Seminar Report

On

"Movie Recommendation System Using Machine Learning"

By

Viraj bhise

72010287B

Under the Guidance
of

MRS. GRISHMA BOBHATE

Sinhgad Institutes

Sinhgad Technical Education Society's

RMD SINHGAD SCHOOL OF ENGINEERING, WARJE, PUNE-58

Department of Computer Engineering

Affiliated to

Savitribai Phule Pune University, Pune

Sinhgad Institutes

RMD SINHGAD SCHOOL OF ENGINEERING,

WARJE, PUNE-58

Department of Computer Engineering

CERTIFICATE

This is to certify that **Viraj Bhise** from Third Year Computer Engineering has successfully completed his / her seminar work titled **"Movie Recommendation system using Machine learning"** at RMD Sinhgad School of Engineering, Warje, Pune in the partial fulfillment of the Bachelors Degree in Engineering.

(MRS. Grishma Bobhate)
Guide
Department of Computer Engineering

(Ms. Vina M. Lomte)
Head,
Department of Computer Engineering

(Dr. V. V. Dixit)
Principal,
RMD Sinhgad School of Engineering Pune – 58

Place:PUNE

Date:

ACKNOWLEGMENT

It is our pleasure to acknowledge sense of gratitude to all those who helped us in making this seminar. I am highly indebted to **(MRS. Grishma Bobhate)** for their guidance and constant supervision as well as for providing necessary information regarding the seminar & also for their support

I would like to express my gratitude towards **Prof. Vina M. Lomte (Head - Department of Computer Engineering)** for their kind co-operation and encouragement which help me for providing the required facilities

Finally we wish to thank and appreciate to all our teachers and friends for their constructive comments, suggestions and guidance and all those directly or indirectly helped us in completing this seminar.

VIRAJ BHISE

ABSTRACT

ABSTRACT A recommendation engine filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behavior of a customer and based on that, recommends products which the users might be likely to buy. If a completely new user visits an e-commerce site, that site will not have any past history of that user. So how does the site go about recommending products to the user in such a scenario? One possible solution could be to recommend the best selling products, i.e. the products which are high in demand. Another possible solution could be to recommend the products which would bring the maximum profit to the business. Three main approaches are used for our recommender systems. One is Demographic Filtering i.e They offer generalized recommendations to every user, based on movie popularity and/or genre. The System recommends the same movies to users with similar demographic features. Since each user is different , this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience. Second is content-based filtering, where we try to profile the users interests using information collected, and recommend items based on that profile. The other is collaborative filtering, where we try to group similar users together and use information about the group to make recommendations to the user.

TABLE OF CONTENTS

1 Introduction

A recommendation system is a type of information filtering system which attempts to predict the preferences of a user, and make suggests based on these preferences. There are a wide variety of applications for recommendation systems. These have become increasingly popular over the last few years and are now utilized in most online platforms that we use. The content of such platforms varies from movies, music, books and videos, to friends and stories on social media platforms, to products on e-commerce websites, to people on professional and dating websites, to search results returned on Google. Often, these systems are able to collect information about a users choices, and can use this information to improve their suggestions in the future. For example, Facebook can monitor your interaction with various stories on your feed in order to learn what types of stories appeal to you. Sometimes, the recommender systems can make improvements based on the activities of a large number of people. For example, if Amazon observes that a large number of customers who buy the latest Apple Macbook also buy a USB-C-toUSB Adapter, they can recommend the Adapter to a new user who has just added a Macbook to his cart. Due to the advances in recommender systems, users constantly expect good recommendations. They have a low threshold for services that are not able to make appropriate suggestions. If a music streaming app is not able to predict and play music that the user likes, then the user will simply stop using it. This has led to a high emphasis by tech companies on improving their recommendation systems. However, the problem is more complex than it seems. Every user has different preferences and likes. In addition, even the taste of a single user can vary depending on a large number of factors, such as mood, season, or type of activity the user is doing. For example, the type of music one would like to hear while exercising differs greatly from the type of music he'd listen to when cooking dinner. Another issue that recommendation systems have to solve is the exploration vs exploitation problem. They must explore new domains to discover more about the user, while still making the most of what is already known about of the user. Three main approaches are used for our recommender systems.

