Applied Machine Learning at Scale (814)

Recommender System

Preston Govender

17560217

Proof of Concept: Report

Contents

Abstract

This report provides a high-level overview of the experiments conducted in the development of a recommender system. The experiments demonstrate the feasibility of building a scalable recommender system. The achieved results indicate the effectiveness of the system, with an RMSE of 0.35 on the training set for experiments conducted on a 100k dataset and an RMSE of 0.65 on the training set for experiments conducted on a 2.5 million dataset. The model's performance was further evaluated on unseen data, specifically the test set, resulting in an RMSE of 0.78, indicating satisfactory model performance. Furthermore, the inclusion of genre weights in the model provided a significant improvement, resulting in an RMSE of 0.72.

Notably, this proof of concept (POC) report omits discussion on prediction response time and integration aspects..

Introduction

Movie recommendation systems are widely used in many online platforms like Netflix, Amazon Prime, Hulu, ShowMax, and Disney Plus. These systems are designed to provide personalised movie recommendations to users based on their viewing history, ratings, and preferences.

From a business perspective implementing a movie recommendation system on a movie website holds significant importance. These systems offer personalised movie recommendations based on user preferences, leading to an enhanced user experience. By providing tailored suggestions, movie websites can drive user retention and engagement, as users are more likely to stay active and explore a wider range of content.

The recommendations also increase user interaction and consumption, positively impacting sales and revenue generation. Moreover, offering superior recommendation capabilities sets a movie website apart from competitors in the crowded streaming industry, attracting new users and fostering loyalty.

Additionally, recommender systems generate valuable user insights and data, empowering businesses to make data-driven decisions.

This document presents a summary of the experiments conducted in two notebooks, namely, *100k_dataset.ipynb* and *25m_dataset.ipynb*. It provides a comprehensive analysis of the results obtained and offers recommendations based on those findings. Moreover, from a technical standpoint, this document serves as a proof of concept (POC) demonstrating the feasibility of addressing the business case. By showcasing the successful implementation of the experiments, it establishes the viability of the proposed solution and its potential to deliver desired outcomes.

Problem Statement

Our objective is to develop a movie recommender system using collaborative filtering techniques applied to the MovieLens dataset. The dataset comprises two subsets: the first containing 100,000 ratings from 1,000 users for 1,700 movies, and the second containing 25 million ratings from 270,000 users for over 62,000 movies. Our aim is to create a system that accurately predicts how a user will rate a movie they haven't seen yet and recommends movies with the highest predicted ratings.

To achieve this, we will employ the maximum likelihood estimation (MLE) method, a widely used statistical technique for estimating model parameters based on observed data likelihood. In our case, the model will be trained on the MovieLens dataset, and the parameters will represent user preferences and movie features.

Our approach will be iterative, starting with a simple model and progressively incorporating additional complexity. The initial model will include basic bias terms for users and movies, along with user-movie interactions. Subsequently, we will enhance the model by incorporating genre as a feature, utilising cosine or Jaccard similarity measures to improve its performance.

Exploratory Data Analysis (EDA)

In this section, we will delve into the exploratory data analysis (EDA) phase of the project. Before proceeding with the experiments, it is essential to gain a comprehensive understanding of the data. Through this process, we aim to make informed decisions regarding data processing and experimentation steps.

During the EDA phase, we will examine various aspects of the dataset, including its structure, statistical properties, and distribution of key variables. By conducting a thorough exploratory analysis, we can uncover patterns, trends, and potential challenges present within the data. This information will serve as a foundation for refining our data preprocessing strategies and guiding our subsequent experiments effectively.

Because the EDA is quite extensive and has been done separately on both datasets, we will provide a concise summary of the key findings and insights obtained. This summary will capture the most important takeaways and learnings derived from our exploratory analysis.

3.1 The Data

For this POC we work with two datasets, namely, *movies.csv* and *ratings.csv*.

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

(a) *movies.csv*

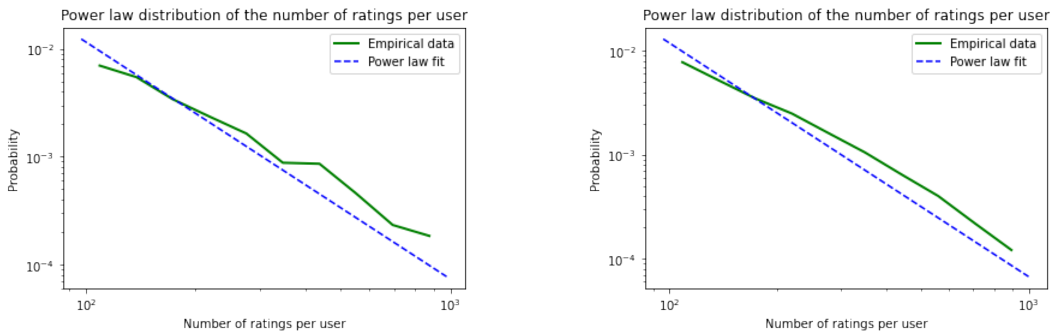
userId	movieId	rating	timestamp
1	1	4.0	964982703
1	3	4.0	964981247
1	6	4.0	964982224
1	47	5.0	964983815
1	50	5.0	964982931

(b) *ratings.csv*

Figure 3.1: Tables for the movies dataset and the ratings dataset

The ratings table serves as a primary source of key data used for training our system. It contains essential information such as user ID, movie ID, and the corresponding rating for each movie the user has watched. Additionally, this dataset includes a timestamp indicating when the user provided the rating for the movie.

3.1.1 Power Law's



(a) Power law on the 100k dataset

(b) Power law on the 2.5M dataset

Figure 3.2: Power law distribution of the number of ratings per user for both datasets

Based on the power law plots of the rating distributions generated above, it appears that there is evidence of a power law in the data. A power law is characterized by a straight line on a log-log plot, and the plot we generated shows a linear relationship between the log of the rating values and the log of their corresponding frequency counts.

However, it's important to note that the presence of a power law in a dataset is not always conclusive evidence of a specific underlying mechanism or process. Power laws are often observed in many different types of natural and social phenomena, and it's possible for a dataset to exhibit a power law distribution even if the underlying mechanism is not truly a power law. In addition, the presence of noise or other confounding factors can also affect the accuracy and interpretation of the power law.

To confirm whether the data follows a power law distribution, we perform a goodness-of-fit test, to compare the distribution of the data to a power law distribution.

The goodness-of-fit tests returns the log-likelihood ratio statistic and the p-value of the test.

In the case of the 2.5M dataset, the log-likelihood ratio statistic is -3698 and the p-value is 0.0. In the case of the 100k dataset, the log-likelihood ratio statistic is -14.5 and the p-value is 0.0005.

The log-likelihood ratio statistic is a measure of the relative goodness of fit of the two distributions being compared (in this case, a power law distribution and a log-normal distribution). A negative value of the statistic indicates that the power law distribution fits the data better than the log-normal distribution.

The p-value is a measure of the strength of evidence against the null hypothesis, which is that the power law distribution and the log-normal distribution fit the data equally well. A p-value less than 0.05 (or some other prespecified significance level) indicates that there is strong evidence against the null hypothesis and that the power law distribution fits the data significantly better than the log-normal distribution. (The null hypothesis is that the log-normal distribution is a better fit to the data).

In summary, the outputs $(-3698, 0.0)$ and $(-14.5, 0.0005)$ suggest that the power law distribution is a significantly better fit to the data than the log-normal distribution, based on the log-likelihood ratio test at a significance level of 0.05.

