if you wish to read more about this cloud package management service. Click Finish.

Testing

7.1. Test cases to test the valid & invalid inputs of the 100kdataset:

Table No. 3. Testcases1

Sr. no.	Test case Title	Description	Valid input	Output	Invalid input	Output Error Message
			Dataset size (symmetric)	Time Required	Asymmetric Dataset	
1.	User-user neighborhood approach	By using User-user Algorithm	1K	1.6	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured data"{0}"for xpath old"{1}"
2.	User-user neighborhood approach	By using User-user Algorithm	10K	2.5		
3.	User-user neighborhood approach	By using User-user Algorithm	100K	4.7		
4	User-user neighborhood approach	By using User-user Algorithm	10M	5.2		
5	Item-item Approach Similarity	By using Item-item Algorithm	1K	1.75	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured data"{0}"for xpath old"{1}"
6.	Item-item Approach Similarity	By using Item-item Algorithm	10K	2.65		
7.	Item-item Approach Similarity	By using Item-item Algorithm	100K	4.76		
8	Item-item Approach Similarity	By using Item-item Algorithm	10M	5.46		

9	Tanimoto Similarity	By using Tanimoto Coefficient	1K	1.82	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured data"{0}"for xpath old"{1}"
10	Tanimoto Similarity	By using Tanimoto Coefficient	10K	2.56		
11	Tanimoto Similarity	By using Tanimoto Coefficient	100K	4.79		
12	Tanimoto Similarity	By using Tanimoto Coefficient	10M	5.5		
13	Pearson Coefficient Similarity	By using Pearson Coefficient	1K	1.64	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured data"{0}"for xpath
14	Pearson Coefficient Similarity	By using Pearson Coefficient	10K	2.40		
15	Pearson Coefficient Similarity	By using Pearson Coefficient	100K	4.4	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured data"{0}"for xpath old"{1}"
16	Pearson Coefficient Similarity	By using Pearson Coefficient	10M	5.32		
17	Slope-One	By using Slope-One algorithm	1K	1.58	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured
18	Slope-One	By using Slope-One	10K	2.36		

		algorithm				data"{0}"for
19	Slope-One	By using Slope-One algorithm	100K	4.34	Asymmetrical Dataset having invalid columns and delimiters.	xpath old"{1}"
20	Slope-One	By using Slope-One algorithm	10M	5.20		
21	SVD	By using SVD Algorithm	1K	1.54	Asymmetrical Dataset having invalid columns and delimiters.	SDT01024 Could no analyze structured data"{0}"for xpath old"{1}"
22	SVD	By using SVD Algorithm	10K	2.31		
23	SVD	By using SVD Algorithm	100K	4.30		
24	SVD	By using SVD Algorithm	10M	5.2		
25	**Hybrid Algorithm**	**Combination of Tanimoto Slope-One And SVD**	**1K**	**0.80**	**Asymmetrical Dataset having invalid columns and delimiters.**	**SDT01024 Could no analyze structured data"{0}"for xpath old"{1}"**
26	**Hybrid Algorithm**	**Combination of Tanimoto Slope-One And SVD**	**10K**	**2.0**		
27	**Hybrid Algorithm**	**Combination of Tanimoto Slope-One And SVD**	**100K**	**4.0**		
28	**Hybrid Algorithm**	**Combination of Tanimoto Slope-One**	**10M**	**4.57**		

7.2. Test cases to test the Sparsity of the 100kdataset:

Table No. 4. Testcases2

Sr. no.	Test case Title	Description	Valid input	Sparsity (before)	Sparsity (after)
			Dataset size (symmetric)	%	%
1.	User-user neighborhood approach	By using User-user Algorithm	100K	93.7	93.7
2	Item-item Approach Similarity	By using Item-item Algorithm	100K	93.7	93.7
3	Tanimoto Similarity	By using Tanimoto Coefficient	100K	93.7	93.7
4	Pearson Coefficient Similarity	By using Pearson Coefficient	100K	93.7	93.7
5	Slope-One	By using Slope-One algorithm	100K	93.7	93.7
6	SVD	By using SVD Algorithm	100K	93.7	93.7
7	**Hybrid Algorithm**	**Combination of Tanimoto Slope-One And SVD**	**100K**	**93.7**	**100**

Deployment

8.1. Readme File:

A] Install JDK:

1) Install jdk 7 in system and confirm the JAVA_HOME environment variable also.
2) Install mysql database having username = hadoop & password = hadoop

B] Eclipse:

1) Open Linux mint.
2) Click on "START" button and search for the eclipse.
3) Once eclipse gets open, Create new Project and firstly imports the all libraries and build the path.
4) Add a new class and your code.

C] Installing Apache Spark:

1) Find the steps of installation in the following link:
https://www.tutorialspoint.com/apache_spark/apache_spark_installation.htm
https://www.dezyre.com/apache-spark-tutorial/apache-spark-installation-tutorial
2) Launch spark shell.

D] Installing Anaconda:

1) Find the steps of installation in the following link:
https://docs.continuum.io/anaconda/ide_integration
https://www.pugetsystems.com/labs/hpc/How-to-Install-Anaconda-Python-and-First-Steps-for-Linux-and-Windows-917
2) Launch notebook using command: jupyter notebook.

Result Analysis & Discussion

9.1. Phase I Implementation:

We have implemented recommendation system algorithms using Item similarity, user-user neighbourhood approach, Tanimoto coefficient Algorithm, Pearson coefficient Algorithm, Slope One Algorithm, SVD Algorithm by using mahout libraries using JAVA platform in Eclipse on Linux platform.

List of Mahout Libraries:

- Mahout-integration-0.12.2.jar
- Mahout-math-0.12.2.jar
- Mahout-hdfs-0.12.2.jar
- Mahout-hdfs-mr-0.12.2_job.jar
- Mahout-commons-math3-3.2.jar
- Mahout-guawa -14.0.1.jar
- Mahout-slf4j-api-1.7.21.jar
- Mahout-slf4j-nop-1.7.21.jar

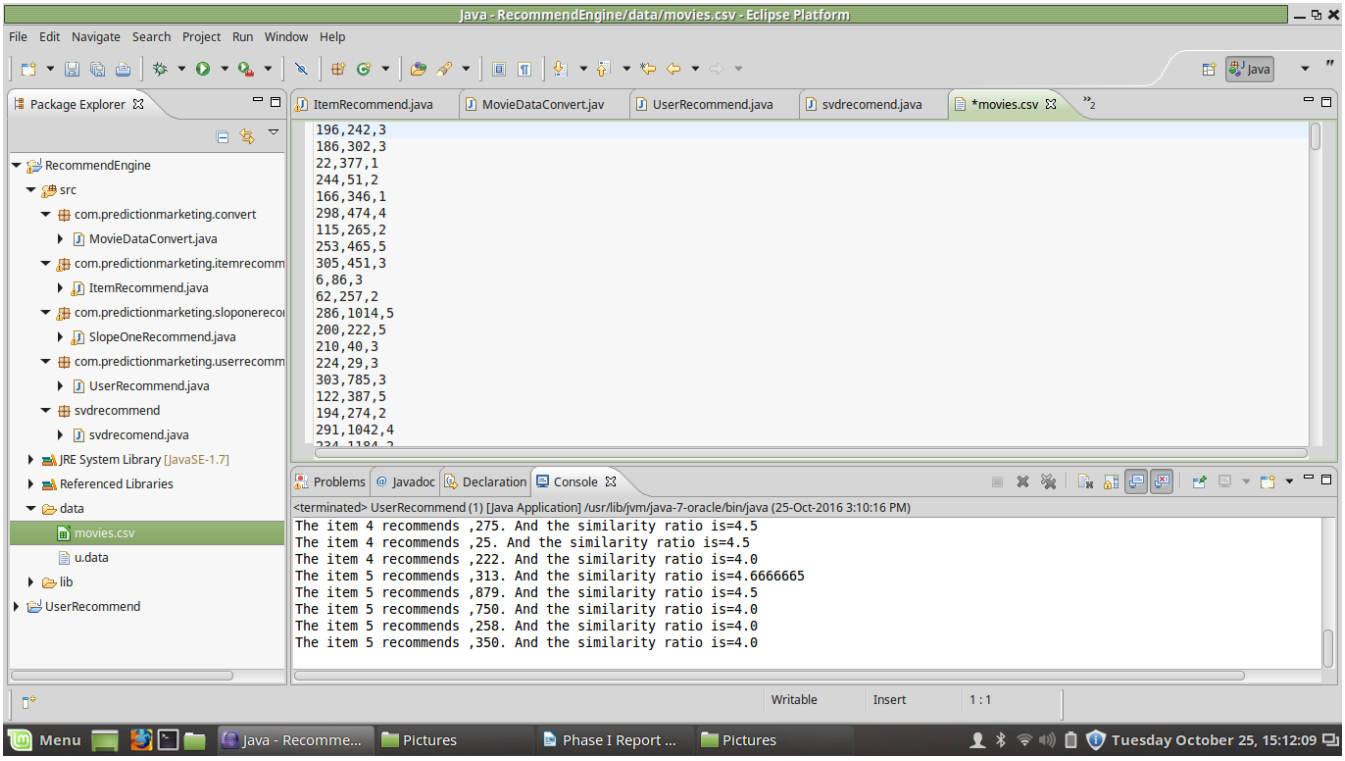
The following table shows the time taken by each dataset to process from asymmetrical data to symmetrical data:

