University of South Wales

BUILDING A PERSONALIZED MOVIE RECOMMENDER SYSTEM

BY

MUHAMMED ADEBISI

September 2023

INTRODUCTION

A **Movie Recommendation System** is a data-driven technology that suggests movies to users based on their preferences, viewing history, and behaviour.

A recommendation system is a system that uses data from the user to make recommendations of relevant content to the user (Falk, 2019). Movie recommendations are algorithms used to make movie suggestions to users based on their preferences and viewing history. The aim of these systems is to make recommendations which are personalized to suit the interests of the user to improve the movie-watching experience (Burke, 2007).

The rapid explosion of information thanks to the internet has made it necessary to create technologies which filter the data and direct users to only suitable and relevant information. The idea of recommendation systems emanated in the mid-90s as a means of offering a user interesting items using his profile information (Ortega et al., 2013). Over the last few decades, there have been several recommendation systems developed using a variety of approaches.

University of South Wales

INTRODUCTION

KEY COMPONENTS

1.User Profile:

- The system collects data on user preferences, including genre, actors, directors, and past viewing history.

2.Movie Database:

- A vast database containing information about movies, including their genres, ratings, release years, and more.

3.Machine Learning Algorithms:

- Sophisticated algorithms process user data and movie information to generate personalized recommendations.

4.Recommendation Engine:

- The heart of the system, which calculates and ranks movie suggestions for each user.

University of South Wales

BENEFITS

•**Personalization**: Users receive movie suggestions tailored to their tastes.

•**Increased Engagement**: Keeps users engaged and encourages more content consumption.

•**Revenue Generation**: Can lead to increased sales and subscriptions for streaming platforms.

CHALLENGES

1.Cold Start Problem:

- Difficulty in recommending movies to new users with limited data.

2.Scalability:

- Handling large user bases and movie catalogues can be challenging.

3.Privacy Concerns:

- Collecting and storing user data must be done with privacy in mind.

University of South Wales

DOMAIN ANALYSIS

TYPES OF RECOMMENDATION SYSTEMS

Collaborative filtering systems make their recommendations using the ratings of users who have similar preferences to the user. They are based on the principle that if two users have had similar preferences or ratings of a movie previously, they are likely to have the same ratings for movie later. Collaborative filtering recommendations are made based on the similarity of the users' preferences (Sarwar et al., 2001).

Content-based filtering systems make recommendations based strictly on the user's preferences in the past by selecting movies which are similar. It relies heavily on item representation and uses the movie attributes to identify similar movies. They operate on the principle that if a user has shown a positive reception to a movie in the past, they would enjoy a movie with similar features (Pazzani & Billsus, 2007).

Hybrid systems combine both approaches, using their strengths and overcoming their weaknesses to make their recommendations. The combination of the strengths of these approaches helps to make the recommendations more accurate (Burke, 2002). This combination to produce hybrid systems can be done in different ways such as the mixed, cascade, feature combination, weighted and switching methods (Burke, 2002).

TYPES OF RECOMMENDATION SYSTEMS

In summary

1.Collaborative Filtering:

- Recommends movies based on the preferences and behaviours of similar users.

2.Content-Based Filtering:

- Recommends movies similar to those a user has previously liked.

3.Hybrid Models:

- Combine collaborative and content-based approaches for improved accuracy.

OBJECTIVE OF STUDY

- The project involves implementing a recommendation system that combines collaborative filtering and content-based filtering, evaluating various machine learning algorithms, preprocessing and analyzing the MovieLens Dataset, creating a user-friendly website for user interaction, and assessing the recommendation system's performance.

University of South Wales

HOW IT WORKS

1.Data Collection:

- User interactions (ratings, likes, searches) are gathered to build user profiles.

2.Data Preprocessing:

- Data is cleaned, transformed, and prepared for machine learning algorithms.

3.Feature Engineering:

- User and movie data are transformed into numerical features for modeling.

4.Machine Learning Models:

- Algorithms like collaborative filtering, content-based, or hybrid models are used to make predictions.

5.Recommendation Generation:

- The system suggests movies by predicting user preferences and ranking them accordingly.

PYTHON IN DATA ANALYSIS

Python is a valuable choice for this project because it is a widely-used programming language in data analysis, known for its simplicity, versatility, and numerous libraries. Python's libraries like Pandas, Seaborn, NumPy, and Matplotlib provide essential tools for data manipulation and visualization. Additionally, Python offers specialized libraries like Surprise and Implicit for recommendation system development, with algorithms like collaborative filtering and implicit feedback support. Moreover, Python's machine learning libraries like Scikit-learn and TensorFlow enable the implementation of content-based and hybrid recommendation systems, providing a wide array of algorithms and deep learning models for feature extraction and preference prediction.

