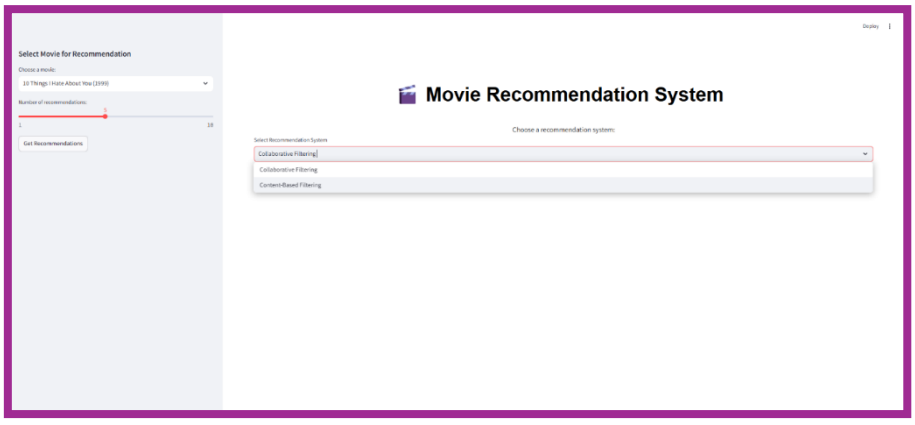
Numaira Zaib

🎬 Movie Recommendation System

Abstract

This project presents an innovative approach to movie recommendation using a combination of Machine Learning (ML) techniques, specifically Collaborative Filtering and Content-Based Filtering, with a focus on enhancing user experience through personalized recommendations. The project leverages datasets obtained from Kaggle and incorporates cutting-edge algorithms such as K-Nearest Neighbors (KNN) and TF-IDF Vectorization to suggest movies based on user preferences and content similarity. Additionally, the project employs Streamlit for the development of a user-friendly web application, enabling users to interactively receive movie suggestions. Power BI is utilized for insightful data visualization, providing an in-depth understanding of movie trends, user preferences, and other significant dataset metrics. The goal of this project is to deliver an AI-driven recommendation system that can assist users in discovering relevant movies while gaining valuable insights from the movie dataset.



By: NUMAIRA ZAIB

1. Introduction

This project aims to build a comprehensive movie recommendation system using a combination of collaborative and content-based filtering approaches. The system uses Python and Streamlit for the web app interface, integrating various datasets for both recommendation methods. Additionally, Power BI has been used to visualize dataset insights in detail.

2. Objective

The objective of this project is to:

1. Build a recommendation system to suggest movies based on user ratings and content similarity.
2. Develop a user-friendly application using Streamlit for displaying movie recommendations.
3. Visualize and analyze movie data using Power BI to gain insights into trends, genres, and user preferences.

3. Datasets Used

Collected data using www.kaggle.com

Collaborative Filtering Datasets

1. **Ratings Dataset (ratings.csv):**
    - Columns: **userId, movieId, rating, timestamp**
    - Contains user ratings for various movies, enabling Collaborative Filtering by identifying similar users or movies based on ratings.
2. **Movies Dataset (movies.csv):**
    - Columns: **movieId, title, genres**
    - Provides details about each movie's title and genre.

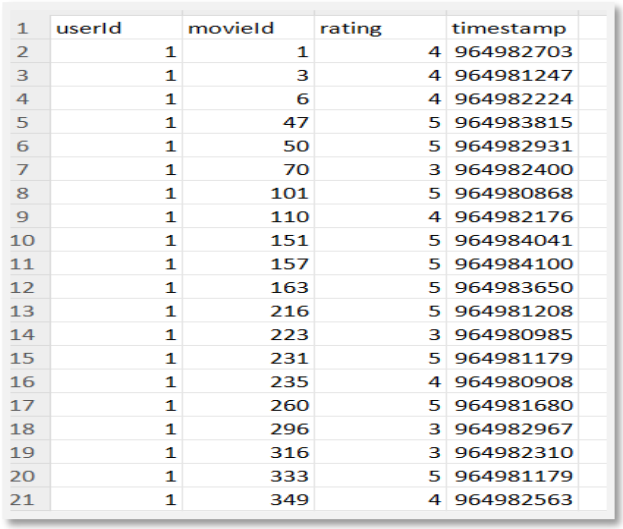
Content-Based Filtering Datasets

1. **TMDB Movies Dataset (tmdb_5000_movies.csv):**
    - Columns: **budget, genres, keywords, overview, popularity, vote_average,** etc.
    - Includes essential movie features for content-based analysis, such as genre, popularity, budget, and description.

