C964: Computer Science Capstone

Task 2 parts A, B, C and D

Western Governors University

Katherine Caudill

January 15, 2024

Part A: Letter of Transmittal

[Today’s date]

[Recipient’s name]

[Company name]

[Address]

To whom this may concern,

Subject: Proposal for Implementing a Machine Learning-Based Movie Recommendation System

I am writing to submit a detailed project proposal for the implementation of a machine learning-based movie recommendation system, as approved in Task 1. This document outlines the key aspects of the project, including its objectives, methodology, and the potential benefits it offers to our customers and the organization.

Enclosed, you will find the comprehensive proposal that addresses the identified need, the planned approach, and the expected impact. I believe that this initiative will significantly enhance our user experience and decision-making processes, aligning with our strategic goals.

I appreciate your consideration of this proposal and am available for any further discussions or clarifications you might require.

Sincerely,

Katherine Caudill

Part B: Project Proposal Plan

Project Overview

This executive summary presents an overview of the development and implementation of a machine learning-based movie recommendation system, aimed at addressing the need for enhanced decision support in content curation and customer engagement in the entertainment industry.

Problem/Opportunity Addressed

The project targets the critical issue of decision fatigue among users navigating extensive movie catalogs. By leveraging machine learning for personalized recommendations, we aim to improve user engagement and satisfaction, thereby addressing a significant opportunity in enhancing the customer experience.

Customer Focus

Our primary customers are users of online streaming platforms seeking a more tailored and efficient movie selection process. This system will fulfill their needs by providing highly relevant and personalized recommendations, enhancing their overall experience.

Existing Gaps

The current data products in the market often rely on basic algorithms that do not adequately consider individual user preferences, leading to less personalized and often irrelevant suggestions. This project aims to bridge this gap by implementing advanced machine learning techniques.

Data Requirements

The system will utilize:

- User demographic data and viewing habits.
- Detailed movie metadata, including genres, ratings, and reviews.
- User feedback for continuous improvement of the recommendation algorithm.

Methodology

The project will follow a structured methodology:

1. **Data Collection and Preprocessing**: Gathering and cleaning relevant data.
2. **Algorithm Development**: Implementing a collaborative filtering algorithm.
3. **Integration and Testing**: Integrating the algorithm into the existing platform and conducting thorough testing.
4. **Feedback Incorporation**: Iteratively improving the system based on user feedback.

Deliverables

Key deliverables include:

- A fully functional recommendation algorithm.
- Integration of the algorithm into the existing platform.
- Comprehensive documentation and a user guide.

Implementation Plan

The implementation will involve:

- Deploying the system in a controlled environment for initial testing.
- Gradual rollout to the user base, monitoring performance and user feedback.
- Full-scale implementation upon successful validation.

Validation and Verification

Methods for validation and verification:

- Rigorous testing phases, including unit testing, system testing, and user acceptance testing.
- Continuous monitoring of user engagement metrics post-implementation.

Programming Environments and Resources

The development will require:

- Programming environments: Python, relevant machine learning libraries.
- Associated costs: Development tools, cloud services for data storage and processing.
- Human resources: Data scientists, developers, QA testers.

Timeline

The projected timeline is as follows:

- **Data Collection and Preprocessing**: Duration - 2 months.
- **Algorithm Development**: Duration - 3 months.
- **Testing and Integration**: Duration - 2 months.
- **Feedback and Iteration**: Duration - ongoing post-implementation.

Each phase will include specific milestones with assigned resources and dependencies clearly outlined. The timeline accounts for buffer periods for unforeseen delays and challenges.

Part C: Application

Part C is your submitted application. This part of the document can be left blank or used to include a list of any submitted files or links.

Please see user guide for dependencies and downloads.

Part D: Post-implementation Report

Solution Summary

- Problem: Users often struggle to find movies that align with their preferences due to the vast number of choices available in online streaming platforms. This leads to decision fatigue and a less satisfying user experience.
- Solution: We developed a machine learning-based movie recommendation system that personalizes suggestions based on user preferences and viewing history. This system aids in discovery and enhances user satisfaction.

Data Summary

- Raw Data Source: The raw data, including movie metadata and user ratings, was sourced from the Full MovieLens Dataset and IMDb.
- Data Management: The data was processed through stages of cleaning, feature extraction, and transformation. Initially, the dataset was filtered to include relevant features like movie titles, genres, keywords, cast, and crew. The data was then transformed using TF-IDF vectorization for use in the recommendation model.