3.1.2 Scale-free Behaviour in the Data

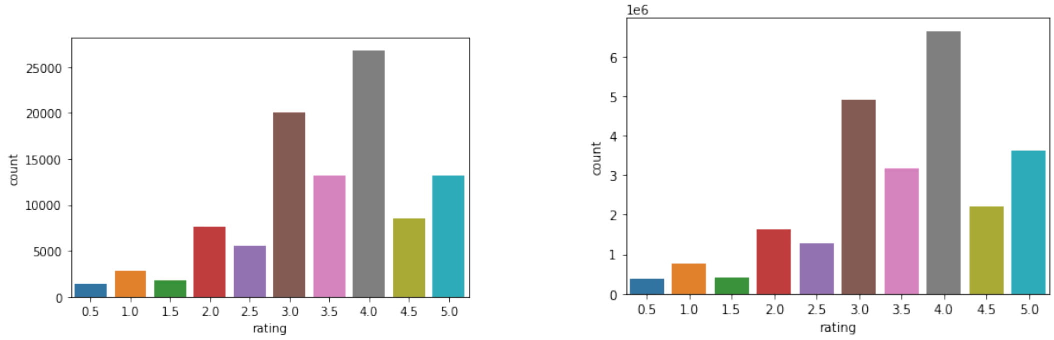
The presence of the power law in the rating distribution would suggest that the distribution of ratings across movies follows a scale-free pattern. In other words, a few movies would receive a large number of ratings, while the majority of movies would receive relatively few ratings.

Similarly, if a power law distribution is observed in the user activity data, it would suggest that the distribution of user ratings follows a scale-free pattern. This means that a few users would rate a large number of movies, while the majority of users would rate relatively few movies.

The observation of a power law distribution in the ratings data or user activity data could have implications for the design of recommender systems based on this dataset. Specifically, it suggests that a few movies and users have a disproportionately large influence on the overall ratings distribution, and therefore their ratings should be given greater or smaller weight in any recommendation algorithm depending on what we are trying to achieve.

3.1.3 More Dataset Plots

Rating Distribution



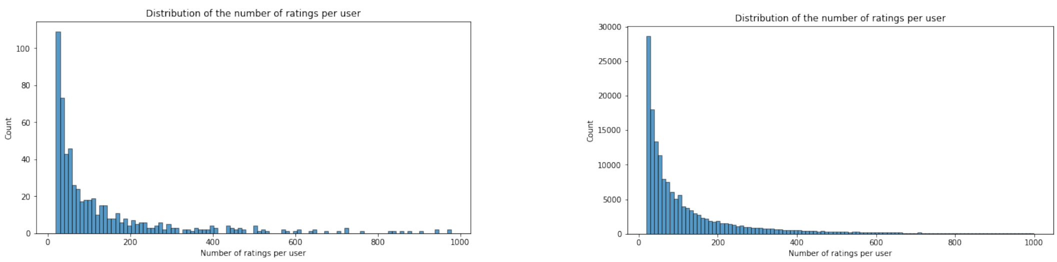
(a) Rating distributions on the $100k$ dataset.

(b) Rating distributions on the $2.5M$ dataset.

Figure 3.3: Rating distribution on the respective datasets.

The distribution of ratings reveals an interesting trend: movies tend to receive relatively fewer low ratings. Users have a tendency to assign 4-star ratings frequently, and they often rate movies higher than the average rating. This observation may indicate that users are more inclined to refrain from rating movies they dislike, while they feel more comfortable providing ratings for movies they enjoy. This behavior suggests a potential positivity bias in the rating patterns, where users are more likely to express positive sentiments through their ratings.

Ratings Per User Distribution



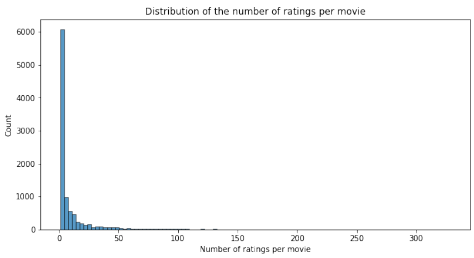
(a) Distribution of the number of ratings per user on the $100k$ dataset.

(b) Distribution of the number of ratings per user on the $2.5M$ dataset.

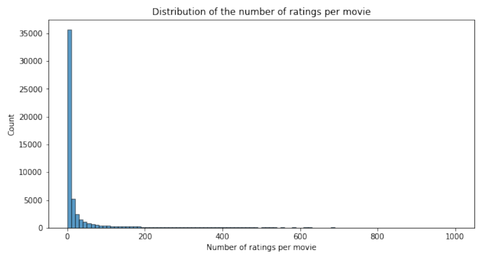
Figure 3.4: Distribution of the number of ratings per user.

When we compare the distribution of the number of ratings per user, we find further evidence supporting our earlier intuition. The analysis reveals that a smaller proportion of users rate a significant number of movies, while a larger proportion of users rate a relatively smaller number of movies. This observation strengthens the notion that a subset of users actively engages in rating a wide range of movies, while the majority of users contribute ratings for a limited selection of films.

Ratings Per Movie Distribution



(a) Distribution of the number of ratings per movie on the 100k dataset.

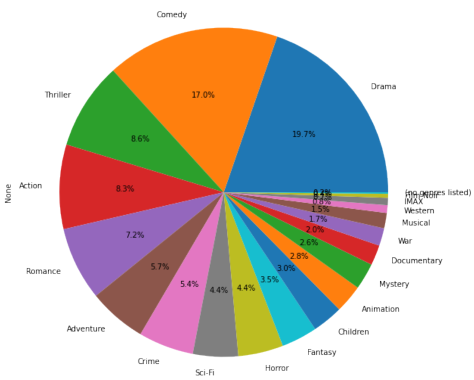


(b) Distribution of the number of ratings per movie on the 2.5M dataset.

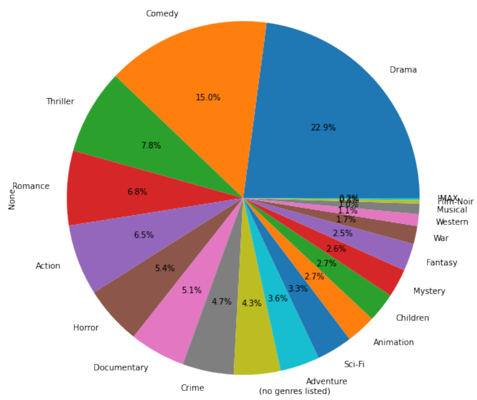
Figure 3.5: Distribution of the number of ratings per movie.

Examining the distribution of the number of ratings per movie reinforces our initial intuition. The distribution shows that a smaller subset of movies receives a significant number of ratings, while a larger number of movies receive relatively fewer ratings. This distribution aligns with our expectations, highlighting the fact that certain movies attract a higher level of user engagement and garner a larger number of ratings, while a larger pool of movies receive comparatively fewer ratings from users.

Genre Distribution

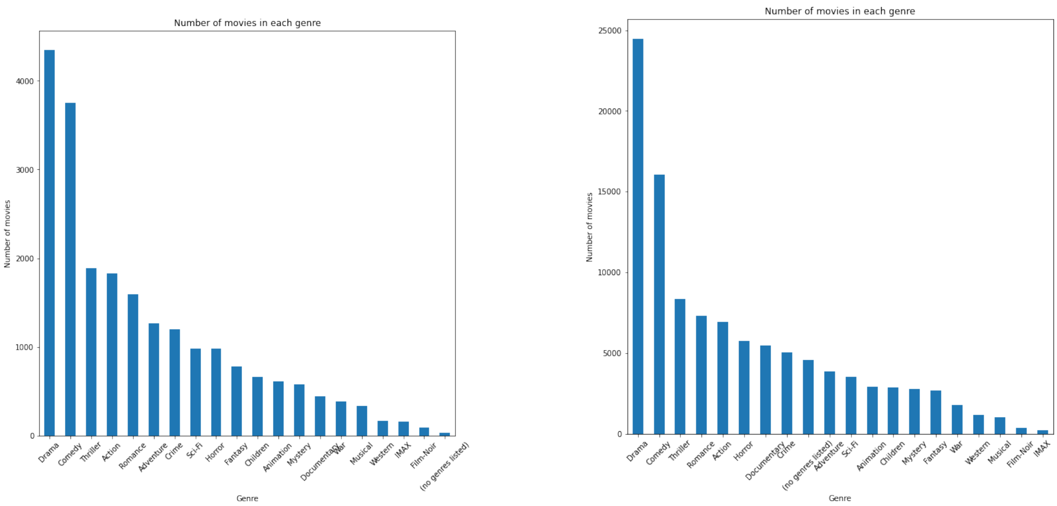


(a) Pie chart representing the genre distribution on the 100k dataset.



