CONTENT BASED MOVIE RECOMMENDATION SYSTEM

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

A. Sri Anvitha (AP21110011551)

Manish Pandey(AP21110011280)

P. Geeth Krishna(AP21110011504)

Kapuganti Gayatri Naga Prishita(AP21110011542)

M. Balaji(AP21110011544)

Under the Guidance of

Abinash Pujahari

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

Nov, 2023

Certificate

Date: 4-Dec-23

This is to certify that the work present in this Project entitled "**CONTENT BASED MOVIE RECOMMENDATION SYSTEM**" has been carried out by **Kapuganti Gayatri Naga Prishita** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Dr. Abinash Pujahari,

Assistant Professor,

Department of CSE,

SRMAP University.

Acknowledgements

We extend our deepest appreciation to the individuals and organizations whose unwavering support and invaluable guidance played a pivotal role in the fruition of our undergraduate research opportunities program.

A profound debt of gratitude is owed to our esteemed faculty mentor, Dr. Abinash Pujahari, whose constant encouragement, constructive feedback, and vigilant supervision were instrumental throughout the entirety of our research endeavor. Dr. Pujahari not only assisted in delineating the project's scope, objectives, and methodology but also provided us with essential resources and tools essential for its execution. His unwavering availability for consultation and guidance proved indispensable whenever we encountered challenges or difficulties during the project.

In expressing our gratitude, we wish to acknowledge the collective effort of all those who contributed to the triumphant culmination of this project. Each member of our dedicated team deserves heartfelt thanks for their diligence, commitment, and collaborative spirit, which significantly contributed to the project's success.

The realization of this project was made possible through the indispensable support of various resources, including the vast expanse of the Internet, research papers, and reference books. These resources played a crucial role in fostering a comprehensive understanding during the demanding phases of our work, thereby enriching the overall quality and depth of our research.

Table of Contents

Abstract

In a world saturated with entertainment choices, navigating the vast landscape of movies to find one that aligns with personal preferences can be a daunting task. This paper introduces a user-friendly content-based movie recommendation system developed in Python, leveraging the extensive MovieLens dataset. The system combines information from both the movies dataset and the user ratings dataset, offering a personalized cinematic journey for users.

The primary aim of this recommendation system is to enhance the overall movie-watching experience by providing tailored suggestions based on individual tastes. Unlike traditional recommendation systems, which often rely solely on collaborative filtering, our approach takes into account the intrinsic features of movies and user preferences, creating a more nuanced and accurate recommendation mechanism.

By analyzing the content of movies, such as genre, director, and cast, the system goes beyond the limitations of collaborative filtering, ensuring that users receive recommendations that align not only with popular choices but also with their unique preferences. This approach acknowledges the diverse nature of individual tastes and aims to cater to a wide range of movie enthusiasts.

The incorporation of user ratings further refines the recommendation process, allowing the system to adapt and evolve over time. As users interact with the platform, the system learns from their preferences and continuously refines its suggestions, creating a dynamic and responsive movie recommendation ecosystem.

This content-based movie recommendation system holds the potential to revolutionize how individuals engage with cinematic content in their daily lives. Beyond its technical underpinnings, the system is designed to empower users in making informed and enjoyable choices, fostering a deeper connection between individuals and the world of cinema. As we navigate the ever-expanding universe of movies, this system serves as a personalized guide, making the pursuit of cinematic delight an effortless and rewarding experience.

Abbreviations

CBF	Content-Based Filtering
CS	Cosine Similarity
W2V	Word2Vec
BOW	Bag of Words
CBOW	Continuous Bag of Words
NLP	Natural Language Processing
Numpy	Numerical Python
CSV	Comma Separated Values

List of Figures

1. Introduction

This project by using Python and the MovieLens dataset, our mission is to refine movie selection. We delve into movie specifics—genres, tags, user ratings—and analyze user ratings for continual improvement. This project seeks to blend these datasets, creating a personalized movie suggestion system.