Machine Learning

- Method Identification: The project employed a content-based filtering method using cosine similarity.
- Development Process: We developed the model by creating a TF-IDF matrix from the combined features of movies and then computed the cosine similarity between movies. The model identifies movies similar to a user’s previously liked movies.

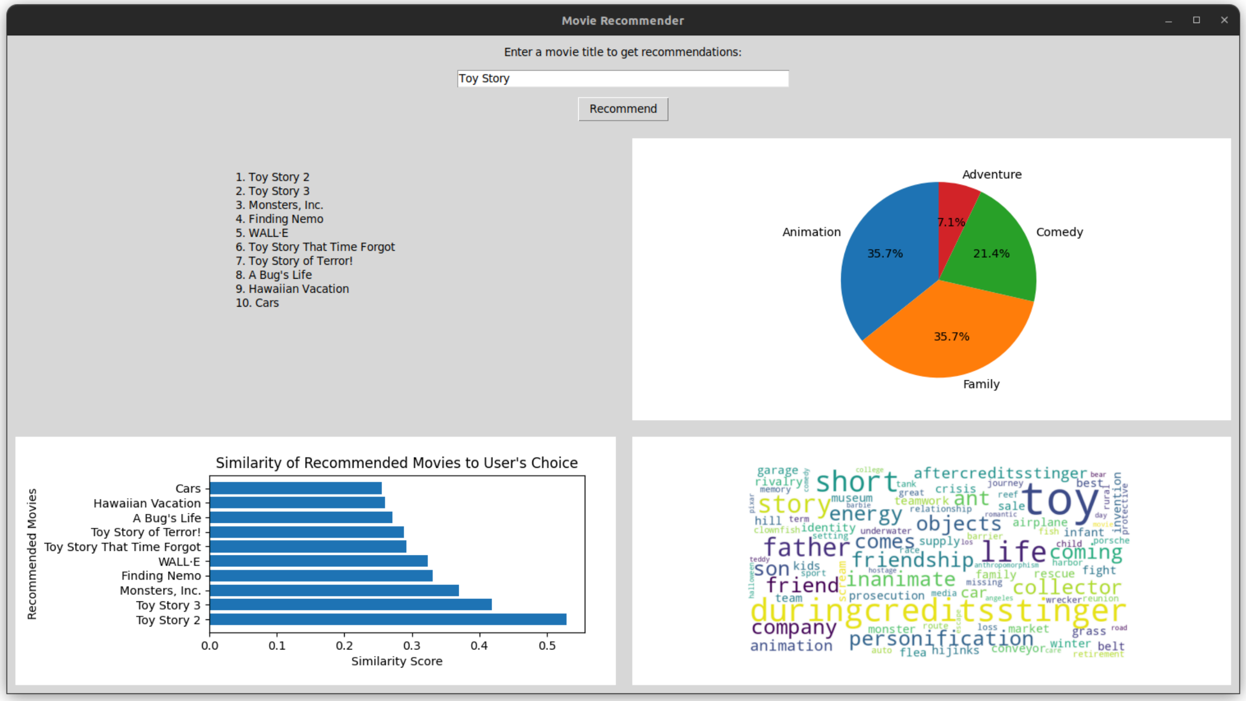
- Justification: Content-based filtering was chosen for its effectiveness in providing personalized recommendations based on individual user preferences.

Validation

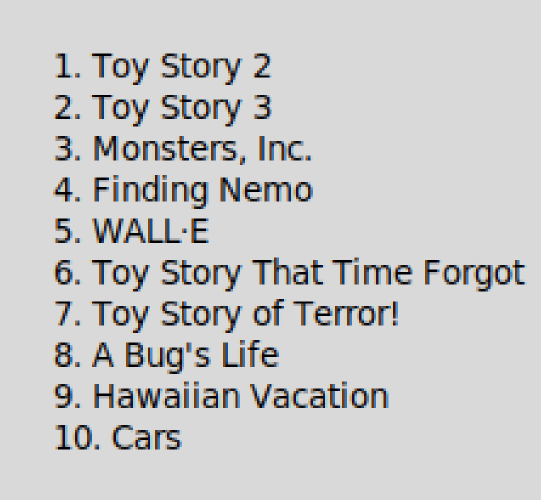
- Method: The validation of the recommendation system was primarily qualitative, relying on user feedback.

- Results/Plan: Initial user feedback was positive, indicating the system's efficacy. Future plans include more quantitative measures, such as tracking engagement metrics to further validate the system's performance.

