

Developing Machine Learning-based Recommender System on Movie Genres Using KNN

ANTHONY EZEH

Department of Computer
and Systems Sciences

Degree project 30 HE credits
Computer and Systems Sciences
Degree project at the master level
Spring term 2023
Supervisor: Dr. Ioanna Miliou



Abstract

With an overwhelming number of movies available globally, it can be a daunting task for users to find movies that cater to their individual preferences. The vast selection can often leave people feeling overwhelmed, making it challenging to pick a suitable movie. As a result, movie service providers need to offer a recommendation system that adds value to their customers. A movie recommendation system can help customers in this regard by providing a process that assists in finding movies that match their preferences. Previous studies on recommendation systems that use Machine Learning (ML) algorithms have demonstrated that these algorithms outperform some of the existing recommendation methods regarding recommendation strategy. However, there is still room for further improvement, especially when it comes to exploring scenarios where users need to spend a considerable amount of time finding movies related to their preferred genres. This prolonged search for the right movies can give rise to problems such as data sparsity and cold start. To address these issues, we propose a machine learning-based recommender system for movie genres using the K-nearest Neighbours (KNN) algorithm. Our final system utilizes a slider bar on a Streamlit web app, allowing users to select their preferred movies and see recommendations for similar movies. By incorporating user preferences, our system provides personalized recommendations that are more likely to meet the user's interests and preferences.

To address our research question: *“How and to what extent can a machine learning-based recommender system be developed focusing on movie genres where movie popularity can be predicted based on its content?”* we propose three main research objectives. Firstly, we investigate the employment of a classification algorithm in recommending movies focusing on interest genres. Secondly, we evaluate the performance of our classification algorithm concerning movie viewers. Thirdly, we represent the popularity of movie genres based on the content and investigate how this representation can inform the movie recommendation algorithm. On the heels of an experimental strategy, we extract and pre-process a dataset of movies and their associated genre labels from Kaggle. The dataset consists of two files derived from The Movie Database (TMDB) 5000 Movie Dataset. We develop a machine learning-based recommender system based on the similarity of movie genres using the extracted and pre-processed dataset. We vary the KNN algorithm with a slider bar to recommend movies of varying similarity to the selected movie, ranging from similar to diverse in genre. This approach can suggest movies with different titles for users with diverse preferences.

We evaluate the performance of the KNN classification algorithm using a user's interest genres, measuring its accuracy, precision, recall, and F1-score. The algorithm's accuracy ranges from low to moderate across different values of **K**, indicating its moderate effectiveness in predicting user preferences. The algorithm's precision ranges from moderate to high, implying that it provides accurate recommendations to the user. The recall score improves with increasing **K** and reaches its maximum at **K=15**, demonstrating its ability to retrieve relevant recommendations. The algorithm achieves a good balance between precision and recall, with an average F1-score of 0.60. This means that the algorithm can accurately identify relevant movies and recommend them to users with a high degree of accuracy. Furthermore, our result shows that the popularity visualization technique using KNN is a powerful tool for analysing and understanding the popularity of different movie genres, which can inform important decisions related to marketing, distribution, and production in the movie industry. In conclusion, our machine learning-based recommender system using KNN for movie genres is a game changer. It allows users to select their preferred movies and see recommendations for similar movies using a slider bar on a Streamlit web app. If confirmed by future research, the promising findings of this thesis can pave the way for developing and incorporating other classification algorithms and features for movie recommendation and evaluation. Furthermore, the adjustable slider bar ranges on the Streamlit web app allow users to customize their movie preferences and receive tailored recommendations.

Keywords: Movie Recommender System, Machine Learning, Content-based Filtering, Collaborative Filtering, KNN Algorithms, Classification Algorithm

Synopsis

Background

This thesis provides an overview of movie recommendation systems and their relevance in helping users find movies that match their tastes. It highlights the three main approaches to recommendation systems: collaborative filtering, content-based filtering, and hybrid systems. The use of ML in designing and developing a movie recommendation system is discussed, and various studies that have explored Artificial Intelligence (AI) methods to solve movie recommendation problems are presented. The thesis also discusses the challenges of data sparsity and cold start problems in recommendation systems and various machine learning algorithms, including collaborative filtering, matrix factorization, content-based filtering, and hybrid recommender systems, that have been used to tackle them. Machine learning algorithms, such as collaborative filtering, content-based filtering, and hybrid systems, have been widely used to address data sparsity and cold start problems in recommender systems. KNN has been utilized in several studies to recommend items based on similarities between users and items. However, there is a lack of studies using KNN with a focus on interest genres, and the goal is to create a machine learning-based movie recommender system using the KNN algorithm to suggest similar movies based on user-selected genres on a Streamlit app. This will provide personalized recommendations for movie enthusiasts.

Problem

The thesis discusses the common cold start problem in recommender systems, where new users with new profiles make it difficult to provide recommendations. Although hybrid techniques have been implemented to address this issue, there is a lack of studies that use the KNN algorithm with a focus on interest genres in overcoming data sparsity and cold start problems. Another aspect of movie recommendation that is often overlooked is the role of movie popularity in content-based filtering, where the popularity of movie genres based on their content is poorly understood. Understanding the popularity of movie genres based on content can provide a more personalized recommendation to users and improve the overall user experience. The significance of this problem lies in creating awareness that can prompt the movie community to pay closer attention to popular genres and invest more time and energy in playing a leading role over their competitors in the movie industry.

Research Question

In this thesis, classification algorithms are employed to create a machine learning-based movie recommender system that prioritizes recommending films based on users' interest genres. With this in mind, our primary research question (**RQ**) is introduced as follows:

How and to what extent can a machine learning-based recommender system be developed focusing on movie genres where movie popularity can be predicted based on content?

For the sake of elaboration, we seek to answer this cogent question with the following sub-questions:

1. *How can a classification algorithm recommend movies to users based on their genres of interest?*

2. *How can the performance of the movie recommendation algorithm, which incorporates the user's interest in various genres, be evaluated?*
3. *How can the popularity of movie genres be represented based on content, and how can this representation inform the movie recommendation algorithm?*

Method

We use an experimental research strategy to proffer answers to our research questions. First, we obtained the movie dataset and cleaned it by removing duplicates, missing values, inconsistent entries, and irrelevant columns. We also extracted additional relevant features that can be used for classification. Next, we employed the movie-genre matrix to store information about movies and their genres in a matrix format. Each row represents a movie, and each column represents a genre. Then, we used the KNN algorithm to select **K** genres with the highest similarity according to the similarity matrix. We trained the KNN algorithm on the extracted features and target variable. After training, we applied pickle to our trained KNN and the similarity and genre parts of our dataset labels. The pickled KNN recommends the **k** movies most similar in genre to the user input or search query. Finally, when a user inputs their movie preferences or searches for a specific movie, the KNN algorithm searches for the **k**-nearest neighbours and recommends them to the user.

Result

Our study on developing a machine learning-based recommender system for movie genres using KNN highlights the importance of the slider bar in allowing users to choose **K** values for movies and similar movies easily. The algorithm's accuracy ranges from **0.5478** to **0.6289** across different **K** values, indicating its effectiveness in predicting user preferences. The precision of our algorithm's results ranges from **0.6165** to **0.6880** across different **K** values, suggesting that it provides accurate recommendations to users. Our results for the recall score demonstrate that the algorithm's ability to retrieve relevant recommendations improves with increasing **K** values, with a maximum of **0.6787** at **K=15**. Additionally, our algorithm's F1-score of **0.60** shows a good balance between precision and recall, which means that the algorithm is able to identify relevant movies and recommend them to the user with a high degree of accuracy.

Discussion

Our study used a classification algorithm to recommend and classify movies based on their genres, achieving a high accuracy rate of **63%**. This approach outperformed a previous study that used a genre correlation algorithm and showed comparable results to other studies that used KNN and K-means algorithms. By utilizing a KNN algorithm, our recommender system can address the data sparsity and cold start challenges, providing users with a list of available movies to choose from. Although the performance of the model is decent, there is still room for improvement, as the accuracy rate is **63%**, and there are false positives and false negatives. Overall, our study highlights the potential of classification algorithms in developing accurate and effective movie recommendation systems that consider user preferences and genre correlations.

Acknowledgement

I sincerely thank my supervisor, **Ioanna Miliou**, for her invaluable guidance and feedback throughout this thesis. I also want to thank my colleagues for their shared experiences and constructive feedback during the seminars.

I would like to seize this opportunity to profoundly thank the reviewers of this thesis for their valuable feedback and insightful comments, which have greatly contributed to the quality and clarity of this thesis. Their suggestions have helped me to refine and improve my work, and I am deeply grateful for their time and effort. Their feedback has not only helped me to produce a better thesis but has also enhanced my understanding of the subject matter. Once again, I would like to express my sincere gratitude to the reviewers and my supervisor for their invaluable support and guidance throughout this process.

I appreciate my family, friends, and partners' unwavering support during this demanding journey. Our DSV peers made the work enjoyable, and I am grateful for their camaraderie.

Finally, I am thankful for this experience at Stockholm University, which allowed me to explore the world of movies and gain practical knowledge.

Table of Contents

1	Introduction	1
1.1	Research Problem	3
1.2	Aim and Objectives.....	3
1.3	Research Question.....	4
1.4	Delimitations	4
1.5	Thesis Structure.....	5
2	Extended Background	6
2.1	Movie Recommender Systems.....	6
2.1.1	Collaborative Filtering	6
2.1.2	Content-based Filtering	7
2.1.3	Hybrid Filtering	8
2.2	Related Work	8
2.2.1	Movie Review.....	9
2.2.2	Movie Genre	9
2.2.3	Movie Rating.....	10
2.2.4	Movie Popularity	11
2.3	Summary	11
2.4	Machine Learning	12
2.4.1	Classification	13
2.4.2	Similarity in Computation	14
2.5	Summary	15
2.6	Novelty of This Thesis	16
3	Research Methodology.....	17
3.1	Research Strategy	17
3.1.1	Experiment.....	17
3.1.2	Design Science	18
3.2	Research Method.....	19
3.2.1	Pre-processing	20
3.2.2	Movie Recommendation	20
3.2.3	Popularity Visualization	22
3.3	Research Methods Applications	22
3.3.1	Data Source and Pre-processing.....	22
3.3.2	Movie Genres and their Features	23
3.3.3	Model Evaluation	24
3.3.4	Software and Techniques	25
3.3.5	Ethical Considerations.....	26

4	Results.....	27
4.1	Evaluating KNN Performance	27
4.1.1	KNN Performance in Recommending Movies	27
4.1.2	KNN Performance at Different Values of K	28
4.1.3	Movie Recommendation using K Values.....	30
4.1.4	Summary	33
4.2	Case Studies.....	33
5	Discussion and Conclusion	37
5.1	Evaluation of Findings	37
5.1.1	Movie Recommendation	37
5.1.2	Evaluating KNN Performance	38
5.1.3	Movie Genre Popularity	39
5.2	Implications and Contribution	40
5.3	Limitations	42
5.4	Future Research	43
5.5	Conclusion.....	43
	References	45
	Appendix A – Poster Outputs for KNN Movie Recommendations	51
	Appendix B – Evaluating KNN Performance in Movie Recommender System	56
	Appendix C – Movie Genre Popularity.....	64
	Appendix D-GitHub Repository	74
	Appendix E – My Thesis Reflection	75

List of Figures

Figure 1: Collaborative Based Filtering adapted from Goyani & Chaurasiya (2020).....	7
Figure 2: Content Based Filtering adapted from Goyani & Chaurasiya (2020).....	8
Figure 3: An Example of KNN Algorithm.....	14
Figure 4: Proposed Experimental Framework.....	19
Figure 5: Movie_with_Genre's with a value of 0 and 1	21
Figure 6: MovieId, Title, and Genres parts of the collected dataset.....	23
Figure 7: MovieId and title parts of the dataset.....	23
Figure 8: Recommended Movies from our System.....	33
Figure 9: Recommended Movies from our System.....	34
Figure 10: Recommended Movies from our System.....	35
Figure 11: Movie Genre Popularity.....	35

List of Tables

Table 1: Reviewed Papers	12
Table 2: Calculating formula of users' similarity	14
Table 3: Movie-Genre Matrix	20
Table 4: Movies with Genre-Related Features in the Dataset	24
Table 5: Performance analysis of Evaluation Metrics on the entire movies	27
Table 6: List of Movies from our Recommender System	28
Table 7: Performance analysis of Evaluation Metrics	29
Table 8: Recommendation Movies using K values	31
Table 9: Movie Titles with Associated Genres	32

List of Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
HAC	Hierarchical Agglomerative Clustering
HCB	Hybrid Collaborative-Based
KNN	K-Nearest Neighbour
ML	Machine Learning
MAE	Mean Absolute Error
MSE	Mean Squared Error
NB	Naïve Bayes
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
SVD	Singular Value Decomposition
SNS	Social Network Service
SLR	Systematic Literature Review
SVM	Support Vector Machine
TMDB	The Movie Database

1 Introduction

Given the vast number of movies readily available worldwide, it is a challenging task for users to find the appropriate movies suitable for their tastes. In other words, people may feel frightened when presented with a vast selection of movies. Consequently, a movie recommendation system is therefore necessary if movie service providers are to provide value to customers. A movie recommendation system comes in handy as it is a process of accomplishing these tasks. Such a system is laced with implications inspired by the recorded successes in different domains such as books (Linden et al., 2003), TV programs (Miller et al., 2003), jokes (Goldberg et al., 2001), and news articles (Resnick et al., 1994). The three approaches to recommendation systems are Collaborative Filtering (Breese et al., 2013), Content-based Filtering (Balabanović & Shoham, 1997), and Hybrid systems (Singh et al., 2020). Collaborative Filtering intends to automatically find groups of similar users from a set of active users. Correlation measures are used to compute the similarities between users. The opinions of the users' groups provide a backdrop for item recommendations to users. In the context of the recommender system, the general term item refers to what movie to watch (Fanca et al., 2020). Content-based Filtering relies upon similarities between the items, that is, between two movies or two purchased items. Hybrid systems combine collaborative and content-based filtering in dealing with the operationalization of data (Quan et al., 2006).

Moreover, state-of-the-art technologies have been used in dealing with numerous challenges that bedevil users of recommender systems. One such technology of interest is ML which is an offshoot of AI. Machine learning makes it possible to use soft technologies for regular operations (Ahuja et al., 2019). It is crucial in recommender systems as it allows them to learn from user behaviour and make predictions. Consequently, designing and developing a movie recommendation system is feasible due to the ML techniques that provide automatic systems, which can learn and improve themselves from experience without being explicitly programmed (Badugu & Manivannan, 2023; Portugal et al., 2018; Wang et al., 2014; Zhang et al., 2019).

Regarding recommendations for movie users or other tasks, there have been a plethora of studies based on AI methods. Chawla et al. (2021) proposed a hybrid movie recommendation system based on the Movie dataset. They combined the user's personalization with the movie's overall features, such as genre and popularity in their content-based, and collaborative filtering model. Pavitha et al. (2022) described an approach to a movie recommendation system using Cosine Similarity to recommend similar movies based on the one the user chose. The authors performed sentiment analysis on the reviews of the movie chosen using Naïve Bayes (NB) classifier and Support Vector Machine (SVM) supervised ML algorithms. Ahuja et al. (2019) discussed how grouping KNN generates recommendation for movies with K-means clustering. They performed experiments and found that the predictive performance decreases with the decreasing number of clusters.

Existing research in the field of recommendation systems using ML has shown that the proposed ML algorithms have outperformed some existing recommendation algorithms in recommendation strategy (Li et al., 2018). However, there is still room for improvement, particularly in exploring the scenario where users need to spend a lot of time finding the right movies related to their interest genre. This lengthy time in finding the right movies to watch, lends credence to issues like data sparsity and cold start problems. To elaborate more specifically, the data sparsity refers to a scenario where it is very hard

to find users that have rated the same items because most of the users do not rate the items (Goyani & Chaurasiya, 2020). Similarly, the cold start problem occurs when there needs to be more information about a new user or item, making it difficult to make accurate recommendations.

There are several machine learning algorithms that have been used to tackle data sparsity and cold start problems in developing recommender systems. One of the commonly used approaches is collaborative filtering, which uses user-item interactions to recommend items to users. Collaborative filtering has been shown to be effective in addressing the cold start problem, where there is a lack of information about new users or items. For example, in their study, Sarwar et al., (2001) used collaborative filtering to develop a movie recommendation system, which demonstrated improved performance in terms of recommendation accuracy.

Another algorithm that has been used to tackle data sparsity and cold start is matrix factorization. Matrix factorization is a technique that reduces high-dimensional data into a lower-dimensional representation, which makes it easier to analyze and process. In the context of recommendation systems, matrix factorization has been used to learn the latent features of users and items, which can be used to make recommendations. For instance, in their study, Koren et al., (2009) used matrix factorization to develop a movie recommendation system, which demonstrated better performance than traditional collaborative filtering algorithms.

Content-based filtering is another approach that has been used to tackle data sparsity and cold start problems. This algorithm uses item features to recommend similar items to users. Content-based filtering has been shown to be effective in addressing the cold start problem, where there is a lack of information about new items. For example, in their study, Pazzani & Billsus, (2007) used content-based filtering to develop a news recommendation system, which demonstrated improved performance in terms of recommendation accuracy.

Finally, hybrid recommender systems, which combine multiple recommendation algorithms, have also been used to tackle data sparsity and cold start problems. These systems use a combination of collaborative filtering, content-based filtering, and other algorithms to make recommendations. For example, in their study, Adomavicius & Tuzhilin, (2005) used a hybrid recommendation system to develop a movie recommendation system, which demonstrated improved performance in terms of recommendation accuracy compared to traditional collaborative filtering algorithms.

Once again, KNN has been used in several studies to address data sparsity and cold start problems in recommender systems. For example, Chen & Lu, (2018), employed KNN to recommend items based on similarities between users and items. While all these algorithms have been widely used in the development of machine learning-based recommender systems and have shown effectiveness in addressing the issues of data sparsity and cold start, there is a paucity of studies that employ the KNN machine learning algorithm with a focus on interest genres, particularly in overcoming issues with data sparsity and cold start problems. Additionally, our goal is to create a machine learning-based movie recommender system that utilizes the KNN algorithm to suggest similar movies based on user-selected genres. This will enable movie enthusiasts to choose movies on a Streamlit ¹ app and find recommendations tailored to their preferences.

¹ Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers.

1.1 Research Problem

Once again, cold start problems are common in instances where the new user's profile is new, as they have not rated any item, and the system is unaware of their preferences, making it challenging to provide recommendations (Goyani & Chaurasiya, 2020; Hawashin et al., 2018; Qian et al., 2013; Zhang et al., 2013). To address this issue, hybrid techniques (Bobadilla et al., 2013) have been implemented, particularly in cases where there are not enough users in the system to find a match. To elaborate specifically, movie recommender systems usually find similar users or similar items and match them with the set of available users or items. Netflix alone, as the world's leading subscription service, can boast more than 30 million streaming members in the United States, Latin America, the United Kingdom, and the Nordics. A million movies and TV programs with different genres are related to their interest. On this premise, You et al.(2013) proposed a clustering method according to the user's interest extracted from Social Network Service (SNS) and the user's rating information system to proffer a solution to the cold-start problem. Amidst this method, there is no visibility of any usage of KNN algorithm in the movie lens data garnered from movie websites about their interest genre, as well as the performance of this algorithm in terms of model metrics. Moreover, cold-start problem is a concept of interest and there is no connection in terms of diverse genres that appeal to users' interest in the movie.

Another aspect of movie recommendation that we should take cognizant of, owing to the dearth of this aspect in the literature, is the pivotal role played by the movie's popularity in the spirit of content-based filtering. Movie popularity is of utmost importance to users, particularly regarding their interest genres (Basu et al., 1998; Halder et al., 2012; Konstan & Riedl, 2012). Despite this popularity, most of the available studies exploring the usage of ML in recommendation systems primarily based the movie's popularity on the type of reviews it gets from the audience (Pavitha et al., 2022). From this standpoint, there is a knowledge gap in visualizing the popularity of movie genres based on the content. To elaborate more specifically, the popularity of movie genres refers to the level of interest or demand for certain types of movies among audiences. This popularity can be determined based on the content of the movies. For example, if a particular movie genre, such as action or comedy, consistently receives high ratings, has high box office returns, or is frequently watched or discussed, it can be considered popular.