We employ the methodology of, utilizing the Word2Vec model to transform movie attributes into meaningful vectors. This enables a understanding of the inherent characteristics like genres and tags. Leveraging the power of cosine similarity, we gauge the likeness between these vectors, refining our content-based movie recommendation system. This approach ensures a refined analysis of movie attributes and user preferences, enhancing the accuracy and personalization of our suggestions.

1.1 Overview of libraries

1.1.1 Python Libraries

NumPy: NumPy means Numerical Python. Inside, you will find things called multidimensional array objects, which are like organized sets of numbers. NumPy also has routines, which are like special methods, for doing things with these sets of numbers. So, in simple terms, NumPy is a special library in Python that is good at handling and working with large sets of numbers.

Pandas: It is a Python library, serves as a robust tool for dataset management within our project. This library features a suite of functions for in-depth data analysis, cleansing, exploration, and manipulation. Its capabilities empower us to efficiently handle diverse data tasks, ensuring a streamlined approach to working with datasets.

1.1.2 Gensim

Gensim is a powerful Python library, primarily known for its capabilities in natural language processing (NLP). It provides tools for topics like word embedding using Word2Vec, Doc2Vec, and other algorithms, making it a go-to choose for NLP tasks. Gensim stands out for its efficiency in handling large text corpora, offering a user-friendly interface for tasks such as document similarity analysis, text summarization, and topic modelling. As a versatile toolkit, Gensim empowers programmers to seamlessly implement and experiment with advanced techniques in the realm of natural language processing.

1.2 Models

1.1.1 Word2Vec

The Word2Vec model, a fundamental part of Gensim, uses neural networks to learn high-dimensional word representations. Using Skip-gram architecture or the

Continuous Bag of Words (CBOW), Word2Vec makes use of a shallow neural network to capture meaningful connections among words

The goal of the CBOW model is to forecast the target word by taking into account the context words that surround it. The Skip-gram model, on the other hand, uses a target word to predict context words. The neural network modifies its weights during training in order to reduce the difference between predicted and actual words. With this modification, brief vector representations—also referred to as word embeddings—are produced in which words with related meanings are positioned adjacent to one another in the vector space.

The trained Word2Vec model not only excels at capturing syntactic and semantic relationships but also allows arithmetic operations on word vectors, unveiling linguistic nuances. In our project, Word2Vec serves as a powerful tool to transform movie attributes into meaningful vectors, fostering a nuanced understanding of the inherent characteristics crucial for enhancing the precision of our content-based movie recommendation system.

2. Methodology

We have to perform preprocessing of the dataset and concatenate the required attributes to make single features. Later, we convert the features to the vector and find the similarity between them. Finally, Get the required output based on the architecture given below:

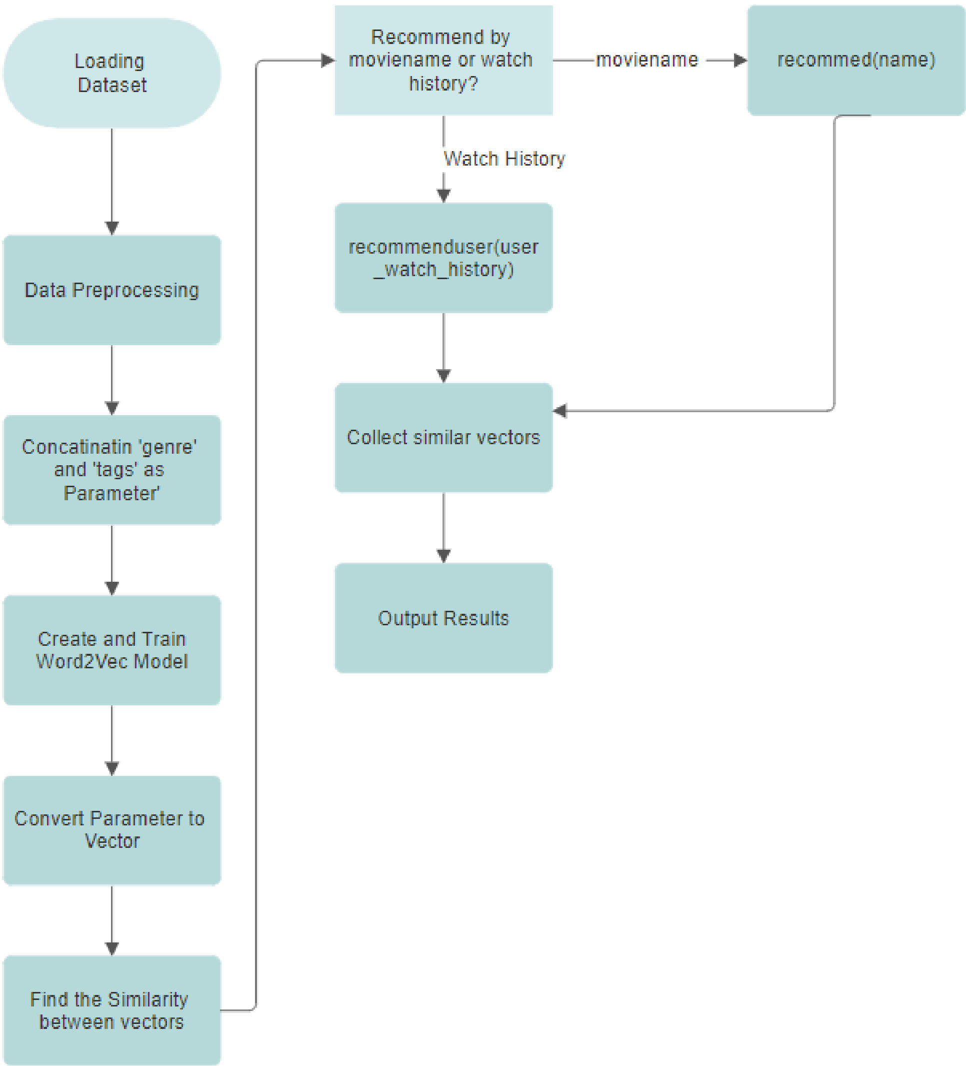

Fig.1.System Architecture

2.1 Data extraction

2.1.1 Importing Libraries: This step involves importing necessary libraries such as Pandas for data handling, NumPy for numerical operations, and Gensim for

Word2Vec model implementation. These libraries enable data manipulation, vector operations, and model training

2.1.2 Loading Datasets: Importing movie and tag data from CSV files into Pandas Data Frames. This step sets the foundation for subsequent data processing by storing the dataset in a structured format for analysis.

The "MovieLens 20M Dataset" is utilized in this study's movie recommendation process. You can access this dataset on Kaggle.com. Six CSV files make up the dataset; only two of them—"movie.csv" and "tag.csv"—are used.

The dataset contains following attribute:

tag.csv: userId, movieId, tag, timestamp

movie.csv: movieId, title, genres

```
<bound method NDFrame.head of        movieId                               title  \
0            1                    Toy Story (1995)
1            2                      Jumanji (1995)
2            3             Grumpier Old Men (1995)
3            4            Waiting to Exhale (1995)
4            5  Father of the Bride Part II (1995)
...        ...                                 ...
27273   131254          Kein Bund für's Leben (2007)
27274   131256         Feuer, Eis & Dosenbier (2002)
27275   131258                    The Pirates (2014)
27276   131260                   Rentun Ruusu (2001)
27277   131262                      Innocence (2014)

                                            genres
0      Adventure|Animation|Children|Comedy|Fantasy
1                       Adventure|Children|Fantasy
2                                   Comedy|Romance
3                             Comedy|Drama|Romance
4                                           Comedy
...                                            ...
27273                                       Comedy
27274                                       Comedy
27275                                    Adventure
27276                           (no genres listed)
27277                      Adventure|Fantasy|Horror

[27278 rows x 3 columns]>
```

