Big Data Technologies and Cloud Computing

Movie Recommendation System Using PySpark

Poulomi Banerjee

Github link for code: https://github.com/BPoulomi02/Big-Data-capstone-Project.git

Abstract

Due to this exponential increase in film productions, it becomes challenging for audiences to make a final decision on their preferred choices of content. This paper aims to describe the creation of a movie recommendation system with data analytics and machine learning capabilities for personalized content suggestions. This is achieved using key attributes, including genres, ratings, and user history to boost the precision of recommendation accuracy.

Introduction

The global propagation of cinema has created a demand for systems which are capable of providing personalized suggestions to watch. This research develops a more robust recommendation engine, as it incorporates the multiple parameters, such as genre, year, actors, and ratings, that will give predictions about users' choices. The aim of this system is to improve users' engagement because recommendations ought to be pertinent and adapted to individual tastes.

Aim

The aim of the Movie Recommendation System project is to build a personalized user-centric system that is going to recommend accurate and meaningful movie suggestions according to individuals' tastes. At present, people face an overwhelming array of movie content. That is, it is no longer about whether movies exist but how one might find movie content aligned with personal preferences. This project seeks to bridge this gap by applying cutting-edge technologies like big data analytics and machine learning to simplify the selection process and enrich the viewing experience.

This endeavor is focused on the simplification of decision-making for users with a huge amount of data that is associated with movies such as genres, release years, directors, actors, and ratings. The system analyzes trends and correlations in users' behavior, such as history of viewing and rating, to guide the system's suggestions. It also aims to make the suggestions as diversified as possible to broaden the user's horizons while still being relevant to the user's preferences.

The project emphasizes a multi-faceted approach to capturing the nuances of user preferences. Each movie attribute, such as genre or rating, is assigned a weighted importance based on its influence on user choices. For example, genres, being a primary determinant of interest, carry a higher weight compared to attributes like the country of production. This hierarchical structuring of features allows the system to balance its recommendations, ensuring they are neither overly simplistic nor unnecessarily complex.

One of the major objectives of the project is to achieve scalable and efficient computational algorithms that can handle large amounts of data. The implementation of Apache Spark allows for distributed processing, which allows the system to handle large movie databases and user interactions in real time. The similarity metrics, such as cosine similarity and Jaccard index, allow for a more elaborate comparison between movies, such that those with the closest resemblance to the user's tastes are identified. This system gives a list of the best recommendations based on the similarities between movies to predict ratings for unwatched movies.

The project also considers challenges inherent in recommendation systems. The system will face issues such as the "cold start" problem when recommending movies to new users with limited data. It also aims to balance diversity and accuracy, ensuring that the recommendations are diverse enough to introduce users to new content but still aligned with their preferences. Moreover, the system aspires to adapt dynamically to changes in user behavior so that the recommendations remain relevant over time.

Ultimately, the project aims to enrich the user experience by transforming how viewers discover content. It aims to reduce decision fatigue, increase satisfaction, and provide a seamless interface in the exploration of the enormous landscape of movies. At a higher level, the system can be used to promote more obscure films and directors and thus

democratize access to cinema and deepen the connections between users and the art of storytelling. While that does contribute to the technological innovation, it also brings about cultural and social appreciation for movies.

Methodologies

Data Collection and Preprocessing

Data was collected from several datasets, including movie information, user ratings, and critic reviews. Preprocessing involved the following steps:

- Conversion of raw data files into CSV format.
- Reading and filtering datasets with PySpark DataFrames.
- Remove duplicates and keep the movies with at least 1800 user reviews and at least 150 critic reviews.
- Standardizing data types and dropping irrelevant columns, like non-English titles.

Feature Engineering

Important attributes were derived and weighted based on their importance in the recommendation process:

- Genres: High, since they significantly influence the preferences of users.
- Country: Low, since most users are not significantly concerned about this attribute.
- Director and Actors: Medium, as it affects the user's familiarity and preference.
- Year of Release: Low, since production technology has advanced.
- Ratings (Audience, Critics, and Top Critics): Medium, as there are different tastes of users.

Model Development

The heart of the system is based on a similarity score calculation between movies for the following features:

- Genre Similarity: Jaccard index of genre sets.
- Director and Country Similarity: Equality indicators.
- Actor Similarity: Jaccard index for the top three actors.

- Rating Similarity: Ratios of minimum to maximum ratings for each category.
- Year Similarity: Decay function based on production year differences.
- Cosine Similarity: Applied for user-specific movie ratings.

Rating Prediction: Ratings for unwatched movies were predicted by correlating similarities with watched movies' ratings. The top 10 movies with the highest predicted scores were recommended to the user.