2. TMDB Credits Dataset (tmdb_5000_credits.csv):

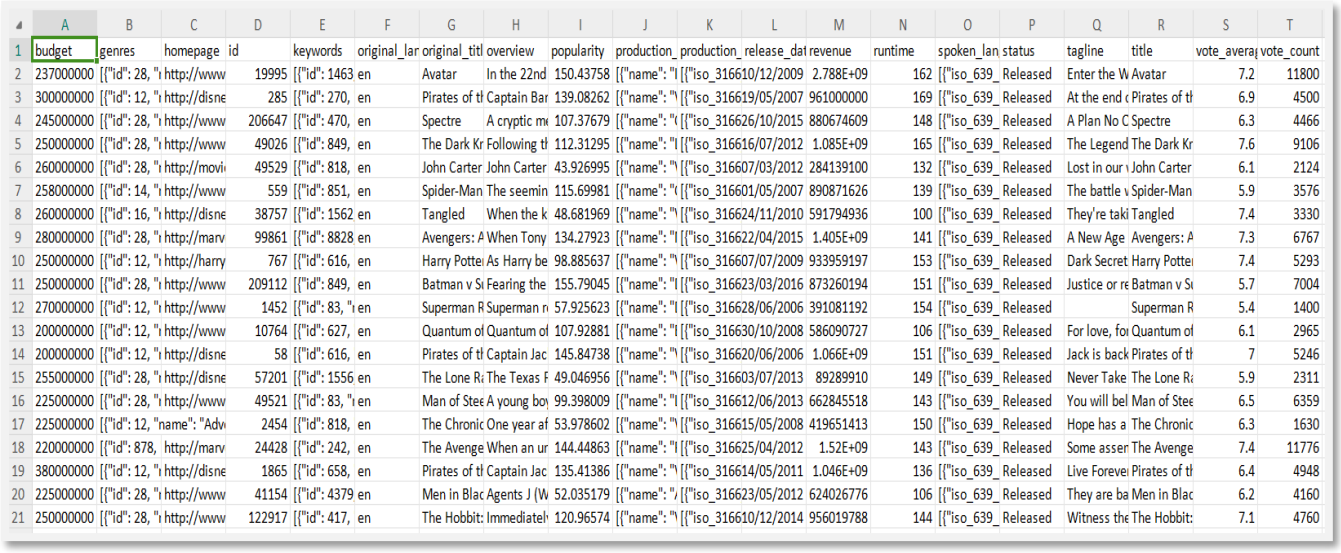
- Columns: **movie_id, title, cast, crew**
- Contains details on the movie cast and crew, which are valuable features for personalized recommendations.

	movieId	title	genres
2	1	Toy Story (1	Adventure|Animation|Children|Comedy|Fantasy
3	2	Jumanji (19	Adventure|Children|Fantasy
4	3	Grumpier Ol	Comedy|Romance
5	4	Waiting to E	Comedy|Drama|Romance
6	5	Father of th	Comedy
7	6	Heat (1995)	Action|Crime|Thriller
8	7	Sabrina (19	Comedy|Romance
9	8	Tom and Hu	Adventure|Children
10	9	Sudden Dea	Action
11	10	GoldenEye (	Action|Adventure|Thriller
12	11	American Pr	Comedy|Drama|Romance
13	12	Dracula: De	Comedy|Horror
14	13	Balto (1995	Adventure|Animation|Children
15	14	Nixon (1995	Drama
16	15	Cutthroat Is	Action|Adventure|Romance
17	16	Casino (199	Crime|Drama
18	17	Sense and S	Drama|Romance
19	18	Four Rooms	Comedy
20	19	Ace Ventura	Comedy
21	20	Money Trai	Action|Comedy|Crime|Drama|Thriller