ABOUT THE DATASET

The MovieLens 25M dataset is a popular and widely used dataset in the field of recommendation systems and movie-related research. It contains a substantial amount of data related to movie ratings and user interactions. Here's some key information about the MovieLens 25M dataset:

1. Dataset Size:

•The "25M" in the dataset's name indicates that it contains 25 million ratings and other interactions. This makes it a large and comprehensive dataset for movie recommendation research.

2. Data Types:

•The dataset includes various types of data, including movie ratings, user IDs, movie IDs, timestamps, and user-generated tags.

3. Ratings:

•Users in the dataset have provided ratings (usually on a scale of 1 to 5) for movies they've watched. These ratings are a fundamental component for building recommendation systems.

4. User IDs and Movie IDs:

•Each user and movie is identified by a unique ID, allowing researchers to track user preferences and movie details.

University of South Wales

ABOUT THE DATASET

5. Timestamps:

•Timestamps are included to indicate when a user rated or interacted with a movie. This temporal information can be useful for understanding how user preferences change over time.

6. Tags:

•Users can also provide tags to describe movies. These tags can be used for content-based recommendation or additional context.

7. Diversity of Movies:

•The dataset covers a wide range of movies from different genres, release years, and popularity levels. This diversity makes it suitable for various recommendation research scenarios.

8. Use Cases:

•The MovieLens 25M dataset is commonly used for training and evaluating recommendation algorithms, such as collaborative filtering, content-based filtering, matrix factorization, and deep learning-based models.

9. Research and Development:

•Researchers and data scientists use this dataset to develop and benchmark recommendation algorithms, conduct experiments, and study user behaviour in recommendation systems.

