DATA REPORT

Revolutionizing the Movie-Watching Experience:

A Data-driven Approach to Personalized Recommendation Systems

Organization: Moringa School

Student pace: DS-full time-Online

Group members:

- Edgar Kiprono
- Sarah Karanja
- Dennis Njogu
- Alice Mumbi
- Moschini Onyango
- Brandon Muraya

Date of publication: 12/04/2024

Business Understanding

In the fast-paced world of entertainment, users often struggle with decision overload when selecting movies to watch. This data report explores the pressing need for personalized recommendation systems to alleviate this challenge, providing users with curated movie suggestions tailored to their tastes and preferences. Through cutting-edge technology and sophisticated algorithms, we aim to transform the movie-watching experience, enhancing user satisfaction and engagement with streaming platforms.

Personalized recommendation systems are essential tools for businesses seeking to enhance user engagement and satisfaction in the competitive streaming landscape. This project at GiggleStream aims to redefine the movie-watching experience by providing tailored recommendations that guide users on a journey of cinematic exploration. By leveraging advanced technology and innovative algorithms, we aim to not only improve user satisfaction but also drive increased viewer retention, platform usage, and revenue generation.

Stakeholder Engagement

For stakeholders invested in GiggleStream's success, this project represents a commitment to excellence and a dedication to providing unparalleled value to users. By harnessing the power of personalized movie recommendations, we aim to set new standards of excellence in the streaming industry, elevating the user experience to unprecedented heights.

Research Questions

- What are the key factors influencing GiggleStream’s users movie preferences and ratings?
- How can collaborative filtering algorithms be effectively applied to generate personalized movie recommendations for GiggleStream’s users?
- What metrics can be used to evaluate the performance and effectiveness of the movie recommendation system in enhancing user satisfaction?
- How can the recommendation algorithm be optimized to improve the accuracy and relevance of movie suggestions over time?

Problem Statement

The objective is to develop a recommendation system that suggests the top 5 movies to GiggleStream’s users based on their ratings of other movies.

Main Objective

Develop a personalized movie recommendation system for GiggleStream’s users to enhance their movie-watching experience and satisfaction with the platform.

Specific Objectives

- Analyze customer ratings of movies to identify patterns and preferences among GiggleStream's users.
- Implement collaborative filtering algorithms to generate personalized movie recommendations based on user ratings and similarities with other users.
- Evaluate the performance of the recommendation system using metrics such as precision, recall, and user satisfaction surveys.
- Optimize the recommendation algorithm to improve the accuracy and relevance of movie recommendations over time.

Data Understanding and Preparation

Data Source

The project's data originates from GroupLens https://grouplens.org/datasets/movielens/latest/, renowned for its MovieLens dataset. While widely utilized in academic circles for various machine learning endeavors, our focus extends beyond academic pursuits. We aim to address real-world business challenges within the movie recommendation domain.

Dataset Overview

For this project, we recommend utilizing the "small" subset of the MovieLens dataset. This subset comprises 100,836 ratings and 3,683 tag applications across 9,742 movies. Generated by 610 users between March 29, 1996, and September 24, 2018, the dataset offers a manageable yet comprehensive scope for deriving meaningful insights and constructing robust recommendation models.

Data Files

The MovieLens dataset consists of four CSV files:

- Links.csv
- Movies.csv
- Ratings.csv
- Tags.csv

Data Preparation

Loading the data

The datasets were loaded in the Jupyter Notebook, where both of them were previewed for a better understanding of their columns and the relationships that exist between them.

The movieLens dataset entails:

- Links.csv: Contains movie identifiers (movieId), IMDb IDs (imdbId), and TMDb IDs (tmdbId).
- Movies.csv: Contains movie identifiers (movieId), titles, and genres.
- Ratings.csv: Contains user identifiers (userId), movie identifiers (movieId), ratings, and timestamps.
- Tags.csv: Contains user identifiers (userId), movie identifiers (movieId), tags, and timestamps.

The shape of each DataFrame is printed to understand the size of the datasets, and the first few rows are displayed to get a glimpse of the data structure. This initial exploration provides insights into the structure and content of the dataset, laying the groundwork for further analysis and modeling.

Data Cleaning

Missing Values

Missing values are identified and displayed for each DataFrame: links.csv, movies.csv, ratings.csv, and tags.csv.

From the output, it appears that the tmdbId column in the links.csv file has 8 missing values. Other files do not contain any missing values.

Merging DataFrames

DataFrames are merged using movieId as the common key.

The merge function is applied to link data from multiple DataFrames: links.csv, movies.csv, and ratings.csv.

After merging, the resulting DataFrame contains 100,823 entries and 8 columns.