(b) Pie chart representing the genre distribution on the 2.5M dataset.

Figure 3.6: Pie chart representing the genre distribution.



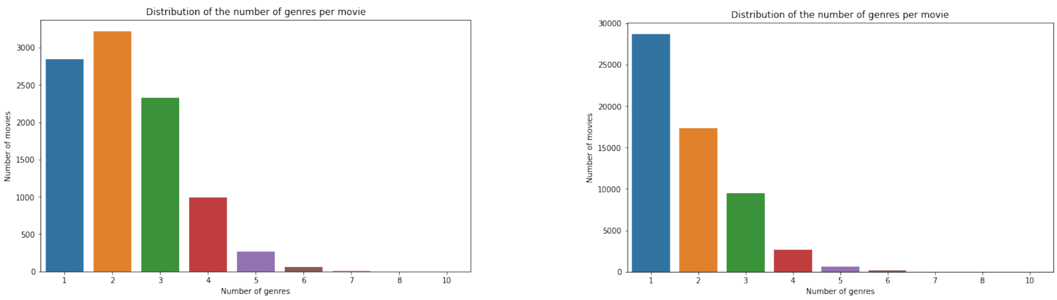
(a) Bar chart representing the genre distribution on the 100k dataset.

(b) Bar chart representing the genre distribution on the 2.5M dataset.

Figure 3.7: Bar chart representing the genre distribution.

The analysis of genre distribution further reveals a pronounced imbalance in the dataset. The genres of drama and comedy emerge as the most popular categories. However, the examination of bar plots also uncovers a number of movies with no listed genres, indicating missing or incomplete genre information. Additionally, we see that genres at the tail end of the bar plots are relatively infrequent in the datasets, suggesting a comparatively smaller representation of such movies in the dataset.

Distribution of Number of Genres per Movie



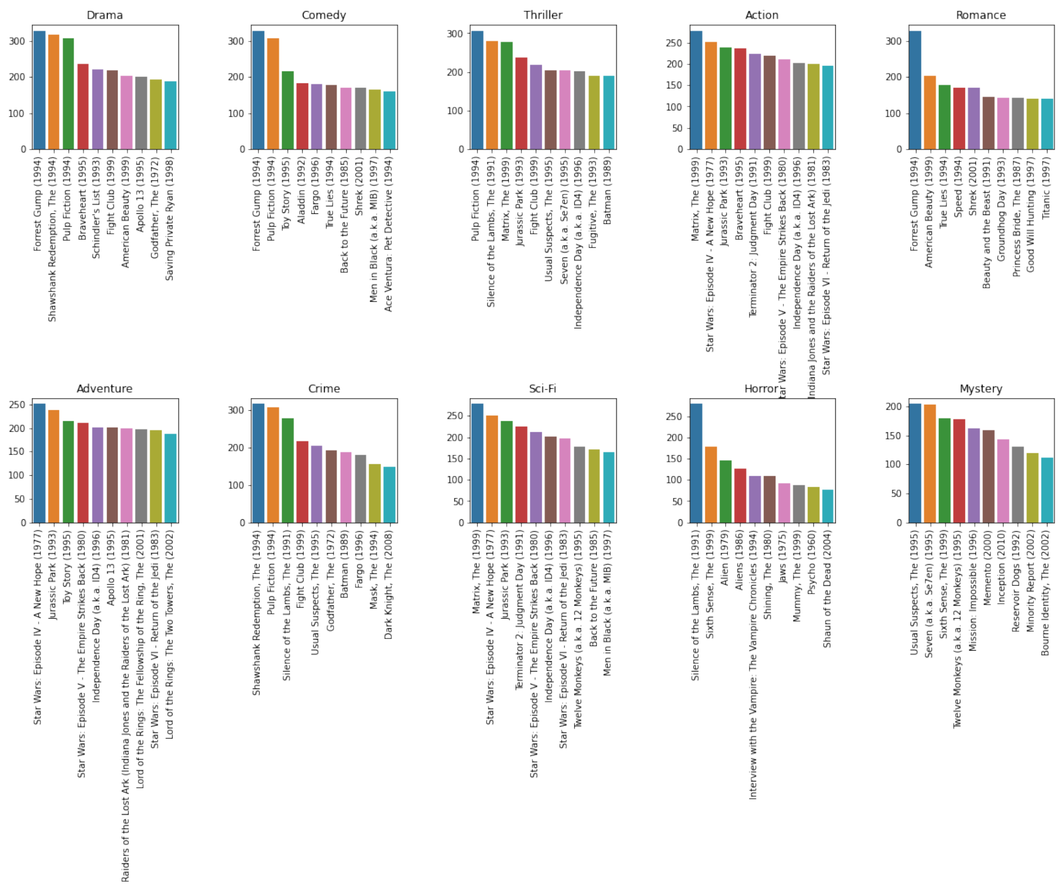
(a) Distribution of the number of genres per movie on the 100k dataset.

(b) Distribution of the number of genres per movie on the 2.5M dataset.

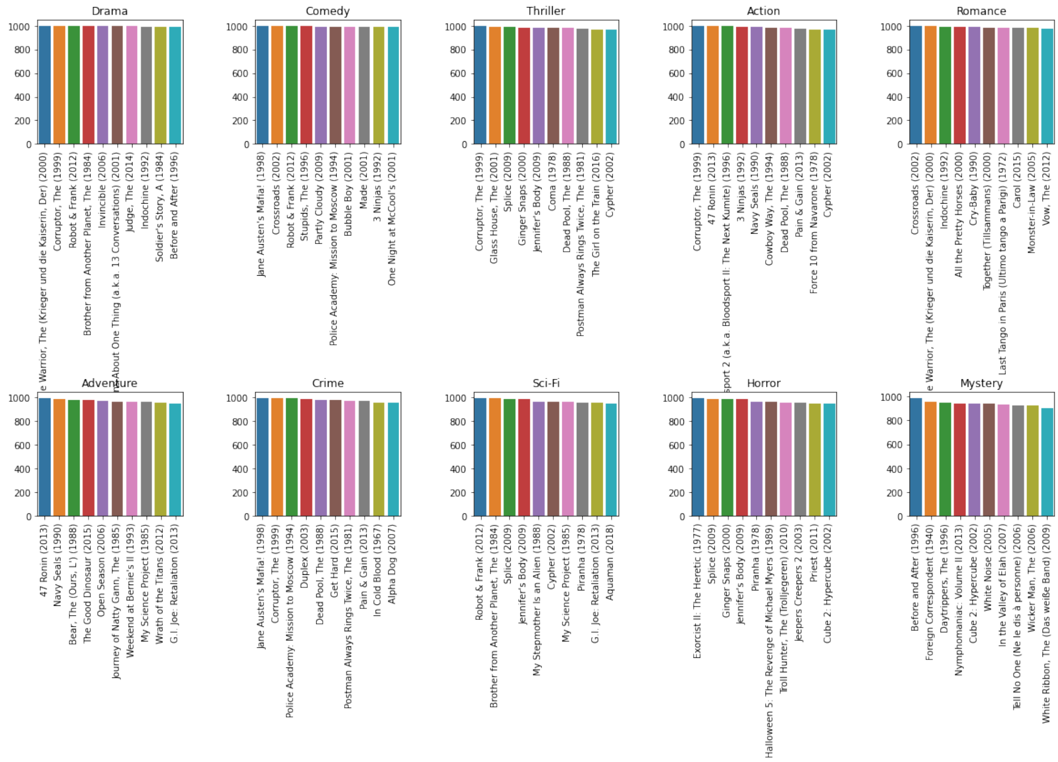
Figure 3.8: Distribution of the number of genres per movie.

Above we see the distribution of how many genres are allocated to a movie, per movie. We conducted an analysis to examine the distribution of the number of genres allocated to each movie. Figure 3.8, shows that the majority of movies are associated with approximately two genre tags, while only a few movies have more than four genres assigned to them. This exploration was driven by curiosity, aiming to gain insights into the diversity and categorisation of genres within the dataset.

Distribution of Top 10 Movies in Terms of Number of Ratings Received for the Top 10 Genres



(a) Distribution of top 10 movies for the top 10 genres on the 100*k* dataset.