University of South Wales

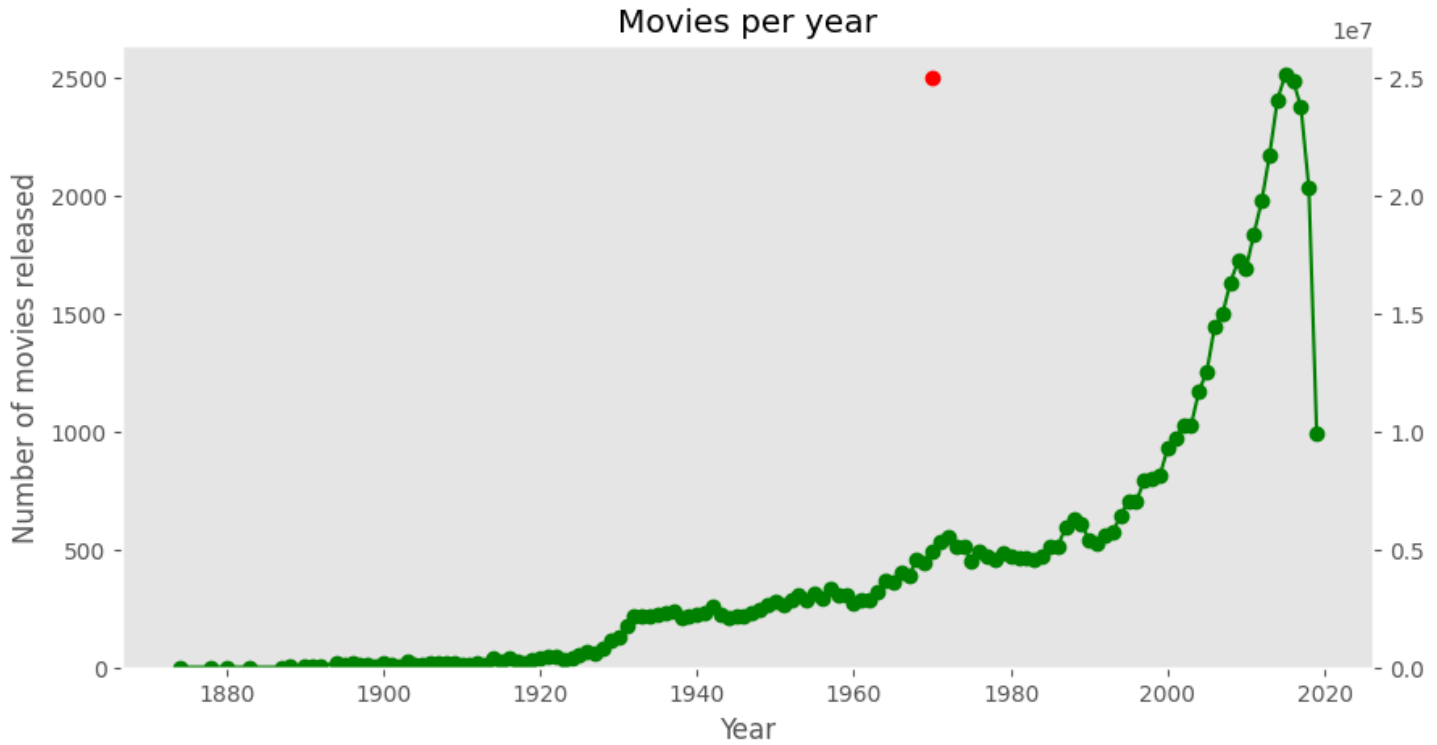
ABOUT THE DATASET

10. Data Exploration: - Researchers can perform exploratory data analysis to gain insights into user preferences, movie popularity, and other factors influencing recommendations.
11. Availability: - The MovieLens 25M dataset is freely available for research purposes and can be downloaded from the MovieLens website.
In summary, the MovieLens 25M dataset is a valuable resource for building and evaluating recommendation systems. Its large size, diverse movie selection, and inclusion of user ratings and tags make it a rich source of data for research in the field of recommendation systems and movie-related analytics.

University of South Wales

VISUALIZATIONS FROM THE DATASET

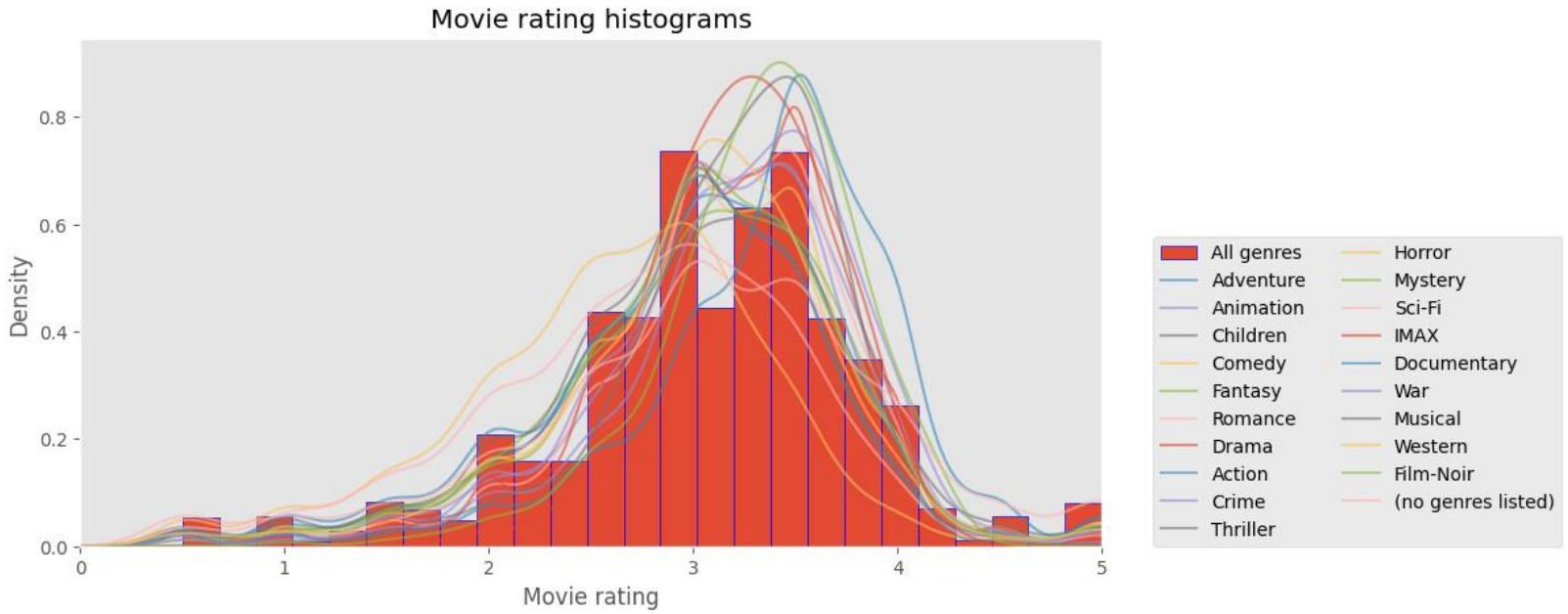
- The line plot below shows the trend in movies per year from 1880 to 2021.
- We can notice a significant increase in movies per year and then a decrease between 2019 and 2021 which can be attributed to COVID pandemic.