The popularity of movie genres can be influenced by a variety of factors, including the quality of the movies, the actors and actresses involved, the marketing and promotion of the movies, and the current trends in society. By analysing the content of movies, researchers and movie industry professionals can gain a better understanding of what types of movies are most appealing to audiences and make more informed decisions about which genres to produce and promote in the future. Additionally, by understanding the popularity of movie genres based on the content, movie streaming platforms and other companies in the industry can provide more relevant and personalized recommendations to their users. This can improve the overall user experience and increase customer satisfaction.

The significance of this problem lies in the awareness it creates, which can help the movie industry become more attuned to the diverse views that appeal to users' interests in movies. This increased awareness can prompt the movie community to pay closer attention to popular genres and invest more time and energy in playing a leading role over their competitors in the movie industry.

1.2 Aim and Objectives

This thesis aims to develop a machine learning-based recommender system, with an emphasis on employing a classification algorithm such as KNN to recommend movies based on interest genres. We

will evaluate the performance of our method in terms of given genres and investigate the popularity of movie genres based on content. For the effective and efficient execution of these thesis aims, our objectives are:

1. *To investigate the employment of a classification algorithm in recommending movies with a focus on interest genres.*
2. *To evaluate the performance of our classification algorithm concerning movies genres.*
3. *To investigate the prediction of the popularity of movie genres based on the content.*

1.3 Research Question

This thesis utilizes classification algorithms for developing a machine learning-based recommender system that focuses on recommending movies based on the interest genres. Given this view, we broach our main **Research Question (RQ)** as follows:

How and to what extent can a machine learning-based recommender system be developed focusing on movie genres where movie popularity can be predicted based on its content?

For the sake of elaboration, we seek to answer this cogent question with the following sub-questions:

1. *How can a classification algorithm be utilized to recommend movies to users based on their genres of interest?*
2. *How can the performance of the movie recommendation algorithm, which incorporates user's interest in various genres, be evaluated?*
3. *How can the popularity of movie genres be represented based on content, and how can this representation inform the movie recommendation algorithm?*

1.4 Delimitations

Owing to the paucity of time and resources which characterize this thesis, we are obliged to use movie lens datasets from Kaggle, which have been used studies such as Pavitha et al.(2022), Liu et al.(2022) and Fanca et al.(2020), to mention but a few. From this standpoint, we resorted to Kaggle to gather datasets, considering that obtaining the current dataset directly from movie production industries such as Netflix, Hulu, and Prime Video might require significant effort. In addition, recommender systems can struggle to make recommendations for new users or items with little or no prior data if adequate considerations are not given to chosen datasets. This premise is true in the sense that recommender systems rely on past interactions and behaviours to make predictions and recommendations. As such, if a new user or item has no prior data, the system has no information to base its recommendations on. This lack of data can result in a cold start problem for the recommender system, making it difficult for it to make accurate recommendations for new users or items. Additionally, as the number of users and items grows, the computation required to generate recommendations can become quite intensive.

In this study, we have considered the limitations of the KNN algorithm, including its potential for decreased performance with larger datasets due to the computational cost of calculating the distance between new points and existing points. We, however, intend to develop a machine learning based recommender system with emphasis on using KNN machine learning algorithms recommending movies

based on genres. The KNN algorithm is well-suited for making recommendations based on genre similarities between movies, as it can identify the closest neighbours to a given movie based on genre and other features. By leveraging the strengths of the KNN algorithm, our recommender system has the potential to provide highly accurate movie recommendations to users based on their preferred genres.

1.5 Thesis Structure

This thesis is structured as follows: **Chapter 1** introduces the background, problem in practice, research problem, aims and objectives, research questions, and delimitations. **Chapter 2** is an extended background that seeks to document the relevant literature to establish a knowledge base. **Chapter 3** broaches the methodology, which comprises the data collection and analysis methods, application of methods, and ethical concerns. In **Chapter 4**, we present our results. In **Chapter 5**, we discuss and conclude this thesis by reflecting on our findings to gain deeper insights. At the same time, we offer the strengths and limitations of this thesis, as well as suggestions for future work.

2 Extended Background

In lieu of the existing aim and objectives, we provide more in-depth information on the related literature and concepts in the following chapters. The next chapters detail the movie recommender system, relevant literature and ideas based on the current aim and objectives. We also highlight machine learning types and concepts as well as the innovative aspects of the approach examined in this thesis.

2.1 Movie Recommender Systems

Movie recommendations function by excluding irrelevant material and including just that which has compatible traits or properties (Çano & Morisio, 2017). As was already said, the world has transitioned from a period of online data scarcity to one of exponential development. The systems operate by altering the data to ensure it is effective for data-driven decision-making. The systems must determine what items match a certain client and which do not in the maze of product information provided. To enhance product viewing and, consequently, the likelihood that customers will make a purchase, the systems go further with target and retargeting marketing (Schafer et al., 2001).

To increase product sales or movie watching, developers must create more effective systems that match user preferences and have greater performance attributes (Deldjoo et al., 2016). Collaborative, content-based, and hybrid filtering are three main categories of filtering techniques.

2.1.1 Collaborative Filtering

Collaborative filtering works by matching the similarities in items and users. It considers the features of the users as well as that of the items the users have watched or searched for before (Shen et al., 2020). Latent characteristics derived from rating matrices are often examined. Recommendations in movie recommendation systems are based on user data and what other users with similar data are watching. For instance, user demographics like age, gender, and ethnicity are selected through collaborative filtering in movie recommender systems (Dakhel & Mahdavi, 2011). An example of recommending movies to users is used to illustrate collaborative filtering in **Figure 1**. The diagram shows that User 1, User 2, and User 3 have rated movies based on their interests. A user-movie matrix is created based on these ratings, and a similarity model is applied to find similarities between users so that recommendations can be made to User 3.

These ratings generate movie suggestions corresponding to individuals with comparable demographic traits and prior user search history. If the user has not entered any data or there is not enough data for any reliable grouping, collaborative filtering suffers from a cold start. It is unsure what to recommend in these situations (Katarya & Verma, 2017). Due to the possibility that individuals with identical demographic traits may not have comparable preferences, the suggestion's accuracy is also constrained (Kumar & Sharma, 2016)

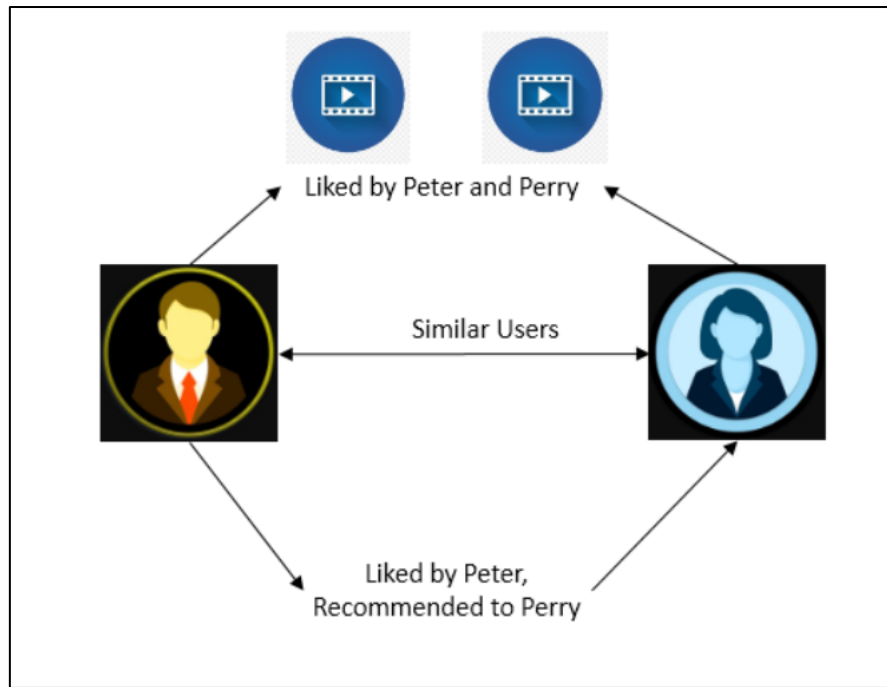


Figure 1: Collaborative Based Filtering adapted from Goyani & Chaurasiya (2020)

2.1.2 Content-based Filtering

Unlike collaborative filtering, content-based techniques employ user and item feature vectors to make recommendations. The key distinction between the two strategies is that content-based algorithms suggest products based on content attributes (no need for data about other users; recommendations about niche items, etc.). Comparatively, collaborative filtering merely considers user behaviour and makes product recommendations based on users that exhibit similar patterns (no domain knowledge; serendipity, etc.). A content-based filtering system operates by making movie recommendations to the user based on the content of the movies. This approach recognizes that clustering in collaborative filtering suggestions may not always align with user preferences (Deldjoo et al., 2016), people with similar demographic features have vastly different interests and preferences; what person X likes may not be same as what person Y enjoys watching. Two examples of the information included in movie suggestions are the main characters and the genre of the movies.

Figure 2 illustrates the working of Content-Based Filtering. The process of content-based filtering is demonstrated by using Geometric Shapes as an example (Patel et al., 2019). In this figure, an Item Profile is created based on the user's preferences. For instance, if the user likes blue circles and triangles, the user profile will be developed based on these items. By matching the user profile with the collection of different movies available, the system determines which movies matches the user's interests. For example, in the movies collection, a blue pentagon matches the user's preference. The user profile is generated based on the data from the item profile.

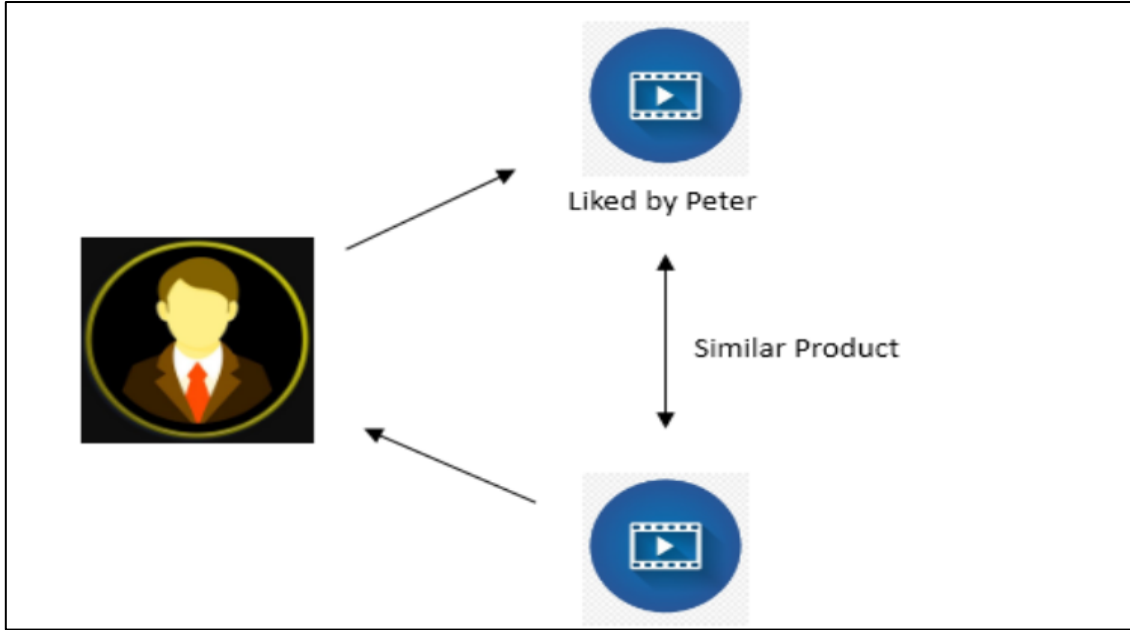


Figure 2: Content Based Filtering adapted from Goyani & Chaurasiya (2020)

2.1.3 Hybrid Filtering

The principles of all the previous algorithms are used in this filtering method. To address the shortcomings of each approach, it integrates collaborative filtering and content-based filtering (Jayalakshmi et al., 2022). It is better since it performs better when giving suggestions and has a quicker computation time (Çano & Morisio, 2017). For instance, content-based filtering may lack knowledge of domain relationships, whereas collaborative filtering may lack knowledge of user preferences (Cami et al., 2017). Combining these solves these problems since suggestions are made using both content data and user behaviour data.

2.2 Related Work

We present literature that resembles tasks regarding our research problem. To identify all the relevant research papers, we conducted a thorough search of the literature based on the guidelines for conducting a Systematic Literature Review (SLR), as presented by Kitchenham (2007). We do not claim that our study is an SLR. We intend to identify and synthesize relevant research papers using a machine learning-based approach in developing a recommender system. We searched mainly SCOPUS databases as well as other relevant databases through the Stockholm University Online Library for such keywords as “*Movie Recommender System*,” “*Machine Learning*,” “*Content-based Filtering*,” “*Collaborative Filtering*,” “*Hybrid Filtering*,” “*KNN Algorithms*,” “*KNN*,” “*K-Nearest Neighbours*,” and “*Machine Learning Algorithms*.” Based on these keywords, we identified some studies that are related to our research problem. Several dataset instances of such movie features of the recommender system as movies’ user ratings, genre, titles, scores, user Id, cast, crew, reviews, and tags are replete in the reviewed papers (Furtado & Singh, 2020; Goyani & Chaurasiya, 2020; Sharma & Yadav, 2020; Vinay et al., 2021; Zhang et al., 2013). A comprehensive list of the literature reviewed mainly in the related work section of this thesis is depicted in **Table 1**.

2.2.1 Movie Review

The ability of the machine learning methods to recommend movies based on any of these instances is shown in the literature. For example, the result of Vinay et al. (2021) depicts the performance improvement of item by item and user to user collaborative filtering using KNN. The authors used movie features such as stars, directors in their content-based filtering review which thereby resulted in the decrease in Root Mean Square Error (RMSE) as the value of K increases. In addition, the integration of content-based filtering and collaborative filtering resulted in the strengthening of the recommendation system (Vinay et al., 2021). However, the authors laid emphasis on the too high ratings issues, such as regularisation which connotes regularizing the parameters that constrain or shrinks the coefficient estimates towards zero.

Still echoing on the review feature of the movie, Pavitha et al., (2022) lend credence to this feature by using two of the supervised ML algorithms, NB Classifier and SVM in performing sentiment analysis on the chosen movie reviews. The outcome is the accuracy score of SVM that turned out to be **98.63%** whereas accuracy of NB is **97.33%**. Consequently, SVM outweighs NB and is proven to be a better fit for sentiment analysis.

2.2.2 Movie Genre

There are many ways to describe movies, including their kinds (such as cinema, animation, documentaries, and flash), length, background music, and emotional or suspenseful content, to name a few (Choi et al., 2012). One of the most vital traits of a movie is its genre. A movie's genre speaks about its basic theme. A movie may belong to only one genre or may have a combination of multiple genres in its several parts. Genres are mostly decided manually by the director of the movie or some experts like critics. It is necessary that a movie's genre be correctly recognized as a major audience decide to watch a movie based on its genre, that reflects the movie content.

The development of movie recommendation systems that offer suggestions to the audience based on their prior movie preferences is the most pertinent application of the movie genre. In literature, a number of these algorithms have been suggested. Choi et al., (2012) suggest a movie recommendation system to address the well-known cold start issue that arises during collaborative filtering using genre correlation algorithm. The recommender method is built on category correlations, which give director and expert-provided movie genres. They calculated genre connections in two distinct approaches, one using a different sample size and the other using films from other decades. They conducted an experiment using the GroupLens movie database and found that decade-based genre correlations may be used to provide exact suggestions. The result is an improved genre correlation algorithm, particularly for small-sized memory devices.

In Liu et al. (2022), we observed the usage of the KNN algorithm in the user-movie genre preference matrix based on users' scores on each type of movie. This usage is apparent in the content-based collaborative filtering where KNN was used to select a certain number of users by calculating the cosine similarity with the target based on how the target would rate a movie and recommend the movie with the highest predicted score to the target. The outcome is the experimental comparison between the content-based collaborative filtering with that of KNN that generally met users' needs despite the differences in the results (Liu et al., 2022). The usage of KNN is affirmed by Cacheda et al., (2011), where its traditional modification by introducing the weighted Person correlation showed a slight improvement in prediction accuracy. Although the weighted Pearson correlation in conjunction with KNN mitigated the data sparsity problem, the actual effect is that highly correlated users disappear as the rating matrix is too sparse.

In addition, some recommender systems have given evidence of instances where users must rate at least six movie genres to get a recommendation. For instance, Furtado & Singh, (2020) proposed a movie recommender model using the K-means algorithm in genre rating with a view to learning the detachment of each point from the centroid. The result of their proposal computes the connection between different clients and relies upon their ratings to prescribe movies to others who have similar tastes.

Aside from placing the requirement on the movie users to use rating as tool when it comes to movie genre, there is one study that lays emphasis on the pivotal role played by movie closed swarms. In this instance, movie swarm means mining a set of movies suitable for a producer for planning a new movie and for new item recommendation, popular and interesting movie mining which can be employed to solve new users problem (Halder et al., 2012). Halder et al., (2012) mine movie databases to collect information, such as, popularity and attractiveness required for recommendation after which similarity metric is used as a machine learning method for mining interested and popular movie genres to recommend movies to a new user. The system achieves high efficiency by optimizing the movie genre correlation algorithm, resulting in faster recommendations without sacrificing accuracy.

Finally, genre-based recommender systems that utilize the cutting-age machine learning method is also conspicuous in some noted studies. For instance, Godhani & Dhamecha, (2017) successfully implemented a simulation of genre based movie recommendation system that uses an ensemble of items that run on Hadoop. The outcome of this simulation is the reduction of the process time as it concerns the used method.

2.2.3 Movie Rating

Recommendation system is used to predict the “rating” or “preference” a user would give to an item (Ahuja et al., 2019). Other than KNN, Indira & Kavithadevi (2019) proposed Novel Recommender Systems in Multi-Cloud by using the Principal Component Analysis (PCA) method and Hierarchical Agglomerative Clustering Algorithm (HAC) while focusing on the user-item aspect of movie feature to enhance the ranking quality and search result quality as well as the minimum rate of ranking accuracy. The selected features are clustered with K-means and ranked using a trust ranking algorithm. The ranked output was evaluated, and the performance measure was analysed to provide an efficient result from the recommender system.

There is one dimension of user item ranking in the proposed architecture movie recommender system by Airen & Agrawal (2022). Here, the authors calculated the similarity between different users using user-item rating matrix. After calculating similarities, the variation of KNN-based collaborative filtering recommendation is used with five cross validations. The outcome is that errors like RMSE, Mean Squared Error (MSE), and Mean Absolute Error (MAE) are stable after the neighbourhood size of 40 neighbours which is the optimized value of K number of nearest neighbour to dataset used.

Moreover, another dimension of user ranking of movies is traceable in the instance where the average rating given by the user is clustered using the K-means clustering to create a utility clustered matrix (Ahuja et al., 2019). The authors calculated the KNN predictions for movie rating with the help of the similarity matrix and utility clustered matrix. The RMSE value of the proposed system, which achieved the same value as the existing technique but with fewer clusters, is far much better than the existing technique.

Furthermore, there is another study on recommender systems based on the movies ratings where the compared approaches of SVD, Co-Clustering, KNN and results are being examined (Gourammolla &

Gokila, 2022). Movie ratings of the user are calculated and movies recommended despite the fact that problems like data sparsity and cold start could not be controlled to certain extent (Gourammolla & Gokila, 2022). The key take away is that the Hybrid Collaborative-Based (HCB) approach produced less error rates compared to other approaches. In Hawashin et al. (2018), the authors proposed a solution to the cold start problem that uses the actual interests of the group to which the target user belongs, to provide movie recommendations to that user. The results of the experiment demonstrate the effectiveness of the proposed approach in terms of search time and space consumption (Hawashin et al., 2018). While this result is worthwhile as it shows the effectiveness of searching time and space use, the target user can be difficult to locate. Li et al., (2018) recognized the need for further improvement in terms of the case of data sparsity and issues surrounding tracking the change of user interest.