One is Demographic Filtering i.e They offer generalized recommendations to every user, based on movie popularity and/or 7 genre. The System recommends the same movies to users with similar demographic features. Since each user is different , this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience. Second is content-based filtering, where we try to profile the users interests using information collected, and recommend items based on that profile. The other is collaborative filtering, where we try to group similar users together and use information about the group to make recommendations to the user.

1.1 OBJECTIVES

Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. ... This article focuses on the movie recommendation systems whose primary objective is to suggest a recommender system through data clustering and computational intelligence.

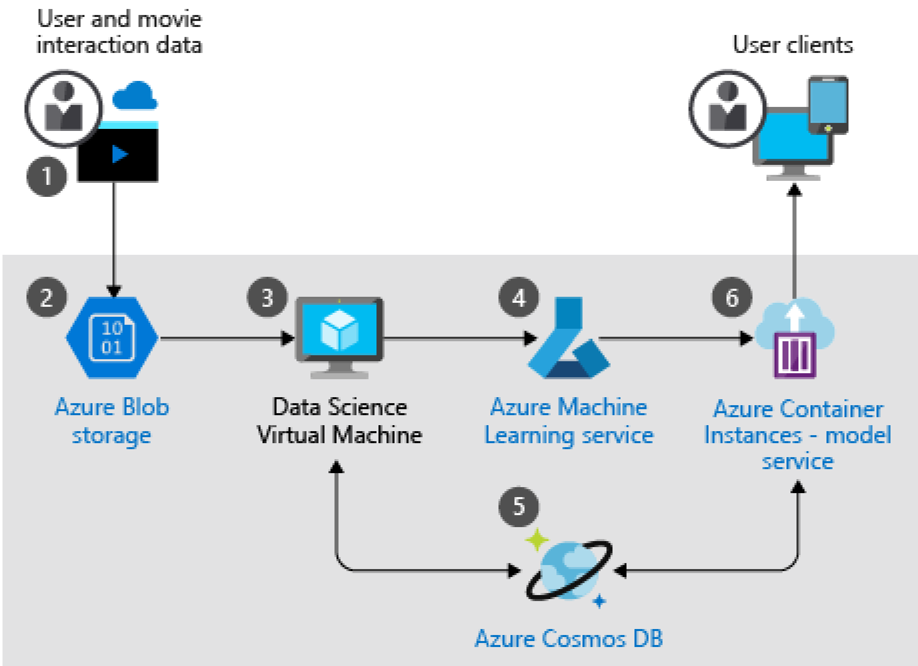
1.2 PROBLEM STATEMENT

Problem Statement : Recommend content to user which is unknown or similar to watching history or based on previous rating for content .

2. Literature Survey

MOVREC is a movie recommendation system presented by D.K. Yadav et al. based on collaborative filtering approach. Collaborative filtering makes use of information provided by user. That information is analyzed and a movie is recommended to the users which are arranged with the movie with highest rating first. Luis M Capos et al has analyzed two traditional recommender systems i.e. content based filtering and collaborative filtering. As both of them have their own drawbacks he proposed a new system which is a combination of Bayesian network and collaborative filtering. A hybrid system has been presented by Harpreet Kaur et al. The system uses a mix of content as well as collaborative filtering algorithm. The context of the movies is also considered while recommending. The user - user relationship as well as user - item relationship plays a role in the recommendation. The user specific information or item specific information is clubbed to form a cluster by Utkarsh Gupta et al. using chameleon. This is an efficient technique based on Hierarchical clustering for recommender system. To predict the rating of an item voting system is used. The proposed system has lower error and has better clustering of similar items. Urszula Kużelewska et al. proposed clustering as a way to deal with recommender systems. Two methods of computing cluster representatives were presented and evaluated. Centroid-based solution and memory-based collaborative filtering methods were used as a basis for comparing effectiveness of the proposed two methods. The result was a significant increase in the accuracy of the generated recommendations when compared to just centroid-based method. Costin-Gabriel Chiru et al. proposed Movie Recommender, a system which uses the information known about the user to provide movie recommendations. This system attempts to solve the problem of unique recommendations which results from ignoring the data specific to the user. The psychological profile of the user, their watching history and the data involving movie scores from other websites is collected. They are based on aggregate similarity calculation. The system is a hybrid model which uses both content based filtering 10 and collaborative filtering. To predict the difficulty level of each case for each trainee Hongli LIn et al. proposed a method called contentboosted collaborative filtering (CBCF).The algorithm is divided into two stages, First being the content-based filtering that improves the existing trainee case ratings data and the second being collaborative filtering that provides the final predictions. The CBCF algorithm involves the advantages of both CBF and CF, while at the same time, overcoming both their disadvantages.