University of South Wales

VISUALIZATIONS FROM THE DATASET

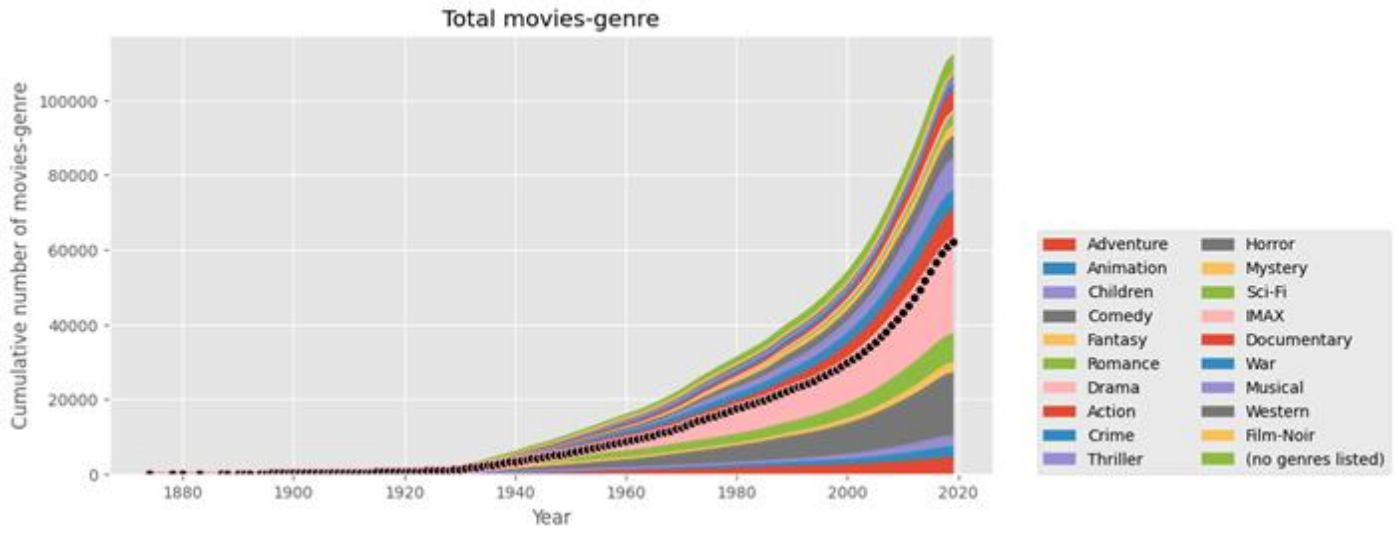
- The histogram below shows the ratings of movies.
- It is noticeable that the 2.9 and 3.4 (most common) have the highest ratings in the dataset
- Some genres have higher ratings on average



University of South Wales

VISUALIZATIONS FROM THE DATASET

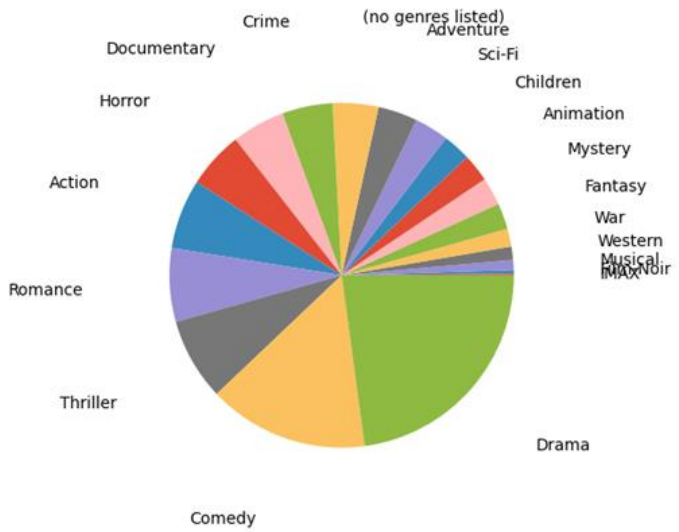
- The cumulative plot below shows the total number of genres in movies per year from 1880 to 2021.
- Sci-fi & Romance have an increasing cumulative over the years which means they have more movies when cumulated.