The user interest vector, on which the similarity matrix is based, is created by combining the user rating matrix with the hybrid recommendation algorithm that the authors presented (Li et al., 2018). The result of the experiment shows that the proposed algorithms solve the problem caused by data sparsity and the change of user's interest, thereby providing more accurate recommendation than some existing algorithms.

2.2.4 Movie Popularity

Movie popularity is a movie feature that encompasses a new feature of popularity gained by the collective activity of the users such as average ratings, number of views, likes, favourites, watchlist additions, and release date when the given Movie lens data is combined together (Chawla et al., 2021). In other words, the popularity of a movie can be predicted in terms of views, ratings, and reviews, just to mention but a few. While movie popularity can provide valuable insight into the likelihood of a movie being well-received, it is important to consider other factors such as personal preferences and individual tastes when making recommendations. Machine learning algorithms can analyse the relationship between movie popularity and other features such as genre, cast, and crew, to make recommendations. As such, content-based and popularity-based models are both good options, particularly in resolving a new user's cold start problem.

Chawla et al. (2021) addressed the limitations of individual models by evaluating the model's hyperparameters tuning, testing accuracy, and performance using SVD and collaborative filtering techniques. This approach resulted in improving the model's accuracy. The authors compiled a list of the most popular movies based on the user's genre preferences, and then ratings are projected for those movies. In terms of quality and diversity of recommendations, the hybrid model performs better than the separate models.

2.3 Summary

Recommender systems have been grouped into collaborative, content-based, and hybrid-based filtering. Several aspects of these groups are noticeable in some of the recommender system related works that characterize this thesis. For instance, the rating matrices which form the basis that can be used to predict some of the outcomes of the matched similarities innate in items or users are well recognized in collaborative filtering. On the other hand, we see the movie genres that form most of the content suggestion as it concerns the content-based filtering. All in all, the related work results in the novel categorization of concepts that we gleaned from sundry literature into such categories as movie review, movie genres, movie rating and movie popularity.

Category	Method	Result	Reported in
Movie Review	KNN	RMSE value that stabilizes around K = 20	(Vinay et al., 2021)
	NB, SVD	Accuracy score of SVM that turns to be 98.63%, and the NB accuracy of 97.33%	(Pavitha et al., 2022)
Movie Genre	Collaborative algorithm	Improved genre correlation algorithm	(Choi et al., 2012)
	KNN, Cosine Similarity	Content-based collaborative filtering that met users' needs	(Liu et al., 2022)
	KNN	A slight improvement in prediction accuracy	(Cacheda et al., 2011)
	K-means	Computes the connection between different clients and relying upon their ratings to prescribe movies	(Furtado & Singh, 2020)
	Similarity metric	Improves movie genre correlation algorithm in terms of efficiency and effectiveness	(Halder et al., 2012)
	Ensemble	Reduction of the process time as it concerns the used method	(Godhani & Dhamecha, 2017)
Movie Rating	PCA, HAC, and K-means	Recommends movies efficiently	(Indira & Kavithadevi, 2019)
	KNN	RMSE value of 1.233 compared to the existing technique	(Airen & Agrawal, 2022)
	K-means	RMSE value far much better with less no of clusters	(Ahuja et al., 2019)
	SVD, KNN	Production of less error rates compared to other approaches	(Gourammolla & Gokila, 2022)
	Similarity metric	Effectiveness in terms of searching time and space consumption	(Hawashin et al., 2018)
	Similarity metric	Improvement in terms of data sparsity and issues with tracking the change of user interest	(Liu et al., 2022)
Movie Popularity	SVD, Collaborative filtering	Enhances the accuracy of the model that performs better than separate models	(Chawla et al., 2021)

Table 1: Reviewed Papers

2.4 Machine Learning

Computers are used to replicate human learning using ML, which enables machines to recognize and learn from the actual world and enhance performance on tasks using this new knowledge. A computer

program is said to learn from experience E regarding some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , increases with experience E , according to a more rigorous definition of ML in Michalski et al., (1984). Machine learning was initially explored as a separate discipline in the 1990s, even though the first notions were developed in the 1950s (Michalski et al., 1984). Today, ML algorithms are employed outside of computer science in a variety of fields, such as business (Apte, 2010), advertising (Cui et al., 2015), and medical (Kononenko, 2001). The process of acquiring information is called learning. Being able to reason, humans naturally learn from their experiences. In contrast, computers use algorithms to learn rather than thinking.

There are now several ML algorithms that have been suggested in the literature. Supervised and unsupervised, semi-supervised are the three main categories. Supervised learning is a type of machine learning algorithm in which the model is trained on a labelled dataset with input-output pairs, such that the algorithm can predict the output for new input data based on the patterns learned from the training data. This approach involves the use of a predefined set of features for each instance, and the model is trained to predict the target variable based on these features (Zhang et al., 2019). Examples of supervised learning algorithms include decision trees, random forests, neural networks, and SVMs (Alpaydin, 2020). Supervised learning has been widely applied in various domains, such as text classification (Yang & Liu, 1999), sentiment analysis (Pang & Lee, 2008), and recommendation systems (Sarwar et al., 2001). However, the performance of supervised learning algorithms can be affected by various factors, such as the quality and quantity of the training data, the choice of features, and the selection of the appropriate model (Zhang et al., 2019). Therefore, careful consideration should be given to these factors during the design and implementation of a supervised learning-based system.

The ML algorithm's task is to learn from training data and then apply what it has learned to actual data. In the recommender system domain, ML algorithms such as NB and SVM have been used in the form of best algorithms to classify the movies reviews because the reviews usually come with huge diversity in them, it is vital to choose the right algorithm for classification (Pavitha et al., 2022).

On the other hand, unsupervised learning takes unlabelled information and finds fresh patterns or groups in the data. In view of the existing movie features where users have to choose their preferred movies, unsupervised learning comes in handy as it provides better methods by the way of clustering algorithms where groups based on genre and tags for movies foster optimization so that each cluster may not significantly increase variance (Cintia Ganesha Putri et al., 2020). In the sub-sections that follow, we elaborate more on the classification and the similarity in computation.

2.4.1 Classification

Classification is a machine learning technique used to predict the class or category of an item or entity based on its features or attributes (Rashid et al., 2020). Classification algorithms, such as the KNN, are widely used in ML for pattern recognition and data classification tasks. KNN is a non-parametric algorithm that identifies the k nearest neighbours to a new data point based on a similarity metric, and then classifies the new point based on the majority class of its neighbours. The choice of similarity metric is critical in the performance of the KNN algorithm, as it determines how similar two data points are. The algorithm identifies the k nearest neighbours based on a similarity metric and predicts the user's preference for a particular movie based on the ratings of these neighbours. Similarity is a fundamental concept in the development of recommendation systems, and it involves comparing the features or attributes of different items to identify those that are most similar. In movie recommendation systems, the similarity metric is often based on genre, which is used to group movies with similar themes and

topics. By using a classification algorithm such as KNN and incorporating similarity measures, it is possible to build a powerful recommendation system that accurately predicts users' movie preferences.

The KNN algorithm's core tenet is that the sample is thought to belong to a certain category if the majority of its k closest neighbours in the feature space do (Bilge et al., 2013). According to **Figure 3**, most of W 's closest neighbours fall into the X category, while w itself is in the X group.

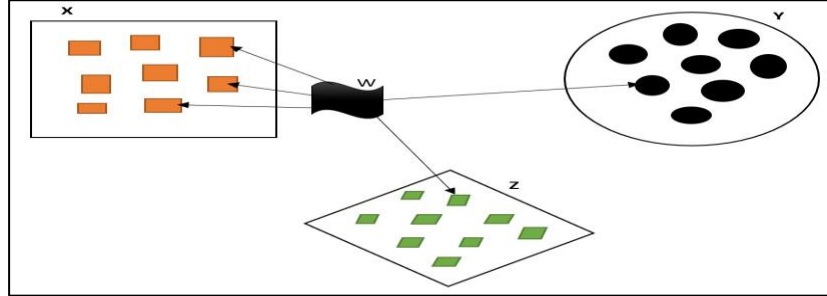


Figure 3: An Example of KNN Algorithm

After calculating the similarity between users as $\text{sim}(u, u^1)$, the algorithm chooses a certain number of users with the highest similarity as u 's neighbours, denoted as u . The algorithm sets a fixed value K for the neighbour selection and chooses only the users with the highest K levels of similarity regardless of the value of the users' neighbour similarity.

2.4.2 Similarity in Computation

In similarity computation, the focus is on measuring similarity. Between item-based and user-based, the similarity computation techniques are essentially the same. There are two fundamental approaches for calculating similarity (Sarwar et al., 2000). In the parlance of recommender systems and similarity computation, item-based can be genre, movie, reviews, score, popularity to mention but a few while user-based encompasses the users that are involved in the movie recommendation.

The similarity between users is calculated by evaluating the value of the items evaluated by two users. For example, to compute the similarity between $U1$ and $U3$, first identify the group of movies that they all scored as $M1$, $M2$, $M4$, and $M5$, as well as the relative scores of these movies (See **Table 2**). Each user uses an N -dimensional vector to represent item score. The score vectors for $U1$ and $U3$ are 1, 3, 4, and 2, 1, 5, respectively. The similarity formula is used to determine how similar $U1$ and $U3$ are (Xie & Meng, 2011).

$U \backslash M$	m1	m2	m3	m4	m5
u1	1	3	3	4	2
u2	3	1	4		
u3	2	4		1	5
u4	2		2		

Table 2: Calculating formula of users' similarity

The formula for expressing the similarity between two users is shown in equation 1 and 2; the most popular methods for doing so are Cosine Similarity and Pearson Correlation Similarity. These two similarities are explained further in the following subsections.

Cosine Similarity

The cosine similarity method calculates the similarity between two users by calculating the cosine of the angle between two vectors as can be depicted mathematically in equation (1). In cosine similarity, vectors are taken as the data objects in data sets, when defined in a product space, the similarity is figured out and computed.

$$Sim(x, y) = \cos(\vec{X}, \vec{Y}) = \frac{\vec{X} * \vec{Y}}{|\vec{X}| * |\vec{Y}|} = \frac{\sum_{s \in s_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in s_{xy}} (r_{x,s})^2} \sqrt{\sum_{s \in s_{xy}} (r_{y,s})^2}} \quad (1)$$

The angle between two vectors determines its direction and is measured in $\cos \emptyset$. This angle \emptyset can be calculated by using equation 1 which is synonymous to $\cos(\vec{X}, \vec{Y})$.

When $\emptyset = 0^\circ$, the “x” and “y” vectors overlap and prove to be similar. When $\emptyset = 90^\circ$, the “x” and “y” vectors are therefore dissimilar.

In terms of equation 1, $r_{x,s}$ and $r_{y,s}$ are the score of goods s scored by user X and Y respectively. s_{xy} is the set of movies that user x and y both scored on. In other words, $s_{xy} = \{s \in Items \mid r_{x,s} \neq \emptyset\}$.

Pearson Correlation Similarity

Pearson correlation similarity is a widely used technique in machine learning for collaborative filtering-based recommender systems (Su & Khoshgoftaar, 2009). It is a measure of the linear correlation between two variables, and in the context of recommendation systems, it is used to find the correlation between the ratings given by users to various items. This correlation can then be used to predict how a user will rate an item they have not yet seen based on their rating history and the rating history of other users who have similar tastes.

Pearson correlation similarity measures linear relationship between two variables. In fact, we measure the correlation between user’s ratings. Rather than considering the distance between feature vectors to estimate similarity, we can consider the correlation between the critics scores.

$$Sim(x, y) = \frac{\sum_{s \in s_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in s_{xy}} (r_{x,s} - \bar{r}_x)^2} \sqrt{\sum_{s \in s_{xy}} (r_{y,s} - \bar{r}_y)^2}} \quad (2)$$

Among them, \bar{r}_x is the average score is $x[3]$, the rest of the symbolic meaning is the same as depicted in equation 1.

2.5 Summary

To summarize, supervised learning is a type of ML where the algorithm is trained with labelled data and expected outputs. In the case of the recommender system, it is commonly used to classify movie reviews by learning from the training data and applying that knowledge to new data. The choice of the right algorithm is important because of the diversity in movie genres or any other type of movie features.

Similarity computation in machine learning provides a backdrop for expressing the similarity between two users or items. After calculating the similarity on users or items, the KNN algorithm is applied with a view to choosing the highest similarity.

2.6 Novelty of This Thesis

Recommending movies that address the issues associated with cold start and data sparsity, particularly as it relates to interest genres is scarce in the reviewed studies we examined in this thesis. To counteract these issues, we aim to develop a machine learning based recommender system using a classification algorithm such as KNN. In other words, we propose a KNN algorithm movie recommender system based on interest genres and at the same time mitigate the challenges associated with cold start and data sparsity. To realize this aim, our objectives comprise investigating the employment of KNN in recommending movies with a focus on interest genres, evaluating the performance of KNN algorithm regarding movie genres and investigating the popularity of movie genres based on its content. Therefore, recommending movies with emphasis on interest genres is a focal point in our thesis.

In addition, there is scant studies that place emphasis on displaying the popularity of movie genres depending on the content. To be more precise, the amount of interest or demand for movie genres among viewers is referred to as the popularity of such movie genres. A novel solution is using Streamlit to create an interactive web application that allows input information about a movie and get a prediction of its popularity based on the genre.

3 Research Methodology

This chapter presents the research methodology as well as the methods executed in this thesis. When we speak of research methodology, we speak in the context and intent of the general theory or strategy of how this study would be conducted whereas methods refer to the definite steps or procedure to achieve the results.

3.1 Research Strategy

Research strategy refers to the plan of action that guides the research process, including the methods, techniques, and tools used to collect and analyse data. A well-designed research strategy helps to ensure that the research is conducted in a systematic, rigorous, and efficient manner, and that the results are valid and reliable. According to Bell et al., (2022), research strategy refers to the overall approach to conducting research, including the methods, techniques, and procedures used to collect and analyse data. A research strategy helps to guide the research process and ensure that it is conducted in a systematic and rigorous manner (Creswell & Creswell, 2017).

Two research strategies, experiment and design science come in handy in terms of proffering answers to the research question on the employment of classification algorithm in recommending movies with a focus on genres of interest. The next subsections shade light on these two concepts and lend credence regarding the choice of the experimental strategy as the more adequate for this thesis.

3.1.1 Experiment

We aim to answer the research question posed in this thesis using the experimental research strategy. This method is vital for exploring the cause-and-effect relationships through an empirical study. Experimental research strategy is a scientific method that involves manipulating one or more variables to observe the effect on another variable, while controlling for extraneous variables that could potentially influence the results. In an experimental research design, researchers randomly assign participants or subjects to different groups or conditions, where each group is exposed to a different level or type of manipulation of the independent variable. The dependent variable is then measured and compared across the different groups or conditions, allowing researchers to determine whether there is a causal relationship between the independent and dependent variables. The experimental research strategy is often used in fields such as psychology, medicine, and biology to test hypotheses and evaluate the effectiveness of interventions or treatments. This type of research can provide strong evidence for causality and is highly valued in scientific inquiry.

Experimental research strategy is a scientific method that involves manipulating one or more variables to observe the effect on another variable, while controlling for extraneous variables that could potentially influence the results. In the context of developing a machine learning-based recommender system on movie genres using KNN, the experimental research strategy would involve manipulating the independent variables such as the selection of features and KNN parameters, while measuring the dependent variable such as the accuracy of the recommendations.

For example, we might conduct an experiment where they randomly assign participants to different groups or conditions, where each group is exposed to a different combination of movie features and

KNN parameters. The dependent variable, accuracy of the recommendations, would then be measured and compared across the different groups or conditions.

To answer the research question: "*How can a classification algorithm be utilized to recommend movies to users based on their genres of interest?*" using experimental research strategy, we could manipulate the independent variables such as the selection of classification algorithm and the types of features used for classification, while measuring the dependent variable such as the accuracy of the recommended movies. For example, researchers could randomly assign participants to different groups or conditions where each group is exposed to a different combination of classification algorithm and features used for classification. The dependent variable, accuracy of the recommended movies, would then be measured and compared across the different groups or conditions.

To answer the research question: "*How can the performance of the movie recommendation algorithm, which incorporates user's interest in various genres, be evaluated?*" using experimental research strategy, researchers could manipulate the independent variables such as the selection of classification algorithm and the types of user's interest genres used for classification, while measuring the dependent variable such as the accuracy of the recommended movies. For example, we could randomly assign participants to different groups or conditions where each group is exposed to a different combination of classification algorithm and user's interest genres used for classification. The dependent variable, accuracy of the recommended movies, would then be measured and compared across the different groups or conditions.

To answer the research question: "*How can the popularity of movie genres be represented based on content, and how can this representation inform the movie recommendation algorithm?*" using experimental research strategy, researchers could manipulate the independent variables such as the types of movie genres and the types of content used for analysis, while measuring the dependent variable which is the popularity of the movie genres. For example, we could randomly select a sample of movies from different genres and analyse their content using natural language processing techniques. The dependent variable, popularity of the movie genres, would then be measured and compared based on the types of content used for analysis. We could also use visualizations such as charts and graphs to display the popularity of the movie genres based on the content analysis results.

3.1.2 Design Science

An alternative strategy could be Design Science. Design science is described by Johannesson & Perjons, (2014) as: "*the scientific study and creation of artefacts as they are developed and used by people with the goal of solving practical problems of general interest*". By using design science, we would develop a movie recommender system based on KNN algorithm that is tailored to the specific needs of users who are interested in movie genres. This approach would involve user-centered design principles, which would ensure that the system is easy to use and effective in recommending movies. The evaluation of the system would provide valuable insights into the effectiveness of KNN algorithm in movie recommendation and guide future development of movie recommender systems based on machine learning algorithms. Design science in essence places emphasis on the development, design, demonstration, and evaluation of an artefact. It would have been better to focus on manipulating the independent variables such as the selection of classification algorithm and the types of user's interest genres used for classification, while measuring the dependent variable such as the accuracy of the recommended movies.

3.2 Research Method

The experimental portion of this thesis is broken up into different parts that are carried out to proffer answer to our research questions:

How and to what extent can a machine learning-based recommender system be developed focusing on movie genres where movie popularity can be predicted based on its content?

Figure 4 displays a schematic representation of our proposed experimental framework. This framework is categorized into three primary components which comprise pre-processing, movie recommendation and popularity visualization.