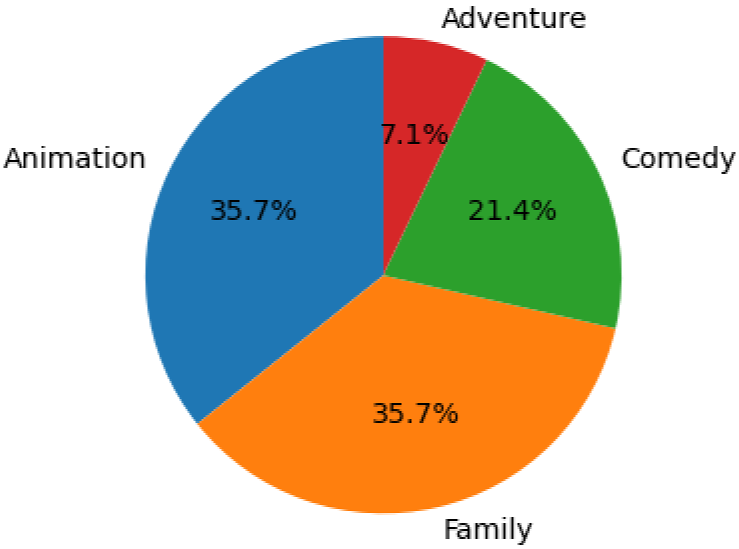
Visualizations



1. Toy Story 2
2. Toy Story 3
3. Monsters, Inc.
4. Finding Nemo
5. WALL·E
6. Toy Story That Time Forgot
7. Toy Story of Terror!
8. A Bug's Life
9. Hawaiian Vacation
10. Cars

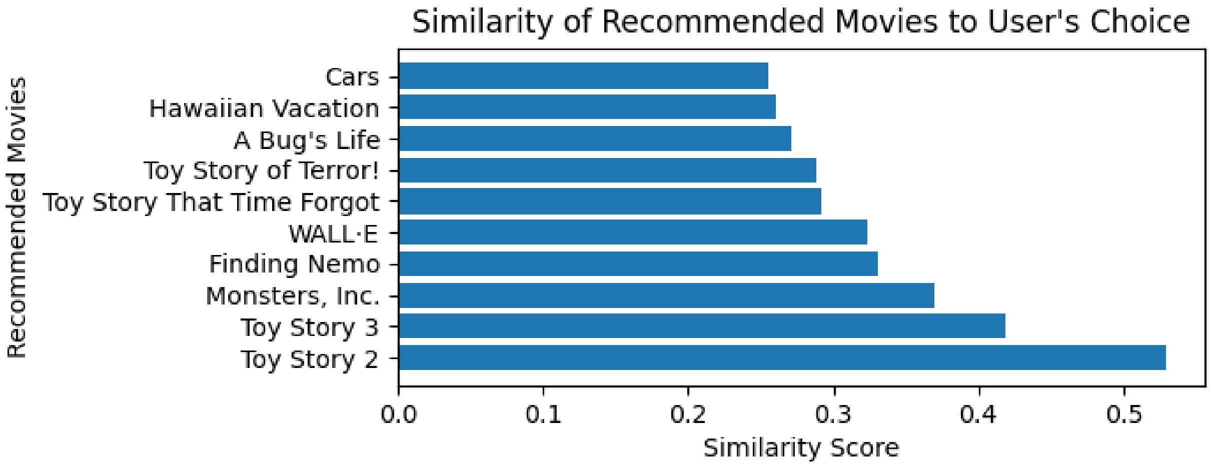


1. Toy Story 2
2. Toy Story 3
3. Monsters, Inc.
4. Finding Nemo
5. WALL·E
6. Toy Story That Time Forgot
7. Toy Story of Terror!
8. A Bug's Life
9. Hawaiian Vacation
10. Cars