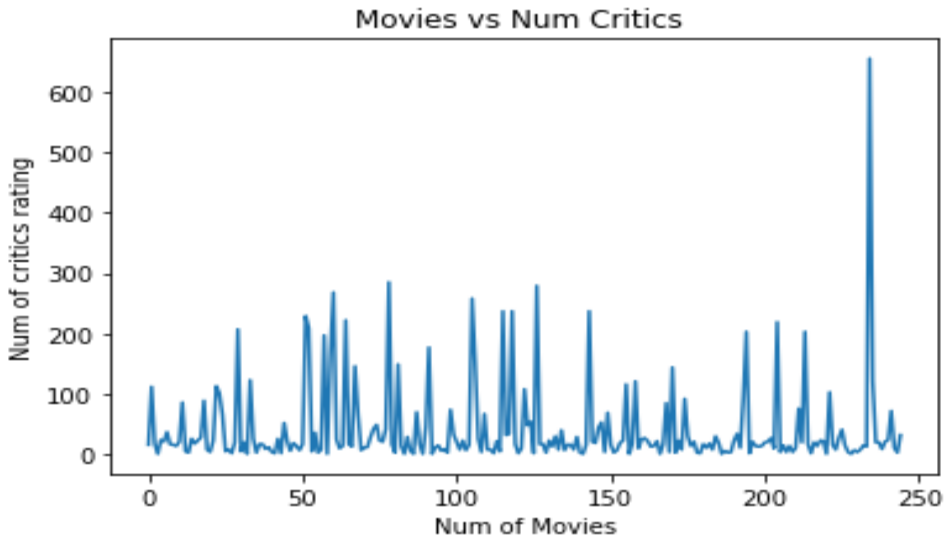
Results

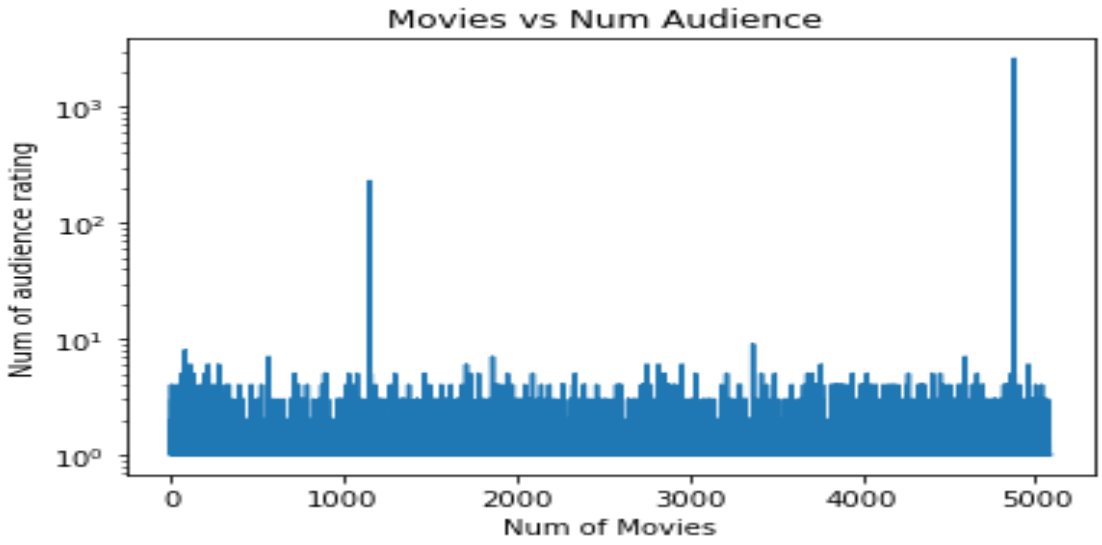
Qualitative performance evaluation was the only way possible as quantitative assessment presented difficulties, given the specifics of personalized recommendations. The runtime to calculate similarity took about 6 minutes, and adding a user ID to predict recommendations took an extra 4 minutes.