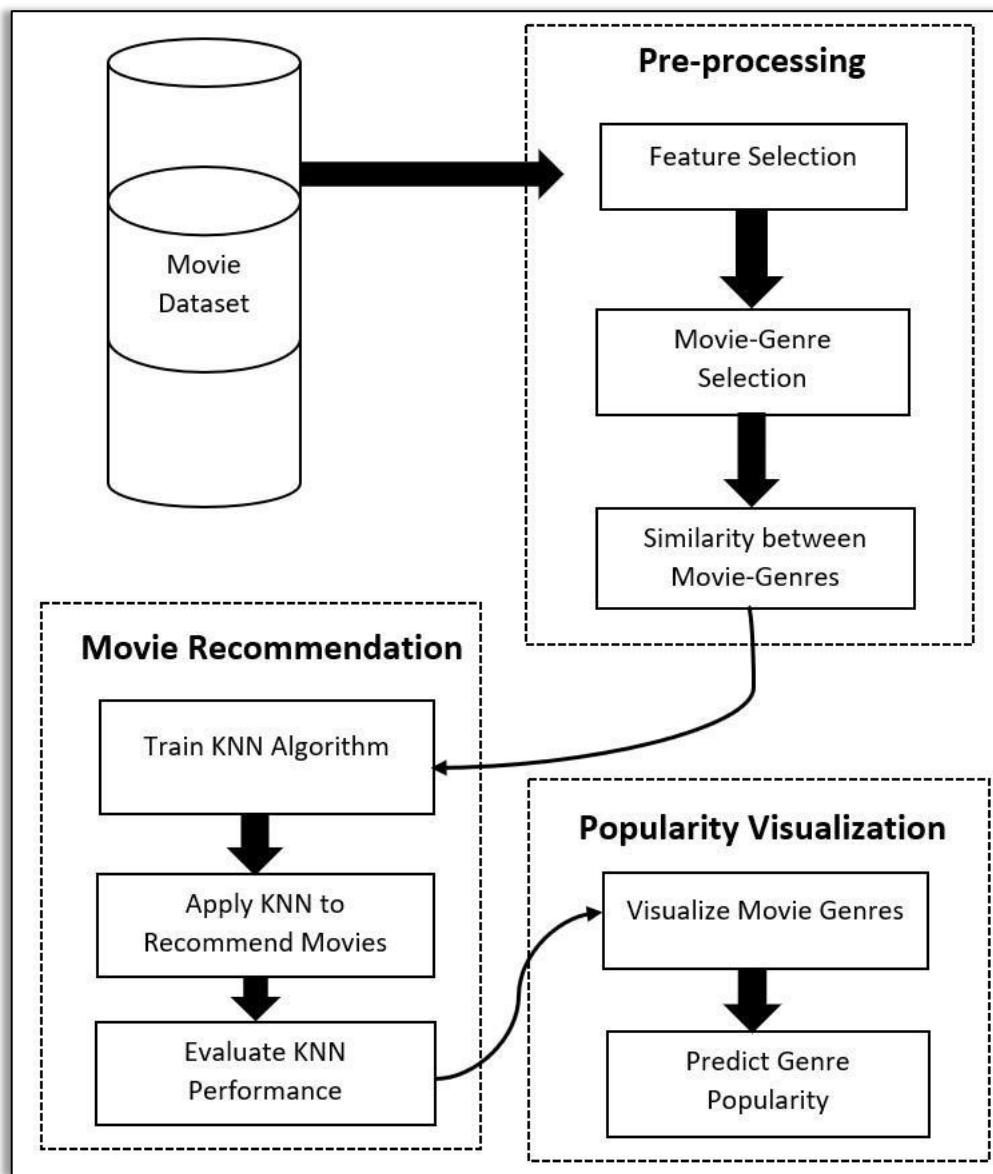


Figure 4: Proposed Experimental Framework

3.2.1 Pre-processing

The pre-processing involves collecting the dataset of movies and their associated genre labels. We obtained the dataset from Kaggle², which consists of two files derived from The Movie Database³(TMDB 5000 Movie Dataset): tmdb_5000_movies.csv and tmdb_5000_credits.csv. The collected dataset is cleaned by removing any duplicates, missing values, or inconsistent entries and irrelevant columns. It could also be feasible to extract additional relevant features that can be used for classification, such as movie ratings or release dates. During feature selection, we select the relevant features that will be used for classification.

In the movie-genre selection step, we employ the movie-genre matrix which is a data structure that stores information about movies and their genres in a matrix format. In this matrix, each row represents a movie, and each column represents a genre. The values in the matrix indicate whether a particular movie belongs to a particular genre. For example, a value of 1 might indicate that a movie is of a particular genre, while a value of 0 might indicate that it is not (See **Table 3**). This type of matrix is commonly used in the development of recommender systems, where it can be used to identify movies that are similar in genre. The matrix can also identify trends and patterns in movie genres over time, which can be useful in developing marketing strategies for movie studios and distributors.

G\M	M1	M2	M3	M4	M5
G1	1	0	0	0	1
G2	0	1	1	1	0
G3	0	0	0	1	0
G4	0	0	0	1	0

Table 3: Movie-Genre Matrix

The movie-genre matrix is typically created by collecting data about movies and their genres from various sources, such as movie databases and user reviews. Once the data has been collected, it can be processed and organized into the matrix format. Cosine similarity could be used in this process to calculate the similarity relationship between movie and genre.

3.2.2 Movie Recommendation

In this category, we used KNN to select k genres with the highest similarity according to the similarity matrix. The basic idea here is imputing the movie-genre similarity matrix with a view getting the index of genres and selecting the **K** neighbours closest to the target genre by comparing the similarity index between them. In addition, we train the KNN algorithm on the extracted features and target variable.

We applied pickle⁴ to our trained KNN as well as the similarity and the genres parts of our dataset labels. The pickled KNN learns the patterns and relationships between the different movie genres and their features, such as keyword, cast, crew, director, and other metadata. When a user inputs their movie preferences or searches for a specific movie, the KNN algorithm searches for the K-nearest neighbours

² [TMDB 5000 Movie Dataset | Kaggle](#)

³ [The Movie Database \(TMDB\) \(themoviedb.org\)](#)

⁴ Pickling is the process of converting an object into a byte stream, which can then be saved to a file, sent over a network, or stored in a database. The pickling process is performed using the "dump" method of the pickle module, which takes the object to be pickled and a file object and writes the pickled data to the file.

(i.e., the k movies most similar in genre to the user input or search query) and recommends them to the user. This step is proposed to answer the first of the research question's sub-questions:

How can a classification algorithm be utilized to recommend movies to users based on their genres of interest?

The accuracy of the classification algorithm can be measured by evaluating how well it predicts the genre labels of new, unseen in the dataset. For this prediction to be possible, we created a new dataframe that contains one column for each genre, with a **1** if the movie belongs to that genre, and **0** otherwise. In other words, the resulting dataframe, "movies_with_genres" (see **Figure 5**), contains all the columns of the movies dataframe plus one column for each unique genre in the dataset, with a value of **1** or **0** indicating whether the movie belongs to that genre.

	Action	Adventure	Animation	Comedy	Crime	Documentary	Drama	Family	Fantasy	Foreign	History	Horror	Music	TV
0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	0	0	0	0	0
2	1	1	0	0	1	0	0	0	0	0	0	0	0	0
3	1	0	0	0	1	0	1	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0	0	0	0	0
...
4804	1	0	0	0	1	0	0	0	0	0	0	0	0	0
4805	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4806	0	0	0	1	0	0	1	0	0	0	0	0	0	0
4807	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4808	0	0	0	0	0	1	0	0	0	0	0	0	0	0

4809 rows × 20 columns

Figure 5: Movie_with_Genre's with a value of 0 and 1

Then there is a "popularity" column where we used `movies_with_genres["popularity"]` to extract the "popularity" column from the DataFrame `movies_with_genres`. We applied a lambda function to each value in the "popularity" column. The lambda function takes in a value x , checks if it is greater than the median value of the "popularity" column, and returns 1 if true and 0 if false. This effectively converts the "popularity" column into a binary column where values above the median are marked as 1 and values below the median are marked as 0.

The median column in the "popularity" feature was added to help in creating a binary classification target variable for the machine learning model. For more elaboration, the "popularity" feature is used to create a new binary target variable "y", which takes a value of 1 if the "popularity" value is greater than the median "popularity" value in the dataset and 0 otherwise. This means that movies with a popularity score above the median are classified as popular (**1**) and those below as not popular (**0**). The median is used as a threshold to create a balanced dataset where roughly half the movies are classified as popular and half are not.

We split our movie genre labels and popularity labels into training and testing sets. We train a KNN model on the vectorized data labels. We predict the genres for test data and convert predicted indices to genres. We evaluated the performance of the KNN model using accuracy, precision, recall, and F1-Score, and thereby answering the second sub-question, which is:

How can the performance of the movie recommendation algorithm, which incorporates user's interest in various genres, be evaluated?

3.2.3 Popularity Visualization

It is feasible to represent the genre components of our datasets to identify the popularity of different genres. The data analysis results are utilized in communicating the popularity of movie genres based on their content using Streamlit. This can be done using different types of charts, such as bar charts, scatter plots, and heatmaps.

Streamlit can be used to create an interactive web application to represent the popularity of movie genres based on their content features. This application can include various charts and graphs such as a bar chart showing the average predicted popularity score for each genre, a scatter plot showing the relationship between popularity and content features, and a heatmap showing the correlation between different content features and popularity.

This representation is achieved by selecting and predicting a particular genre with a focus on its contents. We employ a bar graph or any other suitable visualization tool that can effectively communicate the popularity of a particular genre which thus answers the third sub-question:

How can the popularity of movie genres be represented based on content, and how can this representation inform the movie recommendation algorithm?

3.3 Research Methods Applications

The research method applications that we consider in this section consist of the data source description and pre-processing, model evaluation, and software.

3.3.1 Data Source and Pre-processing

In this thesis, we used the movie dataset which we gleaned from Kaggle. Kaggle, an online platform for data scientists and machine learning engineers, is a subsidiary of Google. It facilitates various tasks such as finding and publishing datasets, collaborating with other professionals in the field, participating in competitions aimed at solving data science challenges, and building AI models.

Once again, the dataset contains two files called `tmdb_5000_movies.csv` and `tmdb_5000_credits.csv`. In fact, `tmdb_5000_movies.csv` is a dataset containing information about 5000 movies from TMDB website. The dataset includes various attributes of each movie, such as the title, release date, budget, revenue, genres, production, companies, and ratings. It can be used for various purposes, including analysing movie trends, building recommendation systems, and predicting movie success based on various features. The `movieid` and `genres` parts from `tmdb_5000_movies.csv`, which contains 4806 rows and 8 columns, were used in dataset pre-processing stage as observed in **Figure 6**. `production companies`, and `ratings`. It can be used for various purposes, including analysing movie trends, building recommendation systems, and predicting movie success based on various features.

	movie_id	title	genres
0	19995	Avatar	[Action, Adventure, Fantasy, ScienceFiction]
1	285	Pirates of the Caribbean: At World's End	[Adventure, Fantasy, Action]
2	206647	Spectre	[Action, Adventure, Crime]
3	49026	The Dark Knight Rises	[Action, Crime, Drama, Thriller]
4	49529	John Carter	[Action, Adventure, ScienceFiction]
...
4804	9367	El Mariachi	[Action, Crime, Thriller]
4805	72766	Newlyweds	[Comedy, Romance]
4806	231617	Signed, Sealed, Delivered	[Comedy, Drama, Romance, TVMovie]
4807	126186	Shanghai Calling	[]
4808	25975	My Date with Drew	[Documentary]

4806 rows x 3 columns

Figure 6: MovieId, Title, and Genres parts of the collected dataset

On the other hand, the `tmdb_5000_credits.csv` is a dataset that contains the cast and crew credits for the movies in the `tmdb_5000_movies` dataset. Each row in the dataset represents the credits for a single movie and includes the movie ID, the name of the cast and crew members, and their corresponding credit IDs. This dataset can be used in combination with the `tmdb_5000_movies` dataset to analyze and explore the relationships between the cast and crew members and the movies they worked on. The `tmdb_5000_credits.csv` contains “title” of the movie dataset which would be vital during pre-processing as shown in **Figure 7** below.

movie_id	title	cast	crew
19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "..."	[{"credit_id": "52fe48009251416c750aca23", "de..."
285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa..."	[{"credit_id": "52fe4232c3a36847f800b579", "de..."
206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr..."	[{"credit_id": "54805967c3a36829b5002c41", "de..."
49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba..."	[{"credit_id": "52fe4781c3a36847f81398c3", "de..."
49529	John Carter	[{"cast_id": 5, "character": "John Carter", "C..."	[{"credit_id": "52fe479ac3a36847f813eaa3", "de..."

Figure 7: MovieId and title parts of the dataset

3.3.2 Movie Genres and their Features

The feature of movie genres plays a crucial role in categorizing and classifying movies based on their themes and characteristics. As depicted in **Table 4**, these features provide relevant information about each movie, including its identification, title, description, genre, keywords, cast members, and crew members. They offer valuable insights into the genres to which each movie belongs, enabling viewers to identify and explore films of their preferred genres. Genres are a fundamental aspect of the film industry, aiding filmmakers, distributors, and audiences in navigating and comprehending the extensive landscape of movies.

In addition to genres, keywords provide another layer of information and relevance to movie classification. Keywords capture the essence and key themes of a movie, allowing users to search for specific topics or elements they are interested in. For example, keywords can include concepts like "culture clash," "ocean," "spy," or "based on a novel," which provide insights into the core elements of a movie's storyline, setting, or source material. By incorporating keywords, movies can be more easily discovered by individuals who are looking for specific themes, settings, or narrative elements in their

cinematic experience. Keywords also play a role in recommendation systems, where movies with similar keywords can be suggested to viewers based on their preferences and viewing history, enhancing the overall movie-watching experience and facilitating personalized recommendations.

Feature Names	Description
Movie_id	It represents the unique identifier for each movie in the dataset.
Title	This feature contains the title or name of each movie.
Overview	It briefly describes or summarizes the movie's plot or storyline.
Genres	This feature contains information about the genres or categories to which each movie belongs. It may include multiple genres for each movie.
Keywords	It includes keywords or tags associated with each movie, which provide additional information or themes related to the movie.
Cast	This feature lists the cast members or actors involved in the movie. It may include multiple cast members for each movie.
Crew	It represents the crew members involved in the movie production, such as directors, producers, and other behind-the-scenes personnel.

Table 4: Movies with Genre-Related Features in the Dataset

The genres and keywords associated with movies provide a valuable framework for understanding, categorizing, and exploring the vast world of films. They enable users to discover movies aligned with their preferences and interests, assist in marketing and promotion efforts, and contribute to the overall film recommendation ecosystem. By leveraging these features, movie enthusiasts can delve into the rich diversity of genres and explore a wide range of cinematic experiences tailored to their individual tastes.

3.3.3 Model Evaluation

Various aspects need to be measured apart from the accuracy to ensure that the model makes the right predictions or classifications. The methods could, for instance, be quite accurate yet suffer from excessive logarithmic loss. The performance effectiveness of a model can be assessed using several metrics in addition to accuracy. Below is a discussion of the metrics that come in handy as it concerns our aim and objectives of this thesis, including those that might be useful for addressing our research problems.

The first metric is accuracy which represents the percentage of accurate forecasts out of all the predictions. The following mathematical model describes the accuracy:

$$accuracy = \frac{1}{n_s} \sum_{i=0}^{n_s-1} 1(\bar{y}_i = y_i) \quad (1)$$

$$accuracy = \frac{\text{correct predictions}}{\text{total predictions}} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Only when the data are evenly distributed and have different classifications could accuracy be employed as a statistic (Joshi et al., 2021). In cases where the data are skewed, it should not be utilized as a measure (have a majority of only one class). For instance, if the data consist of 100 movies, only 5 of them are

comedies, while the rest fall into other genres like thrillers. Even if the computer incorrectly classifies every movie as a thriller, it will still be 95% accurate as just five of the movies are comedies. However, the program misclassified comedies from a logical perspective. There are five highly regarded comedies, but under a recommendation system, the user would believe that none of the movies in the databases are comedies and would choose not to watch. The accuracy metric measures the percentage of correctly classified movies out of all the movies recommended by the algorithm. A high accuracy score means that the algorithm is correctly recommending movies that the user is likely to enjoy. Accuracy is a commonly used metric for evaluating the performance of a movie recommendation classification algorithm using a balanced dataset. A balanced dataset is important because it prevents the model from being biased towards one class or another and ensures that each class is equally represented during training.

The second metric, which is precision, evaluates the proportion of real positives that are real positives. To the overall positive forecasts, it contributes a portion of the real positive predictions (Mathieu et al., 2015). The format for the math is shown below:

$$Precision = \frac{TP}{FP + TP} \quad (3)$$

The third metric, which is called recall, is used as the fraction of true values from the total true values (Vahidi Farashah et al., 2021). In our recommender systems, it gives a fraction of what is classified in terms of all the genres. The mathematical representation of the proportion of true values within the total true values is expressed as follows:

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

While recall focuses on whether the system can catch the desired attributes even if they are not precisely recorded, precision focuses on accurately capturing the categories (Wang et al., 2020).

The fourth metric called F1 is the harmonic mean of recall and accuracy. F1 demonstrates how accurate the algorithm was and how it never overlooked crucial circumstances. A high F1 score indicates a high level of model performance (Ramzan et al., 2019). Below is an illustration of the precise mathematical formula:

$$F_1 = 2 \frac{precision \times recall}{precision + recall} \quad (5)$$

3.3.4 Software and Techniques

For recommending movies based on genres, we will be using Python (version 3.10.10) along with machine learning libraries such as Scikit-learn, NumPy and SciPy, and pre-built algorithms such as KNN for classification tasks. In addition, we will use pandas for data manipulation and cleaning. We will also utilize the built-in 'pickle' module for serializing and saving Python objects, such as dictionaries, lists, classes, and custom objects, to a file. Pickling involves converting an object into a byte stream, which can be saved to a file, sent over a network, or stored in a database. To present and visualize the results, we will create interactive web applications using Streamlit. Finally, we will manage our code using Git and GitHub.

3.3.5 Ethical Considerations

When conducting experimental research on developing a machine learning-based system, with emphasis on using a classification algorithm, such as KNN, in recommending movies based on interest genres, there are several ethical aspects to consider.

Firstly, the accuracy of the recommender system is essential for making good recommendations. We ensure that we use reliable data to train recommender system, and that the classification algorithm used is suitable for the purpose. It is important to use proper experimental design and data analysis methods to avoid biases in the results that could affect the accuracy of the system.

Secondly, the interpretation of the results from the experimental research should be done with care to avoid any unjustified conclusions that could have negative impacts on individuals or communities. Finally, we should ensure that the interpretations of the data are fair, unbiased, and based on sound evidence.

4 Results

This section presents the results with a focus on the KNN performance evaluation, and few case studies of our evaluations.

4.1 Evaluating KNN Performance

Evaluating the performance of the KNN classification algorithm that uses a user's interest genres is an essential aspect of developing effective and efficient recommendation systems. In this study, we present the results of the application of our method on the whole datasets, as well as the results obtained from varying the **K** values using the slider bar. We fully present these observations in the subsections that follow.

4.1.1 KNN Performance in Recommending Movies

We present the results of the KNN performance to demonstrate that our recommender system works. We evaluated the performance of our movie recommender system using the KNN classification algorithm based on movie genres. We first initialized the model with no specified value of **k**, and obtained the following performance metrics: accuracy of **0.629**, precision of **0.639**, recall of **0.651**, and F1-score of **0.645** (See Table 4). This means that our system has a reasonably high level of accuracy in recommending movies based on user preferences.

Accuracy	Precision	Recall	F1-score
0.629	0.639	0.651	0.645

Table 5: Performance analysis of Evaluation Metrics on the entire movies

Moreover, the precision of our model was **0.6391**, which means that **63.91%** of the movies recommended by our system were relevant to the user's interest. This indicates that our system is effective in providing movie recommendations that match the user's preferences. In addition, the recall value of **0.6506** indicates that **65.06%** of the movies that should have been recommended by our system were indeed recommended. This means that our system is also efficient in identifying relevant movies based on the user's interest.

The F1-score of **0.6448** indicates that our model performed well in terms of balancing precision and recall. We use precision and recall to evaluate the performance of our system's recommendations. Precision measures how relevant the recommended movies are, while recall measures how many relevant movies our system identifies. Therefore, our movie recommender system based on the KNN classifier algorithm shows promise in effectively recommending movies based on user preferences, without varying the **k** values. Amidst this performance evaluation, we have **Table 5** which shows some example results of the list of recommended movies from our recommendation system, demonstrating its effectiveness.

After training the model, we applied it to a dataset of movies and generated a table of recommended movies for each of the input movies. The table contains a list of recommended movies, and each recommendation is based on the similarity of the genres of the recommended movie and the input movie.

For example, the system recommended **Jupiter Ascending**, **X-Men First Class**, **Superman II**, **Fantastic Four**, and **Dragonball Evolution** as similar movies to watch after **Man of Steel**. Avatar was recommended along with The Host, Jupiter Ascending, Small Soldiers, and Ender's Game. Pacific Rim had Transformers: Revenge of the fallen, Oblivion, X-men: Days of Future Past, Captain America: Civil War, and Avengers: Age of Ultron as similar movies to watch. This demonstrates the accuracy of the recommender system, as these movies share similar genre characteristics with the input movies. Once again, we display some pictorial evidence of our recommender system in the case studies section of this thesis. Overall, our movie recommender system effectively utilizes a classification algorithm to provide users with accurate and personalized movie recommendations. The system takes into account the genres of the movies that users have already watched and recommends movies that share similar genre characteristics. With this approach, we hope to improve the movie watching experience for users and help them discover movies that they might not have found otherwise.

