Movie Recommendation System

Akhila Reddy Dodda [1] **Akshay Krishna Sheshadri** [1] **Muni Lohith Krishna Mohan Konidala** [1]

Abstract

Recommender systems are important tools that analyze data on user preferences and behavior to make personalized recommendations. They can help improve the user experience, increase engagement, and drive revenue for businesses and they are widely used in e-commerce, social media, and other applications. We plan to use PySpark to build recommender systems to suggest products, or services, or content to users based on their preferences and behavior. In particular, we plan to build a Movie recommendation system and evaluate it using various performance metrics such as accuracy, Mean average precision @ K (MAP@K), and Mean Average Recall @ K (MAR@K).

1 Introduction

Recommender systems are software tools and algorithms that analyze data on user preferences and behavior to make personalized recommendations for items such as products, services, or content. The goal of these systems is to predict what items a user might be interested in and present them with recommendations that are relevant and useful. They are a powerful tool for businesses looking to improve their customer experience and gain a competitive edge.

Recommender systems have become increasingly important in recent years due to the explosion of digital content and the growing amount of data available on user behavior. They are used in a variety of applications, from e-commerce and online advertising to music and video streaming services, and they can help improve customer engagement, increase sales, and enhance the user experience.

For our use-case, the primary objective of our recommendation system is to predict and filter only those movies that a user is likely to prefer, based on relevant user and movie data.

2 Data

We plan to use 'The Movies Dataset'. This consists of different files containing information as shown in the table below:

Feature	Description
User id	Each user has an unique id
Movie id	Each movie had an unique id
Rating	Mapping of user id, movie id, rating and timestamp
Movie Features	Movie metadata including the Genre and actors of the movie.
Tags	Mapping of tags, user id, and movie id.

Table 1. Data present in The Movies Dataset

We will load these different files into pyspark to perform efficient joins and other transformations and prepare the data to process for our recommender system.

3 Methodology

3.1 Data preparation, exploration, and pre-processing

To begin, we make sure that the data that includes information about movies, users, and their interactions (user ratings of movies) is in a format that can be loaded into PySpark. After loading the data, we will apply appropriate joins and transformations based on the different data sources to prepare the dataset in the required format for training our recommender system.

After this, we will perform exploratory data analysis (EDA) using PySpark to visualize and comprehend the data's structure and characteristics.

Finally, we will preprocess and query the data using SparkSQL, which involves tasks such as data cleaning, handling missing values, feature engineering, and creating train-validation-test splits. The processed data will then be used as input by the Machine Learning Model.

[*] Equal contribution [1] Manning College of Information & Computer Sciences, University of Massachusetts Amherst. Correspondence to: Akhila Reddy Dodda <akhilareddyd@umass.edu>, Akshay Krishna Sheshadri <asheshadri@umass.edu>, Muni Lohith Krishna Mohan Konidala <mkonidala@umass.edu>.

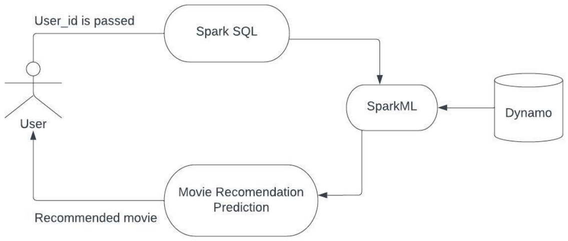

Figure 1. Usecase diagram for the proposed system

The project workflow is presented in Figure 1, and an interactive dashboard will be provided for users to access the system by entering their ID and receiving personalized recommendations.

3.2 Training (filtering strategy)

Recommender systems come in various types, including content-based filtering, collaborative filtering, and hybrid approaches that blend the two techniques. Content-based filtering evaluates item characteristics like genre, category, or keywords to offer recommendations, while collaborative filtering analyzes user preferences and behavior, such as ratings or past purchases, to identify patterns and suggest similar items. Hybrid approaches combine both techniques to suggest items that match both item characteristics and user behavior.

Movie recommendation systems aim to suggest movies that match the user's preferences based on their data. Here, we opt to use collaborative filtering to get movie recommendations. In particular, we plan to use the Alternating Least Squares (ALS) matrix factorization algorithm that is available within the Spark's Machine Learning Library - MLLib.

3.3 Generating Recommendations and evaluation

During training and validation, the model's accuracy is estimated using metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Their formulaes are as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - f(x_i)|$$

where y_i is the actual recommendation, $f(x_i)$ is the predicted recommendation. x_i.

Apart from these accuracy metrics, we also plan to evaluate the recommender system's performance using metrics such as mean reciprocal rank (MRR), mean average precision @ K (MAP@K), and mean average recall @ K (MAR@K) to measure the quality of a ranked list of recommendations.

4 Milestones

4.1 First Milestone

- Our first goal of the project is to download the Movie Data Pyspark.
- We next want to apply appropriate SQL queries to obtain the appropriate format required for our training data.
- We apply pre-processing on the data to remove all the missing variables.

4.2 Final Project Milestones

- Make a smaller dataset from the final dataset and run the Alternating Least Squares (ALS) on the smaller dataset.
- After observing the performance on the smaller dataset we will perform EDA on the data.
- Split the dataset into training and testing
- Fit the ML model to the dataset and build an ML pipeline
- Evaluate the model on the test data

4.3 Project Extension

- If time permits we would like to make a dashboard for the users to interact with the Recommender System.