MOVIE RECOMMENDATION SYSTEM

REPORT

Submitted by

RAGUL M GA (19IT073)

SREESH N (19IT101)

SIDDHARTH S (19IT093)

DIXON J ANTONY A (19IT025)

in partial fulfillment for the completion of course

18ES690 Engineering Design Project

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI–15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

JUNE 2022

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI-15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report **MOVIE RECOMMENDATION SYSTEM** is the bonafide work of **RAGUL M GA (19IT073), SREESH N (19IT101), SIDDHARTH S (19IT093)** and **DIXON J ANTONY A (19IT025)** who carried out the project work under my supervision during the Academic Year 2021 -2022.

COURSE FACULTY SIGN
Ms. PARKAVI R
ASSISTANT PROFESSOR
DEPARTMENT OF IT
TCE, MADURAI-15

COURSE FACULTY SIGN
Dr. SUGANYA R
ASSOCIATE PROFESSOR
DEPARTMENT OF IT
TCE, MADURAI-15

TABLE OF CONTENTS

ABSTRACT

Recommendation System is a major area which is very popular and useful for people to make proper automated decisions. It is a method that helps users to find out the information which is beneficial to him/her from the variety of data available. When it comes to Movie Recommendation System, recommendation is done based on similarity between users (Collaborative Filtering) or by considering particular user’s activity (Content Based Filtering) which he wants to engage with. To overcome the limitations of collaborative and content-based filtering generally, a combination of collaborative and content-based filtering is used so that a better recommendation system can be developed. Also, various similarity measures are used to find out similarity between users for recommendation.

In this report, we have surveyed the state-of-the-art methods of Content Based Filtering, Collaborative Filtering, Hybrid Approach and Deep Learning methods and chosen Collaborative Filtering based method for movie recommendation. We have also reviewed different similarity measures. Various companies like Facebook which recommends friends, LinkedIn which recommends jobs, Pandora recommends music, Netflix recommends movies, Amazon recommends products etc. use recommendation systems to increase their profit and also benefit their customers. We have used Streamlit, an open-source application framework for Machine Learning and Data Science to create our web application, which is the final outcome of this project.

PROBLEM DESCRIPTION

Problem Statement:

- The rapid growth of data collection has led to a new era of information. Data is being used to create more efficient systems and this is where Recommendation Systems come into play.

- Recommendation Systems are a type of information filtering systems as they improve the quality of search results and provide items that are more relevant to the search item or are related to the search history of the user.

- Moreover, companies like YouTube, Netflix, Amazon and Spotify depend highly on the effectiveness of their recommendation engines for their business and success.

- In layman’s terms, we can say that a Recommendation System is a tool designed to predict/filter the items as per the user’s behavior.

Objective:

- To implement a recommendation engine that enables users to receive recommendations based on his/her input to the system.

Subject Domain:

- Machine Learning – To develop a model to recommend movies to the user.
- Python Web Development – To develop a UI for the user to interact with the system.

Societal Relevance:

- ❖ The recommendation system uses data analysis techniques to identify items which match the user's taste and preferences. There are different types of recommendation engines for different applications.

- ❖ These include the product recommendation engine, the content recommendation engine, the e-commerce product recommendation and the movie recommendation engine. Recommendation Engine Market size is projected to reach $12.03 billion by 2025 from $1.14 billion in 2018, with a CAGR of 32.39% during 2020-2025.

- ❖ Growing emphasis on improving consumer experience and increasing trend of digitalization are some of the main driving factors for the demand of recommendation engines. In addition to these factors, the increase in demand for analyzing large volumes of data and increasing use of deep learning technology in recommendation engine is anticipated to provide the market with growth opportunities

BACKGROUND

Literature Review:

S.NO	AUTHOR	TITLE	NAME OF THE JOURNAL/ CONFERENCE	ISSUE YEAR	METHOD	OBSERVATIONS
1	Aminu Da'u Naomie Salim	**Recommendation system based on deep learning methods: a systematic review and new directions.**	Springer	2019	**NLP (Natural Language Processing)** **NN(Neural Networks)**	In this paper, we analyze the various methods of achieving a recommendation system based on deep learning methods. The findings of this paper helped us with getting exposure on different methods used for recommendation systems.
2	SRS Reddy Sravani Nalluri Subramanyam Kunisetti S.Ashok B.Venkatesh	**Content-Based Movie Recommendation System Using Genre Correlation**	Springer	2018	**Content-based filtering** **Genre correlation**	In this paper, the recommendation system has been built on the type of genres that the user might prefer to watch. This paper provided us with exposure in content based filtering using genre correlation and dataset collection (MovieLens Dataset).

3	Ms. Sushmita Roy Prof. (Mr.) Mahendra Sharma Dr.Santosh Kumar Singh.	**Movie Recommendation System Using Semi-Supervised Learning.**	IEEE Xplore	2019	**Supervised recommendation systems**	This paper provided us with knowledge on the real time applications (**Amazon, Netflix, Alibaba etc…**) of our project's core functionality, which is building a recommendation model.

Scope of the Problem:

- Recommendation System is an information tool which helps users to find out the items which they want from the large number of items available.

- Main goal of the recommendation system is to forecast the rating which a specific user gives to an item.

- It helps the user to find the best solution from the available list of items.

Constraints/Limitations:

- People are fickle-minded (i.e) their taste changes from time to time and as this algorithm is based on user similarity it may pick up initial similarity patterns between 2 users who after a while may have completely different preferences.

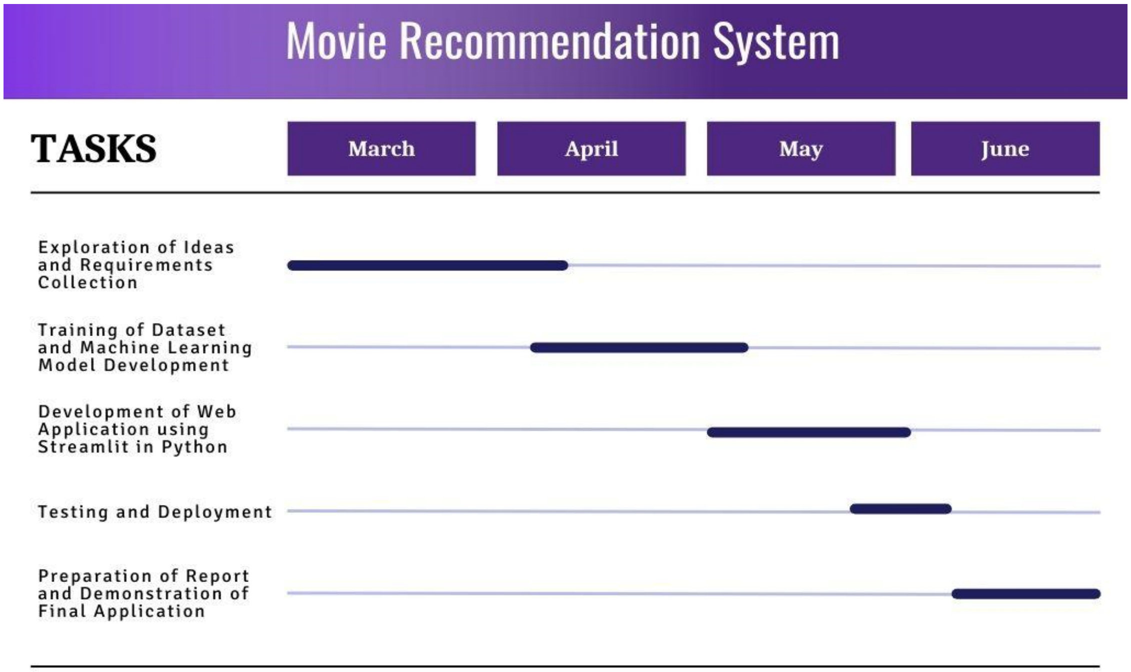
- There are many more users than items therefore it becomes very difficult to maintain each user's preferences in an efficient way.