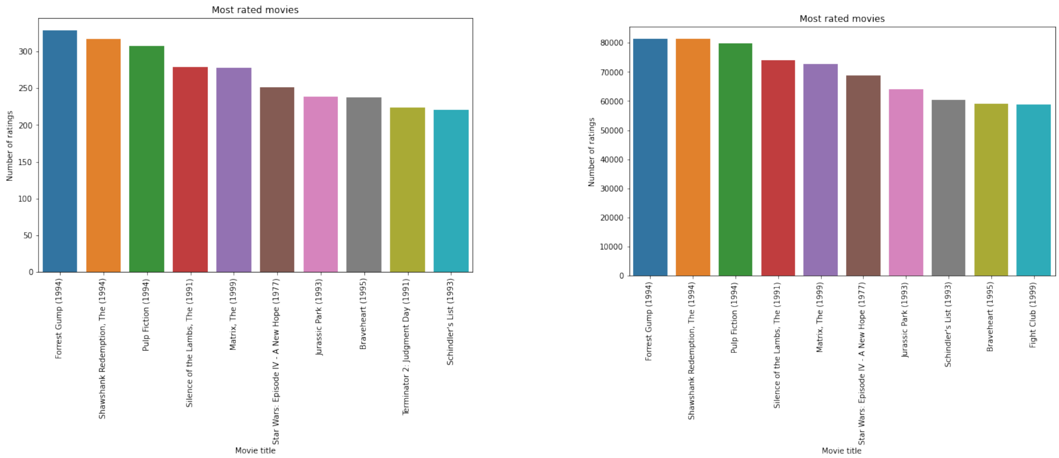


(b) Distribution of top 10 movies for the top 10 genres on the 2.5M dataset.

Figure 3.10: Distribution of top 10 movies in terms of number of ratings received, for the top 10 genres.

Figures 3.10 (a) and (b) show the top 10 movies for each of the top 10 genres. This analysis was conducted out of curiosity, aiming to explore the intersectionality of movie genres and identify movies that stand out within multiple genre categories. Notably, our findings reveal that several movies appear as top-rated choices across multiple genre categories, indicating their broad appeal and ability to resonate with diverse audience preferences.

Most Rated Movies



(a) The top 10 most rated movies on the 100k dataset.

(b) The top 10 most rated movies on the 2.5M dataset.

Figure 3.11: The top 10 most rated movies.

Out of curiosity we plot the top 10 most-rated movies from both datasets. Interestingly, our findings demonstrate an overlap, as both datasets share the same top-rated movies, with the exception of Terminator 2 and Fight Club. This observation highlights the consistency in users' preferences and the enduring popularity of certain movies across different datasets, while also revealing slight variations in the rankings for specific films.

Data Processing

In this section, we delve into the crucial step of transforming and preparing the raw data to make it suitable for any further analysis and modeling. This phase involves various tasks, such as data cleaning, feature engineering, and data integration. By meticulously processing the data, we aim to ensure its quality and optimise its utility for the subsequent stages of the project.

Throughout this section, we will outline the specific data processing techniques employed and the rationale behind them. We will address challenges encountered during the process and discuss the strategies implemented to mitigate potential issues. Additionally, we will highlight any insights gained from the processed data that contribute to a deeper understanding of the underlying movie recommendation problem.

4.1 Data Cleaning

Since the data used in this POC is sourced from an open dataset, it has already undergone substantial cleaning and preparation for modeling purposes. However, building upon our EDA, we recognized the need for additional cleaning steps to ensure the integrity of our analysis.

Although the extent of cleaning required is minimal, we take some precautions to mitigate potential pre-existing biases within the dataset. Specifically, we opted to remove movies with less than 20 ratings and users who have given less than 20 ratings. These threshold values were selected somewhat arbitrarily but with the intention of enhancing the reliability and representativeness of the model.

By implementing this additional cleaning step, we aim to reduce the influence of movies and users with limited ratings, allowing us to focus on more substantial data points and facilitate more accurate modeling results.
The reduced datasets are as follows:

	100k Dataset	2.5M Dataset
Original	100836	25000095
Cleaned	66405	24712388

Table 4.1: Data cleaning

4.2 Constructing the Data Format

This section plays a pivotal role in the project, primarily due to the unique challenges posed by the extremely large and sparse dataset. The conventional data science and machine learning libraries such as pandas or keras/tensorflow proved to be inefficient for data processing and modeling, particularly when dealing with the larger 2.5 million dataset.

To overcome these limitations, we adopt an alternative approach by representing the data using two lists of tuples. The *users* list takes the form:

$$\text{users} = [(\text{userId}, \text{movieId}, \text{rating})]$$

Similarly, the *movies* list is structured as:

$$\text{movies} = [(\text{movieId}, \text{userId}, \text{rating})]$$

By organising the data in this manner, we gain the advantage of efficient indexing into each user or movie and can effectively train our model biases.

To achieve this data structure, we begin by creating mappings that reset the $userId$ index, ensuring it starts from 0. These mappings are represented as dictionaries of the form $\{new\_userId : old\_userId\}$. Likewise, we apply a similar process to the $movieId$, creating a mapping that resets the $movieId$ index to start from 0 and returns a dictionary of the form $\{new\_movieId : old\_movieId\}$.

We subsequently apply these mappings to the corresponding user and movie dataframes. This allows us to easily convert and extract the $userId$, $movieId$, and *rating* information from each dataframe into lists.

This approach simplifies subsequent data processing steps, enabling us to efficiently work with the data and construct our recommendation model.

4.3 Train-Test Split

The train-test split was performed using the widely used **Sci-Kit Learn**'s *train_test_split* library. We faced a decision regarding the selection of data for testing purposes. One option was to set aside a specific group of users and movies, such as the last 1000 entries from each dataset. However, due to the data packaging approach we employed for modeling, where the data is in an "unpacked" state, extracting a random sample of around 1000 entries would not pose any issues during training. It would simply indicate that these particular samples were not present in the original training dataset. This approach allows for a more robust model training and subsequent model analysis, enabling us to evaluate the performance and generalisation of our recommendation model effectively.

Here are the respective splits:

	100k Dataset	**2.5M Dataset**
Training Set	63084	24711152
Test Set	3321	1236

Table 4.2: Train-test split.

4.4 Feature Engineering: Genres

In the second model, we enhance the initial model by incorporating genre information. We utilise the **Sci-kit Learn** library *MultiLabelBinarizer* to perform one-hot encoding of the movie genres. Subsequently, we apply Jaccard similarity to group similar movies together based on their genre representation. This grouping process aims to refine the predictions and offer more accurate recommendations of similar items to the users. By leveraging genre information, we expect to enhance the model's ability to capture and recommend items that share similar thematic characteristics.

Model Building: Biases

This section provides an overview of the model construction process, including the hyperparameters utilised (further elaborated in the subsequent section). Additionally, we delve into the details of our log loss function and outline the calculation of predictions.

We calculate our predictions as follows:

$$prediction = b_u + b_i + p_u^T q_i$$

where b_u is the user bias, b_i is the item bias and $p_u^T q_i$, is the affinity of user u to item i.