4. Architecture of the System



This scenario covers the training and evaluating of the machine learning model using the Spark alternating least squares (ALS) algorithm on a dataset of movie ratings. The steps for this scenario are:

1. The front-end website or app service collects historical data of user-movie interactions, which are represented in a table of user, item, and numerical rating tuples.
2. The collected historical data is stored in blob storage.
3. A Data Science Virtual Machine (DSVM) is often used for smaller workloads to experiment or develop a product based on a Spark ALS recommender model. The ALS model is trained using a training dataset, which is produced from the overall dataset by applying a data splitting strategy. For example, the dataset can be split into sets randomly, chronologically, or stratified, depending on the business requirement.

Similar to other machine learning tasks, a recommender is validated by using evaluation metrics (for example, precision@k, recall@k, MAP, nDCG@k).

4. Azure Machine Learning coordinates the experimentation, such as hyperparameter sweeping and model management.
5. A trained model is saved to Azure Cosmos DB, which can then be applied for recommending the top k movies for a given user.
6. The model is then deployed onto a web or app service by using Azure Container Instances or Azure Kubernetes Service.

For an in-depth guide to building and scaling a recommender service, see the article Build a real-time recommendation API on Azure.

4. TECHNOLOGY USED IN SYSTEM WITH MODEL

1. Python Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

2. Jupyter Notebook The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more

5. ANALYTICAL STUDY

- **Content-based Filtering**

  This filtration strategy is based on the data provided about the items. The algorithm recommends products that are similar to the ones that a user has liked in the past. This similarity (generally cosine similarity) is computed from the data we have about the items as well as the user's past preferences.

  For example, if a user likes movies such as 'The Prestige' then we can recommend him the movies of 'Christian Bale' or movies with the genre 'Thriller' or maybe even movies directed by 'Christopher Nolan'.So what happens here the recommendation system checks the past preferences of the user and find the film "The Prestige", then tries to find similar movies to that using the information available in the database such as the lead actors, the director, genre of the film, production house, etc and based on this information find movies similar to "The Prestige".

  Disadvantages

  1. Different products do not get much exposure to the user.
  2. Businesses cannot be expanded as the user does not try different types of products.

- **Collaborative Filtering**

  This filtration strategy is based on the combination of the user's behavior and comparing and contrasting that with other users' behavior in the database. The history of all users plays an important role in this algorithm. The main difference between content-based filtering and collaborative filtering that in the latter, the interaction of all users with the items influences the recommendation algorithm while for content-based filtering only the concerned user's data is taken into account.

  There are multiple ways to implement collaborative filtering but the main concept to be grasped is that in collaborative filtering multiple user's data influences the outcome of the recommendation. and doesn't depend on only one user's data for modeling.

There are 2 types of collaborative filtering algorithms:

- **User-based Collaborative filtering**

  The basic idea here is to find users that have similar past preference patterns as the user 'A' has had and then recommending him or her items liked by those similar users which 'A' has not encountered yet. This is achieved by making a matrix of items each user has rated/viewed/liked/clicked depending upon the task at hand, and then computing the similarity score between the users and finally recommending items that the concerned user isn't aware of but users similar to him/her are and liked it.

  For example, if the user 'A' likes 'Batman Begins', 'Justice League' and 'The Avengers' while the user 'B' likes 'Batman Begins', 'Justice League' and 'Thor' then they have similar interests because we know that these movies belong to the super-hero genre. So, there is a high probability that the user 'A' would like 'Thor' and the user 'B' would like The Avengers'.

  Disadvantages

  1. People are fickle-minded i.e their taste change from time to time and as this algorithm is based on user similarity it may pick up initial similarity patterns between 2 users who after a while may have completely different preferences.
  2. There are many more users than items therefore it becomes very difficult to maintain such large matrices and therefore needs to be recomputed very regularly.
  3. This algorithm is very susceptible to shilling attacks where fake users profiles consisting of biased preference patterns are used to manipulate key decisions.

