Indian Institute of Information Technology Bhopal

ITW (IT-216) Project

Movie recommendation system

Department- Information Technology

Authors: Jatin Wadhwani(23U030**51**), Yash Zinzala (23U030**60**), Dhruv Gupta(23U030**61**), Prabhakar Singh(23U030**55**)

Submitted To:

Dr. Supriya Agrawal

Asst. Proffesor IIIT Bhopal

Contents

1 Introduction

1.1 Background

Our project centers on exploring traditional machine learning (ML) models commonly used in movie recommendation systems. These models include collaborative filtering[2] techniques such as user-based and item-based filtering, which leverage similarities between users or items to generate recommendations. User-based filtering identifies users with similar preferences and recommends movies based on their collective behavior, on the other hand item-based filtering suggests similar movies given one movie as an input.

By focusing on these established ML models, our objective is to develop a reliable movie recommendation system that provides personalized suggestions based on user behavior and preferences, without relying on deep learning architectures.

1.2 Objective

The primary objective of the movie recommendation system using K-Nearest Neighbors (KNN) is to develop an intelligent system that can suggest relevant movies to users based on their preferences and the preferences of similar users. The system utilizes collaborative filtering, specifically KNN, to identify patterns in user behavior and recommend movies that align **with** individual tastes. The key objectives include:

1. **Personalized Movie Recommendations**: To create a personalized experience for users by recommending movies they are likely to enjoy, based on the ratings and preferences of users with similar tastes.

2. **Use of KNN Algorithm for Collaborative Filtering**: To implement the KNN algorithm for identifying similarities between users or movies based on past ratings. This method helps predict a user's interest in a movie by looking at the ratings of similar users or movies.

3. **Enhancing User Experience**: To improve user engagement and satisfaction by suggesting movies that match the user’s viewing history and preferences, thereby reducing the time spent searching for movies to watch.

1.3 Collaborative filtering

Collaborative filtering is a technique used in recommendation systems to make personalized recommendations based on the preferences and behaviors of other users. The underlying assumption is that if two users have agreed on past preferences (such as liking similar movies or products), they are likely to agree on future preferences as well. It is a data-driven approach that leverages the wisdom of the crowd, where the collective behavior of users is used to make predictions for individual users.

There are two primary types of collaborative filtering:

1. **User-based Collaborative Filtering**: In this approach, the system recommends items (such as movies, products, etc.) to a user based on the preferences of other users who are similar to them. Similarity between users is typically calculated by comparing their past behaviors or ratings. If User A likes movies that are similar to the ones liked by User B, then User B is considered a neighbor of User A, and movies liked by User B that User A hasn't seen are recommended.

    a. **Example**: If two users both liked similar movies, then movies liked by one user can be recommended to the other.

2. **Item-based Collaborative Filtering**: Instead of focusing on user similarity, this method focuses on item similarity. If a user has rated an item (e.g., a movie), the system recommends similar items that other users who liked the same item also liked. Similarity between items is measured by comparing the ratings from different users.

    a. **Example**: If a user likes a particular movie, the system will recommend other movies that are often rated highly by users who liked the same movie.

1.4. Tools Used: Google COLAB, GITHUB, PYTHON

2 Libraries Used

pandas (imported as pd)
Role: Data manipulation and analysis.
Usage: Loading, cleaning, transforming, and manipulating the movie dataset (usually in CSV or other formats) into a suitable format for analysis and modeling. It provides data structures like DataFrames for efficient handling of tabular data.