Fig.2.Glimpse of movie.csv

```
<bound method NDFrame.head of         userId  movieId            tag            timestamp
0           18     4141    Mark Waters  2009-04-24 18:19:40
1           65      208      dark hero  2013-05-10 01:41:18
2           65      353      dark hero  2013-05-10 01:41:19
3           65      521  noir thriller  2013-05-10 01:39:43
4           65      592      dark hero  2013-05-10 01:41:18
...        ...      ...            ...                  ...
465559  138446    55999        dragged  2013-01-23 23:29:32
465560  138446    55999  Jason Bateman  2013-01-23 23:29:38
465561  138446    55999         quirky  2013-01-23 23:29:38
465562  138446    55999            sad  2013-01-23 23:29:32
465563  138472      923  rise to power  2007-11-02 21:12:47

[465564 rows x 4 columns]>
```

Fig.3.Glimpse of tag.csv

2.2 Data Cleaning

2.2.1 Handling Missing Values: Identifying and addressing missing values (NaNs) within the datasets. This process involves checking for missing data points and deciding on suitable strategies like filling missing values or excluding incomplete entries.

2.2.2 Removing Duplicates: Identifying and handling duplicate entries within the datasets. Removing duplicated rows ensures data consistency and prevents bias in subsequent analyses.

2.2.3 Joining Datasets: Joining the tag and movie file to get the dataset to work on given below:

	movieId	title	genres	tag	parameters
0	1	Toy Story (1995)	Adventure,Animation,Children,Comedy,Fantasy	Watched, computer animation, Disney animated f...	Adventure,Animation,Children,Comedy,FantasyWat...
1	2	Jumanji (1995)	Adventure,Children,Fantasy	time travel, adapted from:book, board game, ch...	Adventure,Children,Fantasytime travel, adapted...
2	3	Grumpier Old Men (1995)	Comedy,Romance	old people that is actually funny, sequel feve...	Comedy,Romanceold people that is actually funn...
3	4	Waiting to Exhale (1995)	Comedy,Drama,Romance	chick flick, revenge, characters, chick flick,...	Comedy,Drama,Romancechick flick, revenge, char...
4	5	Father of the Bride Part II (1995)	Comedy	Diane Keaton, family, sequel, Steve Martin, we...	ComedyDiane Keaton, family, sequel, Steve Mart...
...	...	...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy		Comedy
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy		Comedy
27275	131258	The Pirates (2014)	Adventure	bandits, Korea, mutiny, pirates, whale	Adventurebandits, Korea, mutiny, pirates, whale
27276	131260	Rentun Ruusu (2001)	(no genres listed)		(no genres listed)
27277	131262	Innocence (2014)	Adventure,Fantasy,Horror		Adventure,Fantasy,Horror

27278 rows × 5 columns

Fig.4.Working Dataset

2.3 Tokenization

Tokenization is the method of dividing a document into smaller parts, like individual words or phrases. In the code:

The concatenation of movie genres and tags into the 'parameters' column combines different textual elements. While not explicitly tokenizing, this step consolidates diverse textual information into a single field, preparing it for further processing.

During the Word2Vec model training, the 'parameters' column likely underwent splitting (by commas or other delimiters) to create lists of words. This separation contributes to the creation of distinct elements for analysis, resembling a form of tokenization at a higher level.

2.4 Word2Vec Embeddings

2.4.1 Concatenating Features: Combining movie genres and tags into a single 'parameters' column. This consolidation step creates a unified field that includes both genre and tag information for each movie.

2.4.2 Word2Vec Model Training: Using the combined parameters, train a Word2Vec model to produce embeddings. This model captures semantic similarities between words, such as genres and tags, by modeling them as dense vectors in continuous space.