Table No. 5. Time required for data processing

Datasets	1k	10k	100k	10M
Time required	0.80s	2.0s	4.0s	4.57s

9.1.1. Phase I Screenshots and Analysis:

1) Screenshot 1:

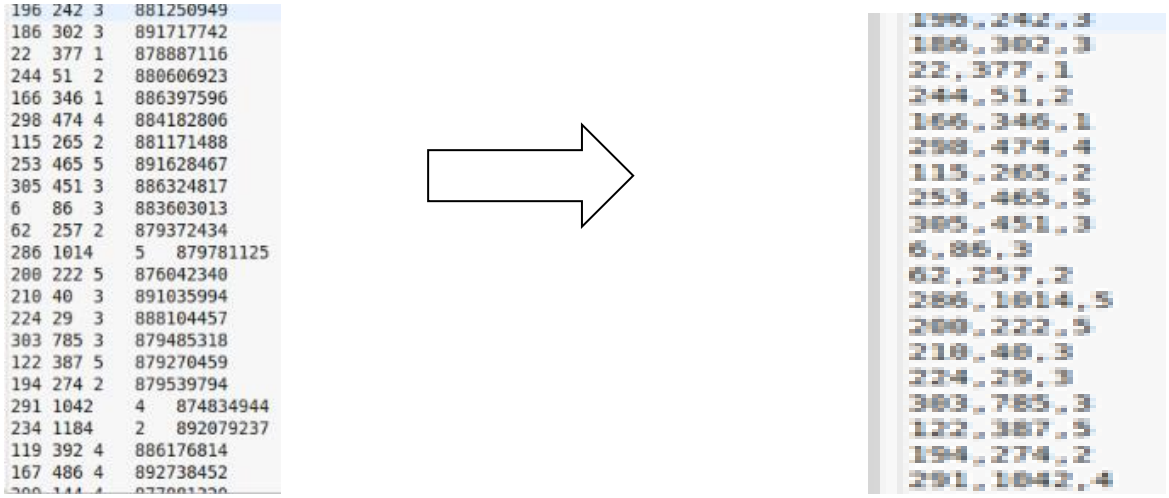


Screenshot 1: Converting asymmetrical dataset into symmetrical

This screenshot shows the result of conversion of 1k dataset into symmetrical form. The dataset before conversion had additional column of timestamp in it which is removed after conversion.

Analysis:

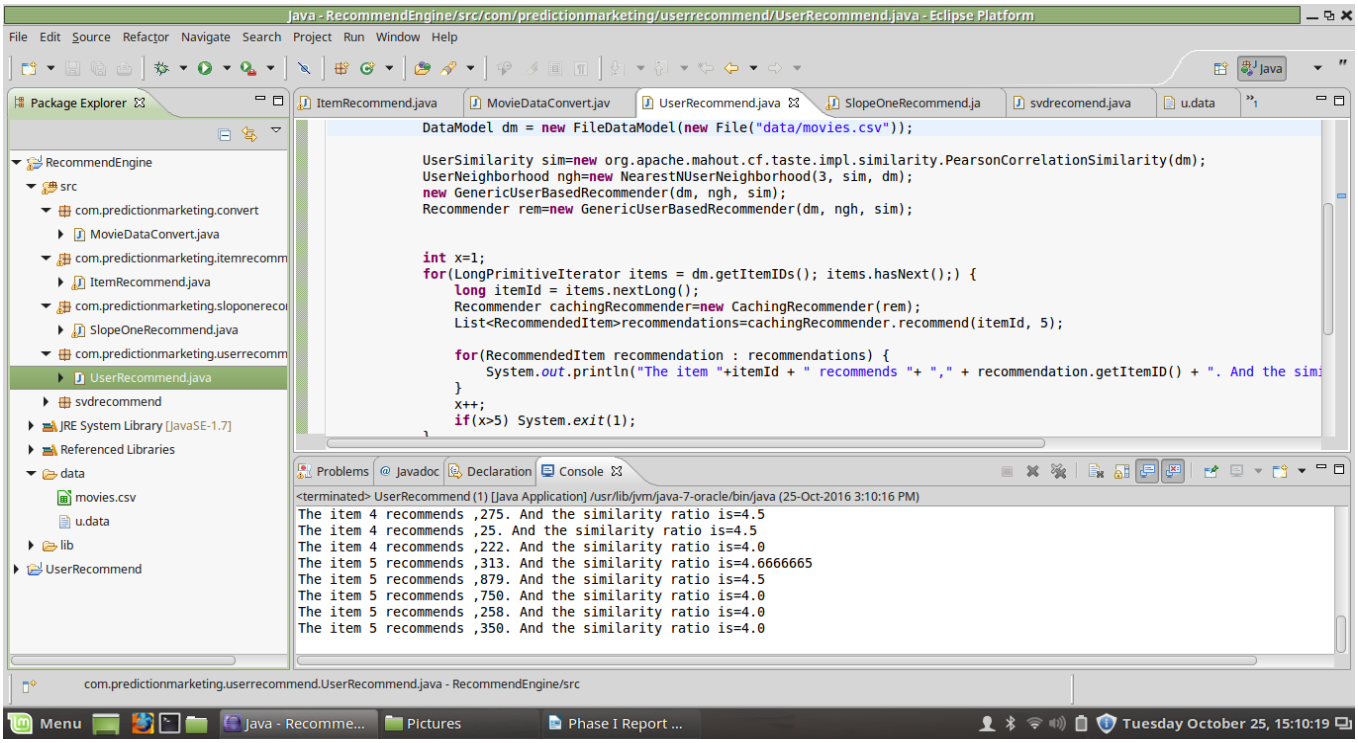
Analysis 1: Data conversion



Before After

The first figure before shows the unstructured data which has four columns viz. 1) Item ID, 2)User ID, 3) Ratings and 4)Timestamp which is then converted to second image which has structured data after conversion viz. 1) Item ID, 2)User ID and 3) Ratings.

2) Screenshot 2:

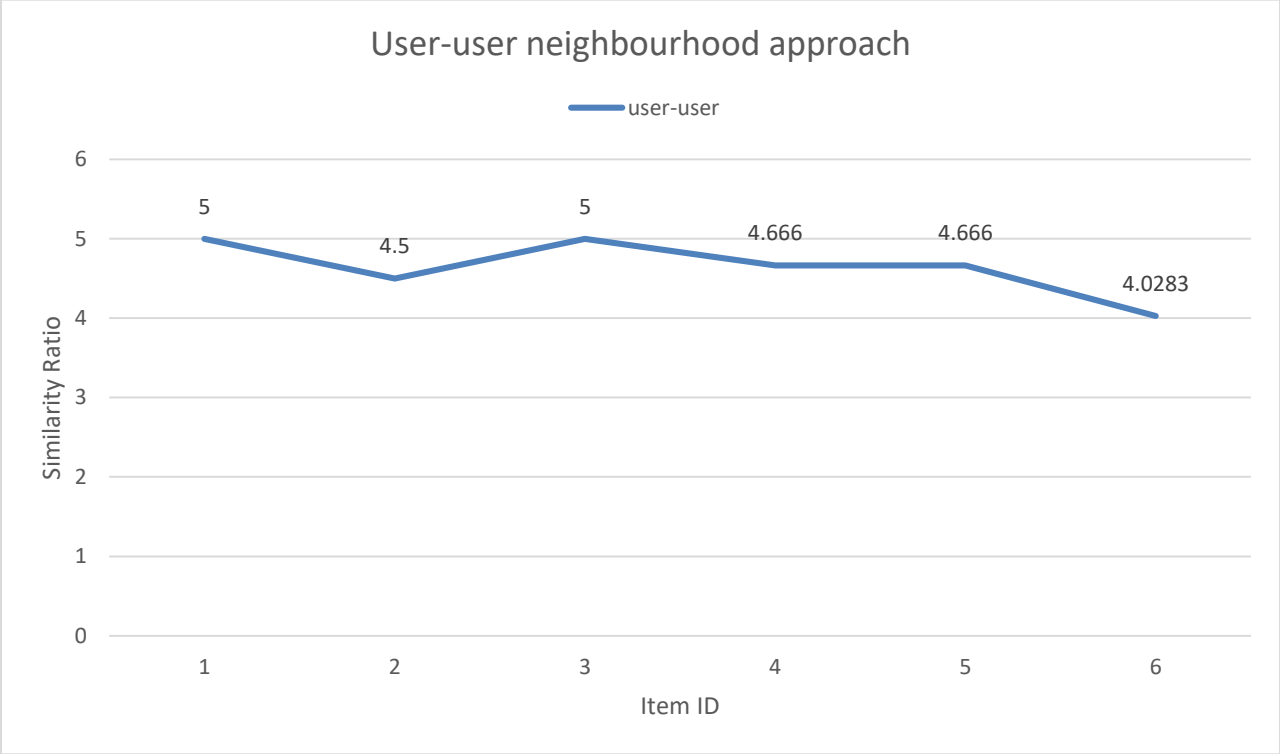


Screenshot 2: User recommendation & o/p

This screenshot shows user-user recommendation and its results.