In the code base, this looks as follows:

$$\mathbf{prediction} = \mathbf{user\_bias} + \mathbf{item\_bias} + \mathbf{user\_factors} \cdot \mathbf{item\_factors}$$

Our log-likelihood is given by:

$$L = -\frac{\lambda}{2} \sum_{m,n} (r_{m,n} - u_m^T v_n)^2 - \frac{\tau}{2} \sum_m u_m^T u_m - \frac{\tau}{2} \sum_n v_n^T v_n$$

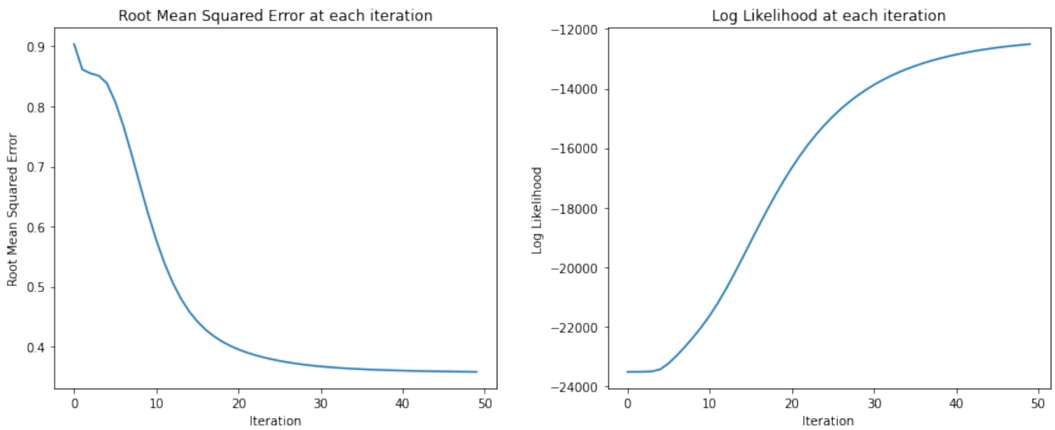
where u_m represents the user vector, v_n represents the movie vector, $r_{m,n}$ represents the rating given by user m to movie n, λ is the regularisation parameter, and τ is the confidence parameter.

This equation is a bit more involved, so please refer to the notebooks if you would like to see how it is implemented.

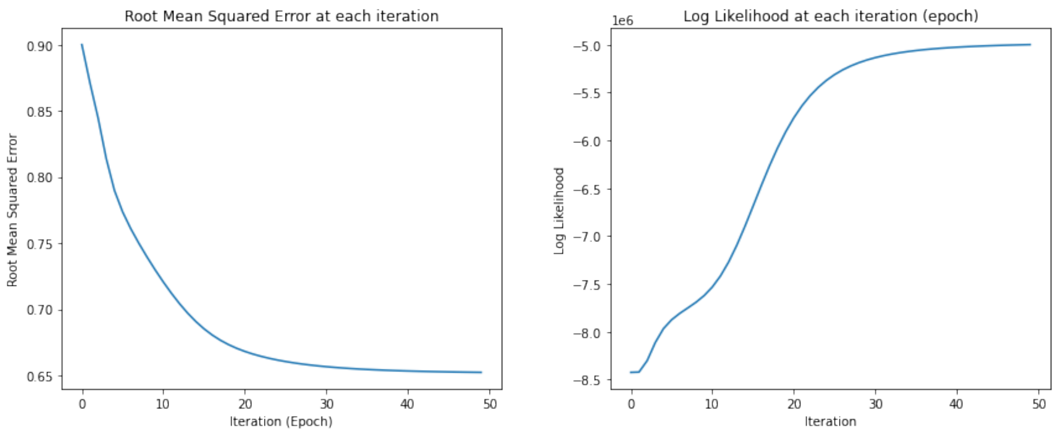
This equation is a bit more involved, so please refer to the notebooks if you would like to see how it is implemented.

We trained our model for 50 epochs using 1000 latent factors for the user and item matrices. The regularization parameter was set to 0.05, and the confidence parameter was set to 0.01.

After training, we achieved an RMSE (Root Mean Square Error) of 0.35 on the 100k dataset and an RMSE of 0.65 on the 2.5M dataset. These values indicate the average difference between the predicted ratings and the actual ratings, with lower values indicating better model performance.



(a) RMSE and log-likelihood for the 100k dataset model. We achieve an RMSE of 0.35.



(b) RMSE and log-likelihood for the 2.5M dataset model. We achieve an RMSE of 0.65.

Figure 5.1: RMSE and log-likelihood for the models trained on the different datasets.

After training, the user biases, item biases, user factors, and item factors are saved as **numpy** arrays. These arrays are then loaded during the prediction phase to make recommendations.

Upon analysing the RMSE and log loss plots, we observe a notable pattern. For both datasets, we achieve a gradual convergence of the RMSE after approximately 30 epochs. Similarly, the log-likelihood exhibits a mostly monotonic increase until around the 30-epoch mark, where it eventually levels off. This indicates that the model's performance stabilizes and further training epochs may not yield significant improvements.

Hyperparameter Tuning: Grid Search

For hyperparameter tuning, we conducted a grid search using the model training algorithm to optimise for the best RMSE and log stability. The grid search was performed on the 100k dataset, and the resulting optimal hyperparameters were then applied to the model training on the 2.5M dataset.

Through the grid search, we discovered that the best combination of hyperparameters consisted of 1000 latent factors for both the user and item matrices. We also determined that a regularisation parameter of 0.05 and a confidence parameter of 0.01 yielded the most favourable results in terms of model performance.

Here are the values that the hyperparameter combinations were generated from:

num_factors	50	100	1000
lambda_value	0.01	0.05	0.1
tau_value	0.0l	0.05	0.1

Table 6.1: Hyperparameter values used for the grid search.

Table 6.3 below presents the results of our grid search, where we explored various combinations of hyperparameters. During the grid search, an important criterion we considered was early stopping based on log stability. Whenever we observed instability in the log loss, we halted the training process and proceeded to the next set of hyperparameters.

We systematically tested all possible combinations and identified the combination that yielded the lowest RMSE while also maintaining log stability. This approach ensured that we selected hyperparameters that not only minimized the RMSE but also provided consistent and stable results.

num_factors	lambda_value	tau_value	RMSE	log_likelihood	log_stability
50	0.001	0.001	1.0280899709345004	-4480.802024567397	False
50	0.001	0.005	1.0280501957330161	-4536.65881705999	False
50	0.001	0.01	1.0280521687681872	-4605.9878436251365	False
50	0.05	0.01	0.3947988456840902	-22342.748464047	True
50	0.005	0.005	0.8321121184822885	-22408.293595283587	False
50	0.005	0.01	0.8324371211878098	-22494.03119174159	False
50	0.1	0.01	0.6318012774415762	-44671.96834560254	True
50	0.01	0.005	0.6319047704320765	-44709.63698050523	False
50	0.01	0.01	0.8466577684141575	-44782.325201895874	False
100	0.001	0.001	1.028549864324458	-4474.850587690902	False
100	0.001	0.005	1.0285455297120143	-4502.6515970948485	False
100	0.001	0.01	1.028546755954432	-4537.47199421012	False
100	0.05	0.01	0.3588585181677033	-22344.22292494143	True
100	0.005	0.005	0.8340637243713622	-22376.781000569717	False
100	0.005	0.01	0.8339833734664837	-22425.53053863848	False
100	0.1	0.01	0.6243957945908065	-44682.53635189754	True
100	0.01	0.005	0.6247859379547057	-44699.63474570428	False
100	0.01	0.01	0.848894625520642	-44737.553373327544	False
1000	0.001	0.001	1.028976734959559	-4469.678861928877	False
1000	0.001	0.005	1.0289767505243421	-4472.457554067982	False
1000	0.001	0.01	1.0289766699841352	-4475.921484792628	False
1000	0.05	0.01	0.35740648808632686	-22345.36400216503	True
1000	0.005	0.005	0.8358001516945825	-22348.655993580687	False
1000	0.005	0.01	0.8357957951405318	-22353.436601512883	False
1000	0.1	0.01	0.6250465935960904	-44690.12208981073	True
1000	0.1	0.05	0.6249020036403468	-44691.865242699845	True
1000	0.01	0.01	0.8509871382671912	-44695.580709150854	False
BEST COMBINATION					
1000	0.05	0.01	0.35740648808632686		

Table 6.2: Grid search results.

Model Evaluation: Biases

In this section, we aim to assess the performance of our trained recommendation model on the test set. Here, we will showcase the results of random predictions made on the test set and report on our evaluation metric, the RMSE.

By examining the random predictions, we can gain insights into the effectiveness of our model in accurately predicting user ratings for unseen movies. Additionally, the RMSE metric allows us to quantify the level of deviation between the predicted ratings and the actual ratings provided in the test set.

In this section we will also gauge the overall performance and reliability of our recommendation system, providing valuable insights for further improvements and fine-tuning.

Note that the primary objective of this POC is to demonstrate the scalability of the model and its applicability to real-world scenarios. Therefore, the evaluation of performance will focus solely on the large dataset consisting of 2.5 million data points. By focusing on this dataset, we aim to showcase the model's capability to handle substantial amounts of data and its potential for practical deployment.