- **Item-based Collaborative Filtering**

  The concept in this case is to find similar movies instead of similar users and then recommending similar movies to that 'A' has had in his/her past preferences. This is executed by finding every pair of items that were rated/viewed/liked/clicked by the same user, then measuring the similarity of those rated/viewed/liked/clicked across all user who rated/viewed/liked/clicked both, and finally recommending them based on similarity scores.

Here, for example, we take 2 movies ‘A’ and ‘B’ and check their ratings by all users who have rated both the movies and based on the similarity of these ratings, and based on this rating similarity by users who have rated both we find similar movies. So if most common users have rated ‘A’ and ‘B’ both similarly and it is highly probable that ‘A’ and ‘B’ are similar, therefore if someone has watched and liked ‘A’ they should be recommended ‘B’ and vice versa.

Advantages over User-based Collaborative Filtering

1. Unlike people’s taste, movies don’t change.
2. There are usually a lot fewer items than people, therefore easier to maintain and compute the matrices.
3. Shilling attacks are much harder because items cannot be faked.

6 . REAL TIME APPLICATIONS

- **Netflix**
- **HULU**
- **Amazon Prime**
- **Youtube**
- **IMDB**
- **Crakle**

7. APPLICATIONS

- e-Commerce

Industry where recommendation systems were first widely used. With millions of customers and data on their online behavior, e-commerce companies are best suited to generate accurate recommendations

- Retail

Target scared shoppers back in 2000s when Target systems were able to predict pregnancies even before mothers realized their own pregnancies. Shopping data is the most valuable data as it is the most direct data point on a customer's intent. Retailers with troves of shopping data are at the forefront of companies making accurate recommendations

- Media

Similar to e-commerce, media businesses are one of the first to jump into recommendations. It is difficult to see a news site without a recommendation system.

- Banking

A mass market product that is consumed digitally by millions. Banking for masses and SMEs are prime for recommendations. Knowing a customer's detailed financial situation, along with their past preferences, coupled by data of thousands of similar users is quite powerful.

- Telecom

Shares similar dynamics with banking. Telcos have access to millions of customers whose every interaction is recorded. Their product range is also rather limited compared to other industries, making recommendations in telecom an easier problem.

8. CONCLUSION

A hybrid approach is taken between context based filtering and collaborative filtering to implement the system. This approach overcomes drawbacks of each individual algorithm and improves the performance of the system. Techniques like Clustering, Similarity and Classification are used to get better recommendations thus reducing MAE and increasing precision and accuracy. In future we can work on hybrid recommender using clustering and similarity for better performance. Our approach can be further extended to other domains to recommend songs, video, venue, news, books, tourism and e-commerce sites, etc

9. FUTURE SCOPE

- In collaborative filtering, we have a problem of sparsity of data. Very few users actually rate the same movie.
- We can use Clustering Algorithms like K-Means to cluster items or users or both based on their attributes.
- In the hybrid approach, we can use more features to get better predictions.
- Neural Networks and Deep Learning have been all the rage the last couple of years in many different fields, and it appears that they are also helpful for solving recommendation system problems.

10. Bibliography : References

- 1. Verma, J. P., Patel, B., & Patel, A. (2015). Big data analysis: Recommendation system with Hadoop framework. In 2015 IEEE International Conference on Computational Intelligence & Communication Technology (CICT). IEEE.
- 2. Katarya, R., & Verma, O. P. (2016). A collaborative recommender system enhanced with particle swarm optimization technique. Multimedia Tools and Applications, 75(15), 9225– 9239.
- 3. https://docs.databricks.com/_static/notebooks/cs100x-2015-introduction-to-big-data/module5–machine-learning-lab.html.
- 4. Wei, J., et al. (2016). Collaborative filtering and deep learning based hybrid recommendation for cold start problem. In 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing,
- 5. Kupisz, B., & Unold, O. (2015). Collaborative filtering recommendation algorithm based on Hadoop and Spark. In 2015 IEEE International Conference on Industrial Technology (ICIT). IEEE.
- 6. Zeng, X., et al. (2016). Parallelization of latent group model for group recommendation algorithm. In IEEE International Conference on Data Science in Cyberspace (DSC). IEEE.
- 7. Ponnam, L. T., et al. (2016). Movie recommender system using item based collaborative filtering technique. In International Conference on Emerging Trends in Engineering, Technology, and Science (ICETETS). IEEE.