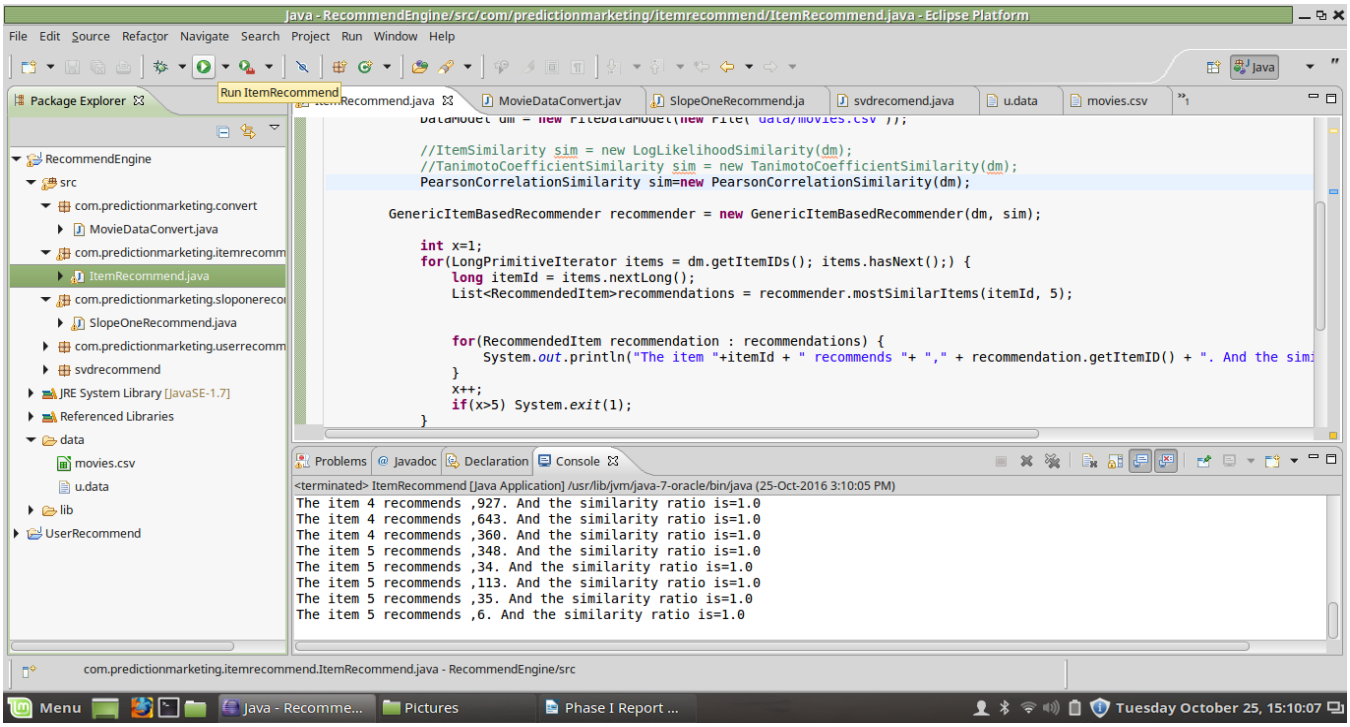
Analysis:

The graph has similarity ratio on y-axis and item ID on x-axis. The similarity ratio of user-user is shown on legend.



Graph 2.User-user recommendation

3) Screenshot 3:



Screenshot 3: Item-item, Tanimoto, Pearson Recommendation & o/p

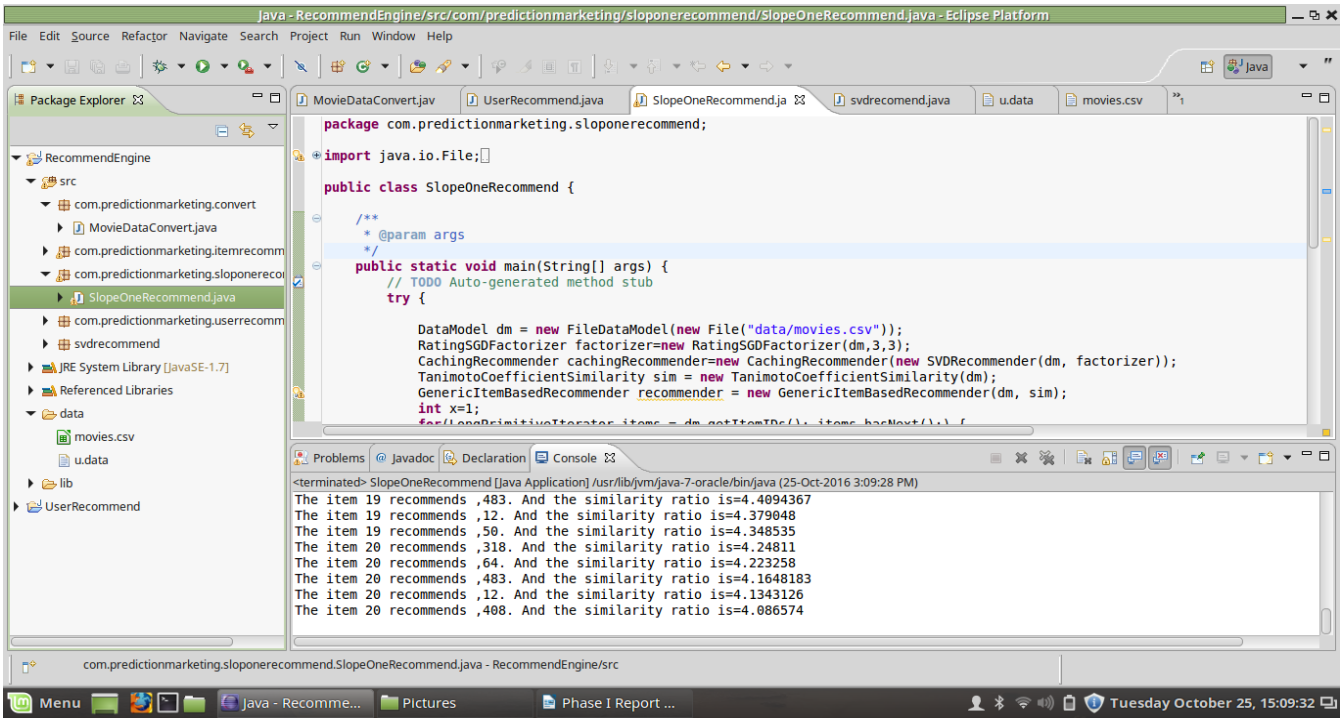
This screenshot shows the Item-item, Tanimoto, Pearson Recommendation and its results.

Analysis:

The graph has similarity ratio on y-axis and item ID on x-axis. The similarity ratio of item-item is shown on legend.

Graph 3: Item-item approach

4) Screenshot 4:

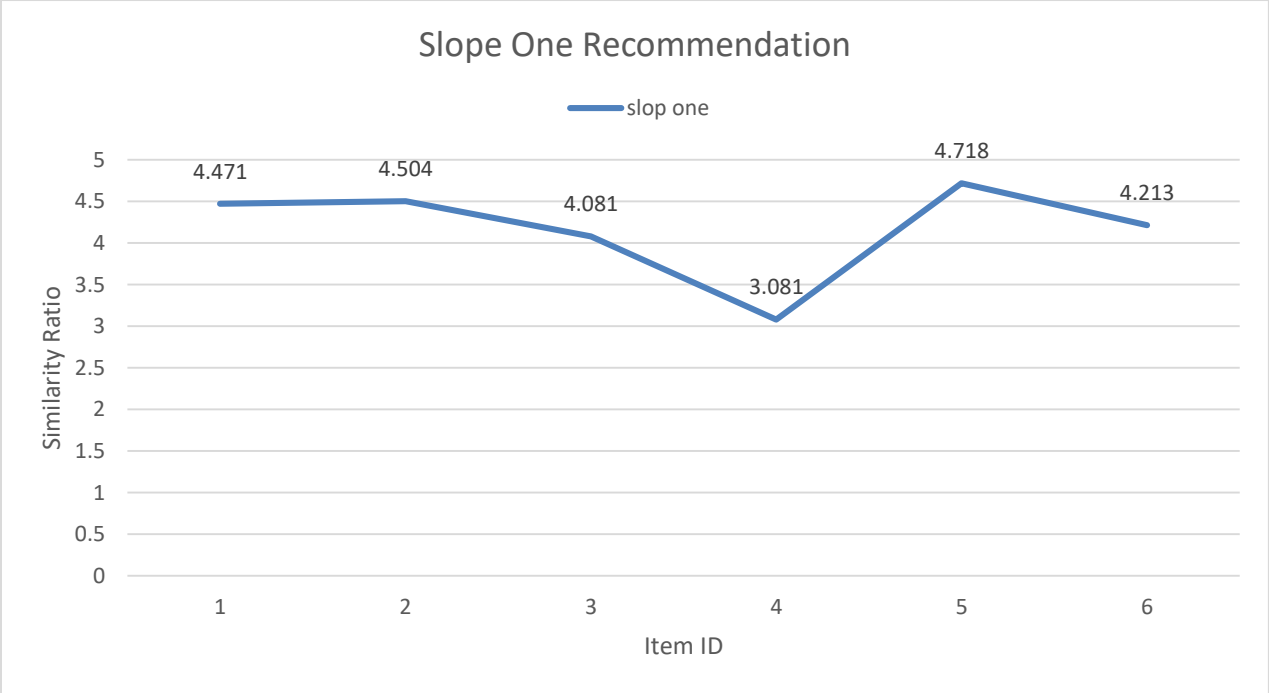


Screenshot 4: Slope one Recommendation & o/p

This screenshot shows the Slope One recommendation and its results.