University of South Wales

VISUALIZATIONS FROM THE DATASET

- The pie chart plot illustrates that the Drama, Comedy, Thriller and Romance genres make up for more than 50% of the movie genres produced.
- Drama genre has the most movies produced followed by the Comedy Genre



University of South Wales

STEPS TO TAKE FOR THE RECOMMENDATION SYSTEM

- Cleaning of the Tags
- Analysing and modelling of the text using TF-IDF vectorization
- Creating a function to search using COSINE SIMILARITY and THE SEARCH FUNCTION
- Facilitation of user interaction with the program called USER INTERACTION WITH IPYTHON WIDGETS
- Combining RECOMMENDATIONS from RATINGS and TAGS
- Calculating RECOMMENDATION PERCENTAGES based on RATINGS
- Calculating RECOMMENDATION PERCENTAGES based on TAGS
- Combining RECOMMENDATIONS from RATINGS and TAGS by combining recommendation percentages from both RATINGS and TAGS
- SORTING and RETURNING RECOMMENDATIONS
- Final RECOMMENDATIONS implemented

University of South Wales

RESULTS

OPERATION: Testing that the clean_title function creates a column with titles containing only alphanumeric characters

EXPECTED RESULT: A column titled "Clean Title" added to the movies dataframe and contain the cleaned movie title

ACTUAL RESULT: The "Clean Title" was created and the values in the column were indeed cleaned movie titles as seen in figure 7.

```
def clean_title(title):
    title = re.sub("[^a-zA-Z0-9 ]", "", title)  # This removes any characters that are not alphanumeric or spaces from the 'title' string
    return title # This returns the cleaned 'title' string
```

[10] ✓ 0.0s

```
# This adds a new column called "clean_title" to the 'movies' DataFrame
# The values in this column are obtained by applying the 'clean_title' function to the values in the "title" column
movies["clean_title"] = movies["title"].apply(clean_title)
movies.head()
```

[14] ✓ 0.4s

	movieId	title	genres	clean_title
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	Toy Story 1995
1	2	Jumanji (1995)	Adventure|Children|Fantasy	Jumanji 1995
2	3	Grumpier Old Men (1995)	Comedy|Romance	Grumpier Old Men 1995
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	Waiting to Exhale 1995
4	5	Father of the Bride Part II (1995)	Comedy	Father of the Bride Part II 1995

University of South Wales

RESULTS

OPERATION: Testing that the clean_title column is indeed used to create a TF-IDF matrix to enable the recommendation system function.

EXPECTED RESULT: A TF-IDF matrix will be formed with values representing the movie titles and their frequency of appearance in the dataframe.

ACTUAL RESULT: A TF-IDF matrix is formed with values representing the movie titles and their frequency of appearance in the dataframe as seen in below.

```
# This imports the TfidfVectorizer class from the sklearn.feature_extraction.text module.
from sklearn.feature_extraction.text import TfidfVectorizer

# This initializes an instance of the TfidfVectorizer class, named "vectorizer".
# This means that the vectorizer will consider both single words(unigrams) and pairs of words (bigrams)
vectorizer = TfidfVectorizer(ngram_range=(1,2))

# This applies the fit-transform method of the vectorizer to the "clean_title" column of the movies dataframe.
# This will convert the text data into a matrix of TF-IDF features
# Each row of the matrix represents a movie title, and each column represents a unique word or bigram.
# The values in the matrix represent the TF-IDF scores for each word or bigram in each movie title
tfidf = vectorizer.fit_transform(movies["clean_title"])
tfidf.shape
```

[20] ✓ 3.0s

```
(62423, 170073)
```

University of South Wales

RESULTS

OPERATION: Testing that the recommendation percentages are calculated for both the ratings and the tags.

EXPECTED RESULT: A dataframe “rec_percentages” dataframe will be created showing the different recommendation percentages

ACTUAL RESULT: A dataframe “rec_percentages” dataframe is created showing the different recommendation percentages as seen in figure below.

	similar_ratings	all_ratings	similar_tags	all_tags
89745	1.00	0.039003	NaN	NaN
26764	1.00	0.000026	NaN	NaN
589	1.00	0.131490	NaN	NaN
59315	1.00	0.052954	NaN	NaN
1240	1.00	0.079131	NaN	NaN
...	...	...	...	...
70599	0.25	0.003024	NaN	NaN
70361	0.25	0.000071	NaN	NaN
70301	0.25	0.000103	NaN	NaN
70183	0.25	0.002145	NaN	NaN
7153	0.25	0.168393	NaN	NaN

University of South Wales

RESULTS

OPERATION: Testing that the recommendation list generated using only the Movie titles and the recommendation list

EXPECTED RESULT: The two recommendation lists generated will contain different movies

ACTUAL RESULT: The recommendation list generated from the Movie Title only approach is different from the recommendation list generated from the final implementation involving Movies, Tags and Ratings as seen in the figures below.