We achieved an RMSE of 0.78 on the test set, compared to the RMSE of 0.65 achieved on the training set. This shows that our model's performance on the unseen data (test set) is relatively close to its performance on the training data. The small difference in RMSE between the two sets indicates that our model generalizes well and can make accurate predictions on new, previously unseen data. We consider this level of performance satisfactory and reinforces the effectiveness of our recommender system in providing accurate and reliable recommendations to users in real-world applications.

Figure 7.1 below illustrates the distribution of the RMSE values for the test set. The plot reveals that the majority of RMSE counts are below 0.8, reaffirming the satisfactory performance of our model. This distribution demonstrates that our recommender system consistently provides good predictions, with the majority of recommendations having relatively low errors. Such results further validate the effectiveness and reliability of our model in generating meaningful recommendations for users.

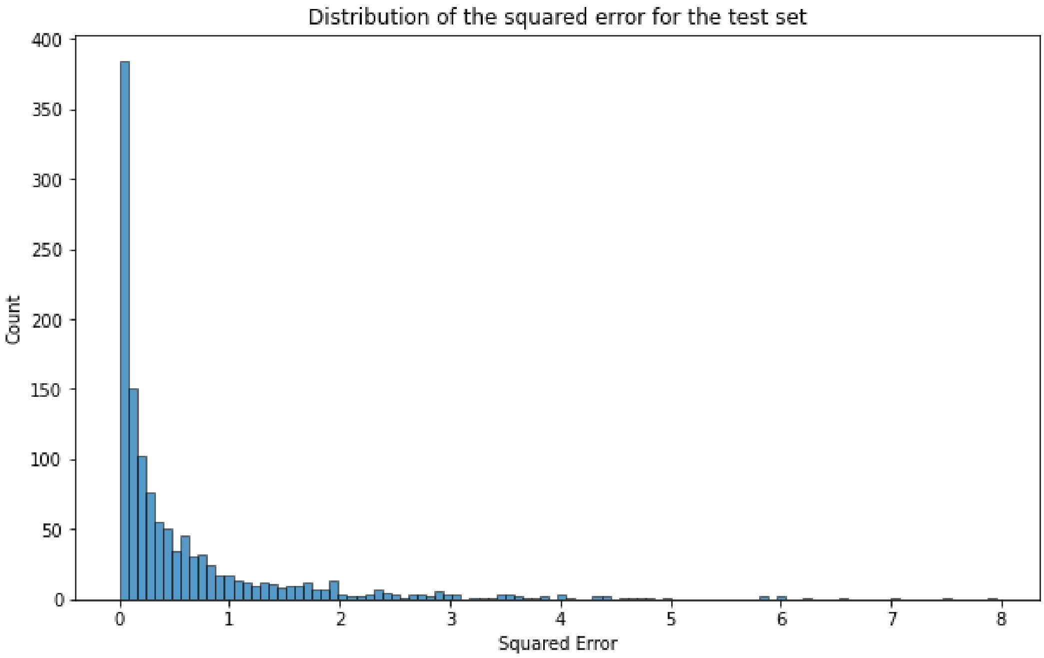


Figure 7.1: Distribution of the squared error for the test set

We further explore the performance of our model by presenting some examples that highlight its efficacy. We randomly select a few users from the test set and generate predictions for them. Tables 7.1, 7.2, and 7.3 showcases three such examples, where we present the top 5 recommended movies for each user.

To provide context and insight into the users' movie preferences, we include the top 20 rated movies in the first column of the table. In the next column, we present the top 10 predictions generated by the model. Finally, in the last column, we present the top movie recommendations that the model suggests the user will enjoy.

Upon analysing the results in the tables, we observe a significant overlap between the predicted movies and the actual rated movies of the users. This indicates the model's ability to accurately understand and capture the users' preferences. Additionally, the algorithm demonstrates proficiency in recommending sequels or prequels to movies that the user has already rated, showcasing its capability to understand and leverage user preferences for related content.

These examples serve as evidence of our model's effectiveness in generating personalised and relevant movie recommendations, catering to the individual tastes and preferences of users.

To further examine the performance of our model, we create three hypothetical users who are new to the platform and have only watched and rated one movie each, either giving it a positive rating of 5 stars or a negative rating of 2 stars. Please refer to Table 7.4 for the specific movies and corresponding predictions.

When analysing the positive rating column for the movie "Toy Story," we observe that the model successfully recommends more movies within a similar genre bracket based on a single positive rating. On the other hand, in the negative rating column, the model suggests a broader range of movies to provide variety. A similar pattern can be observed for "Lord of the Rings." However, the model struggles a bit with "Napoleon Dynamite," which is understandable as this movie has a unique and polarising appeal. Nevertheless, the model suggests movies on either side of the rating spectrum to cater to different preferences.

There are a few notable shortcomings in our current model:

- We would expect the model to recommend the sequels or prequels to "Toy Story" and "Lord of the Rings" based on the user's positive ratings.
- Additionally, we anticipate the model to present more movies of similar genres to cater to the user's preferences.

While these shortcomings may vary depending on specific business cases, we acknowledge the room for improvement in our model. In an upcoming section, we will address these issues by incorporating genres as features or weights in the model, providing more detailed explanations and enhancements in that regard. It's important to note that our current model is a rudimentary implementation, and we have plans to refine it further in subsequent iterations.

User: 124108		
Top 20 Rated Movies	**Top 10 Predictions**	**Top 5 Recommendations (Unseen by User)**
Pulp Fiction (1994)	Godfather, The (1972)	Paths of Glory (1957)
Maltese Falcon, The (1941)	Casablanca (1942)	Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)
Strangers on a Train (1951)	Citizen Kane (1941)	To Kill a Mockingbird (1962)
L.A. Confidential (1997)	Paths of Glory (1957)	On the Waterfront (1954)
Manchurian Candidate, The (1962)	Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	42 Up (1998)
Notorious (1946)	To Kill a Mockingbird (1962)	
Chinatown (1974)	Godfather: Part II, The (1974)	
Godfather: Part II, The (1974)	Chinatown (1974)	
Rear Window (1954)	On the Waterfront (1954)	
Double Indemnity (1944)	42 Up (1998)	
Casablanca (1942)		
Lone Star (1996)		
Godfather, The (1972)		
Citizen Kane (1941)		
Three Colors: Red (Trois couleurs: Rouge) (1994)		
Office Space (1999)		
Insider, The (1999)		
Third Man, The (1949)		
JFK (1991)		
Devil in a Blue Dress (1995)		

Table 7.1: User 124108s predicted and recommended movies.

User: 83		
Top 20 Rated Movies	**Top 10 Predictions**	**Top 5 Recommendations (Unseen by User)**
Like Water for Chocolate (Como agua para chocolate) (1992)	Cry, the Beloved Country (1995)	Cry, the Beloved Country (1995)
Dave (1993)	Something to Talk About (1995)	Rich Man's Wife, The (1996)
First Knight (1995)	Emma (1996)	Firelight (1997)
Speechless (1994)	Rich Man's Wife, The (1996)	Pride and Prejudice (1995)
Mrs. Doubtfire (1993)	Firelight (1997)	Crash (2004)
Circle of Friends (1995)	Pride and Prejudice (1995)	
Immortal Beloved (1994)	Crash (2004)	
Sabrina (1995)	Twice Born (Venuto al mondo) (2012)	
Mask, The (1994)	Curfew (2012)	
Emma (1996)	Sissi (1955)	
Phenomenon (1996)		
Sense and Sensibility (1995)		
Secret of Roan Inish, The (1994)		
Something to Talk About (1995)		
Little Women (1994)		
French Kiss (1995)		
Rob Roy (1995)		
Threesome (1994)		
Bed of Roses (1996)		
Bridges of Madison County, The (1995)		

Table 7.2: User 83s predicted and recommended movies.