	userId	movieId	rating	timestamp
2	1	1	4	964982703
3	1	3	4	964981247
4	1	6	4	964982224
5	1	47	5	964983815
6	1	50	5	964982931
7	1	70	3	964982400
8	1	101	5	964980868
9	1	110	4	964982176
10	1	151	5	964984041
11	1	157	5	964984100
12	1	163	5	964983650
13	1	216	5	964981208
14	1	223	3	964980985
15	1	231	5	964981179
16	1	235	4	964980908
17	1	260	5	964981680
18	1	296	3	964982967
19	1	316	3	964982310
20	1	333	5	964981179
21	1	349	4	964982563

	movie_id	title	cast	crew
2	19995	Avatar	[{"cast_id":	[{"credit_id": "52fe48009251416c750aca23", "department": "Editing", "gender": 0, "id": 1721, "job": "Editor", "name": "Stephen E. Rivkin"}, {"credit_id": "539c47ecc3a36810e30
3	285	Pirates of th	[{"cast_id":	[{"credit_id": "52fe4232c3a36847f800b579", "department": "Camera", "gender": 2, "id": 120, "job": "Director of Photography", "name": "Dariusz Wolski"}, {"credit_id": "52fe423
4	206647	Spectre	[{"cast_id":	[{"credit_id": "54805967c3a36829b5002c41", "department": "Sound", "gender": 2, "id": 153, "job": "Original Music Composer", "name": "Thomas Newman"}, {"credit_id": "52fe4
5	49026	The Dark Kn	[{"cast_id":	[{"credit_id": "52fe4781c3a36847f81398c3", "department": "Sound", "gender": 2, "id": 947, "job": "Original Music Composer", "name": "Hans Zimmer"}, {"credit_id": "52fe4781c
6	49529	John Carter	[{"cast_id":	[{"credit_id": "52fe479ac3a36847f813eaa3", "department": "Writing", "gender": 2, "id": 7, "job": "Screenplay", "name": "Andrew Stanton"}, {"credit_id": "52fe479ac3a36847f81
7	559	Spider-Man	[{"cast_id":	[{"credit_id": "52fe4252c3a36847f80151a5", "department": "Production", "gender": 1, "id": 6410, "job": "Casting", "name": "Francine Maisler"}, {"credit_id": "52fe4252c3a3684
8	38757	Tangled	[{"cast_id":	[{"credit_id": "52fe46db9251416c91062101", "department": "Production", "gender": 2, "id": 7879, "job": "Executive Producer", "name": "John Lasseter"}, {"credit_id": "52fe46dt
9	99861	Avengers: A	[{"cast_id":	[{"credit_id": "55d5f7d4c3a3683e7e0016eb", "department": "Sound", "gender": 2, "id": 531, "job": "Original Music Composer", "name": "Danny Elfman"}, {"credit_id": "55f7d488
10	767	Harry Potte	[{"cast_id":	[{"credit_id": "52fe4273c3a36847f801fab1", "department": "Camera", "gender": 0, "id": 2423, "job": "Director of Photography", "name": "Bruno Delbonnel"}, {"credit_id": "52fe