Note: The Movie used for this test is “The Matrix.”

Movie Title: The Matrix

	movieId	title	genres	clean_title
2480	2571	Matrix, The (1999)	Action|Sci-Fi|Thriller	Matrix The 1999
46345	172255	The Matrix Revisited (2001)	Documentary	The Matrix Revisited 2001
28793	132490	Return to Source: The Philosophy of The Matrix...	Documentary	Return to Source The Philosophy of The Matrix ...
49723	179489	The Living Matrix (2009)	Documentary	The Living Matrix 2009
6247	6365	Matrix Reloaded, The (2003)	Action|Adventure|Sci-Fi|Thriller|IMAX	Matrix Reloaded The 2003

University of South Wales

RESULTS

Final Recommendation using Movie Titles, Ratings and Tags

An interactive movie recommendation widget was then created to allow users to input a movie title and receive recommendations for similar movies that the user might like.

```
# This creates a text input widget for the movie title
movie_name_input = widgets.Text(
    value='Transformers',
    description='Movie Title:',
    disabled=False
)

#This creates an output widget to display the recommendations
recommendation_list = widgets.Output()

# This creates a function to handle the text input
def on_type(data):
    with recommendation_list:
        recommendation_list.clear_output()
        title = data["new"]
        if len(title) > 5:
            results = search(title) # This performs a search based on the movie title
            movie_id = results.iloc[0]["movieId"] # This displays the similar movies based on the search results
            display(find_similar_movies(movie_id))

movie_name_input.observe(on_type, names='value') # This conncects the event handler to the text input widget

display(movie_name_input, recommendation_list)  # This displays the text input widget and the output widget
```

Enter Your Movie Title and Get Recommendations of Similar Movies that I'm sure you would like

Movie Title: The Matrix

	score	title	genres
5337	2.674428	Minority Report (2002)	Action|Crime|Mystery|Sci-Fi|Thriller
1523	2.394242	Men in Black (a.k.a. MIB) (1997)	Action|Comedy|Sci-Fi
10002	2.371422	Batman Begins (2005)	Action|Crime|IMAX
5310	2.354373	Bourne Identity, The (2002)	Action|Mystery|Thriller
12324	2.283005	Iron Man (2008)	Action|Adventure|Sci-Fi
11738	2.281462	Bourne Ultimatum, The (2007)	Action|Crime|Thriller
2451	2.233691	Lock, Stock & Two Smoking Barrels (1998)	Comedy|Crime|Thriller
7299	2.217214	Kill Bill: Vol. 2 (2004)	Action|Drama|Thriller
4857	2.176924	Ocean's Eleven (2001)	Crime|Thriller
1013	2.176718	Die Hard (1988)	Action|Crime|Thriller

University of South Wales

CONCLUSION

It was therefore concluded that from this project, an effective movie recommendation system was implemented using the Python programming language and its libraries such as Pandas, Numpy, Scikit-Learn and ipywidgets, the MovieLens Dataset, and an unorthodox use of cosine similarities for the calculation of recommendation percentages. This project demonstrated a new approach to solving the cold start problem and generating movie recommendations.

FUTURE WORKS AND RECOMMENDATIONS

Also, due to time constraints, not all files in the MovieLens 25m dataset were used in building the recommendation system. Three of the six files in the dataset were unused i.e., the Genome-tag, Genome-Scores, and the Links files. Using the genome-tags and genome-scores creates a student project opportunity for further collaboration to develop the recommendation system further. The original source code of this project could be further extended to contain a vast number of algorithms necessary to create recommendations that surpass even the current industry standards.

A web application could also be developed to allow more users interact with the recommendation system without having to load or setup Python environments or Jupyter Notebooks. This would expand the accessibility of the system, allowing usage of a wider range of devices. To maintain a continuously evolving and efficient system, I would also like to evaluate the performance of the system on a regular basis using feedback from surveys and questionnaires submitted by the users of this research soon.

University of South Wales

THANK YOU

University of South Wales