Maximizing the likelihood that a target word will be predicted by the skip-gram model is its training objective. The aim for a word sequence w1, w2,.. wT can be expressed as the average log probability.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0}\log p(w_{t+j}|w_t)$$

Fig.5.Skipgram objective

where "c" represents the length of context which is used for training, the probability is found by using the following function also called softmax.

$$p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top}v_{w_I}\right)}$$

Fig.6.softmax

where "v'" and "v' " are vector for target and vector for context words respectively and " *W* " is vocabulary size.

2.5 Vectorization of Genres and Tags

2.5.1 Generating Genre Vectors: Utilizing the trained Word2Vec model to convert movie genres and tags into dense vectors. This transformation creates numerical representations for genres and tags, facilitating mathematical operations and comparisons.

2.5.2 Creating Movie Vectors: Computing average vectors for movies by aggregating their respective genre and tag vectors. This step generates comprehensive representations of movies in the vector space based on their constituent genres and tags.

	movieId	title	genres	tag	parameters	genre_vector
0	1	Toy Story (1995)	Adventure,Animation,Children,Comedy,Fantasy	Watched, computer animation, Disney animated f...	Adventure,Animation,Children,Comedy,FantasyWat...	[0.25716612, 0.23065197, 0.12113901, 0.1870318...
1	2	Jumanji (1995)	Adventure,Children,Fantasy	time travel, adapted from:book, board game, ch...	Adventure,Children,Fantasytime travel, adapted...	[0.2479726, 0.08946121, -0.11731894, 0.4538803...
2	3	Grumpier Old Men (1995)	Comedy,Romance	old people that is actually funny, sequel feve...	Comedy,Romanceold people that is actually funn...	[0.050889835, 0.018660491, 0.08484693, -0.1209...
3	4	Waiting to Exhale (1995)	Comedy,Drama,Romance	chick flick, revenge, characters, chick flick,...	Comedy,Drama,Romancechick flick, revenge, char...	[0.24785294, -0.0790424, 0.09795462, -0.270880...
4	5	Father of the Bride Part II (1995)	Comedy	Diane Keaton, family, sequel, Steve Martin, we...	ComedyDiane Keaton, family, sequel, Steve Mart...	[0.117320575, 0.037975755, 0.04198269, -0.1455...
...	...	...	...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy		Comedy	[0.11957534, 0.035187516, 0.18785544, -0.25898...
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy		Comedy	[0.11957534, 0.035187516, 0.18785544, -0.25898...
27275	131258	The Pirates (2014)	Adventure	bandits, Korea, mutiny, pirates, whale	Adventurebandits, Korea, mutiny, pirates, whale	[0.109227166, -0.0138491085, 0.040518932, -0.0...
27276	131260	Rentun Ruusu (2001)	(no genres listed)		(no genres listed)	[-5.589223e-05, 7.9788944e-05, -0.000186327, 0...
27277	131262	Innocence (2014)	Adventure,Fantasy,Horror		Adventure,Fantasy,Horror	[0.2636377, -0.0010528764, 0.018983325, -0.043...

27278 rows × 6 columns

Fig.7.Vectorization of ‘parameter’ as ‘genre_vector’

2.6 Recommendation System

2.6.1 Similarity Calculation: computing the cosine similarity between the movie vectors to compare them. Cosine similarity is used to measure how similar two movies are in terms of on their vector representations..

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Fig.8.Cosine Similarity between two vectors A and B.

2.6.2 Movie Recommendation Functions: Developing functions that leverage the calculated similarity to recommend movies. These functions suggest similar movies based on a given movie title or a user's watch history, enhancing user experience and engagement.

2.7 Identification of Highly Rated Movies

2.7.1 Filtering Highly Rated Movies: Identifying movies highly rated by users based on predefined thresholds. This step focuses on extracting movies with ratings surpassing a certain threshold to understand user preferences.