11	209112	Batman v Su	[{"cast_id":	[{"credit_id": "553bf23692514135c8002886", "department": "Sound", "gender": 2, "id": 947, "job": "Original Music Composer", "name": "Hans Zimmer"}, {"credit_id": "52fe4d5b
12	1452	Superman R	[{"cast_id":	[{"credit_id": "553bef6a9251416874003c8f", "department": "Production", "gender": 2, "id": 3276, "job": "Casting", "name": "Roger Mussenden"}, {"credit_id": "553bee55925141
13	10764	Quantum of	[{"cast_id":	[{"credit_id": "52fe43b29251416c7501aa63", "department": "Writing", "gender": 2, "id": 455, "job": "Screenplay", "name": "Paul Haggis"}, {"credit_id": "52fe43b29251416c7501
14	58	Pirates of th	[{"cast_id":	[{"credit_id": "52fe4211c3a36847f8001873", "department": "Camera", "gender": 2, "id": 120, "job": "Director of Photography", "name": "Dariusz Wolski"}, {"credit_id": "52fe421
15	57201	The Lone Ra	[{"cast_id":	[{"credit_id": "52fe4928c3a36847f818be95", "department": "Directing", "gender": 2, "id": 1704, "job": "Director", "name": "Gore Verbinski"}, {"credit_id": "52fe4928c3a36847f8
16	49521	Man of Stee	[{"cast_id":	[{"credit_id": "52fe4799c3a36847f813e667", "department": "Sound", "gender": 2, "id": 947, "job": "Original Music Composer", "name": "Hans Zimmer"}, {"credit_id": "52fe4799c
17	2454	The Chronic	[{"cast_id":	[{"credit_id": "55a239e69251412979002e8b", "department": "Production", "gender": 1, "id": 1326, "job": "Casting", "name": "Liz Mullane"}, {"credit_id": "52fe4359c3a36847f80
18	24428	The Avenge	[{"cast_id":	[{"credit_id": "52fe4495c3a368484e02b1cf", "department": "Sound", "gender": 2, "id": 37, "job": "Original Music Composer", "name": "Alan Silvestri"}, {"credit_id": "5496018d92
19	1865	Pirates of th	[{"cast_id":	[{"credit_id": "566b4f54c3a3683f56005151", "department": "Camera", "gender": 2, "id": 120, "job": "Director of Photography", "name": "Dariusz Wolski"}, {"credit_id": "52fe431
20	41154	Men in Blac	[{"cast_id":	[{"credit_id": "52fe45b7c3a36847f80d68c7", "department": "Production", "gender": 2, "id": 488, "job": "Executive Producer", "name": "Steven Spielberg"}, {"credit_id": "52fe45b
21	122917	The Hobbit:	[{"cast_id":	[{"credit_id": "548ad49a9251414fa20011ab", "department": "Sound", "gender": 2, "id": 117, "job": "Original Music Composer", "name": "Howard Shore"}, {"credit_id": "553620f