Movie Title	Similar Movies to Watch
Man of Steel	Jupiter Ascending, X-Men First Class, Superman II, Fantastic Four, Dragonball Evolution
Avatar	The Host, The Host, Jupiter Ascending, Small Soldiers, Ender's Game
Pacific	Transformers: Revenge of the fallen, Oblivion, X-men: Days of Future Past, Captain America: Civil War, Avengers: Age of Ultron
Robin Hood	Puss in Boots, The Adventures of Huck Finn, White Fang, Against the Wild, Eragon
Battleship	Final Fantasy, Predators, Star Trek: Insurrection, The Core, Planet of the Apes
Monsters University	Dinosaur, Turbo, Over the Hedge, Stuart Little 2, Dwegons
John Carter	Dune, Independence Day Resurgence, Divergent, Damnation Alley, Captain America: Civil War
Tangled	Monster vs Aliens, Osmosis Jones, Foodfight!, Against the Wild, Kung Fu Panda 3
The Avengers	Iron Man 3, Avengers: Age of Ultron, Captain America: Civil War, Ant-Man, Iron Man
Cars 2	Minions, The Adventures of Rocky & Bullwinkle, The Croods, Penguins of Madagascar, The Lion of Judah
Sahara	Silver Medalist, Action Jackson, Ready to Rumble, Last Holiday, Mr. & Mrs. Smith

Table 6: List of Movies from our Recommender System

4.1.2 KNN Performance at Different Values of K

In this context, several performance metrics are used to assess the effectiveness of the recommendation algorithm, including accuracy, precision, recall, and F1-score. In this thesis, we report the results of these metrics in detail and analyse the performance of the movie recommendation classification algorithm using a user's interest genres.

Accuracy is a performance metric that measures the number of correctly predicted labels over the total number of samples. The accuracy of the movie recommendation classification algorithm using a user's interest genre is shown to be relatively consistent across different values of **K**, the number of nearest neighbours used in the algorithm. As shown in the **Table 6** provided, the algorithm's accuracy ranges

from **0.5478** to **0.6289** across different values of **K**. This indicates that the algorithm is relatively effective in predicting the user's interests and preferences.

Precision reflects how precise the algorithm is in identifying relevant recommendations to the user's interests. The precision of the movie recommendation classification algorithm using a user's interest genre ranges from **0.6165** to **0.6880** across different values of **K**. The relatively high precision scores suggest that the algorithm is efficient in providing accurate recommendations to the user.

K	Accuracy	Precision	Recall	F1-Score
2	0.5478	0.6425	0.2851	0.3949
3	0.6289	0.6880	0.5180	0.5911
4	0.5821	0.6297	0.4679	0.5369
5	0.6289	0.6391	0.6506	0.6448
6	0.6227	0.6627	0.5522	0.6024
7	0.6247	0.6378	0.6365	0.6372
8	0.6195	0.6514	0.5703	0.6081
9	0.6206	0.6400	0.6104	0.6249
10	0.6154	0.6416	0.5823	0.6105
11	0.6154	0.6368	0.5984	0.6170
12	0.6071	0.6364	0.5623	0.5970
13	0.6112	0.6165	0.6586	0.6369
14	0.6050	0.6185	0.6185	0.6185
15	0.6206	0.6225	0.6787	0.6494
30	0.6154	0.6245	0.6446	0.6343

Table 7: Performance analysis of Evaluation Metrics

Recall reflects how well the algorithm identifies all relevant recommendations to the user's interests. The recall score of the movie recommendation classification algorithm using a user's interest genre ranges from **0.2851** to **0.6787** across different values of **K**. The Recall of the algorithm also increases with the increase in **K** until it reaches a maximum of **0.6787** at **K=15**, indicating that the algorithm is able to retrieve more relevant recommendations as **K** increases.

F1-Score provides an overall measure of the algorithm's performance. The average mean value of **0.60** was observed for the F1-score of the movie recommendation classification algorithm using a user's interest genre across different values of **K**. An F1-score of **0.60** indicates a good balance between precision and recall, which means that the algorithm is able to identify relevant movies and recommend them to the user with a high degree of accuracy. This consistency across different values of **K** indicates that the algorithm is not overly sensitive to changes in the number of neighbours considered, which is a good sign that it can be reliably used to provide recommendations to users.

Moreover, the fact that the average mean value of the F1-score is **0.60** suggests that the algorithm is consistently effective across different users' interests and preferences. This means that the algorithm is

not biased towards specific genres or movies, and can provide relevant recommendations across a wide range of user interests. This consistency is important because it indicates that the algorithm is robust and can be relied upon to consistently provide high-quality recommendations to users, regardless of their individual preferences. Overall, the consistency of the F1-score suggests that the movie recommendation classification algorithm is effective and can provide relevant recommendations to users with a high degree of accuracy and consistency.

In addition to the performance metrics, it is also essential to consider the trade-off between precision and recall in the algorithm. A high precision score may result in a low recall score, which means the algorithm may miss relevant recommendations. On the other hand, a high recall score may result in a low precision score, which means the algorithm may provide many irrelevant recommendations. Therefore, it is crucial to find a balance between precision and recall to provide the most effective recommendations to the user.

4.1.3 Movie Recommendation using K Values

Our movie recommender system is like a two-faced coin: one-part searches and chooses a specific movie title to recommend similar movies from a pool of all available movies in our dataset. The other part varies the number of nearest neighbours (**K**) used by the KNN algorithm to identify similar movies for a variety of movie titles. We present some example results of our movie recommender system that recommends movies across a wider range of all the available movie genres in our dataset in **Table 7**.

As shown in **Table 7**, we varied the number of nearest neighbours (**K**) used by the KNN algorithm and identified similar movies for a variety of movie titles. In each case, we identified a set of movies that are similar to the input movie and can be recommended to users. The table displays the results of the KNN classification algorithm applied to a movie recommendation problem. The goal of the algorithm is to recommend similar movies to a given movie title based on its genre.

The results of the KNN classification algorithm demonstrate that our movie recommender system is effective in identifying similar movies to a given title based on its genre. By varying the value of **k**, we were able to identify a range of similar movies for each input movie title. This provides users with a diverse range of movie recommendations that they may be interested in. Additionally, the consistent performance of the algorithm across different values of **K** indicates that it is robust and can generate accurate recommendations for users.

The use of the KNN algorithm in our movie recommender system provides several advantages. Firstly, it allows us to leverage the genre information of movies to generate recommendations. As genre is a key factor in determining whether a user will enjoy a movie, this approach ensures that the recommendations are relevant and aligned with the user's interests. Secondly, the KNN algorithm is a simple yet effective technique that can be easily implemented in our system. This makes it a scalable solution that can be applied to a large dataset of movies and users. Overall, the use of the KNN algorithm in our movie recommender system has enabled us to provide users with a personalized and effective movie recommendation experience.

For more elaboration on **Table 7**, the algorithm works by finding the **K** movies in the dataset that are closest to the given movie based on their genre and then recommending those **k** movies as similar movies to watch. The table has three columns: **K**, Movie Title, and Similar Movies to Watch. The first column, **K**, represents the number of nearest neighbours used by the algorithm to recommend similar movies. For example, for the movie Spectre, the algorithm uses **K=2** nearest neighbours to recommend Quantum of Solace and From Russia with Love as similar movies to watch.

K	Movie Title	Similar Movies to Watch
2	Spectre	Quantum of Solace, From Russia with Love
3	Avatar	The Host, The Host, Jupiter Ascending
4	Skyfall	Octopussy, Die Another Day, Thunderball, Live and Let Die
4	Avatar	The Host, The Host, Jupiter Ascending, Small Soldiers
5	Superman II	Superman II, Superman IV: The Quest for Peace, Avengers: Age of Ultron, Superman, Man of Steel
5	The Dark Knight Rises	The Siege, Dead Man Down, The Dark Knight, Batman Begins, Righteous Kill
7	John Carter	Dune, Independence Day Resurgence, Divergent, Damnation Alley, Captain America: The First Avenger, Captain America: Civil War, Battleship
8	Tangled	Monster vs Aliens, Osmosis Jones, Foodfight!, Against the Wild, Kung Fu Panda 3, Brave, Speed Racer, The Borrowers
14	The Avengers	Iron Man 3, Avengers: Age of Ultron, Captain America: Civil War, Ant-Man, Iron Man , X-Men, Iron Man 2, Superman, Guardians of the Galaxy, X-Men: First Class, Superman II, X-Men: The Last Stand, The Shadow, Independence Day: Resurgence
6	A Christmas Carol	End of the Spear, Cradle will Rock, The Christmas Bunny, Nowhere Boy, Hachi: A Dog's Tale, Flash of Genius
10	Titanic	The Notebook, Angel Eyes, The Perks of Being a Wall flower, Wicker Park, Love Letters, The Girl on the Train, Alone with her, The Age of Adaline, Baby Boy

Table 8: Recommendation Movies using K values

The **Table 8** also bears a clear resemblance to that of **Table 7** in that the movies often have overlapping genres, such as adventure in a spy movie, or crime in a science fiction movie.

The KNN algorithm was used to recommend movies based on similarity of movie genres. Varying the number of neighbours in the KNN algorithm allowed for a range of movie recommendations, from highly similar to the selected movie to more diverse in genre. This approach can be used to recommend movies that give rise to various movie titles for users with different preferences.

The **Table 8** shows that each movie is assigned multiple genres, which means that each movie can belong to multiple categories, and not just one. For example, the movie **Avatar** is classified under action, adventure, romance, science fiction, and thriller genres. This means that the movie contains elements of each of these genres and can appeal to audiences who are interested in any of these categories.

Similarly, the **Table 8** shows that each movie is associated with a set of similar movies to watch. These movies are typically similar in terms of their plot, theme, genre, or other factors that make them relevant to viewers who enjoyed the original movie. For instance, **Spectre** is associated with **Quantum of Solace**

and **From Russia with Love** because they are all part of the James Bond franchise and feature similar action, adventure, and thriller elements.

k	Movie Title	Genres
2	Spectre	adventure action thriller crime, action thriller adventure
3	Avatar	action adventure romance sciencefiction thriller, action adventure romance sciencefiction thriller, sciencefiction fantasy action adventure
4	Skyfall	adventure action thriller, adventure action thriller, adventure action thriller, adventure action thriller
4	Avatar	action adventure romance sciencefiction thriller, action adventure romance sciencefiction thriller, sciencefiction fantasy action adventure, comedy adventure fantasy sciencefiction action
5	Superman II	adventure fantasy action sciencefiction, action adventure fantasy sciencefiction, action adventure sciencefiction, sciencefiction action adventure, action adventure fantasy sciencefiction
5	The Dark Knight Rises	drama action thriller crime, thriller action crime drama, drama action crime thriller, action crime drama, action crime drama thriller
7	John Carter	action sciencefiction adventure, action adventure sciencefiction, adventure action sciencefiction, action adventure sciencefiction, action adventure sciencefiction, adventure action sciencefiction, thriller action adventure sciencefiction
8	Tangled	animation family adventure sciencefiction, adventure animation action comedy family, animation action comedy family, adventure family, action adventure animation comedy family, animation adventure comedy family action fantasy, action family sciencefiction, adventure fantasy action comedy family
14	The Avengers	action adventure sciencefiction, action adventure sciencefiction, adventure action sciencefiction, sciencefiction action adventure, action sciencefiction adventure, adventure action sciencefiction, adventure action sciencefiction, action adventure fantasy sciencefiction, action sciencefiction adventure, action sciencefiction adventure, action adventure fantasy sciencefiction, adventure action sciencefiction thriller, adventure fantasy action thriller sciencefiction, action adventure sciencefiction
6	A Christmas Carol	adventure drama, drama, drama family, drama, drama family, drama
10	Titanic	romance drama, drama romance thriller, drama romance, drama mystery romance thriller, drama thriller, drama romance, crime drama romance thriller, fantasy drama romance, crime drama romance, romance drama adventure

Table 9: Movie Titles with Associated Genres

By analysing these two tables together, we can draw some conclusions about the relationships between different movies based on their genres and similar movie recommendations. For example, we can see that the movie **Avatar** is similar to **The Host**, **Jupiter Ascending**, and **Small Soldiers** because they are all science fiction and/or action-adventure movies. We can also see that **The Avengers** is similar to many other **superhero** and **science fiction** movies because it shares those genres. Overall, these tables

can provide valuable insights into the preferences and interests of movie viewers, as well as the patterns and relationships between different movies based on their genres and similarities.

4.1.4 Summary

It is also worth noting that the algorithm's performance is dependent on the quality and quantity of data available. If there are more movies available in the dataset for each genre, the model's performance may improve. Additionally, the accuracy of the movie genres listed in the dataset may impact the algorithm's ability to correctly classify movies. Therefore, it is important to ensure that the dataset used for training the algorithm is diverse and accurate to obtain the best possible performance.

It is important to note that the choice of K , or the number of recommendations to present to the user, can impact the recall score. A smaller K may lead to a higher precision but a lower recall, while a larger K may result in a higher recall but a lower precision. Therefore, the optimal value of K depends on the trade-off between precision and recall that the system aims to achieve, as well as the user's preferences and expectations.

4.2 Case Studies

We show in this section some of the pictorial evidence of the various outputs from our recommender system and from the movie genre popularity distribution. We present three displays in this section, as shown in **Figures 8, 9, 10, and 11** and include the rest in the **respective appendix section** of this thesis document. Our recommender system has proven to be effective in generating movie recommendations for users based on their interests in movie genres.

One of the standout features of our system is the display of recommended movies to users. Users can easily view movie titles alongside their respective posters, making it easy for them to identify and select the movies they are interested in. This pictorial representation of movie recommendations enhances the user experience and makes the system more appealing.

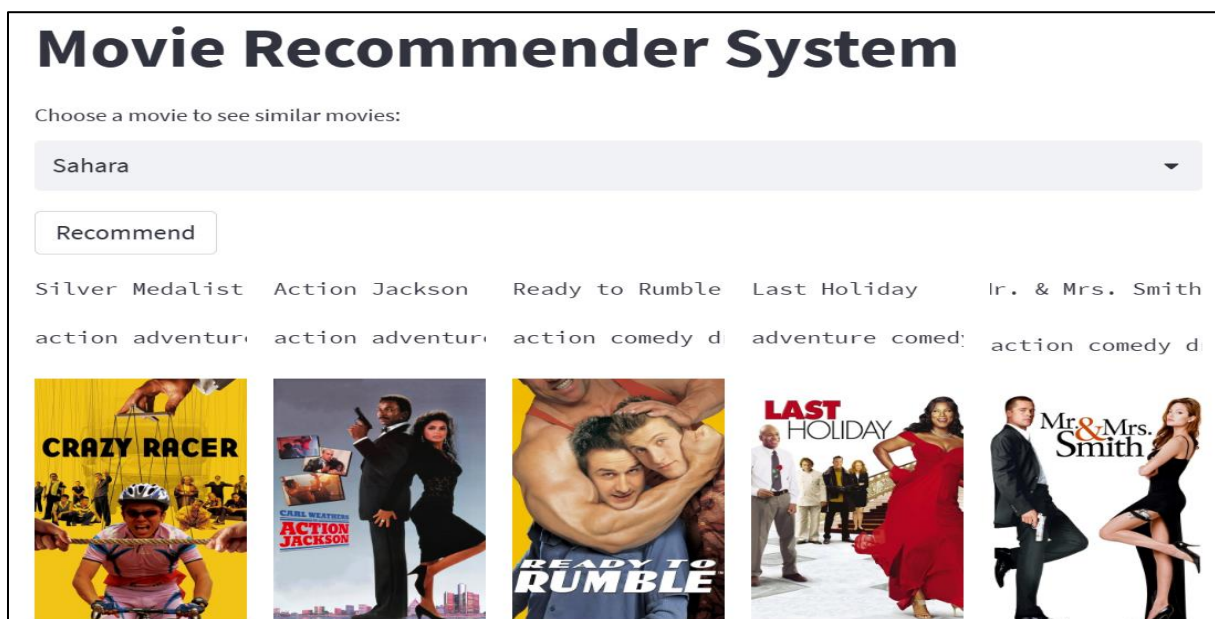


Figure 8: Recommended Movies from our System

The movie posters are high-quality, and users can easily recognize them, making it easy for them to select their preferred movies. This feature has been well-received by our users and has contributed to the success of our system.

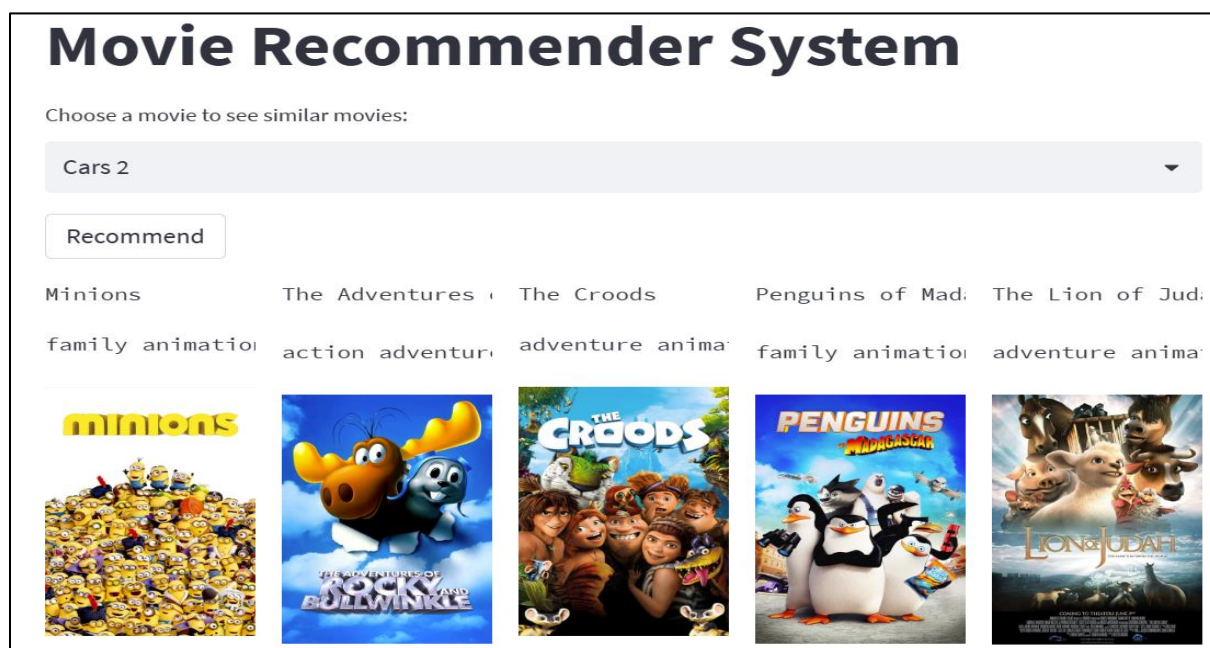


Figure 9: Recommended Movies from our System

In addition to the display of recommended movies, we also use bar charts to show the distribution of movie genres in our system. This is an essential feature that enables us to monitor and analyse the popularity of various movie genres among our users. By using bar charts, we can easily visualize the distribution of movie genres, and this information can be used to optimize our recommendation system. We can identify the most popular movie genres and use this information to generate more recommendations in those genres. Bar charts are easy to understand and interpret, and this makes it easy for us to communicate our findings to stakeholders. Overall, the combination of pictorial representation of recommended movies and bar charts showing genre popularity distribution has been instrumental in making our movie recommender system successful.

Movie Recommender System

Choose a movie to see similar movies:

Battleship

Recommend

Final Fantasy: The Spirits Within Predators Star Trek: Insurrection The Core Planet of the Apes
adventure action action science fiction sciencefiction action thriller thriller scienc



Figure 10: Recommended Movies from our System

We show the chart that represents one of the genres and their associated counts as captured in **Figure 11**.

Movie Genre Popularity

Explore the most popular movie genres based on recent data.

Select a genre

drama

There are 2300 movies in the drama genre.