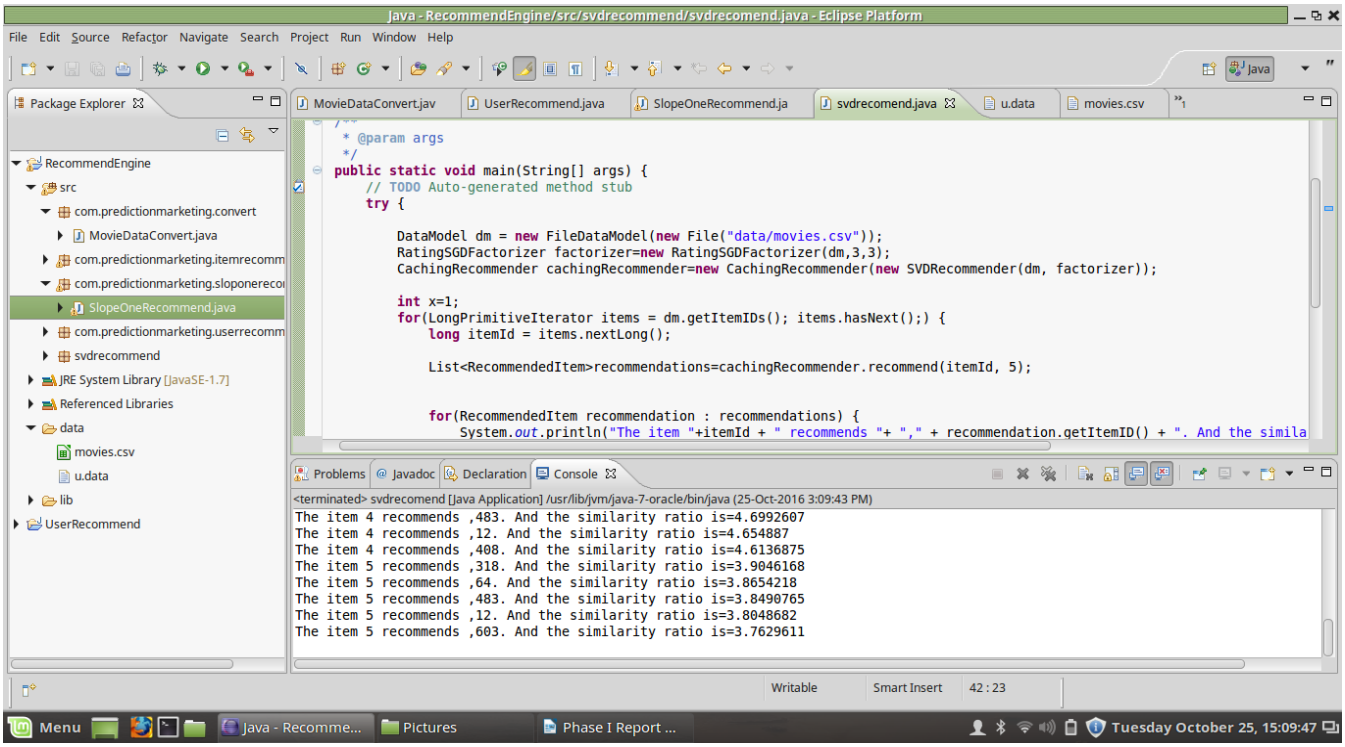
Analysis:

The graph has similarity ratio on y-axis and item ID on x-axis. The similarity ratio of slope one is shown on legend.



Graph4.Slope one recommendation

5) Screenshot 5:

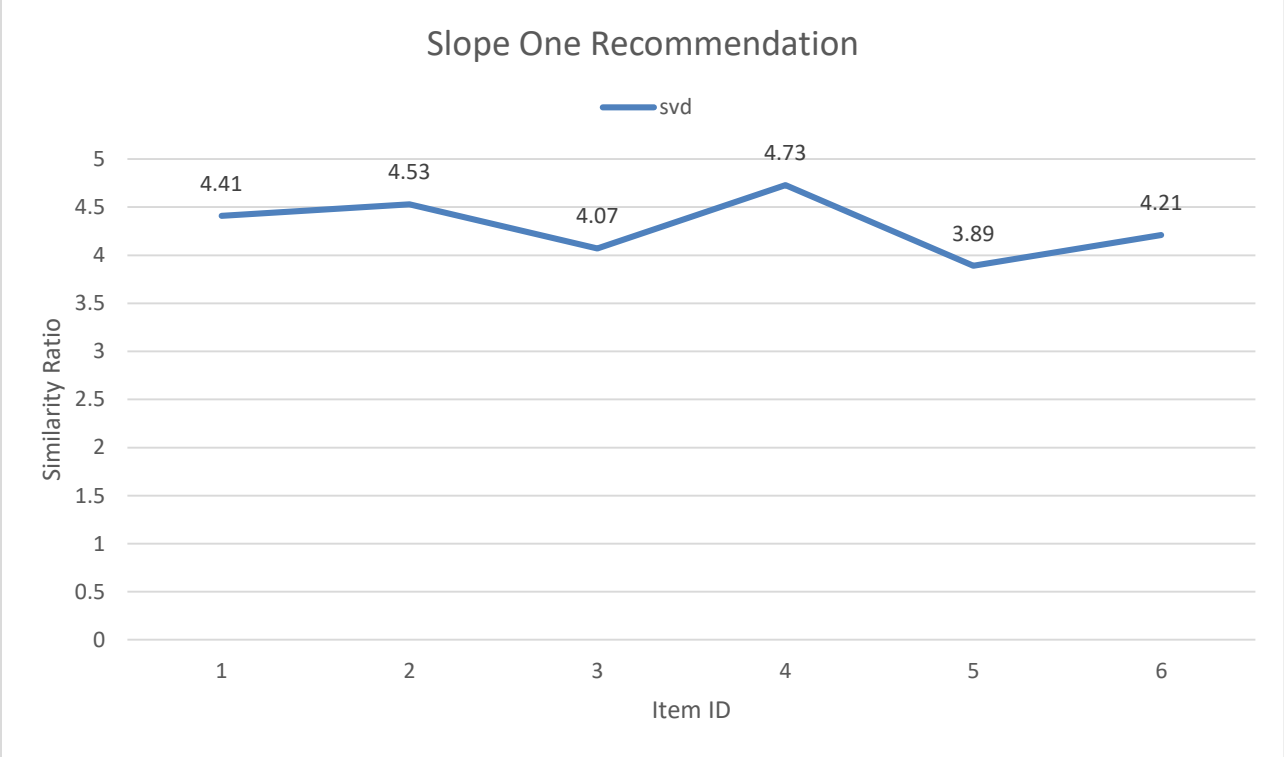


Screenshot 5: SVD Recommendation & o/p

This screenshot shows the SVD recommendation and its results.

Analysis:

The graph has similarity ratio on y-axis and item ID on x-axis. The similarity ratio of SVD is shown on legend.



Graph 5. SVD recommendation

9.2. Phase II:

We have implemented Hybrid recommendation system algorithms using Tanimoto coefficient Algorithm, Pearson coefficient Algorithm, Slope One Algorithm, SVD Algorithm by using Anaconda framework using libraries in Python using conda[root] notebook Windows platform.