Checking for Duplicates

The code snippet checks for duplicated rows in the merged DataFrame.

No duplicates are found in the dataset.

Handling Missing Values

As there are no missing values after merging, no further action is required.

Checking for Maximum and Minimum Ratings

The maximum and minimum ratings in the dataset are printed.

The maximum rating is 5.0, and the minimum rating is 0.5.

Exploratory Data Analysis (EDA)

Distribution of Ratings

Upon examining the ratings, it is apparent that movies rated 4.0 have the highest frequency, totaling 26,816 occurrences, representing about 28.3% of all rated movies. Conversely, movies rated 0.5 have the lowest count, with only 1,367 occurrences, indicating exceedingly low ratings.

Top Ten Movie Titles

The analysis of movie titles reveals that "Forrest Gump (1994)" has garnered the most ratings, with 329 occurrences. Following closely is "Shawshank Redemption, The (1994)" with 317 occurrences.

Top Ten Movie Genres

The genre analysis highlights Comedy as the most prevalent, with 7,194 occurrences, indicating a significant presence of comedic content within the dataset. Other prominent genres include Drama, Comedy|Romance, and Comedy|Drama|Romance.

Top Ten Tags

The tag "In Netflix queue" appears most frequently, occurring 131 times. This suggests that users have tagged certain movies with the intention of watching them or have added them to their Netflix watchlist. Other popular tags include "atmospheric," "thought-provoking," and "superhero," each with varying occurrences.

This comprehensive exploration provides valuable insights into the distribution of ratings, the popularity of movie titles and genres, and the prevalence of user-generated tags. These findings serve as the foundation for further analysis and model development within the recommendation system project.

Modelling Summary

Collaborative Filtering

Collaborative filtering models recommendations based on user-item interactions. Two main types are User-User and Item-Item Collaborative Filtering. User-User recommends items to a user based on similar users' preferences, while Item-Item recommends items based on their similarity to items the user has interacted with.

Item-Item Collaborative Filtering

In this approach, a pivot table is created from the dataset where each row represents a user, each column represents a movie, and the values are the ratings given by users to movies. Nearest neighbors based on cosine similarity are then computed to recommend similar movies to a user.

Hyperparameter Tuning

Hyperparameter tuning is performed to optimize model performance. Grid search and cross-validation techniques are utilized to find the best parameters for the models, such as the number of factors and regularization.

User-Item Collaborative Filtering

This approach employs the Singular Value Decomposition (SVD) algorithm to predict user ratings for movies. By analyzing past ratings, it generates personalized recommendations for users.

Recommendation Model with Surprise

Surprise is used to implement recommendation systems. Various algorithms like KNNBasic and SVD are evaluated for their performance using metrics such as RMSE and MAE. Grid search is employed to find the best parameters for the models.

User Ratings Collection and Incorporation

A function is developed to prompt users to rate movies, and these ratings are incorporated into the recommendation system. The SVD model is then trained with the updated data to generate personalized recommendations.

Overall, these recommendation models leverage collaborative filtering techniques and advanced algorithms to provide personalized movie recommendations to users based on their preferences and past interactions.

Conclusion

In conclusion, our recommendation system, empowered by tuned parameters and rigorous evaluation, represents a significant advancement in the field of personalized recommendation systems. With its superior accuracy, scalability, and potential for further optimization, the system holds great promise for delivering unparalleled user experiences and driving business success in the ever-evolving landscape of recommendation technology.

Recommendations

* To address the identified limitations, we recommend exploring additional data sources and features to enrich the recommendation process and mitigate the cold-start problem.

* Continuous optimization and fine-tuning of algorithmic parameters should be prioritized to further improve recommendation accuracy and relevance.

* Incorporating advanced techniques such as deep learning or reinforcement learning may offer new opportunities for enhancing recommendation quality and addressing sparsity issues within the user-item interaction matrix.

Future Improvement Ideas

* Looking ahead, future iterations of the recommendation system should focus on leveraging emerging technologies and methodologies to deliver more sophisticated and context-aware recommendations.

* Exploring collaborative filtering techniques, hybrid approaches, and context-aware recommendation strategies could unlock new possibilities for enhancing user satisfaction and engagement.

* Continuous monitoring of user feedback and performance metrics will be essential for iteratively refining the recommendation system and ensuring its alignment with evolving user preferences and business objectives.

Deployment

Following extensive trials with our various models, we identified the Recommendation Model with Surprise(with tuned parameters) as the most efficient model as it had the lowest RMSE score. This model showed significant results when we tested it out with our user examples thus saved it using the pickle technique and ran it using the flask environment.

Therefore, we settled on different deployment methods that involve multiple user interfaces. This will enable our recommendation system to be used in both mobile applications and web interfaces depending on the user.