- This algorithm is very susceptible to shilling attacks where fake users profiles consisting of biased preference patterns are used to manipulate key decisions.

- In the case of a new movie, there are not enough user ratings to match.

PROJECT SPECIFICATION / DESIGN REQUIREMENTS

Schedule – Timeline chart:



Budget and Risk Factors:

The budget is negligible as the intended product is a web application built upon data that is available on the internet. Risk factors include web security issues and shilling attacks, which is the influx of fake profiles with biased ratings, which significantly reduces the accuracy of recommendation systems.

Team Members - Roles and Responsibilities:

- Ragul M GA → Development of ML Model
- Dixon J Antony A → UI Design
- Sreesh N → Front-end Development
- Siddharth S → Integration of Web Application

PROPOSED METHODOLOGY

In this implementation, when the user searches for a movie we will recommend the top 10 similar movies using our movie recommendation system. We will be using an item-based collaborative filtering algorithm for our purpose. The dataset used is the MovieLens Small Database.

First, the necessary libraries like pandas, numpy, scikit, matplotlib and seaborn are imported and the datasets movies.csv and ratings.csv are uploaded to the workspace.

Movie dataset has

- movieId – once the recommendation is done, we get a list of all similar movieId and get the title for each movie from this dataset.
- genres – which is not required for this filtering approach.

Ratings dataset has

- userId – unique for each user.
- movieId – using this feature, we take the title of the movie from the movies dataset.
- rating – Ratings given by each user to all the movies using this we are going to predict the top 10 similar movies.

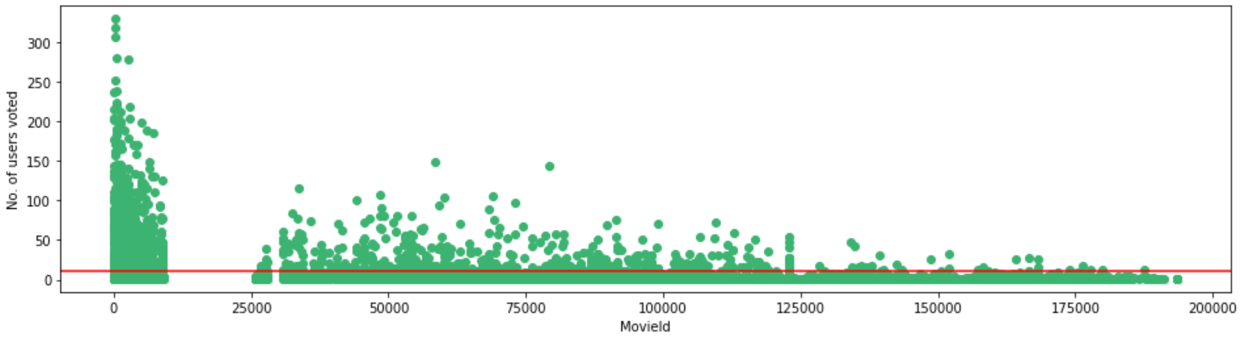
To make things easier to understand and work with, a new dataframe is made where each column would represent each unique userId and each row represents each unique movieId. The NaN values are replaced with 0.0 to make things understandable for the algorithm. This results in the dataset having many number of zeroes or values that will not significantly impact a calculation. This is called sparsity and it must be solved in order to make an efficient model.

In the real-world, ratings are very sparse and data points are mostly collected from very popular movies and highly engaged users. We wouldn’t want movies that were rated by a small number of users because it’s not credible enough. Similarly, users who have rated only a handful of movies should also not be taken into account.

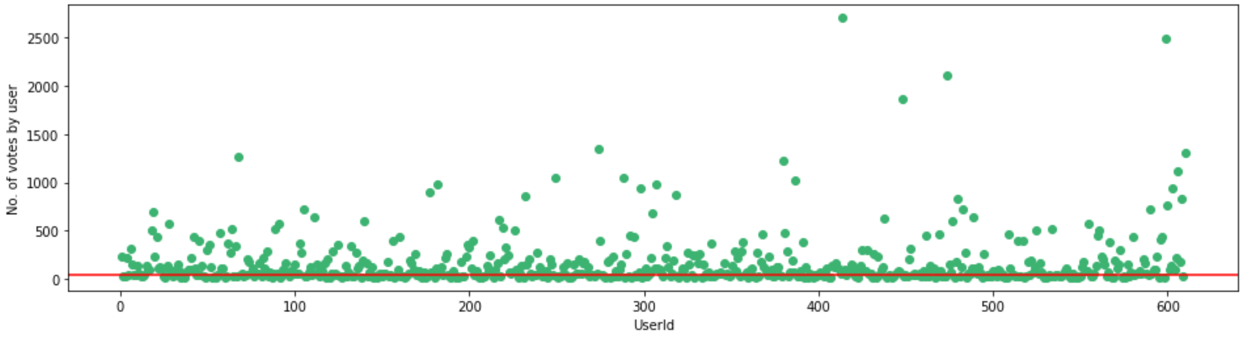
So with all that taken into account and some trial and error experimentations, we will reduce the noise by adding some filters for the final dataset.

- To qualify a movie, a minimum of 10 users should have voted a movie.
- To qualify a user, a minimum of 50 movies should have voted by the user.

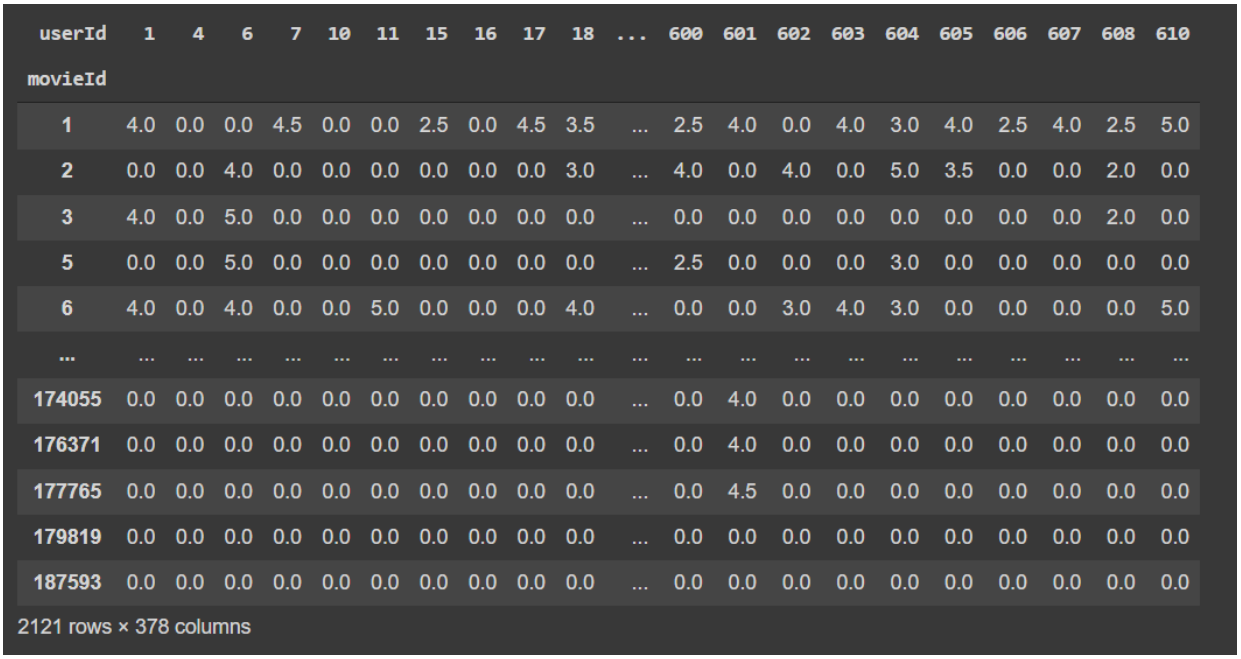
Visualizing the movies with number of users exceeding our threshold of 10, we get:



Visualizing the users with number of votes exceeding our threshold of 50, we get:



After making the necessary modifications as per the threshold set, we get our final dataset which has 2121 x 378 columns (i.e.) 378 users and 2121 movies. Our final_dataset has dimensions of 2121 * 378 where most of the values are sparse. We are using only a small dataset but for the original large dataset of MovieLens which has more than 100000 features, our system may run out of computational resources when that is feed to the model. To reduce the sparsity, the csr_matrix function from the scipy library is used.



userId	1	4	6	7	10	11	15	16	17	18	...	600	601	602	603	604	605	606	607	608	610
movieId																					
1	4.0	0.0	0.0	4.5	0.0	0.0	2.5	0.0	4.5	3.5	...	2.5	4.0	0.0	4.0	3.0	4.0	2.5	4.0	2.5	5.0
2	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	...	4.0	0.0	4.0	0.0	5.0	3.5	0.0	0.0	2.0	0.0
3	4.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0
5	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	2.5	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
6	4.0	0.0	4.0	0.0	0.0	5.0	0.0	0.0	0.0	4.0	...	0.0	0.0	3.0	4.0	3.0	0.0	0.0	0.0	0.0	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
174055	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
176371	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
177765	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	4.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
179819	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
187593	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

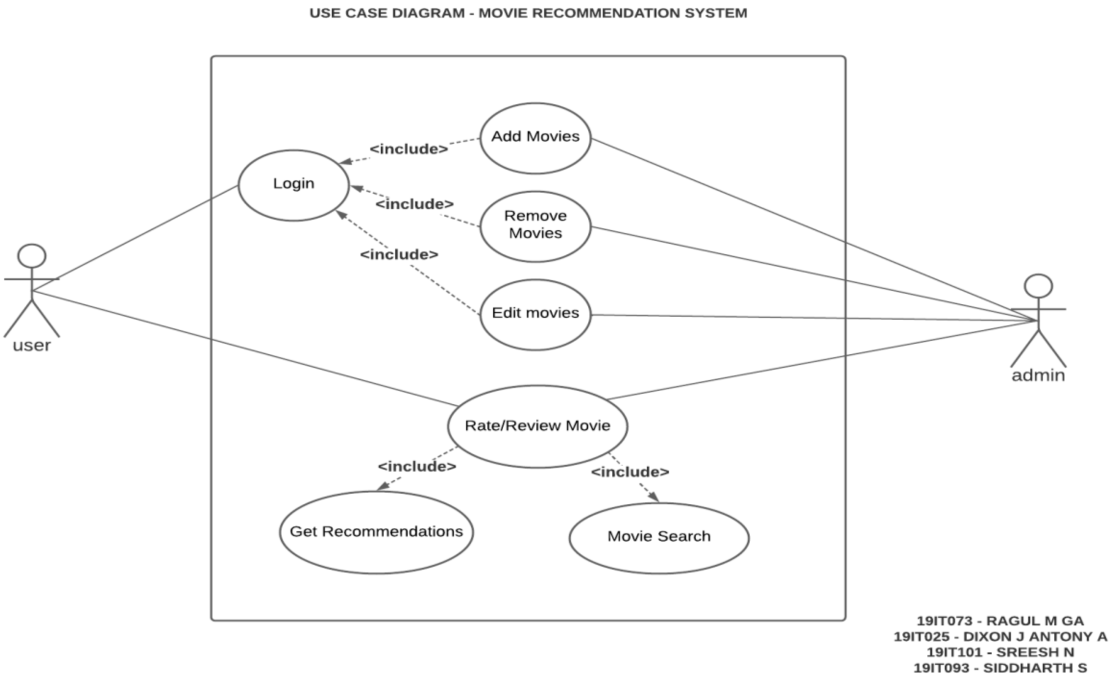
2121 rows × 378 columns

Once the csr_matrix function is used, the dataset is now ready to be fed to the Movie Recommendation System model. We used KNN algorithm to compute similarity with cosine distance metric which is very fast and more preferable than pearson co-efficient.

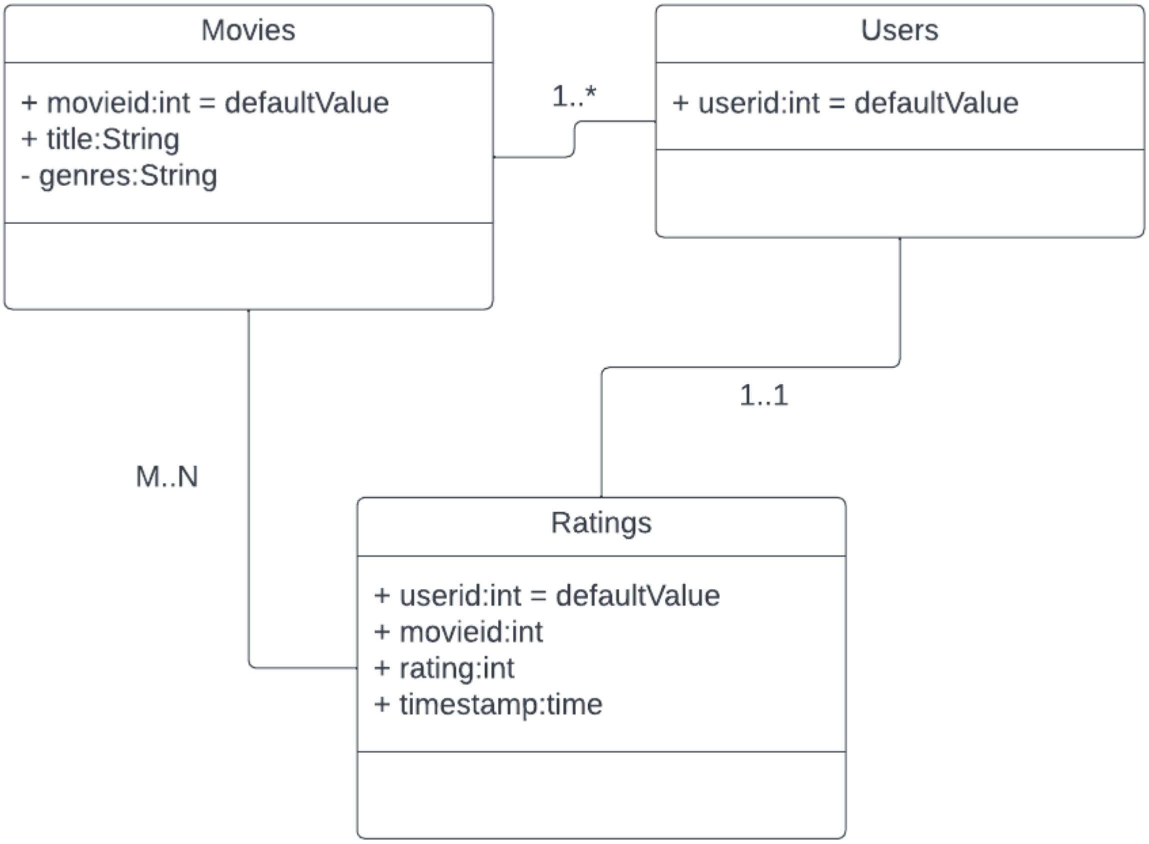
The working principle is very simple. We first check if the movie name input is in the database and if it is we use our recommendation system to find similar movies and sort them based on their similarity distance and output only the top 10 movies with their distances from the input movie. The model is developed and is developed into a web application using Streamlit, a python web framework which is used to develop simple data-driven web applications.