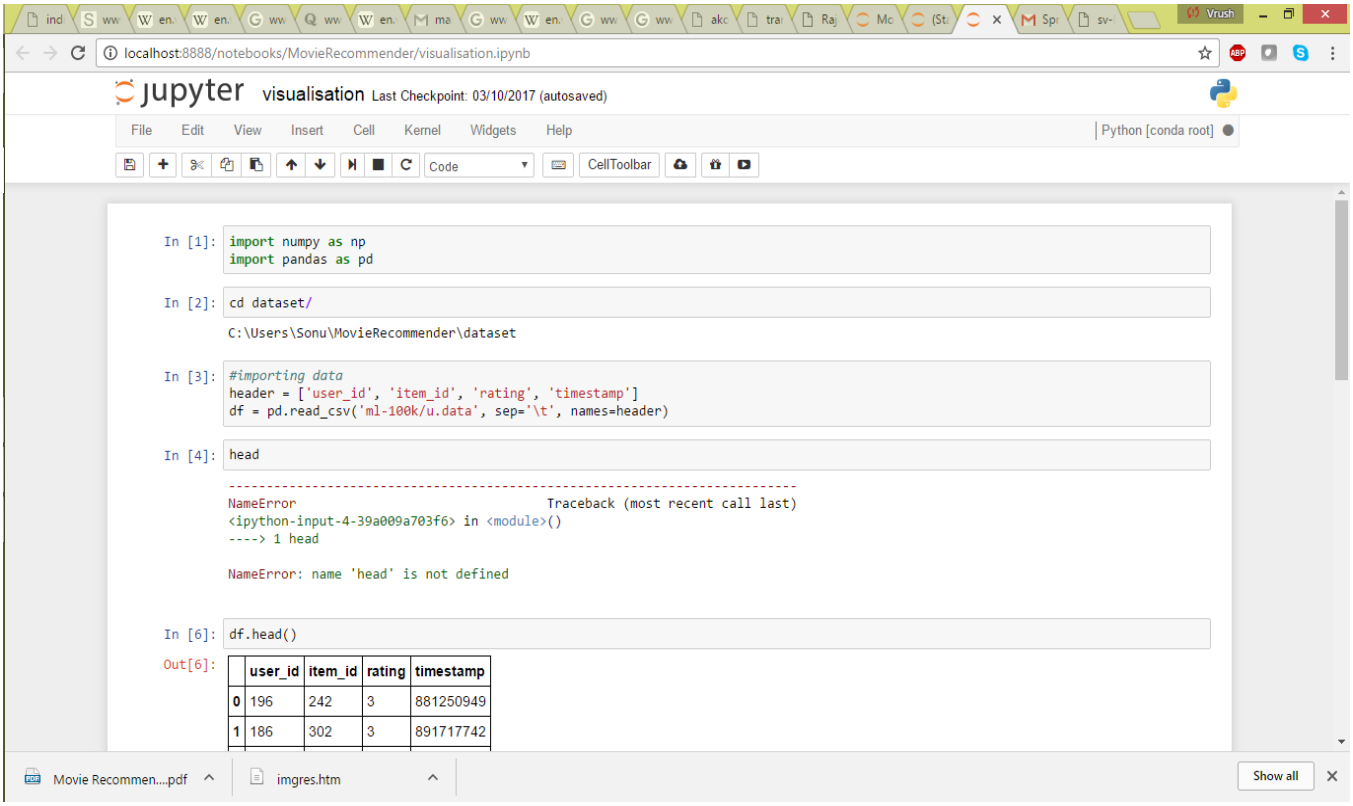
List of Python Libraries:

- Pandas
- Numpy
- Matplotlib
- Sklearn
- Scipy

First we used matplotlib for visualisation of the dataset. Using the Root Mean Square Error we calculated the Sparsity of the dataset which was 93.7%. We applied hybrid algorithm to the dataset and then again calculated the Sparsity which was 100%. Thus, the hybrid algorithm could improve the sparsity of the dataset and give faster recommendations. The difference is almost 6.3% which is massive.

9.2.1. Phase II Screenshots and Analysis:

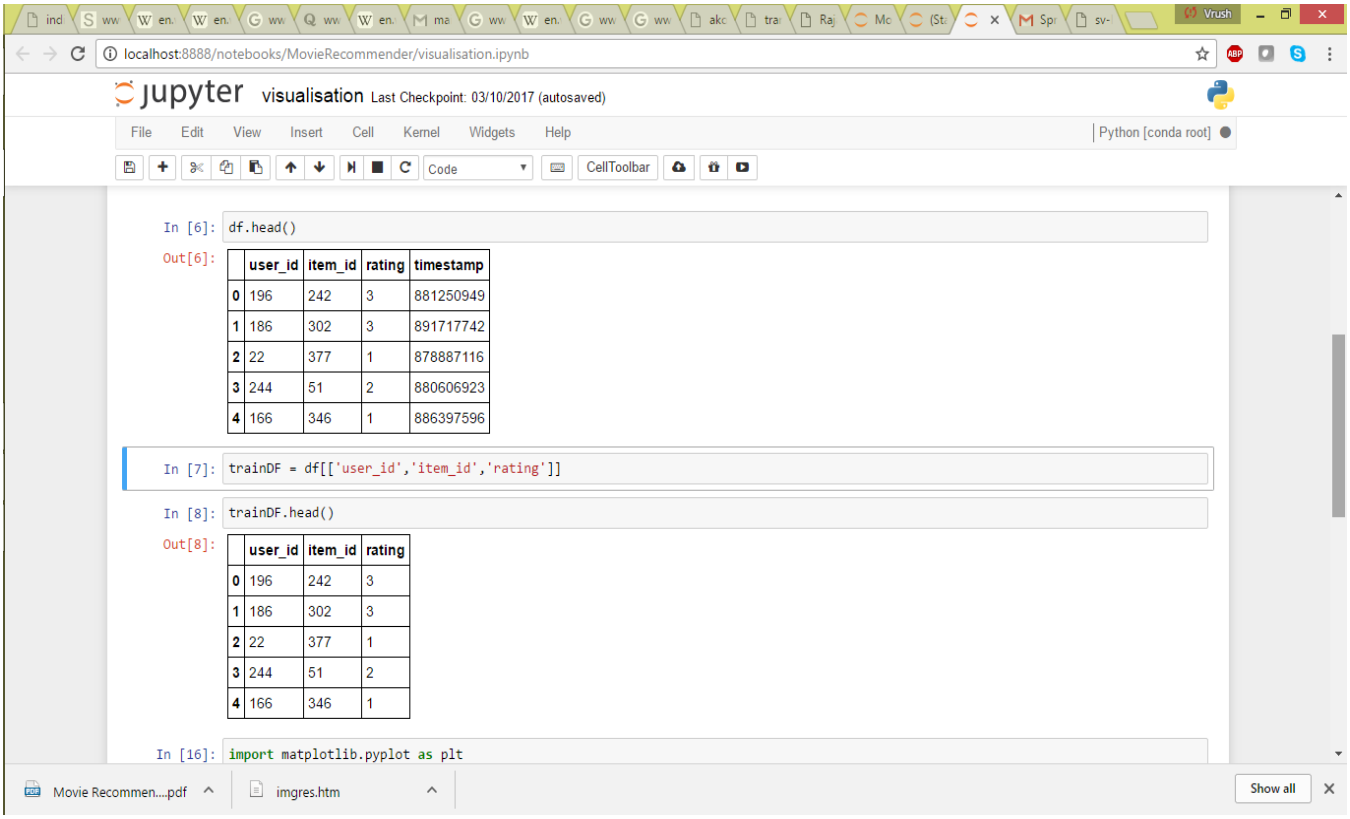
1) Screenshot 7:



Screenshot 7: Input dataset for visualization

This screenshot shows the program that imports python libraries and takes input CSV file which is nothing but dataset and gives column names to it.

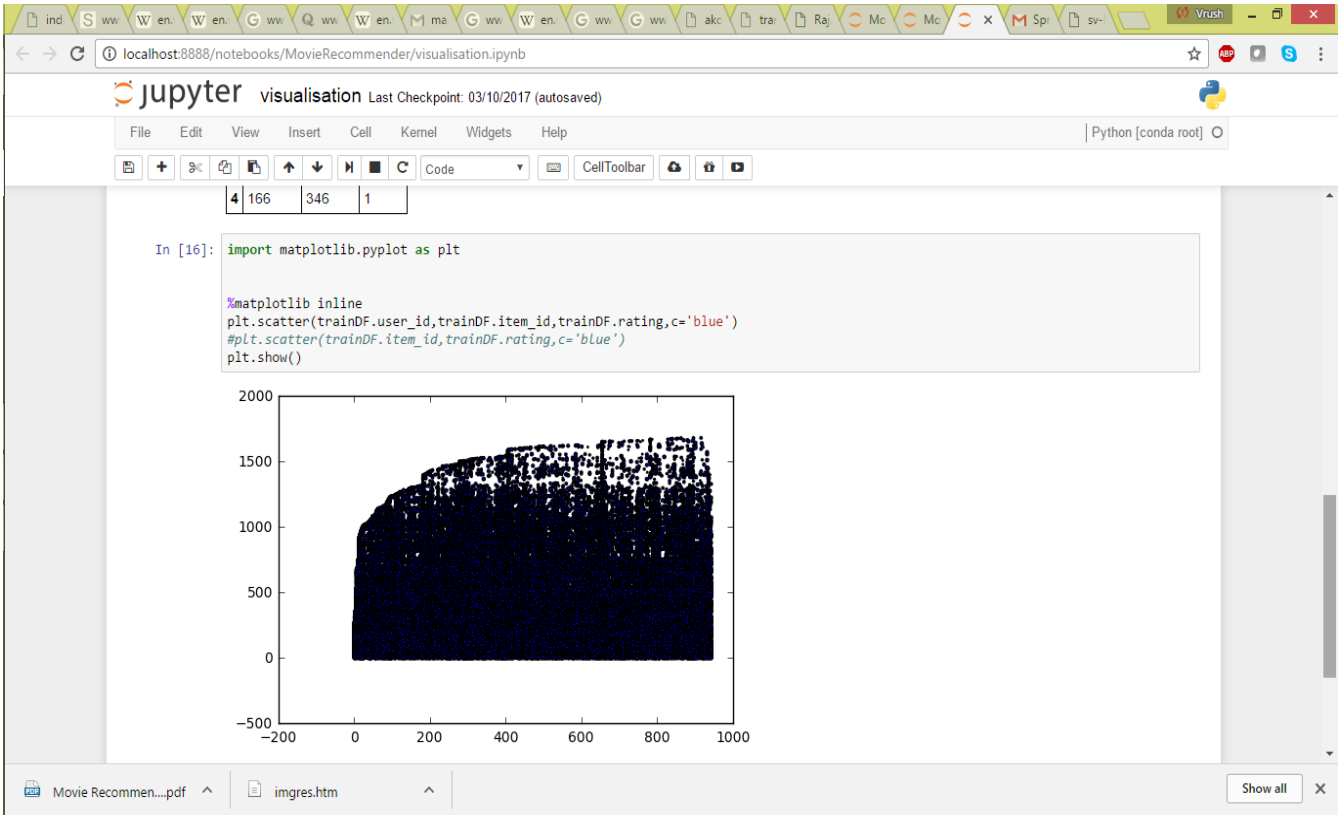
2) Screenshot 8:



Screenshot 8: Conversion of asymmetric dataset to symmetric

This screenshot shows the program that converts the dataset from asymmetric to symmetric i.e. it deletes the last column and trains the dataset.

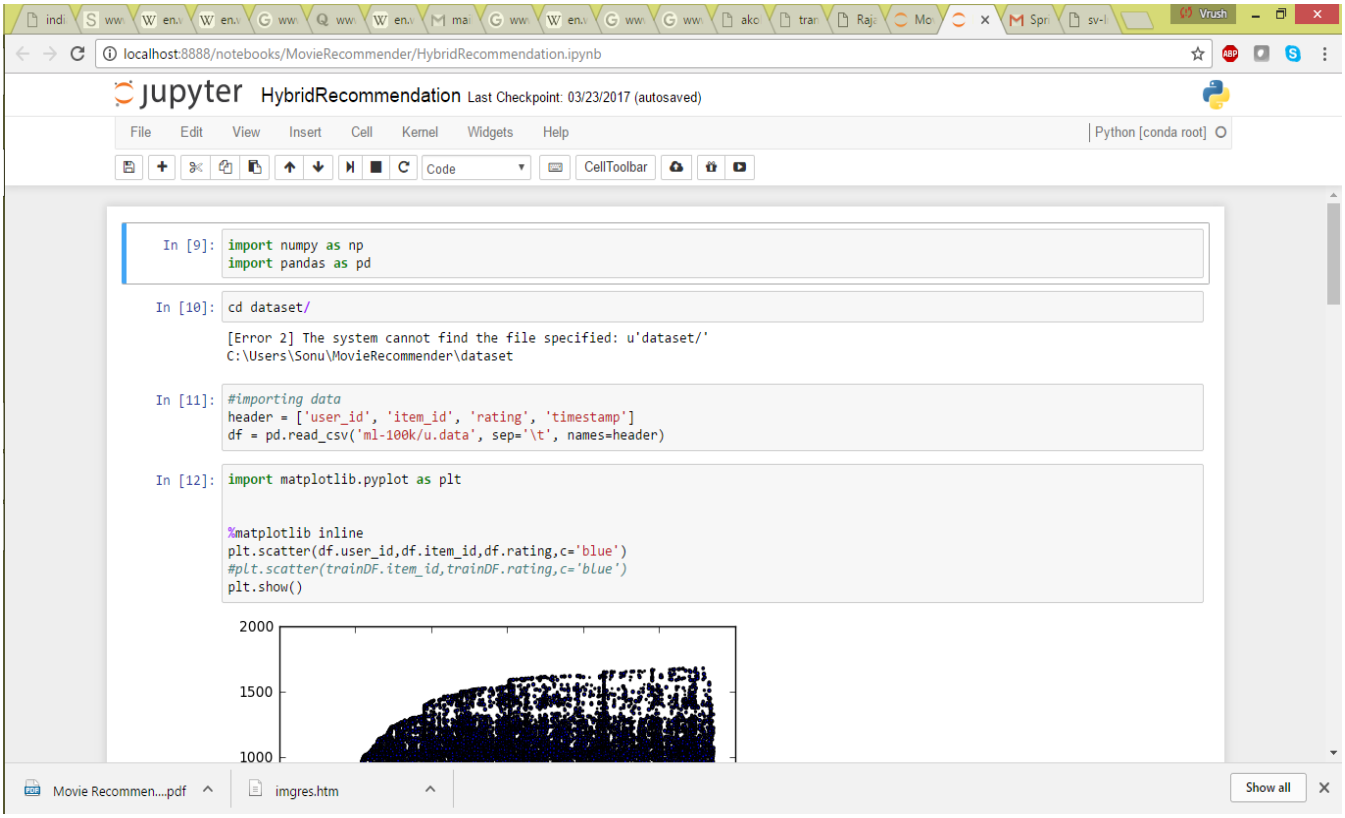
3) Screenshot 9:



Screenshot 9: Visualization of symmetric dataset.

This screenshot shows the program that visualizes the converted the dataset from asymmetric to symmetric.

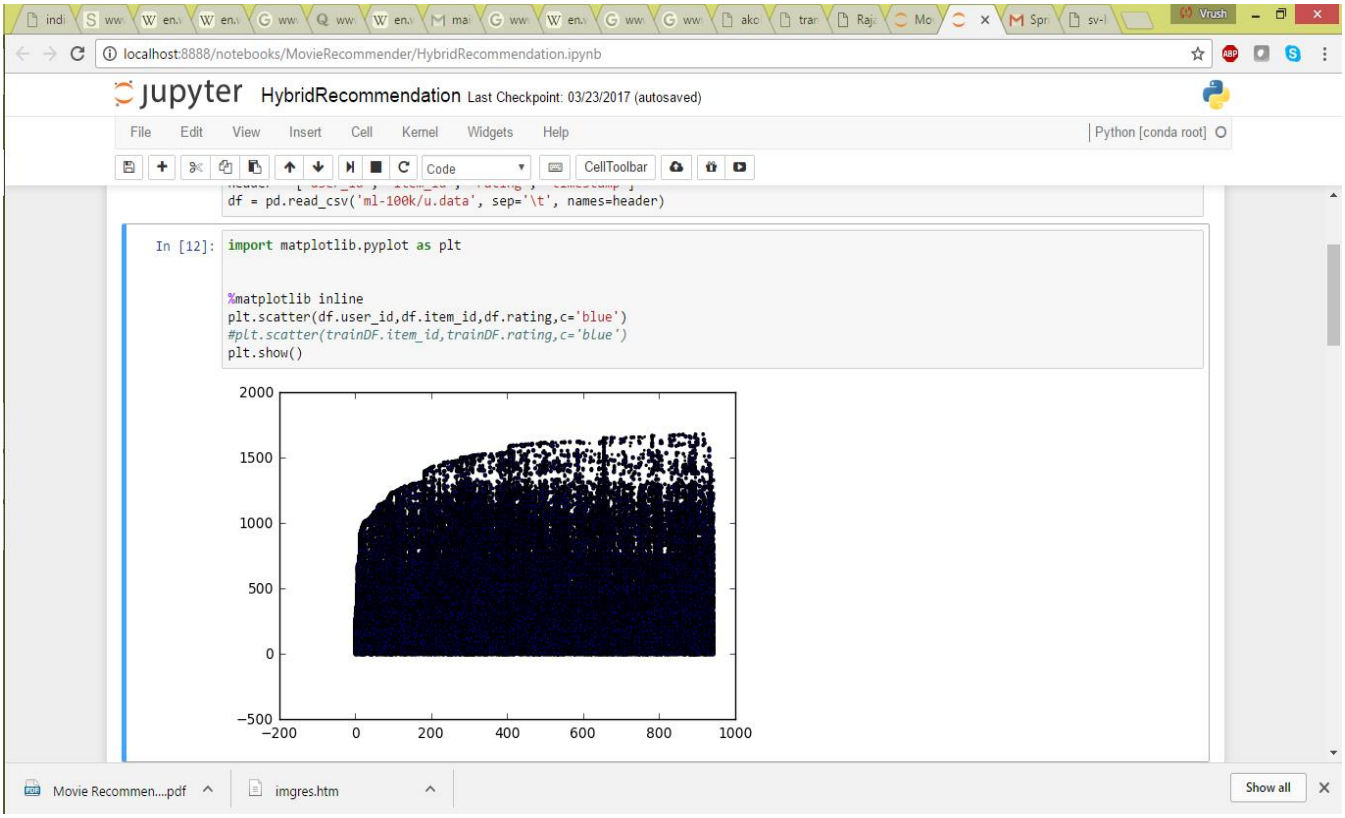
4) Screenshot 10 :



Screenshot 10: Input dataset

This screenshot shows the program that imports python libraries and takes input CSV file which is nothing but dataset.