User: 978		
Top 20 Rated Movies	**Top 10 Predictions**	**Top 5 Recommendations (Unseen by User)**
Napoleon Dynamite (2004)	Wallace & Gromit: The Best of Aardman Animation (1996)	For All Mankind (1989)
Monty Python and the Holy Grail (1975)	Monty Python and the Holy Grail (1975)	Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)
Wallace & Gromit: A Close Shave (1995)	Wallace & Gromit: The Wrong Trousers (1993)	Great Beauty, The (Grande Bellezza, La) (2013)
King of Kong, The (2007)	Princess Bride, The (1987)	Band of Brothers (2001)
Wallace & Gromit: The Wrong Trousers (1993)	For All Mankind (1989)	Planet Earth II (2016)
Wallace & Gromit: The Best of Aardman Animation (1996)	Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)	Cosmos
Memento (2000)	Great Beauty, The (Grande Bellezza, La) (2013)	
Italian Job, The (2003)	Band of Brothers (2001)	
School of Rock (2003)	Planet Earth II (2016)	
Life Is Beautiful (La Vita è bella) (1997)	Cosmos	
Interstellar (2014)		
Grand Day Out with Wallace and Gromit, A (1989)		
Good, the Bad and the Ugly, The (Buono, il brutto, il cattivo, Il) (1966)		
Princess Bride, The (1987)		
Twelve Monkeys (a.k.a. 12 Monkeys) (1995)		
Let the Right One In (Låt den rätte komma in) (2008)		
The Lego Movie (2014)		
Inception (2010)		
Singin' in the Rain (1952)		
Skyfall (2012)		

Table 7.3: User 978s predicted and recommended movies.

Toy Story (1995)		**Lord of the Rings: The Fellowship of the Ring, The (2001)**		**Napoleon Dynamite (2004)**	
(_, 0, 5.0)	**(_, 0, 2.0)**	**(_, 4887, 5.0)**	**(_, 4887, 2.0)**	**(_, 7750, 5.0)**	(_, 7750, 2.0)
Toy Story (1995)	Toy Story (1995)	Debt, The (Dlug) (1999)	Lion King, The (1994)	Golden Bowl, The (2000)	Hitch-Hiker, The (1953)
Paris Was a Woman (1995)	Hard-Boiled (Lat sau san taam) (1992)	Year of the Hare, The (Jäniksen vuosi) (1977)	Ferngully: The Last Rainforest (1992)	Along Came a Spider (2001)	Bug (1975)
Star Maps (1997)	American Virgin (2000)	India's Daughter (2015)	Naqoyqatsi (2002)	Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)	In My Father's Den (2004)
Hangman's Curse (2003)	Smooth Talk (1985)	Curfew (2012)	Jersey Girl (1992)	Whistle Blower, The (1986)	Black Book (Zwartboek) (2006)
Mon Oncle Antoine (1971)	Shivers (They Came from Within) (1975)	I, Claudius (1976)	Paradise Alley (1978)	Bug (1975)	Starbuck (2011)
Elizabethtown (2005)	Summer Storm (Sommersturm) (2004)	Monty Python's Fliegender Zirkus (1971)	Clean (2004)	Obsessed (2009)	Casa de mi Padre (2012)
Young Victoria, The (2009)	Casino Jack (2010)	Planet Earth (2006)	Shame (2011)	Road Trip: Beer Pong (2009)	Warm Bodies (2013)
Thor: Ragnarok (2017)	Dying of the Light (2014)	The Piano Tuner (2011)	Embers (2015)	Dark Shadows (2012)	Passion (2012)
Scooby-Doo and the Cyber Chase (2001)	Solyaris (1968)	Planet Earth II (2016)	Christine (2016)	Red Lights (2012)	Tim's Vermeer (2013)
Listening (2015)	Music for One Apartment and Six Drummers (2001)	His Last Vow	The Bookshop (2017)	The Founder (2016)	Papillon (2018)

Table 7.4: Predictions made on a generated fake user that has only watched one movie and rated it either 5 stars or 2 stars.

Lastly, in Figure 7.2, we present a 2D plot of the item factors for a subset of the movies. Observing the plot, we notice that similar movies tend to cluster together in a row-like manner, indicating some level of grouping based on their factors. However, it is also evident that the movies within each cluster are closely packed - they are close to their neighbours, suggesting limited distinctiveness between them. While there is some separation among the item factors, there is certainly room for improvement in achieving clearer differentiation between movies.

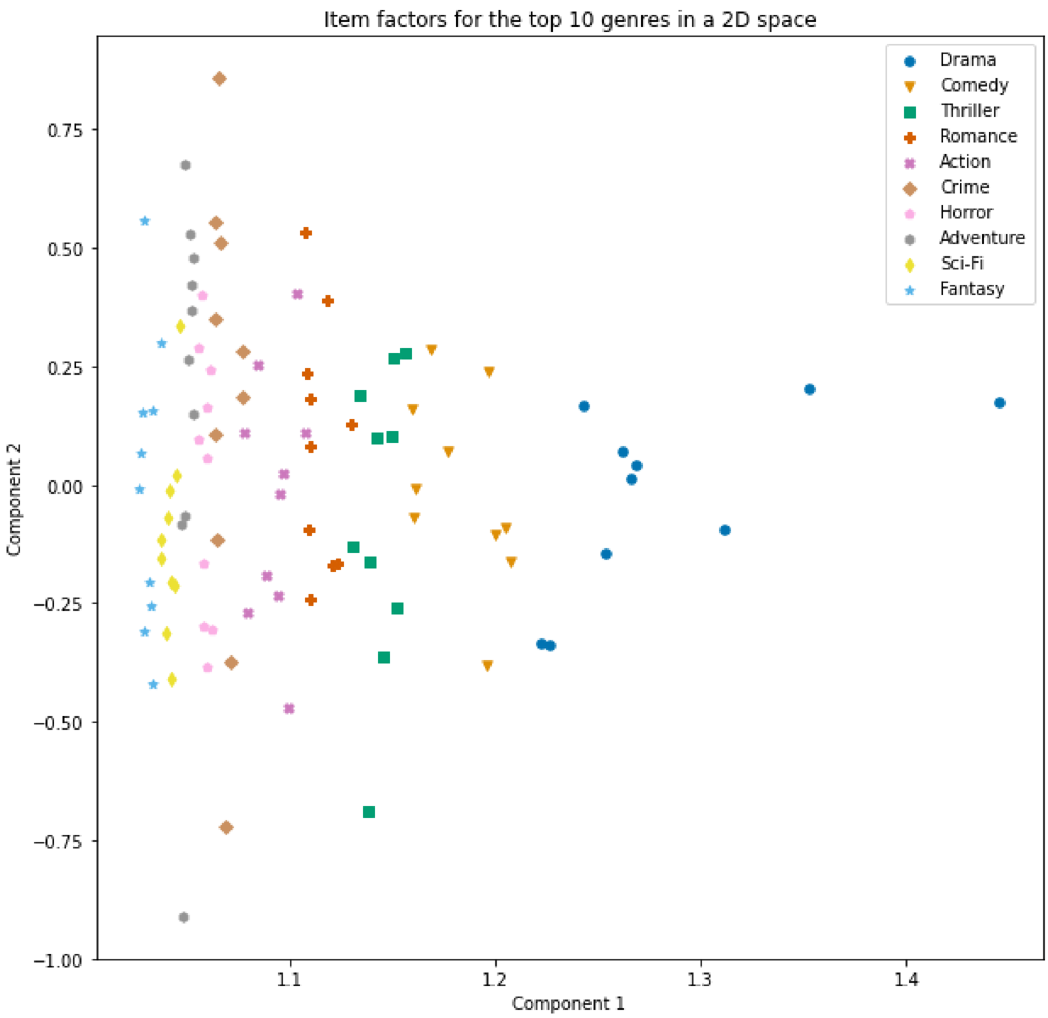


Figure 7.2: Item factors for the top 10 genres in a 2D space.

Model Enrichment: Biases with Genre Weights

In order to enhance the performance of the initial model, we introduce genre weights as an additional feature, as discussed in section 4.4. The incorporation of genre weights aims to provide more prominence to the characteristics of items in the model. By assigning specific weights to the genre feature, we aim to improve the model's ability to capture the inherent properties of items and enhance its overall performance. This adjustment allows the model to effectively leverage the genre information to make more informed and accurate recommendations.

