Link to the repository: ParvinTiks/Movie-Recommendation-system: Datasience project

Movie recommendation system

Karl-Markus Kaiv
Parvin Tiks
Johann Erich Ints

Task 2. Business understanding (0.5 point)

Background

With the overwhelming number of movies available across multiple online platforms, users often face hard decisions of what to watch next. This project leverages a comprehensive dataset of over 10000 IMDb movies to analyze user preferences and provides personal recommendations. Also it provides a short movie description to help the user to decide. Overall aim is to help users discover new movies and different info about the movies that he/she selects.

Business goals

- Build a user-friendly movie recommendation system that offers tailored suggestions based on individual preferences and watch history, which will ensure that the user gets to see the movies that he does not know yet that he/she would like to see.
- Enhance the movie discovery experience by reducing the time users spend searching for content and ideas.
- Create a scalable and efficient solution that leads to a broad and diverse user base, accommodating both new and existing users.

Business Success Criteria

- Increase user engagement by demonstrating repeated usage and a higher selection rate of recommended movies.
- Ensure the system delivers movie recommendations within seconds, maintaining a seamless user experience.
- To achieve high user satisfaction.

Inventory of Resources

- **Dataset**: A rich IMDb dataset containing 13 fields, including metadata such as movie names, genres, ratings, and cast.

- **Technology**: Python for development, Pandas for data handling, and basic algorithms like random selection and content-based filtering for recommendations.

Requirements, Assumptions, and Constraints

- **Requirements**: A recommendation system, an easy-to-use interface, and a mechanism to store and retrieve user history persistently.
- **Assumptions**: Users will engage with the system, providing choices and interacting with recommendations.

Risks and Contingencies

- **Risk**: Insufficient personalization due to limited data or user engagement.
  Mitigation: Employ hybrid recommendation strategies combining content-based filtering with random selections to improve variety and relevance.
- **Mitigation**: Conduct thorough preprocessing, including cleaning and filling in missing data.

Terminology

- **Recommendation System**: Software designed to provide personalized content suggestions.
- **Content-Based Filtering**: A method that uses movie metadata like genres and cast to recommend similar movies.
- **Cold Start Problem**: The challenge of recommending movies to new users with no prior history or insufficient data.

Costs and Benefits

- **Costs**: Time spent on developing, testing, and deploying the system, as well as computational costs for algorithm optimization and user data storage.
- **Benefits**: Improved user experience through personalized movie recommendations, increased engagement.

Data-Mining Goals

- Develop a system capable of identifying user preferences through historical data and metadata analysis.

- Generate accurate and relevant personalized recommendations by analyzing movie similarities in attributes like genre, crew, and language.
- Address the cold start problem by supplementing recommendations for new users with random selections.

Data-Mining Success Criteria

- Effective handling of new users, with random selections providing a satisfactory starting point.
- Fast and efficient processing of large datasets to ensure a smooth user experience, with recommendations delivered in under 3 seconds.

Task 3. Data understanding (1 points)

1. Gathering Data
Outline Data Requirements
Purpose of the data: Predict movies that customers might like by analysing their viewing patterns based on previous behaviors.
Required attributes: Watched movies, actors, directors, ratings, genres, and other relevant metadata.
Format and structure: The data should be structured in a table format to facilitate easier processing and analysis.
Verify Data Availability
Sources: The data is sourced from Kaggle and includes more than 10,000 IMDb movies with relevant metadata for detailed analysis.
Access permissions: This dataset is publicly available, making it accessible for use in research and development.
Formats: The dataset is available in CSV format, which is commonly used for structured data.

2. Describing Data
Metadata inventory: The dataset contains 13 fields with various data types to ensure detailed movie descriptions. Each field is structured to represent essential movie features for predictive analysis.
Field names and data types:

- ID (int), date (date), names (string), score (float), genre (string), overview (string), crew (list), orig_title (string), orig_lang (string), budget (float), revenue (float), country (string).
  Volume: The dataset contains 10,178 rows and 13 columns. The volume ensures sufficient data for predictive analysis across a wide variety of movies and attributes.

3. Exploring Data

Statistics for numeric variables:

- Score
  - Mean: 63.5
  - Median: 65.0

    - Min: 0.0
    - Max: 100
- Budget
    - Mean: 64,882,378.89 USD
    - Median: 50,000,000.0 USD
    - Min: 1 USD
    - Max: 460,000,000.0 USD
- Revenue
    - Mean: 253,140,093.41 USD
    - Median: 152,934,876.5 USD
    - Min: 0 USD
    - Max: 2,923,706,026.0 USD

Occurrences of categories:

- Genre (Top 5 categories with the most occurrences):
    - Drama: 556
    - Comedy: 373
    - Drama, Romance: 268
    - Horror: 260
    - Horror, Thriller: 202

Relationships:

- There are clear relationships between various attributes, such as:
    - Budget and score: Higher budgets tend to correlate with higher scores, as larger productions often result in higher ratings.
    - Score and revenue: Generally, movies with better ratings have a higher chance of generating higher revenues.
    - Budget and revenue: A higher budget typically leads to higher revenue, especially in the case of major studio productions.

Anomalies:

- Some movies have extreme differences in score, budget, or revenue compared to the average values for their genre, indicating potential outliers or extraordinary cases.
Trends:
- The average score for movies has remained fairly stable across the years.
- Certain genres, such as animation or historical films, tend to have higher average scores compared to others, showing a pattern in viewer preferences.

4. Verifying Data Quality

Completeness:

- There are some missing values in certain fields such as genre (85 entries) and crew (56 entries), which represents a minimal portion of the dataset.
- The percentage of missing values is approximately 0.11%, which is considered very low and should not affect overall analysis significantly.

Accuracy:

- Given that this dataset is sourced from Kaggle, a trustworthy platform for data science, we can reasonably trust that the data is accurate and reliable for predictive modeling purposes.
  Consistency:
- A thorough check of the dataset for inconsistencies was performed. Aside from the missing values, no other major issues with data consistency were found.

Timeliness:

- The dataset contains movies collected over an extensive period from May 15, 1903, to December 31, 2023, covering over a century of film production. This long timespan provides a comprehensive view of the evolving movie industry.

Task 4. Planning your project (0.25 points)

(All the hours are roughly estimated)

List of tasks:

1. Think about project ideas and what would be good enough to make into a project. **4 hours (all members)**
2. Write a code that interacts with the user and his/her choices and saves the selected movies via id and username-id to a file. All the communication will be happening in the console, where at the first login the code provides the user with 20 random movies and the user has to select 5 different movies from the selection to determine the initial data, from which the trained model will start to recommend new movies to the user. **5 hours (Karl-Markus Kaiv)**
3. Make the dataset usable. **4 hours (Parvin Tiks)**
4. Train the model to search for the selected movies by actors and genres - **all members 10 hours**
5. Use the trained model to help users choose between different movies. **5 hours (Johann Erich Ints)**
6. Save data that has been entered or chosen by the user to use for the future recommendations. **2 hours (Karl-Markus Kaiv)**
7. Make charts and overall data analysis to see which movies are more popular and so they would be better to suggest to the new users at the beginning. **5 hours (Parvin Tiks, Johann Erich Ints)**
8. Overall testing. **10 hours (all members)**

Methods and tools.

1. **Discord channel** - overall communication.
2. **Visual Studio Code** - IDE.
3. **Python** - coding.
4. **Google** - inspiration and answers to problems.
5. **Kaggle** - dataset.
6. **Github** - Project workflow.