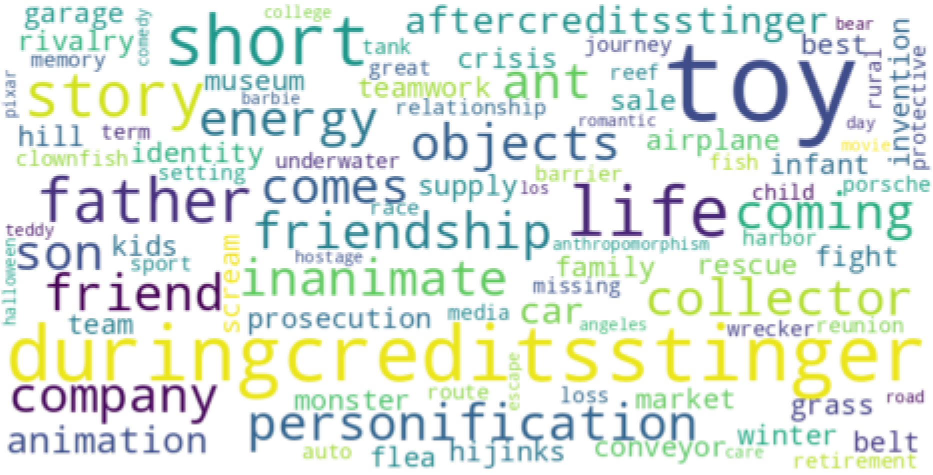
Genre Distribution of Recommended Movies

This pie chart breaks down the genres of the recommended movies. Each slice of the pie represents a different genre, with the size of the slice indicating the frequency of that genre among the recommended movies. The chart provides an at-a-glance view of the genre diversity within the recommendations, helping users understand the predominant genres in their movie suggestions.



Similarity of Recommended Movies to User's Choice

This chart displays a horizontal bar graph representing the similarity scores of the top recommended movies compared to the user's chosen movie. Each bar corresponds to a recommended movie, with the length of the bar indicating the degree of similarity. The longer the bar, the more similar the movie is to the user's choice. The y-axis lists the titles of the recommended movies, and the x-axis shows the similarity scores. This visualization helps users quickly grasp which movies are most closely related to their preference.



Keyword Word Cloud for Recommended Movies

The word cloud visualization illustrates the most prominent keywords associated with the recommended movies. Common keywords are shown in varying font sizes; the larger the word, the more frequently it occurs in the metadata of the recommended movies. This visualization provides insights into thematic elements and topics prevalent in the recommended movie set, offering a quick way to gauge the thematic content.

User Guide

This Movie Recommender Application provides personalized movie recommendations based on a user's input. The system utilizes various data visualization techniques to enhance user experience, offering insights into the similarity, genre distribution, and thematic elements of the recommended movies.

Installation Steps

Clone the Repository or Download the Source Code

- Clone the repository if available, or download the source code to your local machine.

Set Up a Python Environment (Optional but Recommended)

- Create a new Python virtual environment:

```
python3 -m venv movie_recommender_env
```

Activate the environment:

- On Windows:

```
.\movie_recommender_env\Scripts\activate
```

- On macOS and Linux:

```
source movie_recommender_env/bin/activate
```

Install Required Libraries

- Navigate to the directory containing the application's code.
- Install the required libraries using pip:

```
pip install pandas numpy matplotlib sklearn wordcloud networkx
```

1. **Data Files**
   - Ensure you have the necessary data files (movies_metadata.csv, keywords.csv, and credits.csv) in an archive folder in the same directory as your application.
   - Download a copy of the data set from The Movies Dataset on Kaggle (https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data)

Usage

1. **Starting the Application**
   - Run the application using Python:

```
python3 movie.py
```

1. **Using the Application**
   - **Enter a Movie Title**: In the text entry box at the top, type the title of a movie you like.
   - **Get Recommendations**: Click the 'Recommend' button to generate movie recommendations based on your input.
   - **View Results**: The application displays the recommended movies and visualizations:

    - **Recommendations List**: On the top-left, a list of recommended movies is displayed.
    - **Genre Distribution Pie Chart**: On the top-right, a pie chart shows the genre distribution of the recommended movies.
    - **Similarity Bar Chart**: On the bottom-left, a bar chart illustrates the similarity of recommended movies to the user's choice.
    - **Keyword Word Cloud**: On the bottom-right, a word cloud highlights the frequent keywords associated with the recommended movies.

Understanding the Visualizations

- **Similarity Bar Chart**: Understand how similar each recommended movie is to your input.
- **Genre Distribution Pie Chart**: Get a quick overview of the genre makeup of the recommendations.
- **Keyword Word Cloud**: See the common themes and elements present in the recommended movies.

Exiting the Application

- Close the application window or terminate the Python process to exit the application.

Reference Page

Banik, Rounak. “The Movies Dataset.” *Kaggle*, 10 Nov. 2017, www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data.

“Wordcloud Package in Python - Javatpoint.” *Www.Javatpoint.Com*, www.javatpoint.com/wordcloud-package-in-python. Accessed 15 Dec. 2023.