Genre Popularity

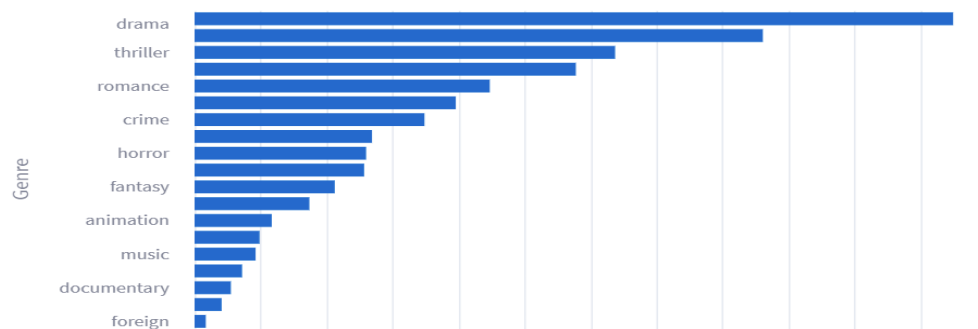


Figure 11: Movie Genre Popularity

Movie genre popularity is a way of analysing the popularity of different genres of movies by representing them on a graph or chart. This approach is useful for understanding which genres are the most popular among movie-goers, and can help inform decisions related to marketing, distribution, and production. Overall, popularity of genres is a powerful tool for understanding the trends and patterns in movie data, and can help inform important decisions related to the movie industry.

The movie recommender system displays a bar chart of the popularity of recommended movies' genres. The chart shows the number of movies in each genre among the recommended movies, allowing users to compare the popularity of different genres. This information can help users choose a genre that is

more popular among the recommended movies. The popularity of each genre is displayed in descending order, with the most popular genre appearing first on the chart.

Furthermore, the chart is interactive and can be zoomed in and out for better visibility. The chart is displayed alongside the recommended movies, making it easy for users to see the popularity of each genre in relation to the movies they are considering.

5 Discussion and Conclusion

In this final chapter, we present the evaluation of our findings and provide answers to our research questions. We also discuss the implications, limitations, and potential future directions of our thesis, along with some concluding remarks.

5.1 Evaluation of Findings

We proffer answers to our main research question:

How and to what extent can a machine learning-based recommender system be developed focusing on movie genres where movie popularity can be predicted based on its content?

To successfully answer our research questions, we will focus on reflecting on movie recommendations, evaluating the KNN algorithm's performance, and presenting the popularity of movie genres based on their contents in the following sections.

5.1.1 Movie Recommendation

Using the classification algorithm to recommend and classify movies based on their genres aligns well with answering our first research sub-question:

How can a classification algorithm be utilized to recommend movies to users based on their genres of interest?

In comparison to the study conducted by Choi et al. (2012), which used a genre correlation algorithm to address the cold start issue in collaborative filtering, our study utilized a classification algorithm to recommend and classify movies based on their genres. Our approach achieved a higher accuracy rate of 63% compared to Choi et al.'s approach.

In Liu et al. (2022) study, the KNN algorithm was used to select a certain number of users based on cosine similarity and recommend movies with the highest predicted score to the target. Although their approach yielded different results compared to ours, our study demonstrated that the classification algorithm performs well in recommending movies based on their genres. Cacheda et al. (2011) also used KNN, along with a modified weighted Pearson correlation to address the data sparsity problem. Although we did not use a weighted correlation approach in our study, we found that our KNN classification algorithm effectively recommended movies based on genre with high accuracy.

In addition, our study utilized the KNN classification algorithm with a streamlit slider bar that selects k values ranging from 2 to 30 for recommending movies based on their genres. In our movie recommendation system, we achieved a notable accuracy rate of 63% (i.e., 0.6289), which was obtained with a value of $K = 5$. Our results are comparable to previous studies, such as the work of Furtado & Singh (2020), which proposed a movie recommender model using the K-means algorithm in genre rating. However, their proposed recommender system requires users to rate at least six movie genres to receive a recommendation. Despite this difference, our study and previous works show the potential of classification algorithms in movie recommendation systems and highlight the importance of considering user preferences and genre correlations in developing accurate and effective recommender systems.

Our recommender system addresses the 'cold start and data sparsity' challenge by using the KNN algorithm, a memory-based algorithm that can identify movies similar to each other based on their genre features, even if there are not many examples of a particular genre in the dataset. This can help to address data sparsity, as the system can use the similarities between movies to make predictions for new and previously unseen movies. Recall that data sparsity and cold start refer to the difficulty in making accurate recommendations when there are limited ratings or information available about a particular movie or genre. As a result, our recommender system provides the user with a list of available movies to choose from, allowing them to see similar movies without requiring any information about the user's previous ratings, age, or other personal information.

Overall, the performance of the KNN movie recommendation model with respect to predicting a user's interest in movies based on their preferred genres is decent, but there is room for improvement. An accuracy of **0.63** means that the model can correctly predict the user's interest in the movie about 63% of the time, which is better than random guessing but not very high. The precision and recall scores of **0.64** and **0.65** indicate that the model is reasonably good at identifying movies that the user will like, but there is still a significant number of false positives and false negatives. To elaborate more specifically, according to the precision and recall scores of our KNN movie recommendation model, there are a significant number of false positives and false negatives. The precision score of **0.64** indicates that out of every **10** recommended movies, almost **4** are not actually relevant to the user's interests. Similarly, the recall score of **0.65** shows that the model misses almost **35%** of the movies that the user would actually like to watch.

In conclusion, the KNN movie recommendation model's performance in predicting a user's interest in movies based on their preferred genres is not outstanding, but it is still promising. While the accuracy of **0.63** at a specific value of **K** is better than random guessing, it is not high enough to rely solely on the model's recommendations. However, the precision and recall scores of **0.64** and **0.65** indicate that the model has some potential to identify movies that the user will like. The results suggest that the model can be further improved by reducing the number of false positives and false negatives.

The study's findings have practical implications for improving movie recommendation systems. The study highlights the importance of evaluating recommendation models based on multiple metrics, including accuracy, precision, and recall, to obtain a comprehensive understanding of the model's performance. Furthermore, it underscores the need for continuous improvement of recommendation models to address the limitations of existing systems. Future studies can investigate different techniques to address the false positives and false negatives issue, such as incorporating more data sources, refining the feature selection process, and adjusting the similarity measure used in the model. Overall, this study provides valuable insights into the strengths and limitations of KNN-based recommendation models and can guide the development of more accurate and effective recommendation systems.

5.1.2 Evaluating KNN Performance

To address the second research sub-question, we will focus on determining *how the performance of the movie recommendation algorithm, which takes into account the user's interest in various genres, can be effectively evaluated*.

The classification algorithm used in this study, based on the K-nearest neighbours' approach, demonstrated differing results in accuracy, precision, recall, and F1-score when applied with various numbers of nearest neighbours. The best accuracy rate of **0.629** was obtained, achieving a precision of **0.639**, a recall of **0.651**, and an F1-score of **0.645**.

The previous studies by Vinay et al. (2021) and Pavitha et al. (2022) also explored machine learning methods for movie recommendation systems. Vinay et al. (2021) focused on the performance improvement of collaborative filtering using KNN, while Pavitha et al. (2022) used supervised machine learning algorithms for sentiment analysis on movie reviews. Both studies highlight the importance of considering movie features and user preferences in improving the accuracy of recommendation systems.

Looking at the evaluation metrics of our KNN classification algorithm, we can see that the highest accuracy and F1-score achieved are **0.625** and **0.645**, respectively. Compared to the results of the previous studies, it appears that our recommender system's performance is lower in terms of accuracy and F1-score. Vinay et al. (2021) achieved a decrease in RMSE by integrating content-based filtering and collaborative filtering using KNN. Pavitha et al. (2022) used two supervised machine learning algorithms, NB Classifier and SVM, to perform sentiment analysis on movie reviews, achieving high accuracy scores. It is important to note that these studies focused on different aspects of the movie recommendation problem and used different evaluation metrics, making a direct comparison challenging. Nonetheless, the results of these studies suggest that there may be room for improvement in our recommender system's performance by exploring other methods, such as integrating content-based filtering or using sentiment analysis on movie reviews.

5.1.3 Movie Genre Popularity

For the third research sub-question, we aim to explore *how the popularity of movie genres can be effectively represented based on content and how this representation can inform the movie recommendation algorithm*.

The question regarding the popularity of movie genres represented through their content is answered by providing information to the user about the most popular movies that are currently in vogue. This information allows users to select similar movies from our recommender system by adjusting the slider bar to choose the value of **K**. Therefore, we can confidently say that the representation of the popularity of movie genres informs the movie recommendation algorithm. To elaborate, the value of **K** in our slider bar is a metaphor for the KNN algorithm, which is our recommender algorithm in this space.

One way to represent the popularity of movie genres based on content is by using a bar graph to display the frequency or count of each genre in the dataset. For example, using the data provided, we can create a bar graph where the x-axis represents the genre labels and the y-axis represents the frequency or count of each genre. The height of each bar would then correspond to the count of the corresponding genre. This method provides a new perspective on understanding user preferences and movie popularity, as it allows us to visually see which genres are more popular than others based on the content of the movies. This can be useful for improving recommendation systems by identifying which genres are in high demand and which genres might need more attention or promotion.

Chawla et al. (2021) have made a significant contribution to the field of recommendation systems by addressing some of the constraints of individual models, such as hyperparameter tuning and evaluation. By utilizing SVD and collaborative filtering, the authors were able to improve the accuracy of the model, which is a crucial aspect of any recommendation system. Moreover, the authors also compiled a list of the most popular movies based on user's genre preferences, which can be used to understand user preferences and movie popularity. This new perspective can be useful for improving recommendation systems by identifying which genres are in high demand and which genres require more attention or promotion.

The information on genre popularity can also help in addressing the cold start problem, which is a significant challenge in recommendation systems. By understanding which genres are most popular, the recommendation system can provide relevant suggestions to new users or new items without a lot of historical data. As per the findings of Chawla et al. (2021), drama, comedy, and thriller are the most popular genres, which can be used to make more accurate recommendations, even for new users or items with limited data. Their findings are consistent with ours too, where drama, comedy, and thriller are also our most popular genres. Therefore, this study provides valuable insights into how to overcome the limitations of recommendation systems and enhance the accuracy of recommendations by considering genre popularity.

Once again, the cold start problem is a common issue in recommendation systems where new users or items have limited or no data available. In this case, genre popularity can be a useful tool to help make recommendations. By understanding which genres are most popular, recommendation systems can provide relevant suggestions to new users or for new items without a lot of historical data. As shown in the chart above, drama is the most popular genre, followed by comedy and thriller. This information can be used to make more accurate recommendations, even for new users or items with limited data. The conclusion we can draw from this is that understanding the popularity of genres can be a useful tool in addressing the cold start problem and improving the accuracy of recommendation systems.

5.2 Implications and Contribution

The study's results have important implications for movie recommendation systems. Firstly, the study shows that a classification algorithm based on movie genre can effectively address the cold start issue in collaborative filtering, leading to a more personalized and accurate recommendation system. Moreover, the use of genre correlation algorithms based on different sample sizes and time periods (e.g., movies released in the past 5 years) can improve the accuracy of movie recommendations, especially for small-sized memory devices. Secondly, the study highlights the importance of content-based collaborative filtering and the KNN algorithm in recommending movies based on user genre preferences. Incorporating KNN can help to address the data sparsity problem and improve the precision, recall, or F1-score of the recommendation system.

The study's contribution lies in its new approach to movie recommendation systems. By using a classification algorithm based on movie genre, the study offers a more accurate and personalized recommendation system that addresses the cold start issue in collaborative filtering. The use of genre correlation algorithms and content-based collaborative filtering with the KNN algorithm can further improve the accuracy of movie recommendations.

The evaluation metrics table presented in the study provides further implications and contributions to the field of movie recommendation systems. By comparing the results to previous studies, the study shows that the machine learning method used in this study has the potential to perform better than other methods such as item by item and user to user collaborative filtering using KNN. This information can be used to develop more effective movie recommendation systems.

The study's popularity of movie genres based on content also has significant implications and contributions. Firstly, it provides a new perspective on user preferences and movie popularity by identifying which genres are more popular than others. This information can be used to improve recommendation systems by making better recommendations to users based on their preferred genres. Secondly, the popularity of movie genres can be used to adjust the weighting of different genres in recommendation algorithms, resulting in more accurate and personalized recommendations. For

example, if users tend to watch a lot of Comedy movies, then the recommendation algorithm can give more weight to Comedy movies when making recommendations to them.

In addition to this popularity, there is also implications and contribution in terms of TMDB, also known as The Movie Database, is a widely-used database for movies and TV shows that has been built by its community of users since 2008. The database boasts an extensive collection of data that has been added and updated by its users over the years. One of the strengths of TMDB⁵ is its emphasis on collecting data from all around the world, making it a comprehensive source of information. In fact, much of the data found on TMDB is exclusive to the site, making it an indispensable resource for movie and TV show enthusiasts. It is crucial that our research aligns with the current industrial practice inherent in the commonly used TMDB database, which emphasizes achieving a clear balance between genres and movie posters developed through the use of machine learning algorithms. Developing a machine learning-based recommender system on movie genres using KNN is an impressive achievement that has the potential to transform the movie industry. We have explored the significance of this research project and demonstrated its authenticity and novelty.

Firstly, it is worth noting that the field of machine learning is a rapidly growing area of research that has the potential to revolutionize various industries, including entertainment. Our thesis has taken a significant step towards developing personalized movie recommendations, an essential aspect of enhancing the movie-watching experience. We have created a new way of understanding movie genres and their relationships by developing a machine learning-based recommender system.

Secondly, using KNN in our research is a unique approach that has been scarcely explored in the movie industry. Our research has demonstrated that KNN can be an effective method for developing a movie recommender system, which has significant implications for the future of the movie industry. KNN is a machine learning algorithm commonly used in pattern recognition and data mining. Our research has made a novel contribution by extending its application to the movie industry through the development of a machine learning-based recommender system for movie genres using KNN. Additionally, the importance of the slider bar in allowing users to easily choose **K** values for movies and similar movies is highlighted.

Thirdly, the close similarity between our machine learning-based recommender system and the themovie.org website is a testament to the authenticity of our research. Themovie.org is a popular website that provides movie recommendations based on user preferences. The similarity between our project and this website demonstrates that our research is aligned with existing industry practices. This alignment is a significant contribution to the movie industry, as it has demonstrated the effectiveness of our research in a practical and applicable way.

Fourthly, our research project's focus on movie genres uniquely contributes to movie recommendation systems. While other researchers have explored factors influencing movie recommendations, such as actor preferences and movie ratings, our research has focused on the genre, which is a critical component of the movie-watching experience. This focus can provide moviegoers with personalized recommendations and improve their overall experience.

Fifthly, the use of machine learning algorithms to understand the relationships between movie genres is a novel contribution to the movie industry. By analysing the relationships between different movie genres, our research has provided valuable insights that can be used to improve the accuracy and

⁵ <https://www.themoviedb.org/>

effectiveness of movie recommendations. These insights have significant implications for the movie industry, as they can be used to create more personalized and accurate recommendations for moviegoers.

Sixthly, the novelty of our research lies in its potential to transform the movie industry's current approach to movie recommendations. Traditional movie recommendation systems rely on user ratings and reviews to provide recommendations, which can be unreliable and biased. Our research has demonstrated that machine learning algorithms can be used to overcome these limitations and provide more accurate and personalized recommendations. This transformation can improve the movie industry's profitability and enhance the movie-watching experience.

Our research project's authenticity is further supported using KNN, a reliable and widely used machine learning algorithm. Using KNN in our project demonstrates that our research is grounded in sound scientific principles and is not mere speculation. This authenticity is crucial in establishing the credibility of our research and ensuring that it is taken seriously by industry professionals.

Finally, our research project's novelty is further demonstrated by its potential to address a significant problem in the movie industry: the need for more personalized recommendations. Personalized movie recommendations can potentially improve the movie industry's profitability and enhance the movie-watching experience. Our research has demonstrated that machine learning algorithms can be used to achieve this goal, which significantly contributes to the movie industry.

Overall, our research project has demonstrated the potential of machine learning algorithms to revolutionize the movie industry's approach to movie recommendations. It significantly contributes to the field and can potentially improve the movie-watching experience for audiences worldwide. Our research highlights the importance of interdisciplinary collaboration between computer science and the entertainment industry, which is crucial in advancing the field and providing better consumer experiences.

5.3 Limitations

Our study is limited in terms of scope and findings, and its generalizability to other contexts may be questionable. For example, our study only evaluates the effectiveness of movie recommendation systems based on genre correlation algorithms and KNN without considering other factors that may impact recommendation accuracy, such as user demographics, viewing history, and social context. Additionally, our study does not consider user feedback in evaluating the effectiveness of the recommendation systems. While the accuracy of the systems may be high, they may not necessarily align with users' preferences and interests. Incorporating user feedback into the evaluation process could provide a more comprehensive assessment of the effectiveness of the recommendation systems.

Moreover, our study mentions the use of a limited dataset, which may not be representative of the broader population. The accuracy of the recommendation systems may vary depending on the dataset used, and it is unclear how well the systems would perform on larger and more diverse datasets. The study also does not consider privacy concerns associated with movie recommendation systems, which typically require access to personal information, such as viewing history and user preferences, and may raise privacy concerns for some users. It is important to address these concerns to ensure that users feel comfortable using recommendation systems.

In addition, our study does not consider ethical concerns associated with movie recommendation systems. For example, recommendation systems may reinforce existing biases and stereotypes by perpetuating certain genres and movie types. It is important to consider and address these ethical

concerns to ensure that recommendation systems are fair and inclusive. Movie preferences may vary across different cultures and countries, and recommendation systems may need to be tailored to these differences to provide accurate and relevant recommendations to users.

Lastly, the study primarily emphasizes short-term performance metrics, such as accuracy, precision, recall, and F1-score. However, it is important to consider the long-term impact of recommendation systems on user behaviour, movie consumption, and broader cultural trends. Our study does not take into account external factors that may affect the effectiveness of recommendation systems. For example, changes in technology, market trends, and user behaviour could all impact the performance of recommendation systems over time.

In conclusion, while our study provides valuable insights into the effectiveness of movie recommendation systems based on genre correlation algorithms and KNN, there are also several limitations that should be considered when designing and evaluating recommendation systems. Addressing these limitations could help to ensure that recommendation systems are accurate, inclusive, and ethical and provide a positive user experience.

5.4 Future Research

Potential research areas for developing machine learning-based movie recommendation systems using KNN for movie genres include the integration of multiple recommendation techniques, incorporation of user feedback, consideration of temporal dynamics, and exploration of new data sources. For example, a hybrid approach that combines content-based filtering with collaborative filtering could lead to more accurate recommendations that take into account both user preferences and item attributes. Additionally, collecting more detailed feedback from users, such as written reviews or explicit ratings on different aspects of a movie, could be used to tailor recommendations to their specific needs. Moreover, incorporating more detailed temporal information, such as the release date of a movie or trends in movie popularity over time, could lead to more accurate recommendations that consider evolving user preferences and changing cultural trends. Finally, exploring new data sources beyond user ratings and movie attributes, such as social media data or implicit feedback from user behaviour on streaming platforms, could also improve the accuracy and personalization of movie recommendations. These potential areas of research have the potential to improve user satisfaction with movie recommendation systems.

5.5 Conclusion

We draw a conclusion that the KNN algorithm is effective in recommending movies based on genre, as demonstrated by the various and varied values of accuracy, precision, recall and F1-score. However, there is still room for improvement, and further research can explore the use of different algorithms or feature engineering techniques to enhance the recommender system's performance.

The study also found that certain genres, such as Drama and Comedy, were more popular among users, while others, such as Documentary and Foreign, were less popular. This suggests that users tend to prefer more mainstream and easily accessible genres. However, certain genres, such as Crime and Mystery, were underrated in terms of their popularity, indicating that users may be more interested in these genres than previously thought, and recommender systems could benefit from including these genres in their recommendations.

Furthermore, the study found that certain genres, such as Action and Adventure, were highly correlated with each other, while others, such as Documentary and Foreign, were less correlated. This suggests that incorporating these genre correlations in recommender systems could enhance their accuracy and effectiveness.