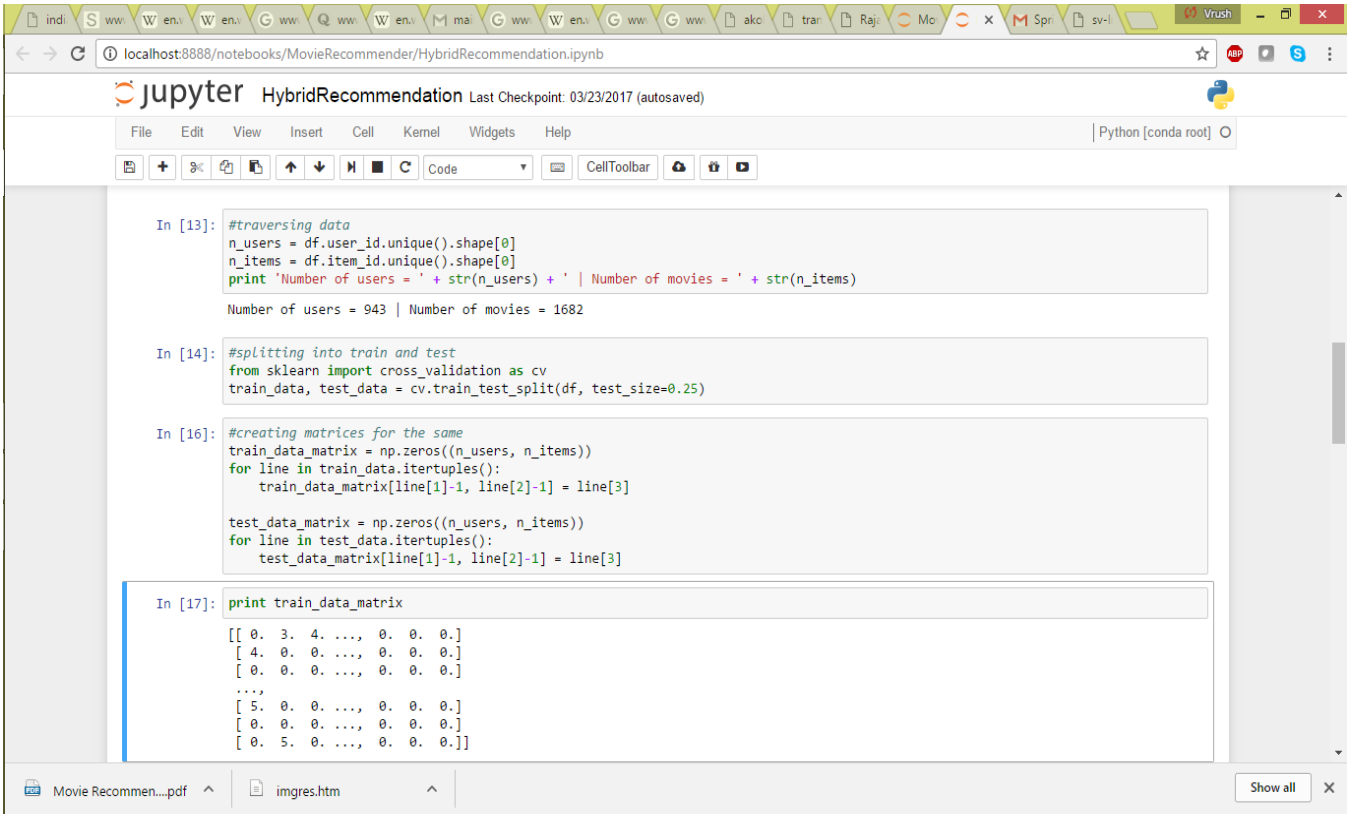
5) Screenshot 11:



Screenshot 11: Visualization (before)

This screenshot shows the program that imports python libraries and visualizes the dataset before applying the hybrid algorithm.

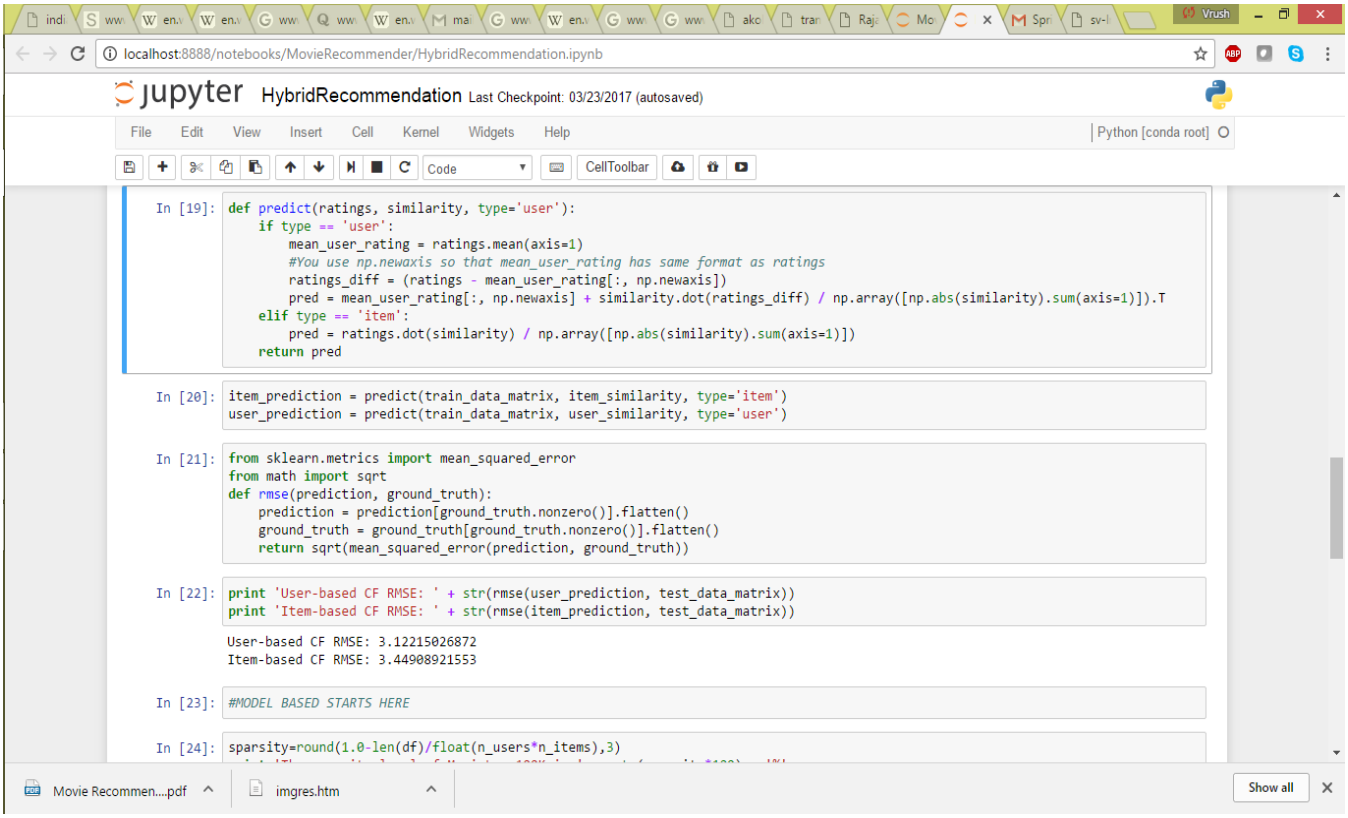
6) Screenshot 12:



Screenshot 12: Splitting dataset

This screenshot shows the program that imports python libraries and traverses the dataset and shows the result. After traversing the dataset it splits the dataset into test data and train data matrix to perform the testing of hybrid algorithm on the same.