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	budget	genres	homepage	id	keywords	original_lar	original_titl	overview	popularity	production_	production_	release_dat	revenue	runtime	spoken_lan	status	tagline	title	vote_avera	vote_count
2	237000000	[{"id": 28, "n	http://www	19995	[{"id": 1463	en	Avatar	In the 22nd	150.43758	[{"name": "I	[{"iso_3166	10/12/2009	2.788E+09	162	[{"iso_639_	Released	Enter the W	Avatar	7.2	11800
3	300000000	[{"id": 12, "n	http://disne	285	[{"id": 270,	en	Pirates of th	Captain Bar	139.08262	[{"name": "I	[{"iso_3166	05/2007	961000000	169	[{"iso_639_	Released	At the end (	Pirates of th	6.9	4500
4	245000000	[{"id": 28, "n	http://www	206647	[{"id": 470,	en	Spectre	A cryptic m	107.37679	[{"name": "I	[{"iso_3166	26/10/2015	880674609	148	[{"iso_639_	Released	A Plan No C	Spectre	6.3	4466
5	250000000	[{"id": 28, "n	http://www	49026	[{"id": 849,	en	The Dark Kr	Following th	112.31295	[{"name": "I	[{"iso_3166	16/07/2012	1.085E+09	165	[{"iso_639_	Released	The Legend	The Dark Kr	7.6	9106
6	260000000	[{"id": 28, "n	http://movi	49529	[{"id": 818,	en	John Carter	John Carter	43.926995	[{"name": "I	[{"iso_3166	07/03/2012	284139100	132	[{"iso_639_	Released	Lost in our v	John Carter	6.1	2124
7	258000000	[{"id": 14, "n	http://www	559	[{"id": 851,	en	Spider-Man	The seemin	115.69981	[{"name": "I	[{"iso_3166	01/05/2007	890871626	139	[{"iso_639_	Released	The battle v	Spider-Man	5.9	3576
8	260000000	[{"id": 16, "n	http://disne	38757	[{"id": 1562	en	Tangled	When the k	48.681969	[{"name": "I	[{"iso_3166	24/11/2010	591794936	100	[{"iso_639_	Released	They're taki	Tangled	7.4	3330
9	280000000	[{"id": 28, "n	http://marv	99861	[{"id": 8828	en	Avengers: A	When Tony	134.27923	[{"name": "I	[{"iso_3166	22/04/2015	1.405E+09	141	[{"iso_639_	Released	A New Age	Avengers: A	7.3	6767
10	250000000	[{"id": 12, "n	http://harry	767	[{"id": 616,	en	Harry Potte	As Harry be	98.885637	[{"name": "I	[{"iso_3166	07/07/2009	933959197	153	[{"iso_639_	Released	Dark Secret	Harry Potte	7.4	5293
11	250000000	[{"id": 28, "n	http://www	209112	[{"id": 849,	en	Batman v Su	Fearing the	155.79045	[{"name": "I	[{"iso_3166	03/2016	873260194	151	[{"iso_639_	Released	Justice or re	Batman v Su	5.7	7004
12	270000000	[{"id": 12, "n	http://www	1452	[{"id": 83, "n	en	Superman R	Superman n	57.925623	[{"name": "I	[{"iso_3166	28/06/2006	391081192	154	[{"iso_639_	Released		Superman R	5.4	1400
13	200000000	[{"id": 12, "n	http://www	10764	[{"id": 627,	en	Quantum of	Quantum of	107.92881	[{"name": "I	[{"iso_3166	10/10/2008	586090727	106	[{"iso_639_	Released	For love, for	Quantum of	6.1	2965
14	200000000	[{"id": 12, "n	http://disne	58	[{"id": 616,	en	Pirates of th	Captain Jac	145.84738	[{"name": "I	[{"iso_3166	06/2006	1.066E+09	151	[{"iso_639_	Released	Jack is back	Pirates of th	7	5246
15	255000000	[{"id": 28, "n	http://disne	57201	[{"id": 1556	en	The Lone Ra	The Texas F	49.046956	[{"name": "I	[{"iso_3166	07/2013	89289910	149	[{"iso_639_	Released	Never Take	The Lone Ra	5.9	2311
16	225000000	[{"id": 28, "n	http://www	49521	[{"id": 83, "n	en	Man of Stee	A young boy	99.398009	[{"name": "I	[{"iso_3166	06/2013	662845518	143	[{"iso_639_	Released	You will bel	Man of Stee	6.5	6359
17	225000000	[{"id": 12, "name": "Adv		2454	[{"id": 818,	en	The Chronic	One year af	53.978602	[{"name": "I	[{"iso_3166	05/2008	419651413	150	[{"iso_639_	Released	Hope has a	The Chronic	6.3	1630
18	220000000	[{"id": 878,	http://marv	24428	[{"id": 242,	en	The Avenge	When an ur	144.44863	[{"name": "I	[{"iso_3166	04/2012	1.52E+09	143	[{"iso_639_	Released	Some asser	The Avenge	7.4	11776
19	380000000	[{"id": 12, "n	http://disne	1865	[{"id": 658,	en	Pirates of th	Captain Jac	135.41386	[{"name": "I	[{"iso_3166	04/2011	1.046E+09	136	[{"iso_639_	Released	Live Forever	Pirates of th	6.4	4948
20	225000000	[{"id": 28, "n	http://www	41154	[{"id": 4379	en	Men in Blac	Agents J (W	52.035179	[{"name": "I	[{"iso_3166	05/2012	624026776	106	[{"iso_639_	Released	They are ba	Men in Blac	6.2	4160
21	250000000	[{"id": 28, "n	http://www	122917	[{"id": 417,	en	The Hobbit:	Immediatel	120.96574	[{"name": "I	[{"iso_3166	10/12/2014	956019788	144	[{"iso_639_	Released	Witness the	The Hobbit:	7.1	4760