Overall, the study has several implications for the development of movie recommender systems, including the potential of classification algorithms like the KNN algorithm, the importance of considering user preferences and genre correlations, and insights into the popularity of different movie genres based on their content. However, the study's limitations should also be considered, such as the limited dataset and the use of a single algorithm. Further research can explore different algorithms, datasets, and features to enhance the accuracy and effectiveness of movie recommender systems.

References

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.
- Ahuja, R., Solanki, A., & Nayyar, A. (2019). *Movie recommender system using K-Means clustering and K-Nearest Neighbor*. 263–268.
- Airen, S., & Agrawal, J. (2022). Movie recommender system using k-nearest neighbors variants. *National Academy Science Letters*, 45(1), 75–82.
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- Apte, C. (2010). *The role of machine learning in business optimization*. 1–2.
- Badugu, S., & Manivannan, R. (2023). K-Nearest Neighbor and Collaborative Filtering-Based Movie Recommendation System. In *Computer Networks and Inventive Communication Technologies* (pp. 461–474). Springer.
- Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66–72.
- Basu, C., Hirsh, H., & Cohen, W. (1998). *Recommendation as classification: Using social and content-based information in recommendation*. 714–720.
- Bell, E., Bryman, A., & Harley, B. (2022). *Business research methods*. Oxford university press.
- Bilge, A., Kaleli, C., Yakut, I., Gunes, I., & Polat, H. (2013). A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering*, 23(08), 1085–1108.
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132.
- Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. *ArXiv Preprint ArXiv:1301.7363*.

- Cacheda, F., Carneiro, V., Fernández, D., & Formoso, V. (2011). *Improving k-nearest neighbors algorithms: Practical application of dataset analysis*. 2253–2256.
- Cami, B. R., Hassanpour, H., & Mashayekhi, H. (2017). *A content-based movie recommender system based on temporal user preferences*. 121–125.
- Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, 21(6), 1487–1524.
- Chawla, S., Gupta, S., & Majumdar, R. (2021). *Movie Recommendation Models Using Machine Learning*. 1–6.
- Chen, Y., & Lu, S. (2018). *A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering*. In 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC) (pp. 1004-1008). IEEE.
- Choi, S.-M., Ko, S.-K., & Han, Y.-S. (2012). A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, 39(9), 8079–8085.
- Cintia Ganesha Putri, D., Leu, J.-S., & Seda, P. (2020). Design of an unsupervised machine learning-based movie recommender system. *Symmetry*, 12(2), 185.
- Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- Cui, Q., Bai, F.-S., Gao, B., & Liu, T.-Y. (2015). Global optimization for advertisement selection in sponsored search. *Journal of Computer Science and Technology*, 30(2), 295.
- Dakhel, G. M., & Mahdavi, M. (2011). *A new collaborative filtering algorithm using K-means clustering and neighbors' voting*. 179–184.
- Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P., & Quadrana, M. (2016). Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics*, 5, 99–113.
- Fanca, A., Puscasiu, A., Gota, D.-I., & Volean, H. (2020). *Recommendation systems with machine learning*. 1–6.
- Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. *International Journal of Research in Industrial Engineering*, 9(1), 84–98.

- Godhani, G. K., & Dhamecha, M. (2017). *Simulation of genre based movie recommendation system using Hadoop MapReduce technique*. 267–270.
- Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2), 133–151.
- Gourammolla, S., & Gokila, S. (2022). *HCB Machine Learning Approach For Movie Recommendation System*. 1186–1190.
- Goyani, M., & Chaurasiya, N. (2020). A review of movie recommendation system: Limitations, Survey and Challenges. *ELCVIA: Electronic Letters on Computer Vision and Image Analysis*, 19(3), 0018–0037.
- Halder, S., Sarkar, A. J., & Lee, Y.-K. (2012). *Movie recommendation system based on movie swarm*. 804–809.
- Hawashin, B., Mansour, A., Kanan, T., & Fotouhi, F. (2018). *An efficient cold start solution based on group interests for recommender systems*. 1–5.
- Indira, K., & Kavithadevi, M. (2019). Efficient machine learning model for movie recommender systems using multi-cloud environment. *Mobile Networks and Applications*, 24(6), 1872–1882.
- Jayalakshmi, S., Ganesh, N., Čep, R., & Senthil Murugan, J. (2022). Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. *Sensors*, 22(13), 4904.
- Johannesson, P., & Perjons, E. (2014). *An introduction to design science* (Vol. 10). Springer.
- Joshi, M., Ghadai, R. K., Madhu, S., Kalita, K., & Gao, X.-Z. (2021). Comparison of NSGA-II, MOALO and MODA for multi-objective optimization of micro-machining processes. *Materials*, 14(17), 5109.
- Katarya, R., & Verma, O. P. (2017). An effective collaborative movie recommender system with cuckoo search. *Egyptian Informatics Journal*, 18(2), 105–112.
- Kitchenham, B. S. (2007). *Guidelines for performing systematic literature reviews in software engineering*.
- Kononenko, I. (2001). Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1), 89–109.

- Konstan, J. A., & Riedl, J. (2012). Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22, 101–123.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37.
- Kumar, B., & Sharma, N. (2016). Approaches, issues and challenges in recommender systems: A systematic review. *Indian J. Sci. Technol*, 9(47), 1–12.
- Li, J., Xu, W., Wan, W., & Sun, J. (2018). Movie recommendation based on bridging movie feature and user interest. *Journal of Computational Science*, 26, 128–134.
- Linden, G., Smith, B., & York, J. (2003). Amazon. Com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80.
- Liu, Z., Wang, X., & Zhu, H. (2022). *A New Machine Learning Algorithm for Users' Movie Recommendation*. 1–4.
- Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. *ArXiv Preprint ArXiv:1511.05440*.
- Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (1984). *Machine learning an artificial intelligence approach*. Springer.
- Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). *Movielens unplugged: Experiences with an occasionally connected recommender system*. 263–266.
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2), 1–135.
- Patel, A., Thakkar, A., Bhatt, N., & Prajapati, P. (2019). *Survey and evolution study focusing comparative analysis and future research direction in the field of recommendation system specific to collaborative filtering approach*. 155–163.
- Pavitha, N., Pungliya, V., Raut, A., Bhonsle, R., Purohit, A., Patel, A., & Shashidhar, R. (2022). Movie Recommendation and Sentiment Analysis Using Machine Learning. *Global Transitions Proceedings*.
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. *The Adaptive Web: Methods and Strategies of Web Personalization*, 325–341.

- Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97, 205–227.
- Qian, X., Feng, H., Zhao, G., & Mei, T. (2013). Personalized recommendation combining user interest and social circle. *IEEE Transactions on Knowledge and Data Engineering*, 26(7), 1763–1777.
- Quan, T. K., Fuyuki, I., & Shinichi, H. (2006). *Improving accuracy of recommender system by clustering items based on stability of user similarity*. 61–61.
- Ramzan, B., Bajwa, I. S., Jamil, N., Amin, R. U., Ramzan, S., Mirza, F., & Sarwar, N. (2019). An intelligent data analysis for recommendation systems using machine learning. *Scientific Programming*, 2019, 1–20.
- Rashid, T., Hossain, M. S., & Muhammad, G. (2020). *Rashid, T., Hossain, M. S., & Muhammad, G. (2020). A hybrid approach for movie recommendation using deep learning and collaborative filtering. Journal of Ambient Intelligence and Humanized Computing, 11(1), 295-305.*
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *Grouplens: An open architecture for collaborative filtering of netnews*. 175–186.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). *Application of dimensionality reduction in recommender system-a case study*. Minnesota Univ Minneapolis Dept of Computer Science.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. 285–295.
- Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5, 115–153.
- Sharma, P., & Yadav, L. Y. (2020). Movie Recommendation System Using Item Based Collaborative Filtering. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, ISSN, 2347–5552.
- Shen, J., Zhou, T., & Chen, L. (2020). Collaborative filtering-based recommendation system for big data. *International Journal of Computational Science and Engineering*, 21(2), 219–225.
- Singh, R. H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020). Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology*, 9(5), 556–559.

- Su, X., & Khoshgoftaar, T. M. (2009). *A survey of Collaborative Filtering Techniques*", Hindawi Publication Corporation-Advances in Artificial Intelligence. 2009.
- Vahidi Farashah, M., Etebarian, A., Azmi, R., & Ebrahimzadeh Dastjerdi, R. (2021). A hybrid recommender system based-on link prediction for movie baskets analysis. *Journal of Big Data*, 8, 1–24.
- Vinay, D., Kumaraswamy, B., & Basavaraddi, C. C. S. (2021). *Machine learning based recommendation system on movie reviews using KNN classifiers*. 1964(4), 042081.
- Wang, Ma, Y., Zhao, K., & Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 1–26.
- Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing*, 25(6), 667–675.
- Xie, H.-T., & Meng, X.-W. (2011). A personalized information service model adapting to user requirement evolution. *ACTA ELECTONICA SINICA*, 39(3), 643.
- Yang, Y., & Liu, X. (1999). *A re-examination of text categorization methods*. 42–49.
- You, T., Rosli, A. N., Ha, I., & Jo, G.-S. (2013). Clustering method based on genre interest for cold-start problem in movie recommendation. *Journal of Intelligence and Information Systems*, 19(1), 57–77.
- Zhang, D., Hsu, C.-H., Chen, M., Chen, Q., Xiong, N., & Lloret, J. (2013). Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems. *IEEE Transactions on Emerging Topics in Computing*, 2(2), 239–250.
- Zhang, Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1), 1–38.

Appendix A – Poster Outputs for KNN Movie Recommendations

Movie Recommender System

Choose a movie to see similar movies:

Spectre



Number of neighbors:

2 30

Recommend

Quantum of Solace
adventure action thriller crime

From Russia with Love
action thriller adventure



Movie Recommender System

Choose a movie to see similar movies:

Avatar

Number of neighbors:




3 30

Recommend

The Host
action adventure romance sc

The Host
action adventure romance sc

Jupiter Ascending
sciencefiction fantasy acti



Movie Recommender System

Choose a movie to see similar movies:

Skyfall

Number of neighbors:



Recommend

Octopussy

Die Another Day

Thunderball

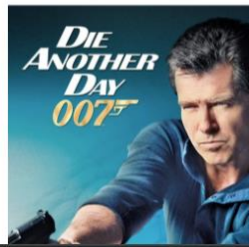
Live and Let Die

adventure action thr

adventure action thr

adventure action thr

adventure action thr



Movie Recommender System

Choose a movie to see similar movies:

Avatar

Number of neighbors:



Recommend

The Host

The Host

Jupiter Ascending

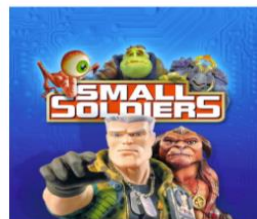
Small Soldiers

action adventure rom

action adventure rom

sciencefiction fanta

comedy adventure far



Movie Recommender System

Choose a movie to see similar movies:

Superman II

Number of neighbors:



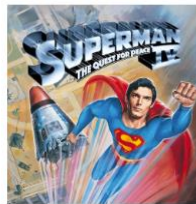
Recommend

Superman Return: Superman

Superman IV: Th Ant-Man

Man of Steel

adventure fanta: action adventure action adventure sciencefiction : action adventure



Movie Recommender System

Choose a movie to see similar movies:

The Dark Knight Rises

Number of neighbors:



Recommend

The Siege

Dead Man Down

The Dark Knight

Batman Begins

Righteous Kill

drama action th thriller action drama action cr action crime dr action crime dr



Movie Recommender System

Choose a movie to see similar movies:

John Carter

Number of neighbors:



Recommend

Dune Independence Day Divergent Damnation / Captain Am Captain Am Battleship
action sci action adv adventure action adv action adv adventure thriller a



Movie Recommender System

Choose a movie to see similar movies:

Tangled

Number of neighbors:



Recommend

Monsters Osmosis J Foodfight Against t Kung Fu P Brave Speed Rac The Borro
animation adventure animation adventure action ad animation action fa adventure



Movie Recommender System

Choose a movie to see similar movies:

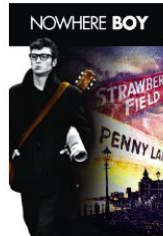
A Christmas Carol

Number of neighbors:



Recommend

End of the Spear Cradle Will Rock The Christmas Bunny Nowhere Boy Hachi: A Dog's Tale Flash of Genius
adventure drama drama drama family drama drama family drama



Movie Recommender System

Choose a movie to see similar movies:

Titanic

Number of neighbors:



Recommend

The Notebook Angel Eyes The Perks of Being a Wallflower Wicker Park Love Letter The Girl on the Train Alone with You The Age of Adaline Baby Boy I Dream of Jeannie
romance drama drama drama drama drama crime fantasy crime romance



Appendix B – Evaluating KNN Performance in Movie Recommender System

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 2

Accuracy: 0.5478170478170478

Precision: 0.6425339366515838

Recall: 0.285140562248996

F1-score: 0.39499304589707923

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 3

Accuracy: 0.6288981288981289

Precision: 0.688

Recall: 0.5180722891566265

F1-score: 0.5910652920962199

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 4

Accuracy: 0.5821205821205822

Precision: 0.6297297297297297

Recall: 0.4678714859437751

F1-score: 0.5368663594470046

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 5

Accuracy: 0.6288981288981289

Precision: 0.6390532544378699

Recall: 0.6506024096385542

F1-score: 0.6447761194029851

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 6

Accuracy: 0.6226611226611226

Precision: 0.6626506024096386

Recall: 0.5522088353413654

F1-score: 0.6024096385542168

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 7

Accuracy: 0.6247401247401247

Precision: 0.6378269617706237

Recall: 0.6365461847389559

F1-score: 0.6371859296482412

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 8

Accuracy: 0.6195426195426196

Precision: 0.6513761467889908

Recall: 0.570281124497992

F1-score: 0.6081370449678801

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 9

Accuracy: 0.6205821205821206

Precision: 0.64

Recall: 0.6104417670682731

F1-score: 0.6248715313463514

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 10

Accuracy: 0.6153846153846154

Precision: 0.6415929203539823

Recall: 0.5823293172690763

F1-score: 0.6105263157894738

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 11

Accuracy: 0.6153846153846154

Precision: 0.6367521367521367

Recall: 0.5983935742971888

F1-score: 0.6169772256728779

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 12

Accuracy: 0.6070686070686071

Precision: 0.6363636363636364

Recall: 0.5622489959839357

F1-score: 0.5970149253731343

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 13

Accuracy: 0.6112266112266113

Precision: 0.6165413533834586

Recall: 0.6586345381526104

F1-score: 0.6368932038834952

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 14

Accuracy: 0.604989604989605

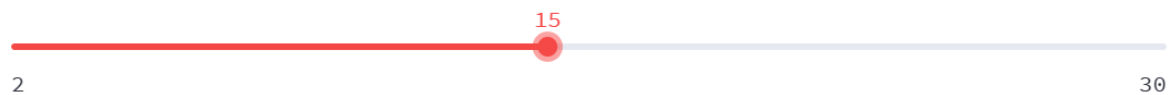
Precision: 0.6184738955823293

Recall: 0.6184738955823293

F1-score: 0.6184738955823293

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 15

Accuracy: 0.6205821205821206

Precision: 0.6224677716390423

Recall: 0.678714859437751

F1-score: 0.6493756003842459

Evaluating KNN Performance in Movie Recommender System

Number of neighbors:



Evaluation Metrics for k = 30

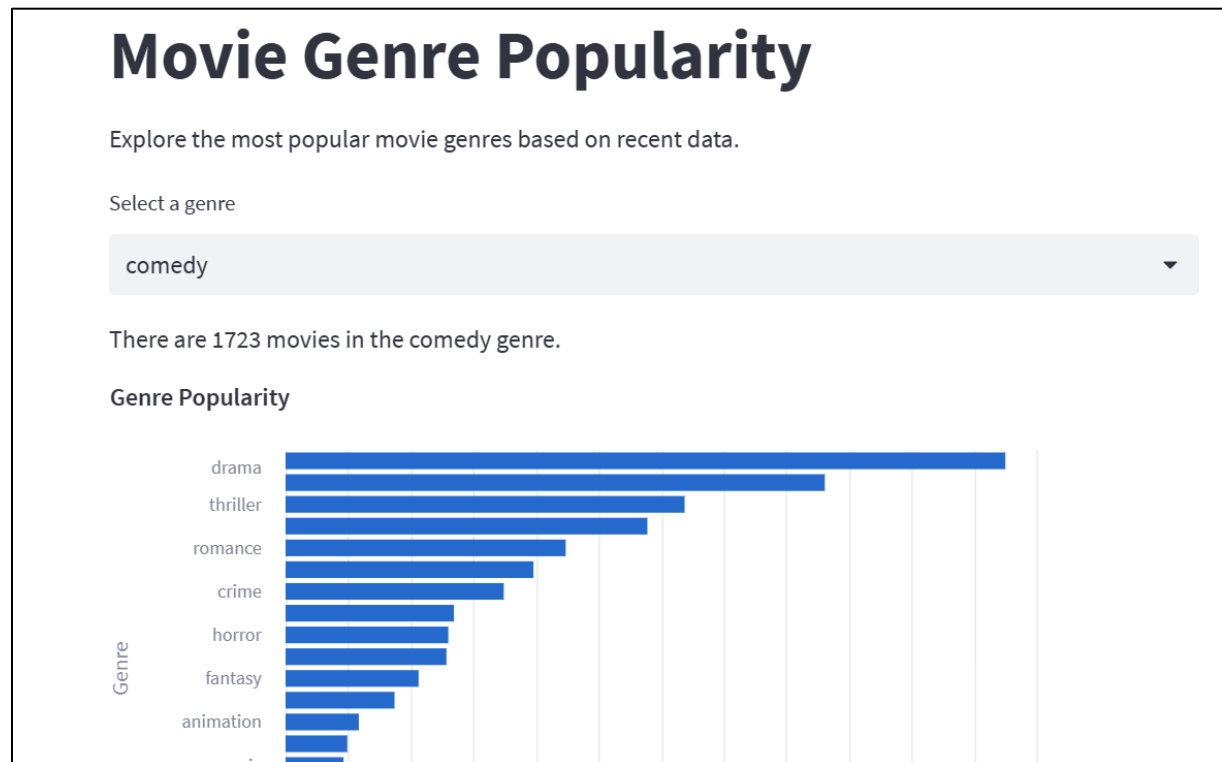
Accuracy: 0.6153846153846154

Precision: 0.6245136186770428

Recall: 0.6445783132530121

F1-score: 0.6343873517786561

Appendix C – Movie Genre Popularity



Movie Genre Popularity

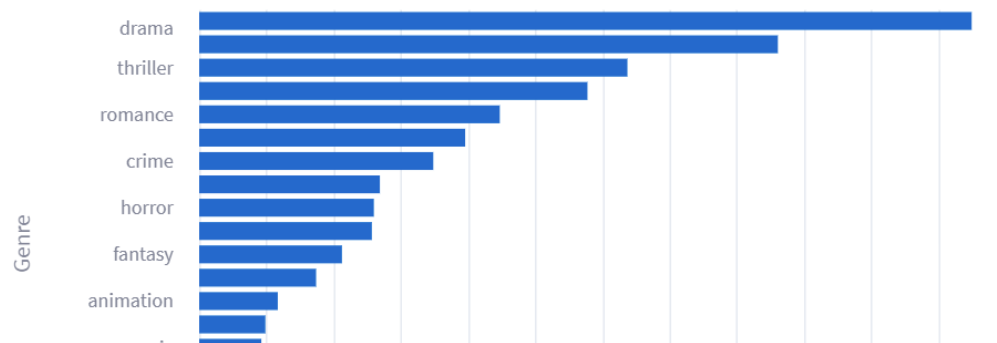
Explore the most popular movie genres based on recent data.

Select a genre

thriller

There are 1275 movies in the thriller genre.