2.7.2 Recommending Similar Movies: Recommending movies similar to those highly rated by users. Leveraging the similarity metrics, the system suggests movies that align with the taste of users who have rated movies positively.

2.8 Output and Interaction

2.8.1 Output Generation:

Providing lists of recommended movies based on the implemented recommendation systems. These outputs offer users a curated selection of movies based on their preferences or a specific movie's characteristics.

2.8.2 User Interaction:

Designing functions that allow interaction with users watch histories and preferences. These functions facilitate personalized movie suggestions, enhancing user engagement and satisfaction.

This methodology here emphasizes the use of Word2Vec embeddings to represent movie features (genres and tags) in a vector space, enabling the calculation of similarities between movies for recommendation purposes. It employs cosine similarity as a measure to quantify the similarity between movie vectors, ultimately providing movie suggestions based on this similarity metric.

3. Discussion

➢ The Dataframe originally looks like:

```
[14]: movies
```

	movieId	title	genres	tag
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	[Watched, computer animation, Disney animated ...
1	2	Jumanji (1995)	Adventure|Children|Fantasy	[time travel, adapted from:book, board game, c...
2	3	Grumpier Old Men (1995)	Comedy|Romance	[old people that is actually funny, sequel fev...
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	[chick flick, revenge, characters, chick flick...
4	5	Father of the Bride Part II (1995)	Comedy	[Diane Keaton, family, sequel, Steve Martin, w...
...	...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy	[nan]
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy	[nan]
27275	131258	The Pirates (2014)	Adventure	[bandits, Korea, mutiny, pirates, whale]
27276	131260	Rentun Ruusu (2001)	(no genres listed)	[nan]
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror	[nan]

27278 rows × 4 columns

Figure 9. Original Dataframe.

➢ The Dataframe after joining all the required parameters into a single column:

```
[24]: movies
```

	movieId	title	genres	tag	parameters
0	1	Toy Story (1995)	Adventure,Animation,Children,Comedy,Fantasy	Watched, computer animation, Disney animated f...	Toy Story (1995),Adventure,Animation,Children,...
1	2	Jumanji (1995)	Adventure,Children,Fantasy	time travel, adapted from:book, board game, ch...	Jumanji (1995),Adventure,Children,Fantasy,time...
2	3	Grumpier Old Men (1995)	Comedy,Romance	old people that is actually funny, sequel feve...	Grumpier Old Men (1995),Comedy,Romance,old peo...
3	4	Waiting to Exhale (1995)	Comedy,Drama,Romance	chick flick, revenge, characters, chick flick,...	Waiting to Exhale (1995),Comedy,Drama,Romance,...
4	5	Father of the Bride Part II (1995)	Comedy	Diane Keaton, family, sequel, Steve Martin, we...	Father of the Bride Part II (1995),Comedy,Dian...
...	...	...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy		Kein Bund für's Leben (2007),Comedy,
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy		Feuer, Eis & Dosenbier (2002),Comedy,
27275	131258	The Pirates (2014)	Adventure	bandits, Korea, mutiny, pirates, whale	The Pirates (2014),Adventure,bandits, Korea, m...
27276	131260	Rentun Ruusu (2001)	(no genres listed)		Rentun Ruusu (2001),(no genres listed),
27277	131262	Innocence (2014)	Adventure,Fantasy,Horror		Innocence (2014),Adventure,Fantasy,Horror,

27278 rows × 5 columns

Figure 10. Updated Dataframe

➢ The Dataframe after finding the corresponding vectors for each movie:

```
[29]: movies
```