4. Models and Algorithms Used

4.1 Collaborative Filtering

Collaborative Filtering provides recommendations based on user interaction data. In this project, we implemented K-Nearest Neighbors (KNN) with cosine similarity for rating-based recommendations.

Algorithm: K-Nearest Neighbors (KNN)

- Purpose: KNN identifies similar movies or users based on past rating data.
- Functionality:
    - A pivot table of users vs. movies is created, where each cell represents a user's rating for a movie.
    - Cosine similarity is used to calculate the similarity between movies based on ratings.
    - For a given movie, KNN finds the top-K movies with similar ratings, and these similar movies are recommended to the user.

Steps in Code for Collaborative Filtering:

1. **Data Loading and Preparation:**
    - The ratings and movie data are merged based on movieId.
    - Movies with fewer ratings are filtered out to avoid sparsity issues.
2. **Pivot Table Creation:**
    - A pivot table with userId and title is created.
    - This matrix is fed into the KNN model to find movies with similar user interaction patterns.
3. **Model Training and Prediction:**
    - Using KNN with cosine similarity, similar movies are identified for each movie in the dataset.
    - This allows recommending top-K movies based on user similarity for Collaborative Filtering.

4.2 Content-Based Filtering

Content-based filtering recommends movies by identifying items with similar characteristics. This approach utilizes metadata such as genre, keywords, cast, and crew.

Algorithm: TF-IDF Vectorization and K-Nearest Neighbors (KNN)

- TF-IDF Vectorization: The overview, keywords, genres, cast, and crew columns are vectorized to create a content feature set.
    - TF-IDF (Term Frequency-Inverse Document Frequency) assigns weights to words based on their frequency, making less common but relevant words more impactful.
- K-Nearest Neighbors (KNN) with Cosine Similarity:
    - KNN identifies movies with the closest vectorized content, suggesting those with similar characteristics.

Steps in Code for Content-Based Filtering:

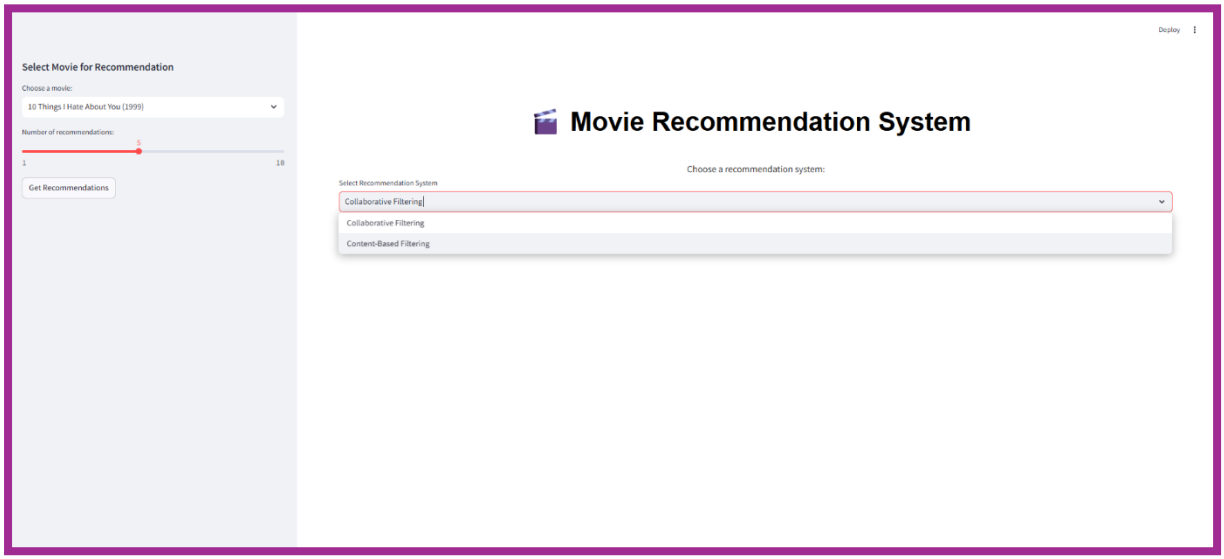
1. **Data Preprocessing:**
    - The genres, keywords, cast, and crew columns are converted to text.
    - A combined column metadata is created by concatenating these columns to represent the movie content.
2. **TF-IDF Vectorization:**
    - TF-IDF is applied to metadata, creating a sparse matrix of feature weights.
    - This matrix serves as the input for KNN to calculate similarity scores.
3. **Similarity Calculation and Recommendation:**
    - The KNN model finds the top-K most similar movies based on cosine similarity.
    - Given a movie, the model recommends other movies with similar content profiles.

