رواد مصر الرقمية

وزارة الاتصـــــــالات
وتكنولوجيا المعلومات

Graduation Project Document

Movie Recommendation System

Prepared By

Mariam Mahmoud

Hanin Hossam

Hadeer Ashraf

Perihane Tarek

Supervised By

Eng. Ahmed Essam Azab

Netflix Movie Recommendation System

This project aims to design and implement a personalized movie recommendation system based on machine learning techniques, leveraging collaborative filtering and data analysis. The system predicts user preferences and recommends movies accordingly. The project incorporates Azure Blob Storage for data handling, SQL Server for warehousing, and Python for machine learning model training and deployment.

Overview

The Netflix recommendation system is designed to provide users with personalized movie suggestions based on their viewing history and ratings. By utilizing collaborative filtering, the system enhances user engagement by offering a tailored experience. It integrates Natural Language Processing (NLP), machine learning, and data science to understand user preferences and predict ratings.

1. Data Management and ETL Process

1.1 Data Warehouse and Schema Design

To manage large-scale movie and user data, the project employs a structured data warehouse in SQL Server. The warehouse is organized using a star schema, which includes fact and dimension tables for efficient querying and analytics.

- **Fact Table**:
  FactRatings: Contains the ratings provided by users for different movies.
- **Dimension Tables**:
  DimUsers: Stores user details like demographic data.
  DimMovies: Holds information about movies, such as titles, genres, and release dates.

1.2 Data Collection and Preprocessing

The dataset used in this project includes user ratings for thousands of movies. The data is extracted from CSV files, transformed into a structured format, and loaded into the SQL database.

- **Data Preprocessing**:
  - Tokenized and standardized movie titles and user inputs.
  - Handled missing values in the dataset.
  - Split the data into training, validation, and test sets.

1.3 ETL Process

The ETL (Extract, Transform, Load) process was handled by SQL scripts. Data was extracted from CSV files, transformed using data cleaning techniques, and then loaded into the appropriate SQL tables.

---

2. Machine Learning Model

2.1 Model Selection

The recommendation system utilizes collaborative filtering with a machine learning algorithm to predict user preferences. The collaborative filtering model relies on matrix factorization, which identifies patterns in user behavior and movie ratings.

2.2 Training the Model

The machine learning model was trained using historical user ratings to predict future ratings. A collaborative filtering approach with matrix factorization was applied to detect similarities between users and movies.

2.3 Model Architecture

The architecture includes the following components:

- **Embedding Layers**: Convert user and movie IDs into vectors that represent their unique features.
- **Interaction Layer**: Learns relationships between user and movie features through matrix factorization.
- **Output Layer**: Predicts movie ratings on a continuous scale, typically from 1 to 5 stars.

2.4 Model Evaluation

The model's performance was evaluated using metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to determine the accuracy of the predicted ratings.

---

3. Personalized Recommendations

The system generates personalized movie recommendations based on the user’s past behavior and preferences. It identifies movies that the user is likely to rate highly by analyzing similarities between users and movies.

3.1 How the Recommendation Works

- **User Input**: The system collects user ratings for a subset of movies.

- **Prediction**: Using collaborative filtering, the system predicts ratings for movies the user has not seen.
- **Recommendations**: Movies with the highest predicted ratings are recommended to the user.

---

4. Deployment

4.1 Model Deployment Using Azure

The trained machine learning model was deployed on Azure using the Streamlit framework. This allows users to interact with the model through a user-friendly web interface, where they can input their user ID to receive movie recommendations.

4.2 User Interface Design

- **Input Field**: Allows users to enter their user ID.
- **Recommendation Button**: Triggers the model to fetch recommendations.
- **Output**: Displays the list of recommended movies.

---

5. Benefits for Users

5.1 Personalized Experience

The system enhances the user experience by providing personalized movie recommendations based on individual preferences, making it easier for users to discover new content tailored to their tastes.

5.2 Improved User Engagement

By offering relevant recommendations, the system encourages users to explore more content and engage more frequently with the platform.

5.3 Continuous Improvement

The model is continuously fine-tuned with new data, improving its accuracy and relevance over time, leading to better user satisfaction.

---

6. Implementation Steps

- **Step 1**: Data collection from CSV files and preprocessing.
- **Step 2**: Building and training the collaborative filtering model.

- **Step 3**: Evaluating the model using appropriate metrics.
- **Step 4**: Deploying the model using Streamlit on Azure.
- **Step 5**: Testing the deployment and making final adjustments.

7. Conclusion

The Netflix Movie Recommendation System combines machine learning and collaborative filtering to offer a personalized and dynamic user experience. By leveraging user data, it provides accurate recommendations, boosting user engagement and satisfaction. The model can be further refined with additional data, making it a powerful tool for content personalization.