[29]:

	movieId	title	genres	tag	parameters	genre_vector
0	1	Toy Story (1995)	Adventure,Animation,Children,Comedy,Fantasy	Watched, computer animation, Disney animated f...	Toy Story (1995),Adventure,Animation,Children,...	[0.24317917, 0.25684926, 0.09635146, 0.0443296...
1	2	Jumanji (1995)	Adventure,Children,Fantasy	time travel, adapted from:book, board game, ch...	Jumanji (1995),Adventure,Children,Fantasy,time...	[0.11852338, 1.1255654, -0.5972555, -0.0623051...
2	3	Grumpier Old Men (1995)	Comedy,Romance	old people that is actually funny, sequel feve...	Grumpier Old Men (1995),Comedy,Romance,old peo...	[-0.16571058, -0.19244501, 0.1499135, -0.07413...
3	4	Waiting to Exhale (1995)	Comedy,Drama,Romance	chick flick, revenge, characters, chick flick,...	Waiting to Exhale (1995),Comedy,Drama,Romance,...	[-0.6350109, -0.1790633, 0.04793664, 0.4221134...
4	5	Father of the Bride Part II (1995)	Comedy	Diane Keaton, family, sequel, Steve Martin, we...	Father of the Bride Part II (1995),Comedy,Dian...	[0.16135903, 0.24405129, 0.092645004, -0.11805...
...	...	...	...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy		Kein Bund für's Leben (2007),Comedy,	[1.0687699, 0.26957676, -0.88112736, -0.234533...
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy		Feuer, Eis & Dosenbier (2002),Comedy,	[1.089233, 0.2506709, -0.8946547, -0.22491048,...
27275	131258	The Pirates (2014)	Adventure	bandits, Korea, mutiny, pirates, whale	The Pirates (2014),Adventure,bandits, Korea, m...	[0.30850446, 0.764403, -0.64108473, 0.6663325,...
27276	131260	Rentun Ruusu (2001)	(no genres listed)		Rentun Ruusu (2001),(no genres listed),	[-0.49922466, -0.2995272, 0.19498444, -0.37964...
27277	131262	Innocence (2014)	Adventure,Fantasy,Horror		Innocence (2014),Adventure,Fantasy,Horror,	[0.45099866, 2.3822725, 0.5468839, 1.7561836, ...

27278 rows × 6 columns

Figure 11. Dataframe with movie vectors added.

➢ Movie recommendations based on a one movie:

```
[67]: recommed('Harry Potter and the Deathly Hallows: Part 2 (2011)')

1) Harry Potter and the Chamber of Secrets (2002)
2) Lord of the Rings: The Two Towers, The (2002)
3) Lord of the Rings: The Fellowship of the Ring, The (2001)
4) Harry Potter and the Order of the Phoenix (2007)
5) Harry Potter and the Goblet of Fire (2005)
6) Lord of the Rings: The Return of the King, The (2003)
7) Tales from Earthsea (Gedo Senki) (2006)
8) Spiderwick Chronicles, The (2008)
9) Harry Potter and the Deathly Hallows: Part 1 (2010)
10) Chronicles of Narnia: Prince Caspian, The (2008)
```

Figure 12. Movie recommendations based on one movie.

- Movie recommendations based on a list of movies:

```
[75]: movs = ['Thor (2011)', 'Harry Potter and the Deathly Hallows: Part 2 (2011)', 'Clash of the Titans (1981)', 'Harry Potter and the Deathly Hallows: Part 1
recommend_user(movs) #hard coded
```

```
1) Sinbad: The Fifth Voyage (2014)
2) Perils of the Sentimental Swordsman (1982)
3) Conquest (1983)
4) Highlander: Endgame (Highlander IV) (2000)
5) Legend of the Eight Samurai (Satomi hakken-den) (1983)
6) The Golden Voyage of Sinbad (1973)
7) Curse of the Ring (Ring of the Nibelungs) (2004)
8) In the Name of the King 2: Two Worlds (2011)
9) In the Name of the King III (2014)
10) Ator, the Fighting Eagle (Ator l'invincibile) (1982)
11) Sex and Zen (Rou pu Tuan zhi tou Qing bao Jian) (1992)
12) Karate-Robo Zaborgar (Denjin Zabôgâ) (2011)
13) Holy Flame of the Martial World (Wu lin sheng huo jin)(1983)
14) Sorceress (1982)
15) Dragon Hunter (2009)
```

Figure 13. Movie recommendations based on a list of movies.