NumPy (imported as np)
Role: Numerical computations and array operations.
Usage: Performing mathematical calculations on movie ratings, creating and manipulating arrays for data representation, and handling numerical data efficiently.

scikit-learn (imported as sklearn)
Role: Machine learning algorithms and tools.
Usage: Provides the core implementation of the kNN algorithm (NearestNeighbors class) for finding similar users. It also offers tools for data preprocessing, model evaluation, and other machine-learning-related tasks.

scipy.sparse
Role: Handling sparse matrices efficiently.
Usage: In scenarios where the user-item rating matrix is sparse (many missing ratings), csr_matrix from scipy.sparse is used to store and manipulate the data more efficiently.

matplotlib.pyplot (imported as plt)
Role: Data visualization.
Usage: Creating plots and charts to visualize movie ratings, user similarities, and other patterns in the data. This helps in exploratory data analysis and understanding the data better.

seaborn (imported as sns)
Role: Statistical data visualization.
Usage: Building on top of matplotlib, seaborn provides more advanced and statistically informative visualizations, such as heatmaps for visualizing user-item rating matrices, which are helpful for understanding relationships and patterns in the data.

3 Approach used

Collaborative filtering using KNN

K-Nearest Neighbors (KNN) is a simple, non-parametric, and lazy supervised learning algorithm used for classification and regression tasks. It works by finding the **K** closest data points (neighbors) to a given query point in the feature space and making predictions based on those neighbors. The key principle behind KNN is that similar data points tend to be close to each other in the feature space, meaning that if a data point is similar to others, it is likely to share the same class or value.

Using KNN we have implemented both user and item based collaborative filtering. Firstly, movie-user matrix is created to represent the ratings given by each user for a particular movie. By identifying the k nearest neighbors to a user's vector, we personalize movie recommendations to align with similar users' tastes. This aspect is particularly beneficial for streaming platforms, where tailored content suggestions on a user's homepage to recommend the movies on a general basis without any genre bias.

Now, this is further extended to give genre wise recommendations, for this we added only movies of particular genre in movie-user matrix and implemented KNN similarly.

On the other hand, our model gives item-based recommendations by identifying similar movies using KNN, instead of focusing solely on user preferences. This functionality proves invaluable when users search for specific movies or upon completing a movie, where recommending related content enhances the overall user experience.

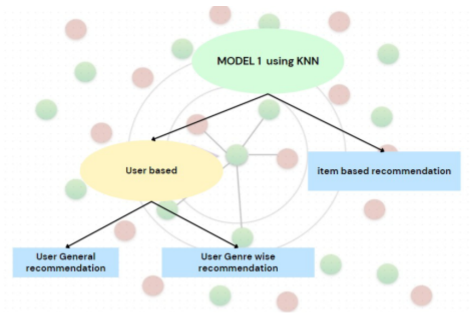
3.2. About dataset

The dataset used in this project describes 5-star rating and free-text tagging activity from **MovieLens**, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. All selected users had rated at least 20 movies. It has 4 files links.csv, movies.csv, ratings.csv and tags.csv. Detailed description can be found in the readme file.
Two dataset files imdb.csv and tmdb.csv were genrated by scrapping data for all the movies from the corresponding sites to use some important features like rating, user and critic reviews etc.

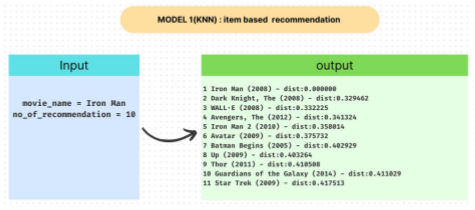
4. Result and key outcomes

Our KNN-based movie recommendation system successfully generated personalized recommendations by identifying similar users based on their movie ratings. The system demonstrated good accuracy, showing that the predicted ratings were close to the actual ratings.

We experimented with different values of K, and found that K=10 produced the best results, balancing the need for specific and general recommendations. Smaller values of K gave more personalized suggestions, while larger values reduced relevance.


MODEL 1 using KNN
User based
item based recommendation
User General recommendation
User Genre wise recommendation