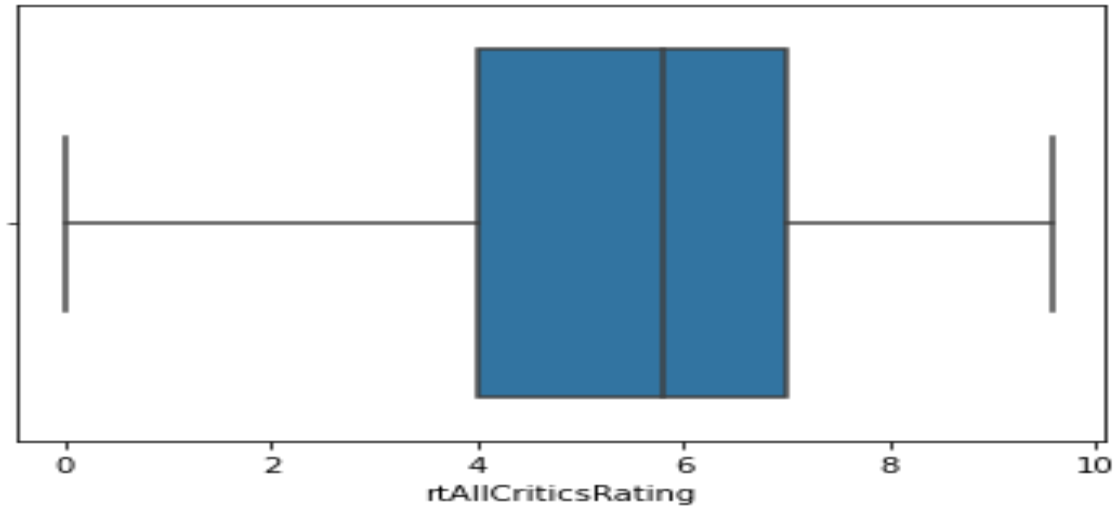
Visualization

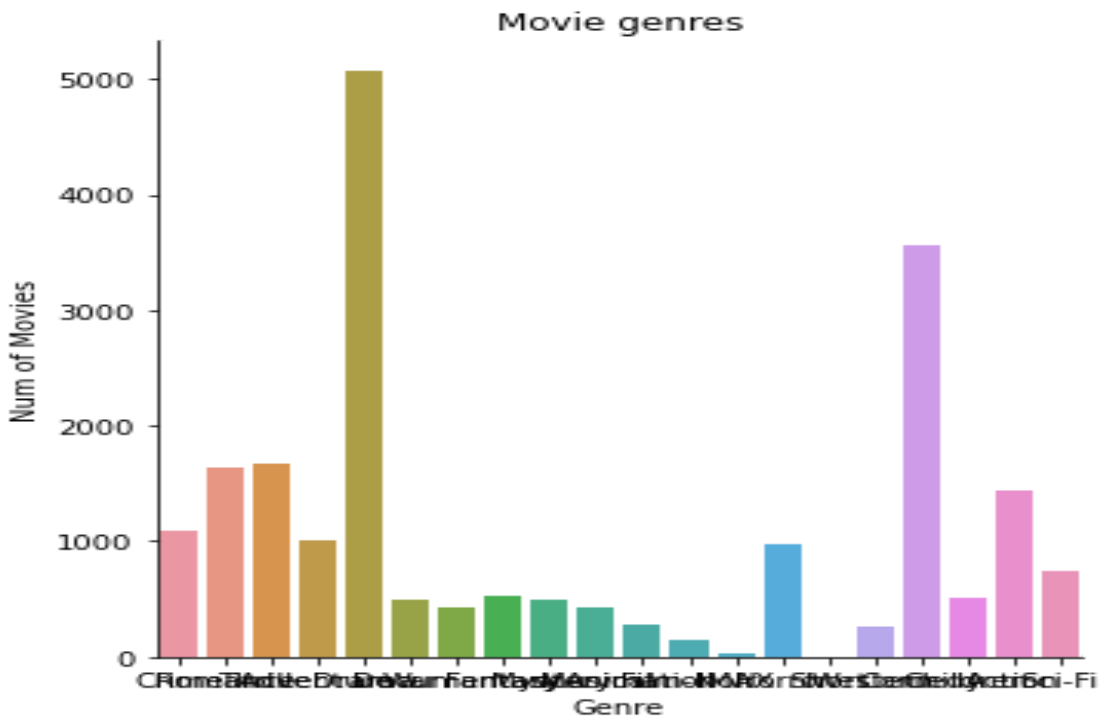
Graphics and charts pointed out major insights:

- Movies vs No. of critics rating
- No. of Movies vs Movie Genres
- All Critics' ratings
- No. of Movies by NO. of Audience.




Movies vs Num Audience
Num of audience rating
10^3
10^2
10^1
10^0
0
1000
2000
3000
4000
5000
Num of Movies


0
2
4
6
8
10
rtAllCriticsRating



Challenges and Improvements

Challenges

- Computational bottlenecks in constructing item similarity matrices due to high dimensionality.
- Scalability was low in the initial implementation of the matrix computation.

Proposed Improvements

- Fully distributed execution of the model to speed up computation and process larger data sets.
- Additional user behavior data, including browsing patterns and social interactions, to make more accurate predictions.

Conclusions

The Movie Recommendation System indeed does demonstrate the possibility of using large amounts of data and machine learning techniques in providing recommendations of

personalized contents. Future work will indeed focus on the optimization of scalability and incorporating additional user-specific data for enhanced performance.