DESIGN

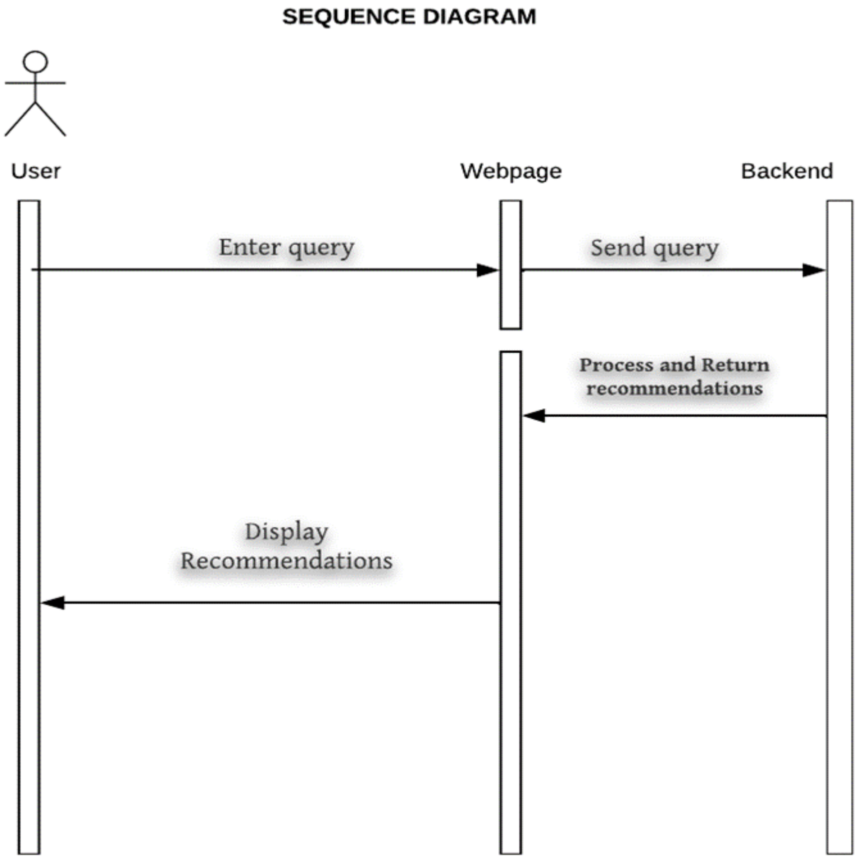
Use Case Diagram:



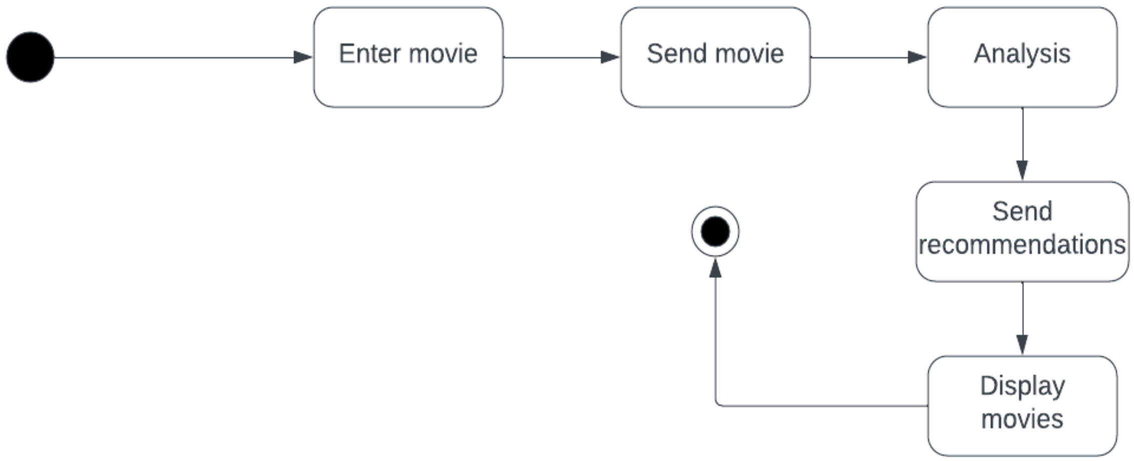
Class Diagram:



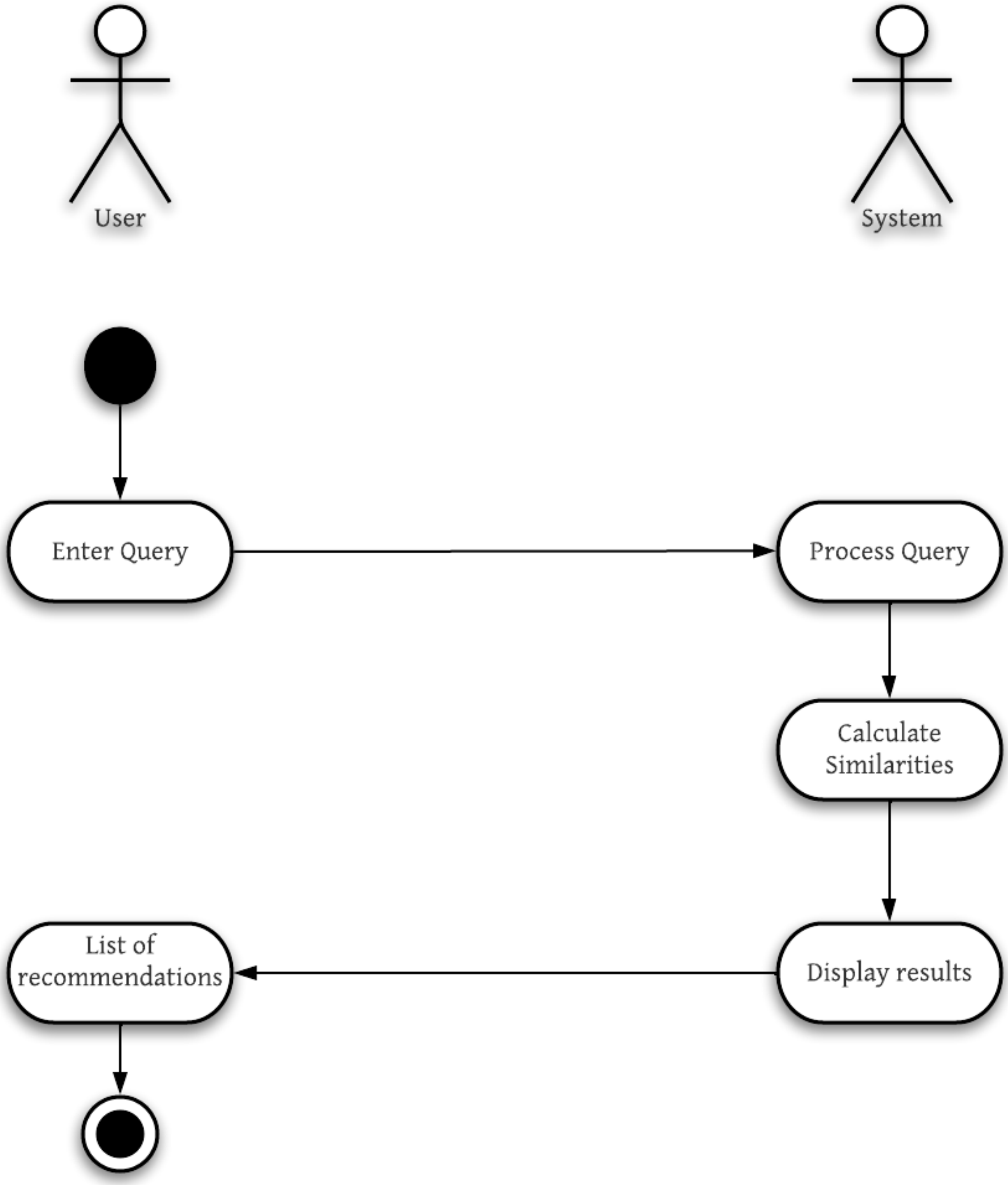
Sequence Diagram:



State Transition Diagram:



Activity Diagram:



BUSINESS ASPECTS

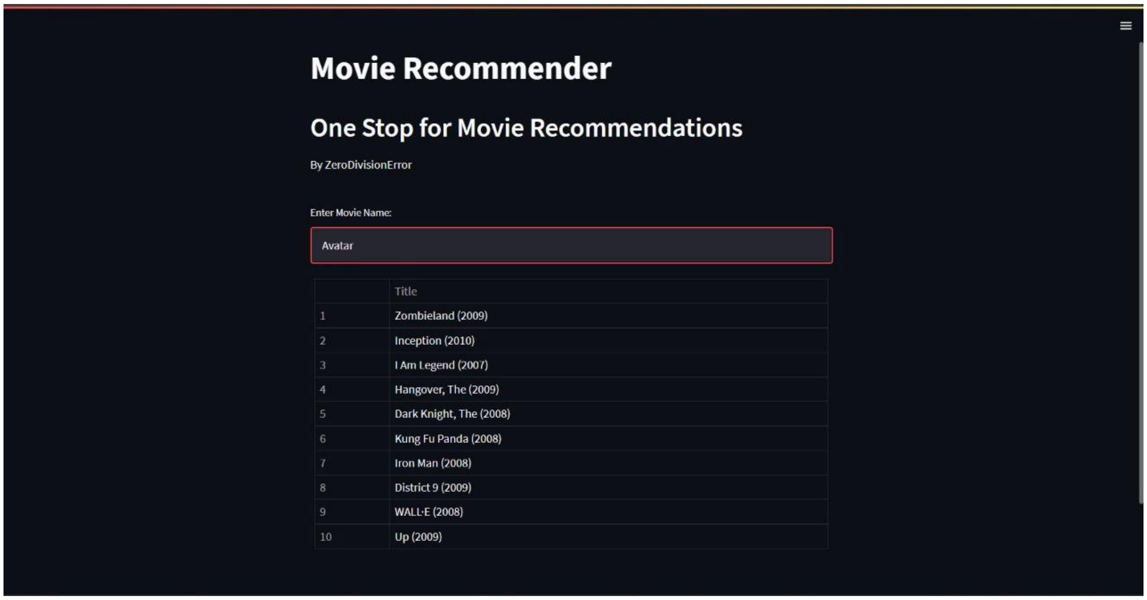
The internet is a go-to space for businesses seeking to access the global marketplace. The online market is rapidly growing: Statista estimates that e-commerce revenue is set to grow to 4.88 trillion USD by 2021, so to expand commercial activity and make a digital business successful, it is vital to not neglect a critical part of any internet business: recommender systems.

When it comes to sites - such as Netflix and Disney+ - that are based on monthly subscriptions, there has to be a reason why the customer should pay for another month. Recommender systems are most commonly used to ensure the user always discovers something new to watch. By ensuring the user receives regular recommendations that suit their taste, they are more likely to renew their subscription for another month. At the end of the day, the user experience is what keeps them coming back to you.

Often overlooked as a way to analyze a market, a recommender can be used to discover user preferences and see what people are most interested in. By utilizing user ratings and the number of users watching a show, businesses can ensure they offer similar products. Much like the previous section, if you offer your user something they may be interested in, they are much more likely to regularly use your business. If you believe your business would benefit from such a system then might be able to help you.

If you have ever been to a website and been recommended a product that would suit your needs, then you understand how powerful this tool can be. As a business, you can use recommender systems to ensure your user is always provided with the best possible experience. Even in industries where there isn't a large amount of competition, a good user experience will still lead to more sales and more profit. By using the right software, such as buyer intent models or purchasing models, you can provide your users with the right recommendations and ensure they keep coming back.

FINAL IMPLEMENTATION



Movie Recommender

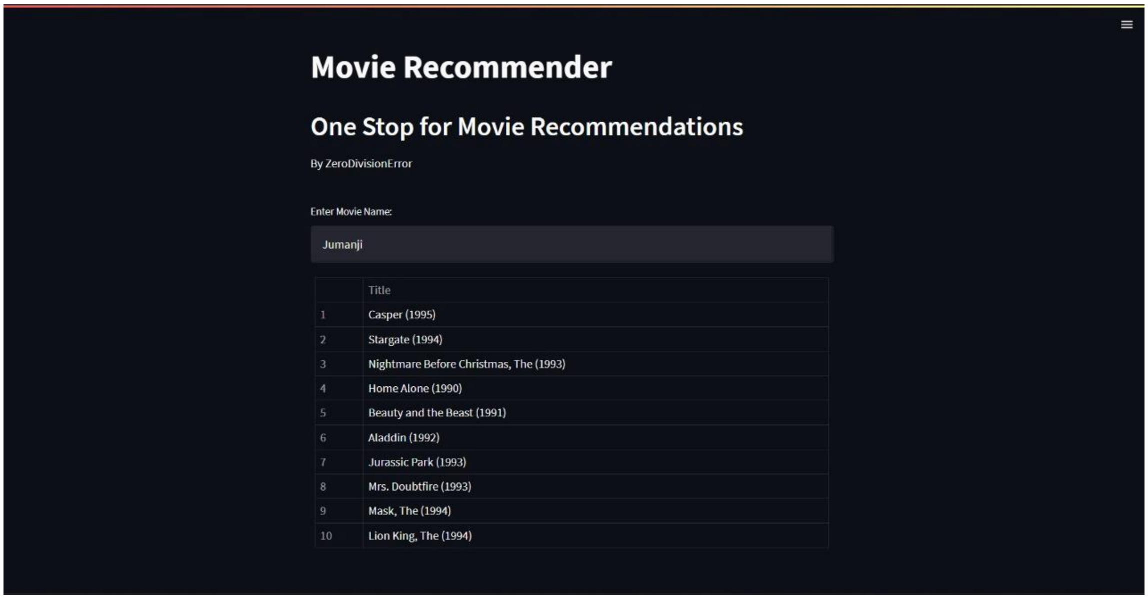
One Stop for Movie Recommendations

By ZeroDivisionError

Enter Movie Name:

Avatar

	Title
1	Zombieland (2009)
2	Inception (2010)
3	I Am Legend (2007)
4	Hangover, The (2009)
5	Dark Knight, The (2008)
6	Kung Fu Panda (2008)
7	Iron Man (2008)
8	District 9 (2009)
9	WALL·E (2008)
10	Up (2009)



Movie Recommender

One Stop for Movie Recommendations

By ZeroDivisionError

Enter Movie Name:

Jumanji

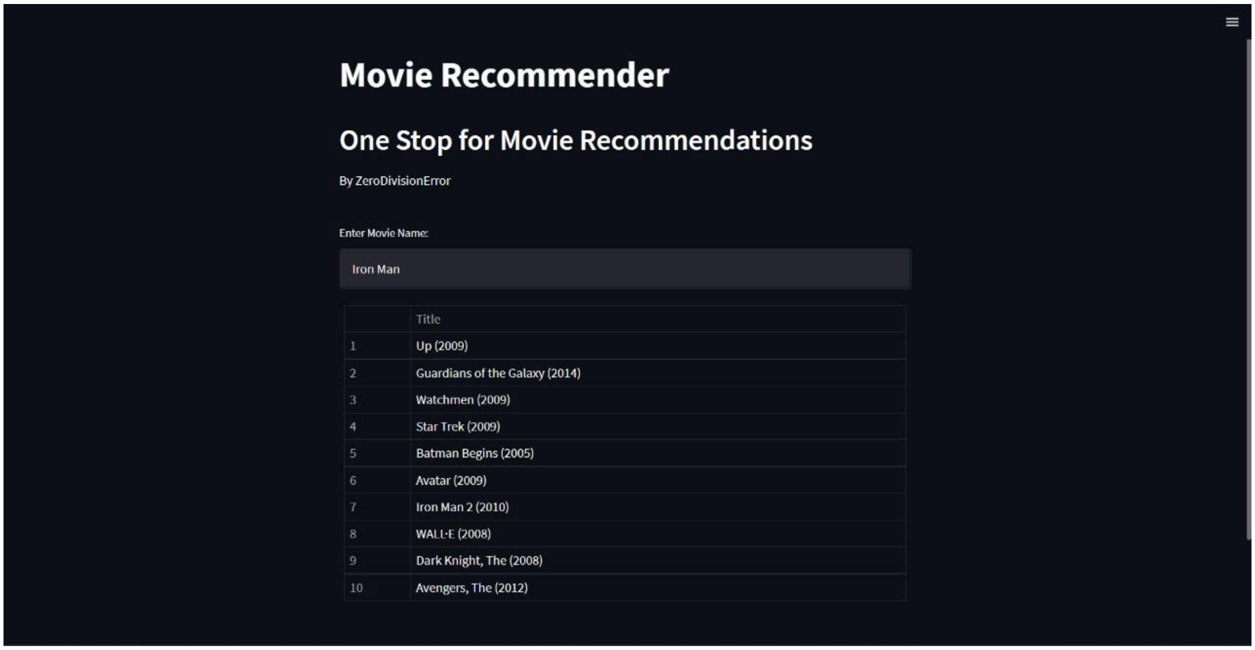
	Title
1	Casper (1995)
2	Stargate (1994)
3	Nightmare Before Christmas, The (1993)
4	Home Alone (1990)
5	Beauty and the Beast (1991)
6	Aladdin (1992)
7	Jurassic Park (1993)
8	Mrs. Doubtfire (1993)
9	Mask, The (1994)
10	Lion King, The (1994)

Technology Stack:

1. **Dataset** – MovieLens Small
2. **Algorithms** – Item based Collaborative Filtering and k-nearest neighbours
3. **FrontEnd -** StreamLit (Python)
4. **Backend -** Pandas, NumPy, SciPy, Sklearn **(**Python)

TESTING AND VALIDATION

Let us test our Movie Recommendation System by giving different inputs. Let us begin with ‘Iron Man’. Iron Man is a 2008 American superhero film based on the Marvel Comics character of the same name. It is the first film in the Marvel Cinematic Universe (MCU). All the movies recommended are superhero or animation movies which are ideal for kids as is the input movie Iron Man.



Movie Recommender

One Stop for Movie Recommendations

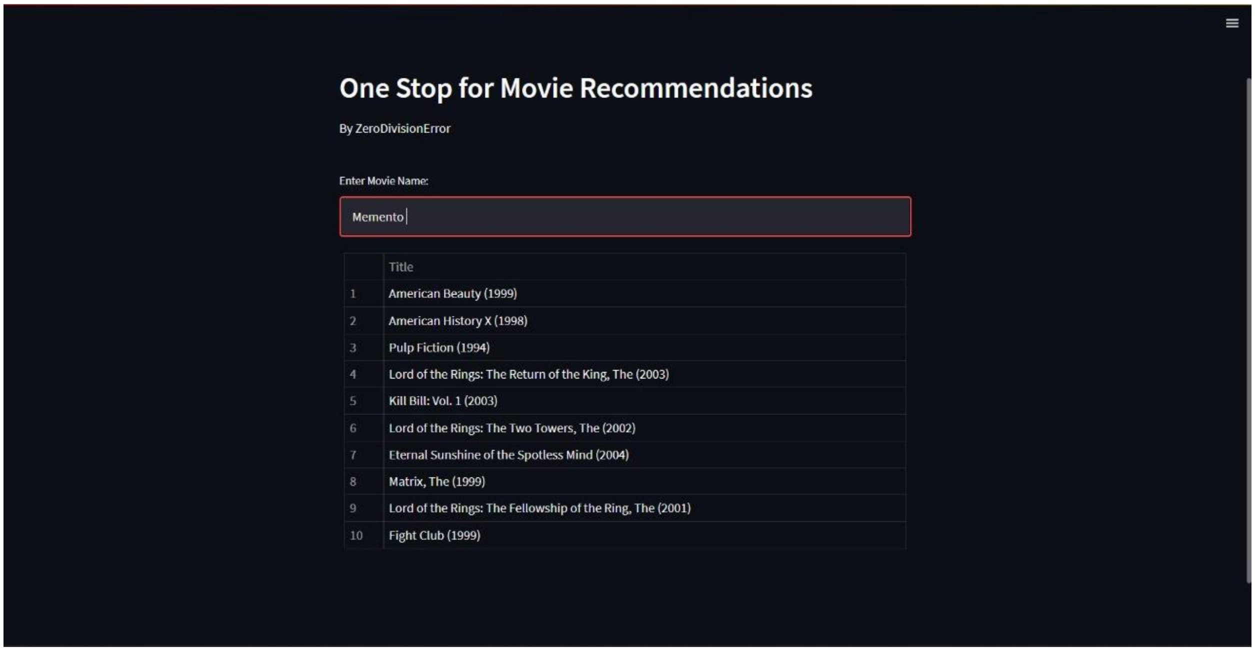
By ZeroDivisionError

Enter Movie Name:

Iron Man

	Title
1	Up (2009)
2	Guardians of the Galaxy (2014)
3	Watchmen (2009)
4	Star Trek (2009)
5	Batman Begins (2005)
6	Avatar (2009)
7	Iron Man 2 (2010)
8	WALL·E (2008)
9	Dark Knight, The (2008)
10	Avengers, The (2012)

Next, let us give the input ‘Memento’. Memento is a 2000 American psychological mystery crime thriller film directed and written by Christopher Nolan. All the movies in this list are serious and mindful movies just like Memento itself, which is a really good result.



One Stop for Movie Recommendations

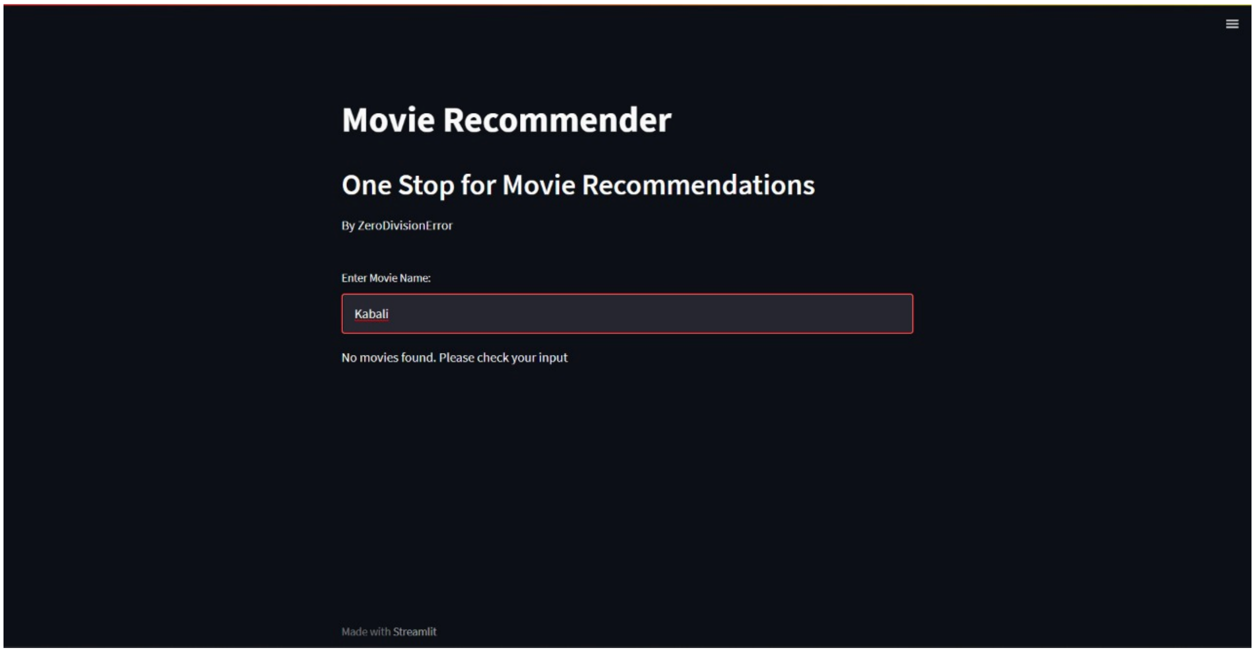
By ZeroDivisionError

Enter Movie Name:

Memento

	Title
1	American Beauty (1999)
2	American History X (1998)
3	Pulp Fiction (1994)
4	Lord of the Rings: The Return of the King, The (2003)
5	Kill Bill: Vol. 1 (2003)
6	Lord of the Rings: The Two Towers, The (2002)
7	Eternal Sunshine of the Spotless Mind (2004)
8	Matrix, The (1999)
9	Lord of the Rings: The Fellowship of the Ring, The (2001)
10	Fight Club (1999)

When a movie which is not present in the database is given as input (for example: Kabali), we get the message “No movies found. Please check your input” as an error message from the system.



Movie Recommender

One Stop for Movie Recommendations

By ZeroDivisionError

Enter Movie Name:

Kabali

No movies found. Please check your input

Made with Streamlit

Thus, our Movie Recommendation System is tested and validated successfully, which implies that our model works quite well.

DELIVERABLES

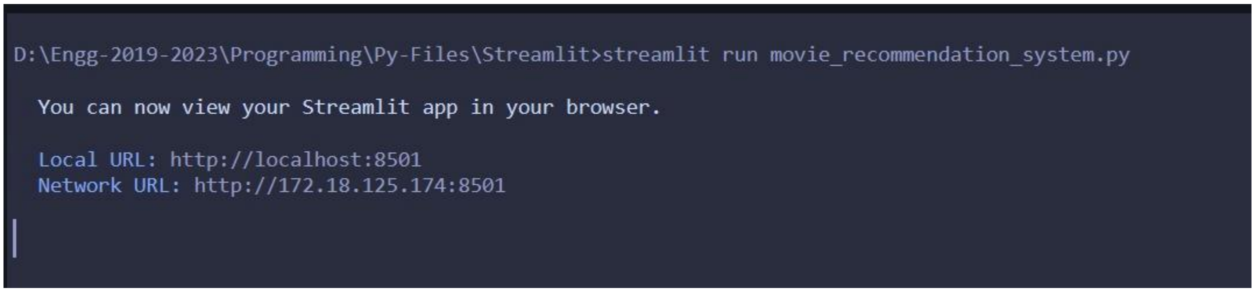
- Design Documents
- A Fully Developed Machine Learning Model
- Web Application developed with Streamlit
- Friendly User Interface
- Immediate Response Time
- High Quality Recommendations

FINANCIAL CONSIDERATION

Since the project is a python web project, the financial consideration for the project is close to zero. The only costs involved with the project are web hosting charges or web security precautions as the dataset is available for free on the Internet.

DEPLOYMENT

To kickstart the application, the following command is typed in the Terminal: *streamlit run movie_recommendation_system.py*



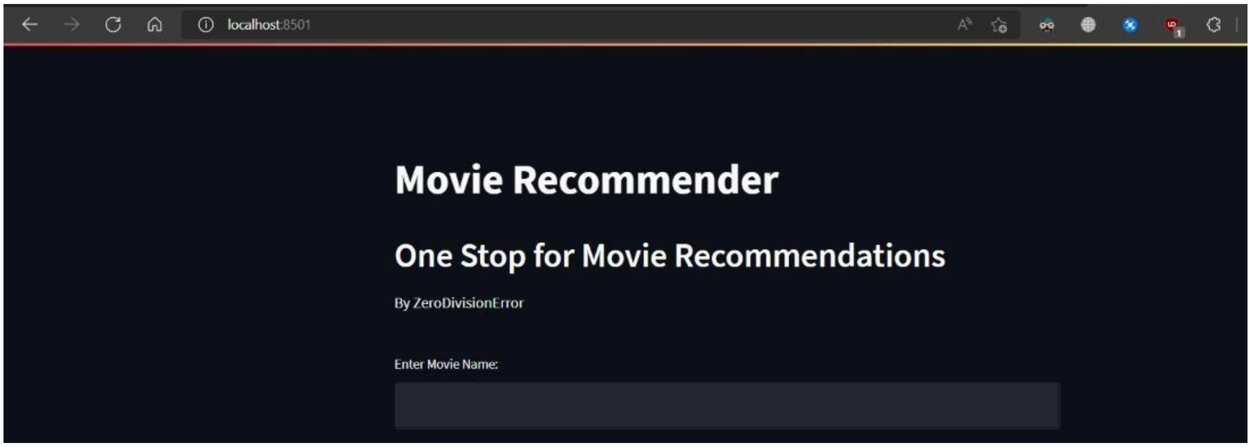
```
D:\Engg-2019-2023\Programming\Py-Files\Streamlit>streamlit run movie_recommendation_system.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.18.125.174:8501

```

Upon running the above command, we get the Local URL, which is http://localhost:8501 and if we type this URL in our browser, our deployed web application is shown as follows:



Movie Recommender

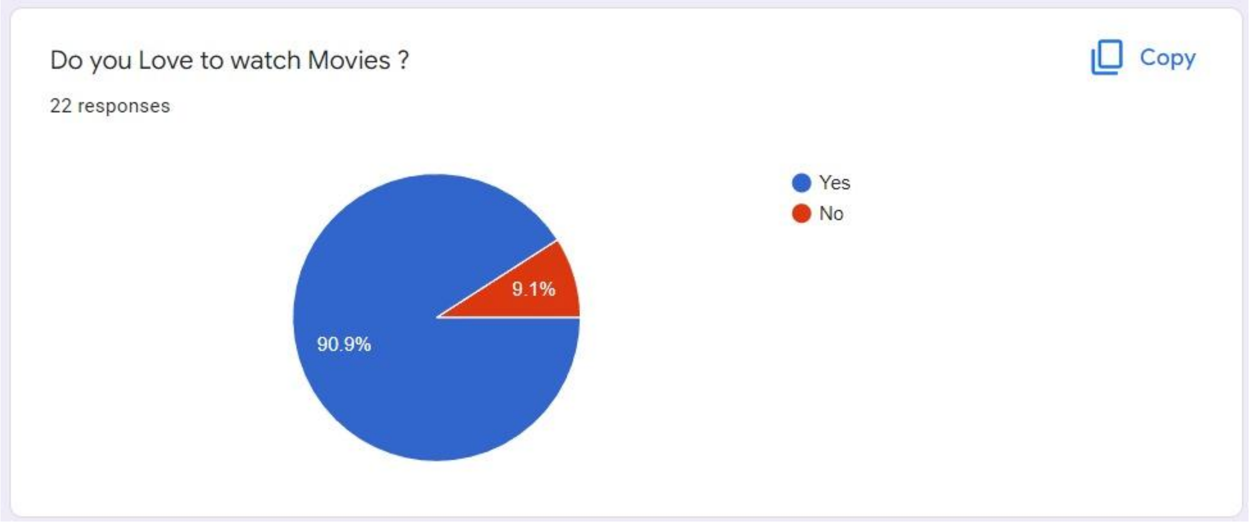
One Stop for Movie Recommendations

By ZeroDivisionError

Enter Movie Name:

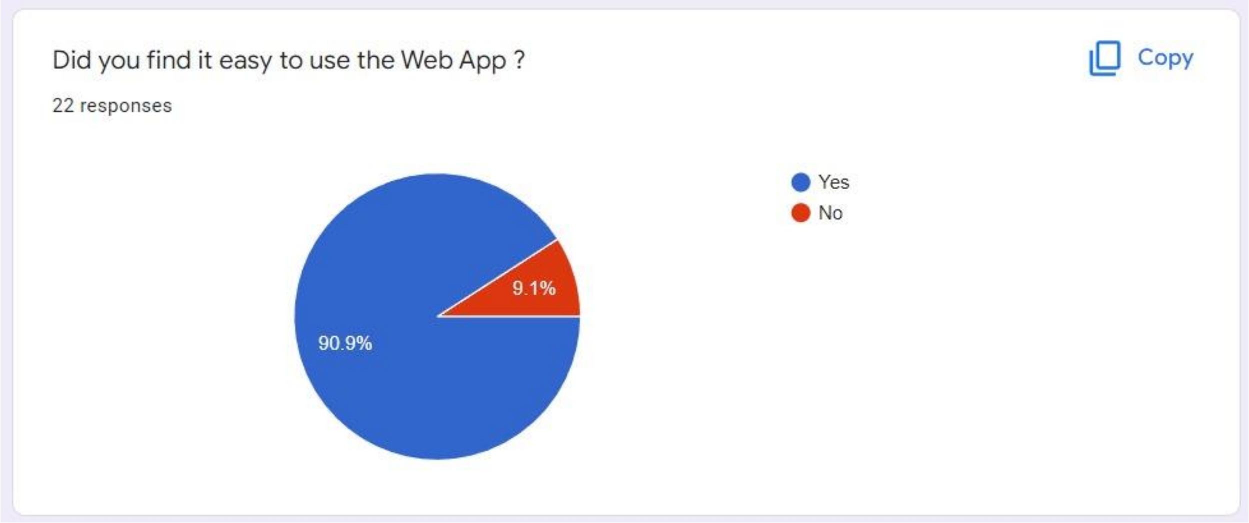
PERFORMANCE OF THE PROJECT

Feedback from Google Forms:



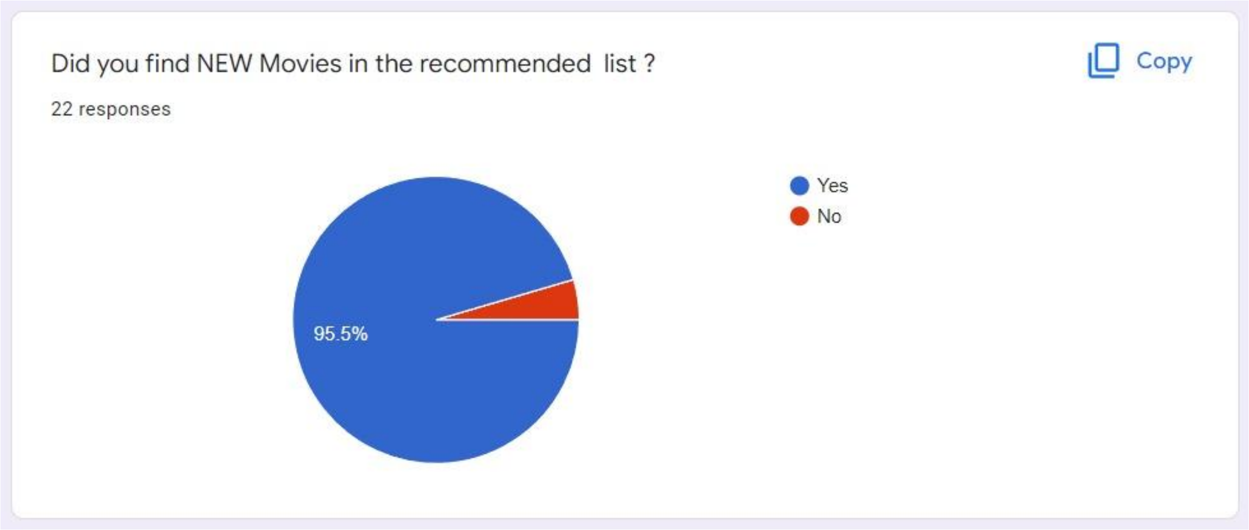
Do you Love to watch Movies ?

22 responses

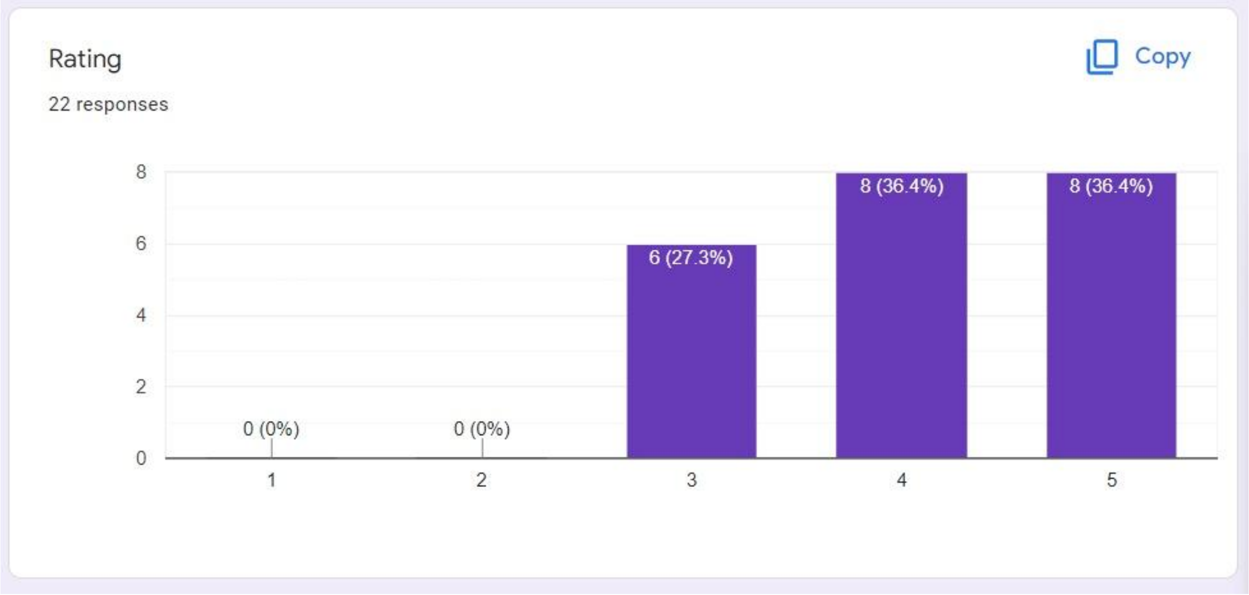


Did you find it easy to use the Web App ?

22 responses


Did you find NEW Movies in the recommended list ?

22 responses


Rating

22 responses

Based on the above responses, it is inferred that our application has been developed in a way that is useful to the end-users (i.e.) common people who watch movies regularly and are looking for recommendations.

CONCLUSION

- Everyone loves movies irrespective of age, gender, race or geographical location. We all in a way are connected to each other through this amazing medium.

- Yet what’s interesting is the fact that our choices are unique are in terms of movie preferences.

- Recommendations are not a new concept. It has been a component of real-time applications since ages, even before e-commerce emerged into existence.

- With all that said, it is still seen that similar movies are liked by a specific part of the society and that’s where our movie recommendation system comes into play.

REFERENCES

- https://data-flair.training/blogs/data-science-r-movie-recommendation/
- https://www.analyticsvidhya.com/blog/
- www.geeksforgeeks.org/python-implementation-of-movie-recommender-system/
- https://techvidvan.com/tutorials/
- https://www.mygreatlearning.com/blog/
- https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system
- https://link.springer.com/article/10.1007/s10462-019-09744-1
- https://link.springer.com/chapter/10.1007/978-981-13-1927-3_42
- https://ieeexplore.ieee.org/document/8978353
- https://colab.research.google.com/
- https://www.datacamp.com/tutorial/streamlit
- https://streamlit.io/