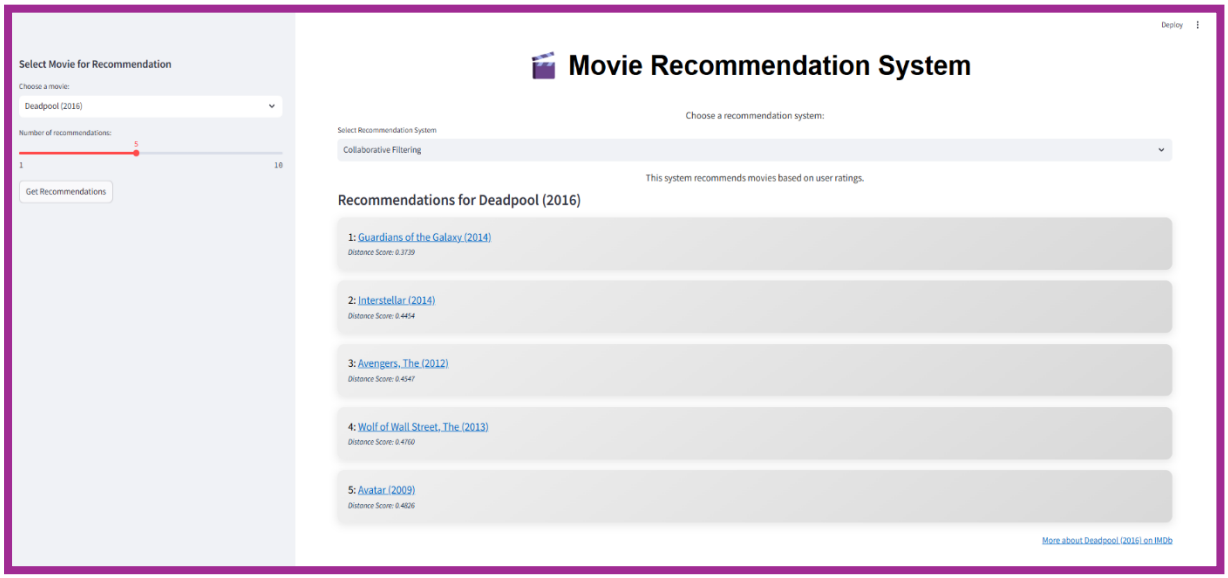
5. Streamlit Application

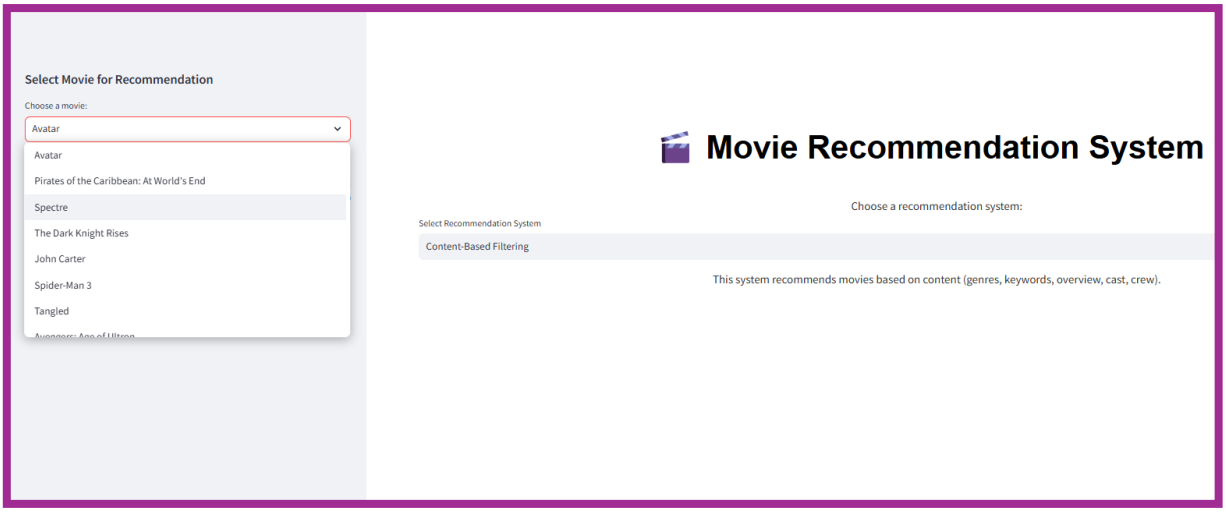
The Streamlit app provides an interactive interface for users to obtain recommendations based on collaborative or content-based filtering. Key components of the Streamlit app include:

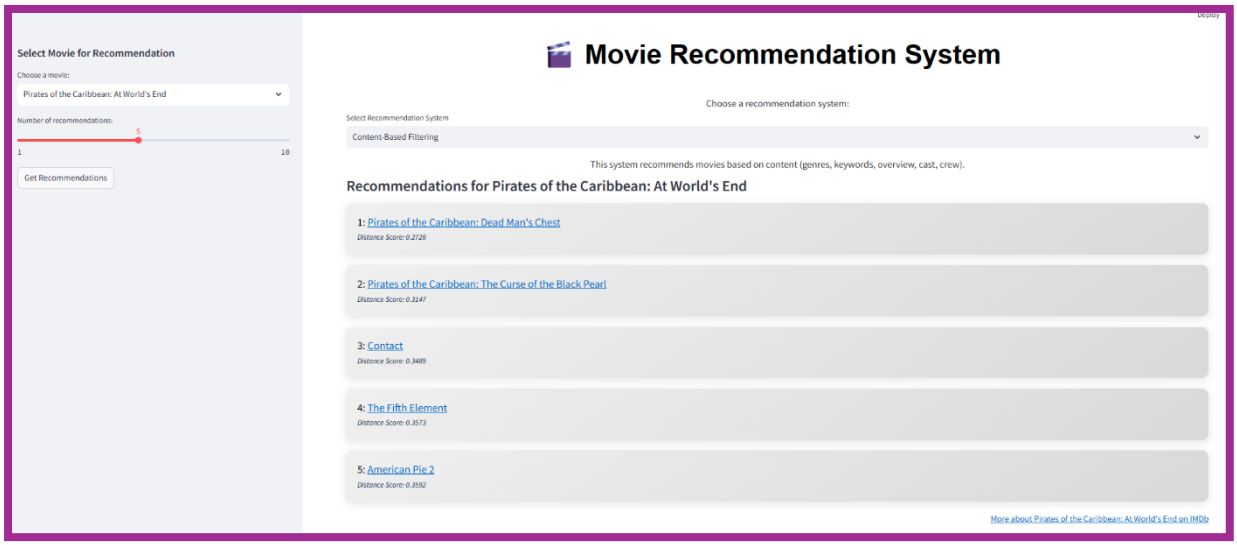
5.1 Features and User Interface

1. **Collaborative Filtering Section:**
    - Allows users to select a movie and receive recommendations based on user rating patterns.
    - An input slider lets users specify the number of recommendations.
2. **Content-Based Filtering Section:**
    - Allows users to select a movie and receive recommendations based on similar metadata features.
    - Displays detailed information about each recommended movie.
3. **Custom Styling:**
    - Custom CSS styling is used to enhance the visual layout and create hover effects for interactivity.









6. Power BI Dashboard for Data Visualization

Power BI is used to create data visualizations that provide insights into the movie dataset. Key metrics and charts include:

6.1 Measures and Calculations

- AvgRating = AVERAGE(ratings[rating])
  Calculates the average rating given by users across all movies

- MaxRating = MAX(Ratings[rating])
  Retrieves the highest rating value from the ratings dataset, highlighting the maximum score given by any user

- RatingCount = COUNT(Ratings[rating])
  Counts the total number of individual ratings submitted by users for movies

- St.DevRating = STDEV.P(Ratings[rating])
  Calculates the population standard deviation of all movie ratings, providing insight into rating variability

- AvgVote = AVERAGE(tmdb_5000_movies[vote_average])
  Calculates the average number of votes received across all movies in the tmdb_5000_movies dataset

- PopularityScore= AVERAGE(tmdb_5000_movies[popularity])
  Calculates the average popularity score for movies in the dataset, representing user engagement

6.2 Visualizations

1. **Treemap of Rating Count by Rating:**
   - Shows the distribution of rating counts, indicating popular rating ranges.
2. **Donut Chart (Budget vs Revenue):**
   - Compares the average budget against revenue, offering insights into profitability across movies.
3. **Clustered Bar Chart (Revenue by Genre):**
   - Displays the total revenue generated by different genres, highlighting successful genres.
4. **Line and Stacked Column Chart (Vote Count by Vote Average):**
   - Shows user engagement through vote count and average rating.
5. **Genre Slicer:**
   - Enables filtering by genre, allowing users to analyze genre-specific trends.