To incorporate the genre information, we employ Jaccard similarity to calculate a similarity score. This score is then used as a weight that is added to the original prediction. By incorporating the genre similarity in this manner, we introduce some variability to the predictions, allowing for the recommendation of items that may not be immediately deemed relevant by the model. This approach is particularly useful when users have a preference for a specific type of movie, as it provides an opportunity for diverse and personalised recommendations. By considering the genre affinity in the weighting scheme, we aim to enhance the model's ability to cater to individual user preferences and deliver more tailored recommendations.

In summary, our approach involves the following steps:

- We begin by calculating the predictions for a specific user using the initial model.
- From the prediction results, we leverage Jaccard similarity on the movie genres to identify similar movies.
- By employing Jaccard similarity on genres, we aim to find movies that not only share the same genre but also align with the user's preferences.

This approach allows us to prioritize and retain the user's preferences while expanding the recommendations to include movies that are both genre-similar and well-aligned with the user's predicted preferences. By incorporating the Jaccard similarity-based genre matching, we enhance the model's ability to generate personalized recommendations that take into account both user preferences and genre similarities, resulting in a more tailored and diverse set of recommendations.

Model Evaluation: Biases with Genres

By incorporating genre weights into our model, we have observed an improvement in performance. The RMSE score has decreased from 0.78 to 0.72, indicating a positive impact on prediction accuracy.

Figure 9.1 visualizes the distribution of RMSE values for the test set with the inclusion of genre weights. The plot clearly illustrates that the majority of RMSE counts fall below the threshold of 0.8, reaffirming the satisfactory performance of our model. This distribution highlights the consistent delivery of accurate predictions by our recommender system, with most recommendations exhibiting relatively low errors. These findings provide robust evidence supporting the effectiveness and reliability of our model in generating meaningful and relevant recommendations for users, particularly with the integration of genre weights.

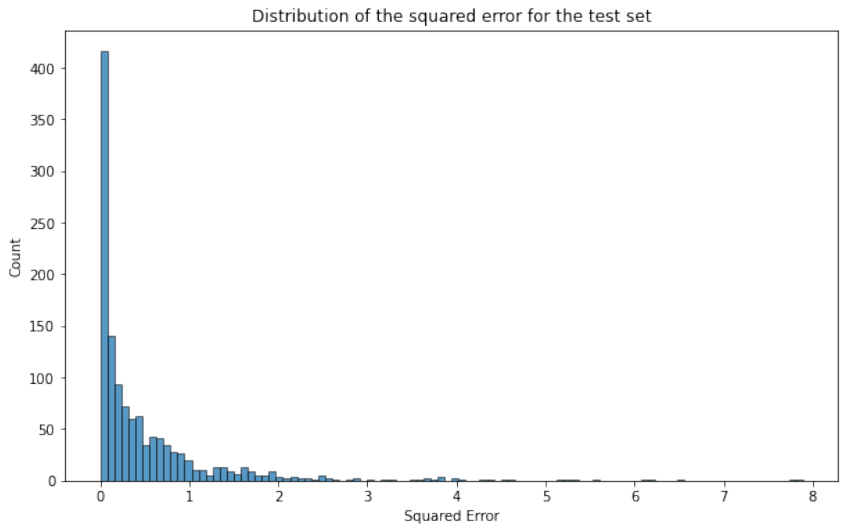


Figure 9.1: Distribution of the squared error for the test set with genre weights.

Model Comparison

User: 33523	
Crucible, The (1996)	
Basic Model	**Model with Genres**
Before the Rain (Pred dozhdot) (1994)	Othello (1995)
Shawshank Redemption, The (1994)	Cry, the Beloved Country (1995)
Schindler's List (1993)	Restoration (1995)
Fargo (1996)	Georgia (1995)
Cinema Paradiso (Nuovo cinema Paradiso) (1989)	Before the Rain (Pred dozhdot) (1994)
Paths of Glory (1957)	Shawshank Redemption, The (1994)
Boot, Das (Boat, The) (1981)	Schindler's List (1993)
Graduate, The (1967)	Cinema Paradiso (Nuovo cinema Paradiso) (1989)
Lives of Others, The (Das leben der Anderen) (2006)	Graduate, The (1967)
Civil War, The (1990)	Civil War, The (1990)

Table 10.1: Predictions for a random user and movie on both models.

User: 135337	
Harry Potter and the Order of the Phoenix (2007)	
Basic Model	**Model with Genres**
Reservoir Dogs (1992)	Apollo 13 (1995)
Princess Mononoke (Mononoke-hime) (1997)	Clan of the Cave Bear, The (1986)
Green Mile, The (1999)	Harry Potter and the Prisoner of Azkaban (2004)
Dear Zachary: A Letter to a Son About His Father (2008)	Sophie's World (Sofies verden) (1999)
	Valerie and Her Week of Wonders (Valerie a týden divu) (1970)
	Nibelungen: Siegfried, Die (1924)
	Harry Potter and the Order of the Phoenix (2007)
	Spiderwick Chronicles, The (2008)
	Where the Wild Things Are (2009)
	Thor (2011)

Table 10.2: Predictions for a random user and movie on both models.

User: 152637	
Lilo & Stitch (2002)	
Basic Model	**Model with Genres**
Usual Suspects, The (1995)	Wallace & Gromit: The Wrong Trousers (1993)
Léon: The Professional (a.k.a. The Professional) (Léon) (1994)	Princess Bride, The (1987)
Wallace & Gromit: The Wrong Trousers (1993)	Grand Day Out with Wallace and Gromit, A (1989)
Princess Bride, The (1987)	Iron Giant, The (1999)
Kolya (Kolja) (1996)	Transformers: The Movie (1986)
Life Is Beautiful (La Vita è bella) (1997)	Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)
Run Lola Run (Lola rennt) (1998)	Lilo & Stitch (2002)
Almost Famous (2000)	Treasure Planet (2002)
Cry Freedom (1987)	Kaena: The Prophecy (Kaena: La prophétie) (2003)
Amelie (Fabuleux destin d'Amélie Poulain, Le) (2001)	WALL·E (2008)

Table 10.3: Predictions for a random user and movie on both models.

To compare the performance of the enhanced model with the basic model, we conducted predictions on the same random movies from the test set. The results clearly demonstrate the substantial improvement achieved with the inclusion of genres. This improvement is particularly evident in Table 10.2 and Table 10.3.

In Table 10.2, we provided recommendations for a random user who had watched ”Harry Potter and the Order of the Phoenix.” We observed that the basic model struggled to find relevant recommendations, while the enhanced model had no difficulty at all. Notably, the enhanced model suggested a prequel to the Harry Potter movie, addressing one of the shortcomings mentioned earlier. Furthermore, the recommended movies align well with the genre spectrum of Harry Potter films.

Similarly, in the case of ”Lilo & Stitch”, we observed that the basic model provided reasonably good predictions, but the enhanced model outperformed it. The enhanced model's recommendations were more accurate and aligned with the user's preferences and the genre of the movie.

These examples clearly highlight the significant improvement achieved by the enhanced model, demonstrating its ability to provide more relevant and tailored recommendations compared to the basic model.

Conclusion

In conclusion, our experiments have demonstrated the potential of a basic recommender system in delivering personalised movie recommendations based on user preferences, thereby enhancing the overall user experience. By incorporating a single additional feature, we were able to further improve the personalisation of these recommendations.

The implementation of tailored suggestions holds significant business value for movie websites. It can greatly contribute to user retention and engagement, as users are more likely to remain active and explore a wider range of content when presented with relevant recommendations. This increased user interaction and consumption have a direct positive impact on sales and revenue generation.

Furthermore, the ability to offer superior recommendation capabilities gives a movie website a competitive edge in the crowded streaming industry. By providing personalized and meaningful suggestions, these platforms can attract new users and foster loyalty among existing customers. This differentiation allows them to stand out from competitors and establish themselves as go-to destinations for movie enthusiasts.

In conclusion, our experiments not only validate the effectiveness of the recommender system in generating personalised recommendations but also highlight its potential to drive business growth by improving user satisfaction, increasing engagement, and maximising revenue opportunities.