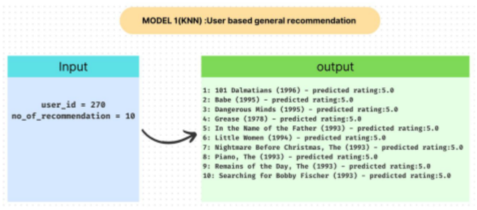
(a) model



```
movie_name = Iron Man
no_of_recommendation = 10
```

```
1 Iron Man (2008) - dist:0.000000
2 Dark Knight, The (2008) - dist:0.329462
3 WALL·E (2008) - dist:0.332225
4 Avengers, The (2012) - dist:0.341324
5 Iron Man 2 (2010) - dist:0.358014
6 Avatar (2009) - dist:0.375732
7 Batman Begins (2005) - dist:0.402929
8 Up (2009) - dist:0.403264
9 Thor (2011) - dist:0.410508
10 Guardians of the Galaxy (2014) - dist:0.411029
11 Star Trek (2009) - dist:0.417513
```

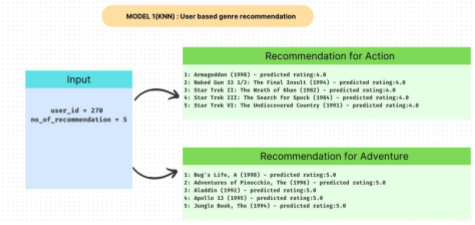
(b) Movie(item) based



```
user_id = 270
no_of_recommendation = 10
```

```
1: 101 Dalmatians (1996) - predicted rating:5.0
2: Babe (1995) - predicted rating:5.0
3: Dangerous Minds (1995) - predicted rating:5.0
4: Grease (1978) - predicted rating:5.0
5: In the Name of the Father (1993) - predicted rating:5.0
6: Little Women (1994) - predicted rating:5.0
7: Nightmare Before Christmas, The (1993) - predicted rating:5.0
8: Piano, The (1993) - predicted rating:5.0
9: Remains of the Day, The (1993) - predicted rating:5.0
10: Searching for Bobby Fischer (1993) - predicted rating:5.0
```

(c) User based general Recommendations


MODEL 1(KNN) : User based genre recommendation
Input
Recommendation for Action
Recommendation for Adventure

```
user_id = 270
no_of_recommendation = 5
```

```
1: Armageddon (1998) - predicted rating:4.0
2: Naked Gun 33 1/3: The Final Insult (1994) - predicted rating:4.0
3: Star Trek II: The Wrath of Khan (1982) - predicted rating:4.0
4: Star Trek III: The Search for Spock (1984) - predicted rating:4.0
5: Star Trek VI: The Undiscovered Country (1991) - predicted rating:4.0
```

```
1: Bug's Life, A (1998) - predicted rating:5.0
2: Adventures of Pinocchio, The (1996) - predicted rating:5.0
3: Aladdin (1992) - predicted rating:5.0
4: Apollo 13 (1995) - predicted rating:5.0
5: Jungle Book, The (1994) - predicted rating:5.0
```

(d) User based genre wise recommendations

4.1 Key Learnings from the project

Importance of Data Quality: The accuracy of recommendations heavily depends on the quality and quantity of user data. Sparse datasets, where users have rated only a few movies, made it challenging for the model to make accurate predictions.

Choosing the Right K Value: Selecting the optimal value of K is crucial for balancing between personalized and generalized recommendations. We learned that K=10 provided the best performance, as it offered relevant yet diverse suggestions.

Model Limitations: While KNN is effective for small to medium-sized datasets, it struggles with sparse data and cold start issues. This reinforced the need for exploring more advanced techniques in future iterations, such as matrix factorization or deep learning-based approaches.

Practical Application of KNN: The project provided hands-on experience in implementing a real-world recommendation system and deepened our understanding of collaborative filtering and user-based models.

5 Summary

The project outlines a movie recommendation system based on traditional machine learning models, specifically focusing on collaborative filtering techniques like KNN. The system aims to provide personalized movie suggestions by analyzing user behavior and preferences. Further developments can be done by using advanced ML techniques like neural network to increase the scope and accuracy of model.

References

1. Herv´ Abdi. Singular value decomposition (svd) and generalized singular value decomposition (gsvd). URL https://www.cimat.mx/~alram/met_num/clases/Abdi-SVD2007-pretty.pdf.
2. Yuliia Kniazieva. Introduction to the movie recommendation system architecture. URL https: // labelyourdata.com/articles/movie-recommendation-with-machine-learning