- Movie recommendation based on user's top-rated movies based on userId:

```
[65]: get_high_rated_movies(126) #give the userId as the parameter here
```

```
1) Sherlock Holmes: A Game of Shadows (2011)
2) Player, The (1992)
3) Nacho Libre (2006)
4) Female Trouble (1975)
5) Creator (1985)
6) Trading Places (1983)
7) Falling Down (1993)
8) Working Girl (1988)
9) Ladykillers, The (1955)
10) M*A*S*H (a.k.a. MASH) (1970)
11) Dead Men Don't Wear Plaid (1982)
12) Not on Your Life (Verdugo, El) (Executioner, The) (1963)
13) Moscow on the Hudson (1984)
14) Apartment, The (1960)
15) Casino Royale (1967)
```

Figure 14. Movie recommendations based on user's ratings.

4. Concluding Remarks

Content-based movie recommendation systems offer a personalized approach to movie suggestions, tailoring recommendations to each user's individual preferences and tastes based on their watch history. These systems are able to recognize trends and anticipate which movies a user might like by analyzing several aspects of movies, including genre, stars, directors, plot summaries, and user ratings.

Key advantages of content-based movie recommendation systems include:

- Personalization: The ability to tailor recommendations to each user's unique preferences, ensuring that suggestions are relevant and engaging.
- Diversity: the ability to broaden consumers' cinematic horizons by introducing them to fresh and fascinating films that they might not have otherwise found.

The development of this content-based movie recommendation system has several implications. Firstly, it demonstrates the feasibility of using movie attributes to make accurate recommendations. Second, it offers consumers a useful resource for finding new films that suit their interests. Thirdly, by investigating content-based filtering as a successful recommendation method, it advances the field of recommender systems as a whole.

Overall, the content-based movie recommendation system can significantly improve the way users discover and enjoy movies.

5. Future Work

A content-based movie recommendation system recommends items to users with help of the features of the items they have liked or interacted with in the past. Future works for improving content-based movie recommendation systems can include:

1.**Incorporating Textual Analysis**:

Analyze and incorporate textual information such as movie reviews, summaries, and user reviews. Natural Language Processing (NLP) techniques is very useful to extract meaningful information from text. Utilize sentiment analysis to understand the emotional context of reviews and better capture user preferences.

2.**Dynamic Recommendation**:

Implement techniques for real-time or near-real-time recommendation to adapt to changing user preferences. Explore the incorporation of temporal dynamics to capture how user preferences evolve over time.

3.**Hybrid Recommendation Systems:**

Combine content-based approaches with collaborative filtering or other recommendation techniques to create hybrid systems that leverage the strengths of multiple methods. Investigate the use of reinforcement learning for dynamically adapting the recommendation strategy based on user feedback.

4.**User Feedback Integration**:

Provide techniques for actively looking out and utilizing user feedback so that over time, recommendations will become more accurate and meaningful..

5.**Personalized Contextual Recommendations**:

To make recommendations that are more individualized and context-aware, take into account contextual elements like the user's mood, time of day, and location.

References

1. [Hastie, T., Tibshirani, R., & Friedman, J.], 2009. The Elements of Statistical Learning. Springer. https://doi.org/10.1007/978-0-387-84858-7

2. [Koren, Y., Bell, R., & Volinsky, C.], 2009. Matrix Factorization Techniques for Recommender Systems. IEEE Computer Society. 42(8), 30-37. https://doi.org/10.1109/MCSE.2009.33

3. [Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J.], 2013. Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems. 26. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

4. [Harper, F. M., & Konstan, J. A.], 2016. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS). 5(4), Article 19. https://doi.org/10.1145/2898361