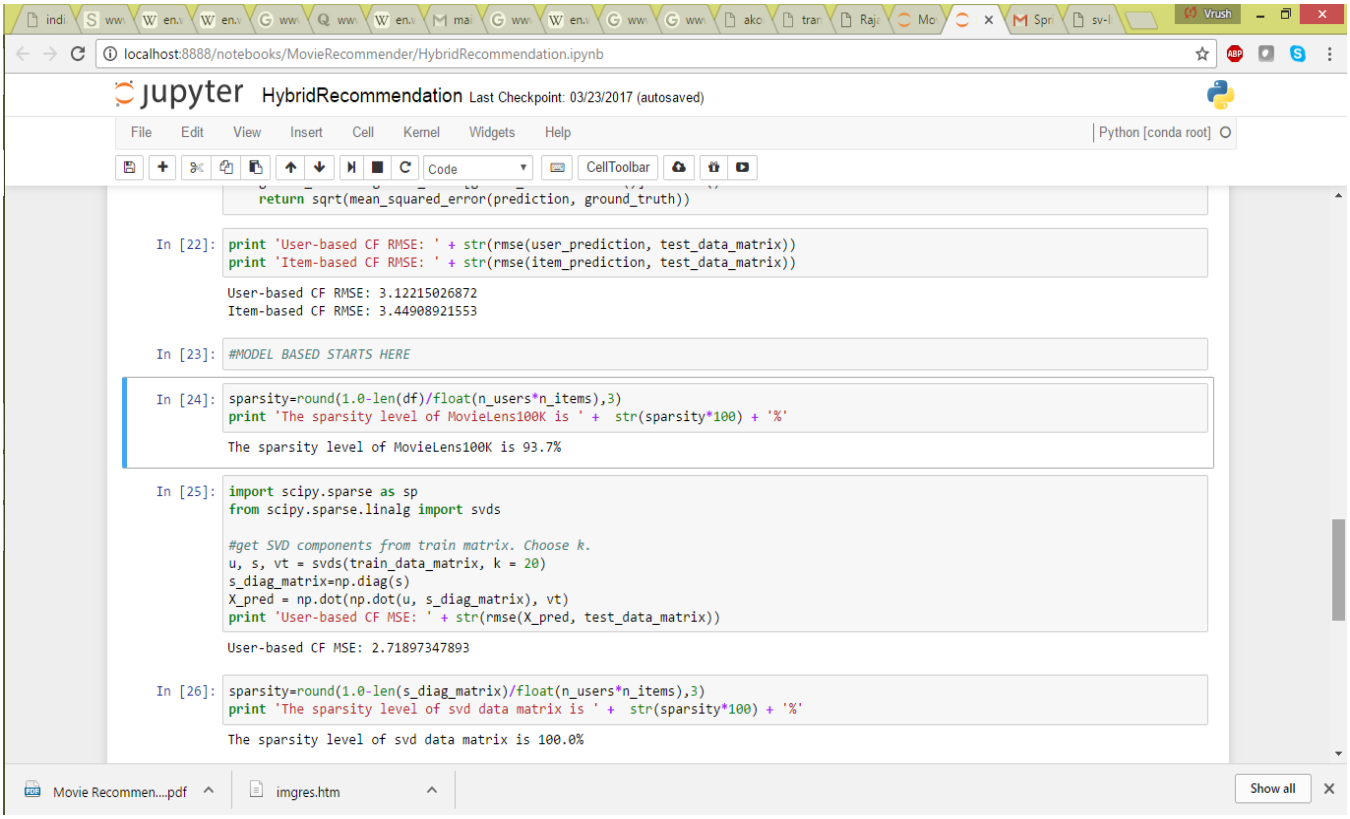
7) Screenshot 13:



Screenshot 13: Collaborative Filtering Algorithm

This screenshot shows the program that imports python libraries, defines functions and uses collaborative filtering algorithm for prediction of recommendations and calculates root mean square error value of both user and item prediction.

8) Screenshot 14:



Screenshot 14: Content based Algorithm & Sparsity(after)

This screenshot shows the program that imports python libraries, defines functions and uses content based SVD algorithm for prediction of recommendations and calculates root mean square error value and sparsity.

Conclusion and Future Work

7.1. Conclusion:

- We successfully implemented basic algorithms in java, using eclipse & mahout libraries. The implemented algorithms are: item-item, user-user similarity, tanimoto, Pearson coefficient, Slope one, SVD recommendation.
- We have successfully installed the softwares which are needed for Recommendation System viz Hadoop, Scala, SBT and Apache Spark and practiced basic programs of each.
- We studied and filtered Collaborative filtering algorithms and chose the three algorithms which gave best results for implementation. The algorithms are Pearson coefficient algorithm and SVD algorithm.
- Successfully implemented Hybrid Algorithm on dataset using collaborative filtering and content based filtering.
- Analyzed the Sparsity of the dataset before and after which is 6.3% accurately, increase which denotes that the hybrid algorithm which we used reduces the sparsity and increases the efficiency of the recommender system.
- Further implementation of modules is under work.

7.2. Future Work:

Future works are desirable in order to keep comparing the recommendation algorithms implementations available in the newer releases of Apache Spark and R language, since both engines for large-scale data processing are rapidly evolving. It will help to introduce new clustering algorithm using the scala.

Spark is now at version 2.0.0, released on July 2016. Since release 1.3 a new Data-Frames API was introduced, that provides powerful and convenient operators when working with structure datasets. Since release 1.4 they provide the SparkR, an R binding for Spark based on Spark's new Data Frame API. SparkR gives R users access to Spark's scale-out parallel runtime along with all of Spark's input and output formats. It also supports calling directly into Spark SQL.

Finally, a direct way to improve our work is by validation the algorithms on larger datasets, with millions or billions of ratings, and if possible from different recommendation domains such as music, movie, products and news.

Also a comparative study between serial implementation of our algorithms and parallel implementation can be done as a future work and new research can be found in this field using different tools and languages like R, Scala.

References:

[1]Ungar LH, Foster DP. "*Clustering methods for collaborative filtering*". In AAAI workshop on recommendation systems (Vol. 1, pp.114-129), Jul 26 1998

[2]Scala. *http://www.scala-lang.org*

[3] Apache Hive. *http://hadoop.apache.org/hive*

[4] http://grouplens.org/datasets/movielens/

[5]Ungar LH, Foster DP. "*Clustering methods for collaborative filtering*". In AAAI workshop on recommendation systems (Vol. 1, pp.114-129), Jul 26 1998

[6]J. Jiang, J. Lu, G. Zhang, and G. Long, "*Scaling-up item-based collaborative filtering recommendation algorithm based onhadoop*," in Services (SERVICES), 2011 IEEE World Congress on. IEEE, 2011,pp. 490–497.

[7]Adomavicius, Gediminas, and Alexander Tuzhilin. "*Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*." Knowledge and Data Engineering, IEEE Transactions on 17.6 (2005): 734-749.

[8] Burke, Robin. "*Hybrid web recommender systems*." In The adaptive web, pp. 377-408. Springer Berlin Heidelberg, 2007.

[9]Spark Programming Guide - Spark 1.6.0 Documentation, *http://spark.apache.org/docs/latest/programming-guide.html*

[10] "*A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark*" Sasmita Panigrahia*, Rakesh Ku. Lenkaa, Ananya Stitipragyana.

[11] Globo.com, "*Porquetrabalharna globo.com?," [Online].* Available: https://github.com/globocom/IwantToWorkAtGloboCom.

[12] Kulkarni, Swapna, "*A Recommendation Engine Using Apache Spark*" (2015). Master's Project,456

[13] Francesceo Ricci , Lior Rokach and Bracha Shapira "*Introduction To Recommender System Handbook*" DOI 10.1007/978-0-387-85820-3_1, @ Springer Science+Business Media ,LLC 2011

[14] Burke Robin, "*Hybrid Web Recommender Systems*" In The adaptive web pages 377-408 Springer Berlin Heidelberg ,2007

[15] Jianwen Chen, Ling Feng ,"*Efficient Pruning Algorithm For Top-K Ranking On Database With Value Uncertinty*" CIKM-13 Pages ,2231-2236 ,2013 ,ISBN-978-1-4503-2263-8

[16] Jiawei Han, "*Data Mining:Concepts And Techniques*" ,2005 ISBN-1558609016

[17] Ungar LH, Foster DP. "*Clustering Methods For Collaborative Filtering*". In AAAI Workshop On recommendations systems (Vol. 1,pp 114-129) July 26 1998

[18] "*Big Data Product Watch 8/28/15 : Streaming Analytics high Performance Computing And More.*" ICT Monitor Worldwide, August 29 2015 Issue.

[19] Zhcng Wen "*Recommendation System Based On Collaborative Filtering*" Dec 12,2008

[20] Pagare Reena, Patil Shalmali A. "*Study of Collaborative Filtering Recommendation Algorithm-Scalability Issue*" International Journal of Computer Applications , Volume 67-Number 25,2013

[21] Sasmita Panigrahi, Rakesh Ku. Lenka And Ananya Stitipragyan ,"*A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark*" Procedia Computer Science 1000-1016,2016

[22] Qun Liu,Xiaobing Li* "*A New Parallel Item-Based Collaborative Filtering Algorithm Based On Hadoop*" doi:10.17706/jsw.10.4.416-426, 2014

[23] A.H.M. Ragab , A.F.S.Mashat and A.M.Khedra, "*HRSPCA: Hybrid Recommender System For Predicting College Admission,*" doi:10.1109/ISDA.2012.6416521 pp 107-113,2012

Activity Chart:

Sr. No:	Activity	Duration in Days	End Date	Month- 2016									
				7	8	9	10	11	12	1	2	3	4
1	Team Formation, Project Proposal/ Problems Identification (at least 3 as per format), Background work & Finalization	11											
2	Project Title Finalization & Literature Survey Review	7											
3	System Requirements Specification	7											
4	Synopsis Submission & Presentation	8											
5	Project Design: high Level Project Design: low Level, System Requirements	22											
6	Project Phase- I: Report Making	7											
7	Coding 50%	13											
8	Final Presentation and evaluation of Project Phase-I	8											

9	Self study on relevant technologies	7											
10	Self study on relevant technologies	7											
11	Self study on relevant technologies	30											
12	100% Coding & Preparation for Special Achievements	15											
13	Participation in competitions & presentations, Report making	10											
14	Final Report making & submission	15											
15	Final Presentation	7											