Genre Popularity



Movie Genre Popularity

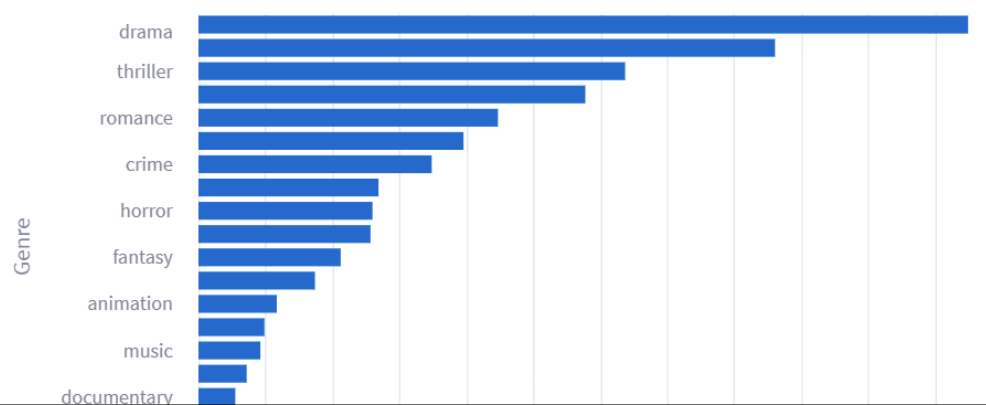
Explore the most popular movie genres based on recent data.

Select a genre

action

There are 1156 movies in the action genre.

Genre Popularity



Movie Genre Popularity

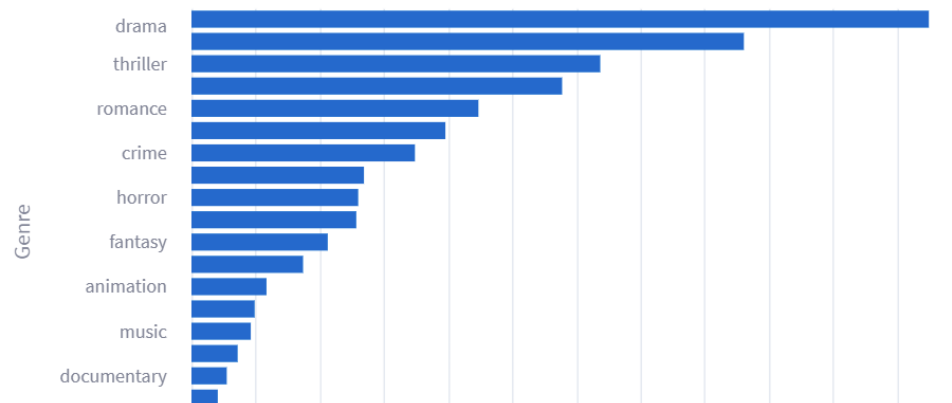
Explore the most popular movie genres based on recent data.

Select a genre

romance

There are 895 movies in the romance genre.

Genre Popularity



Movie Genre Popularity

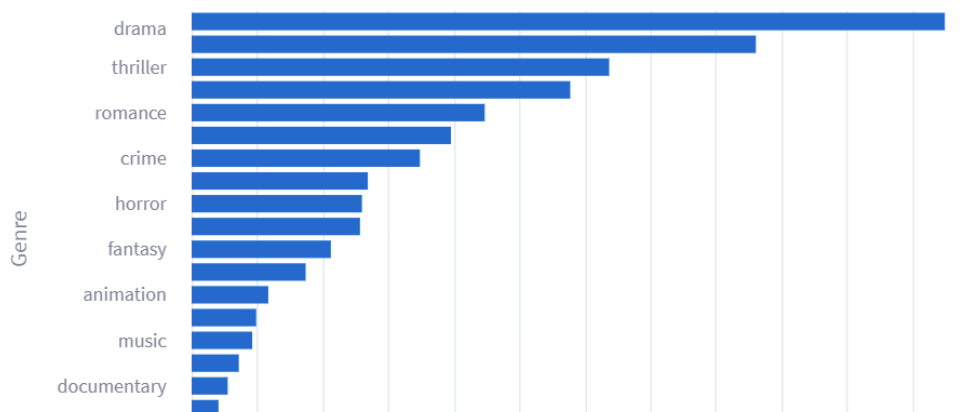
Explore the most popular movie genres based on recent data.

Select a genre

adventure

There are 792 movies in the adventure genre.

Genre Popularity



Movie Genre Popularity

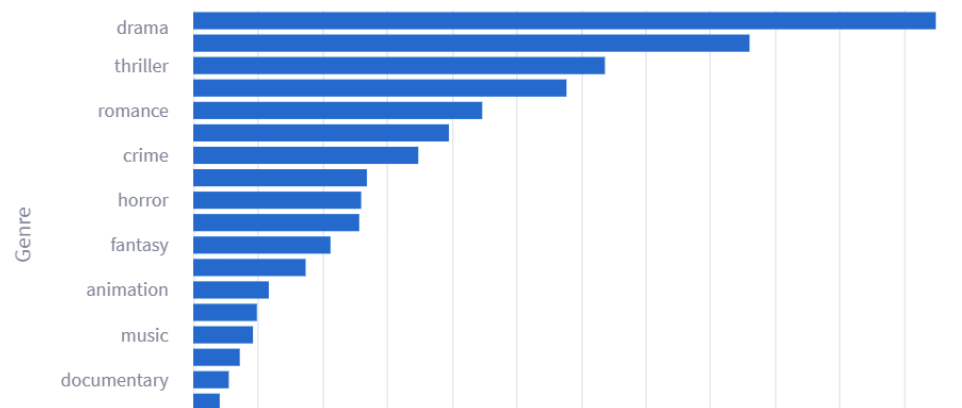
Explore the most popular movie genres based on recent data.

Select a genre

crime

There are 697 movies in the crime genre.

Genre Popularity



Movie Genre Popularity

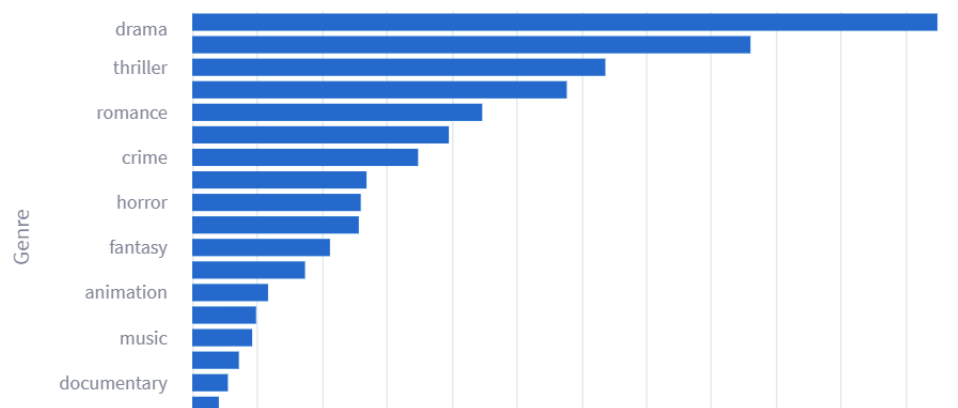
Explore the most popular movie genres based on recent data.

Select a genre

sciencefiction

There are 538 movies in the sciencefiction genre.

Genre Popularity



Movie Genre Popularity

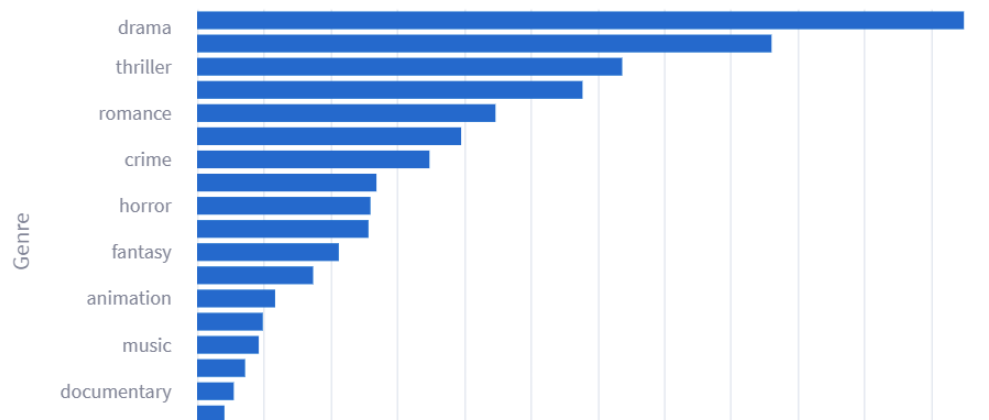
Explore the most popular movie genres based on recent data.

Select a genre

horror

There are 520 movies in the horror genre.

Genre Popularity



Movie Genre Popularity

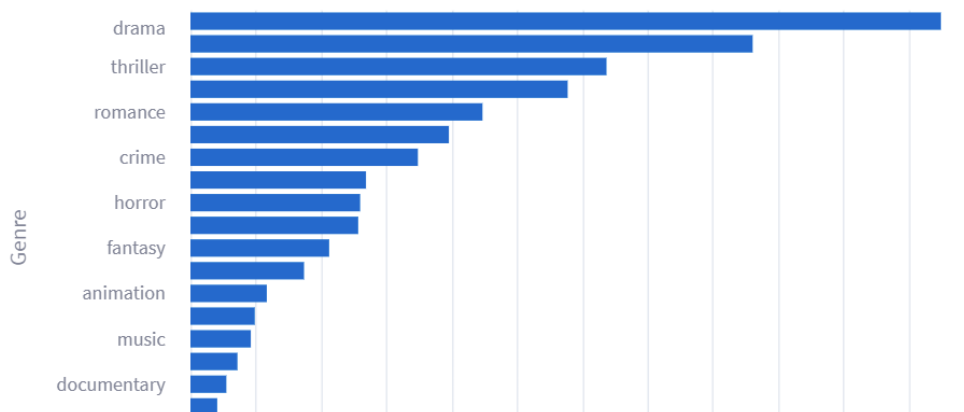
Explore the most popular movie genres based on recent data.

Select a genre

family

There are 514 movies in the family genre.

Genre Popularity



Movie Genre Popularity

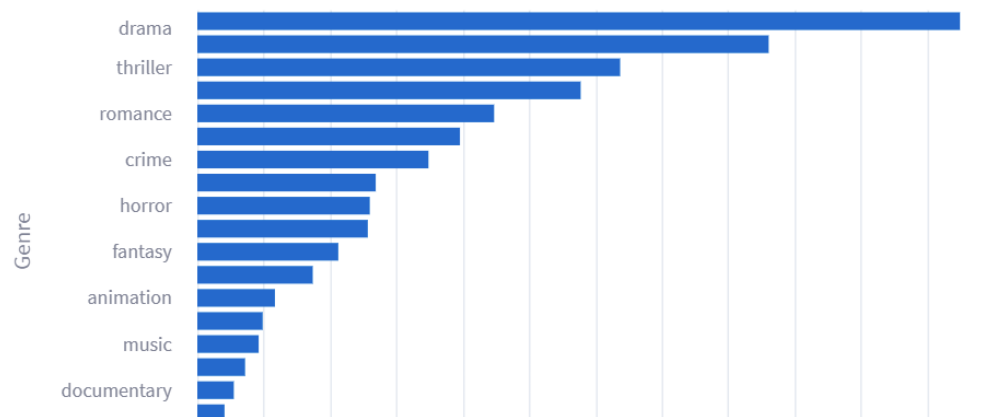
Explore the most popular movie genres based on recent data.

Select a genre

fantasy

There are 425 movies in the fantasy genre.

Genre Popularity



Movie Genre Popularity

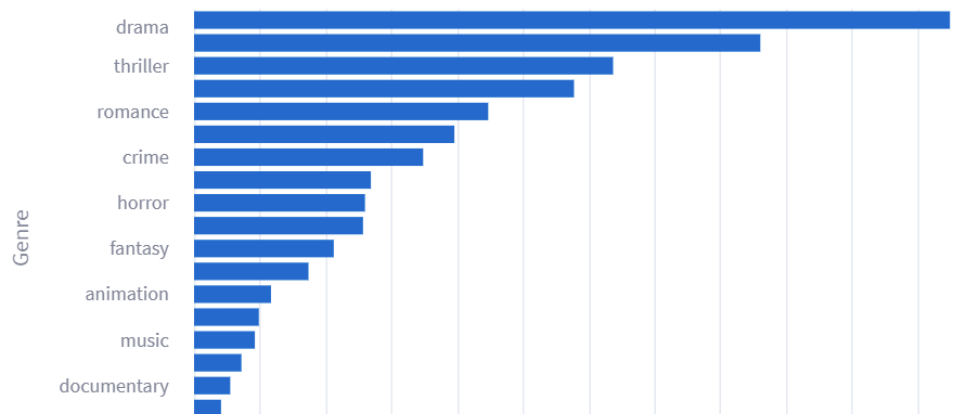
Explore the most popular movie genres based on recent data.

Select a genre

mystery

There are 348 movies in the mystery genre.

Genre Popularity



Movie Genre Popularity

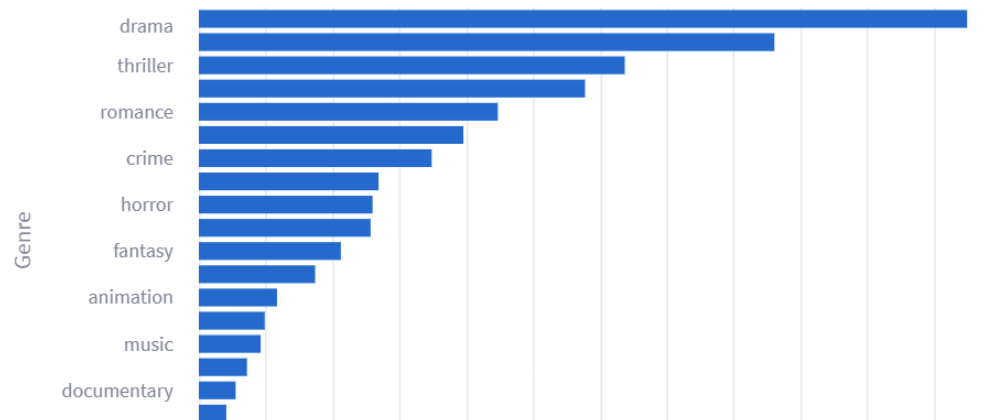
Explore the most popular movie genres based on recent data.

Select a genre

animation ▼

There are 234 movies in the animation genre.

Genre Popularity



Movie Genre Popularity

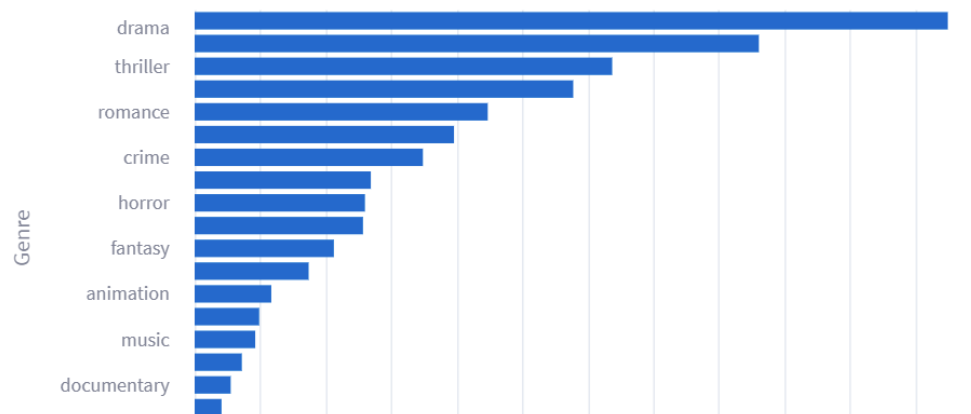
Explore the most popular movie genres based on recent data.

Select a genre

history ▼

There are 197 movies in the history genre.

Genre Popularity



Movie Genre Popularity

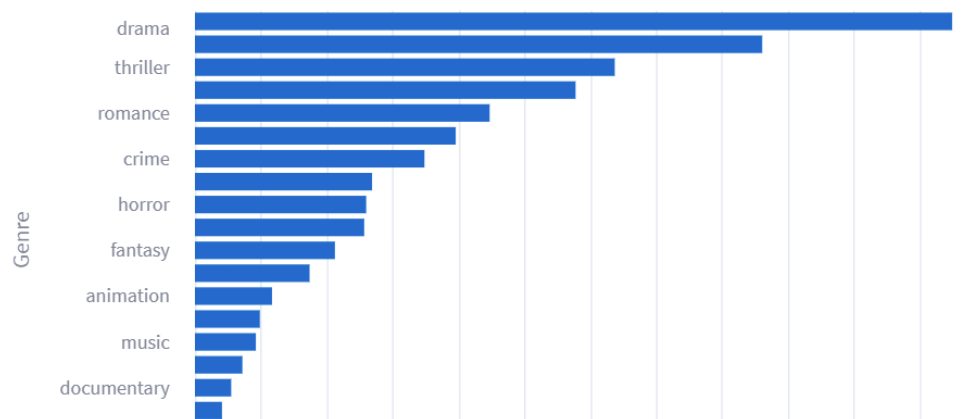
Explore the most popular movie genres based on recent data.

Select a genre

music

There are 185 movies in the music genre.

Genre Popularity



Movie Genre Popularity

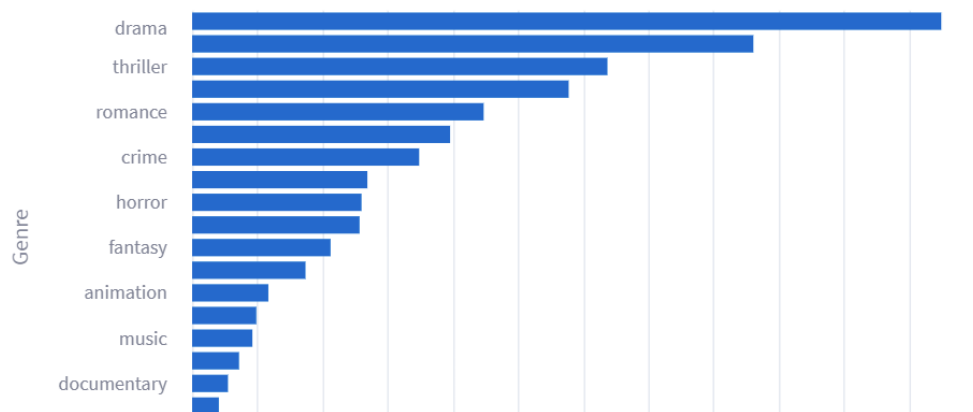
Explore the most popular movie genres based on recent data.

Select a genre

war

There are 144 movies in the war genre.

Genre Popularity



Movie Genre Popularity

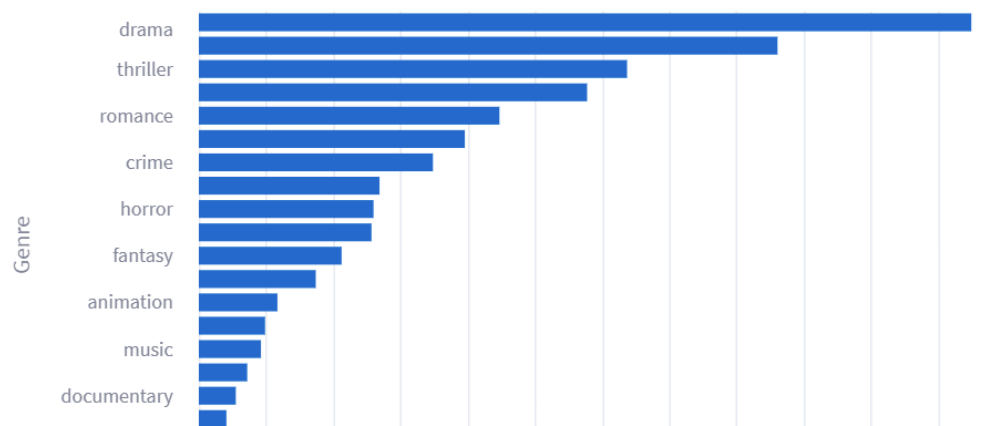
Explore the most popular movie genres based on recent data.

Select a genre

documentary ▼

There are 110 movies in the documentary genre.

Genre Popularity



Movie Genre Popularity

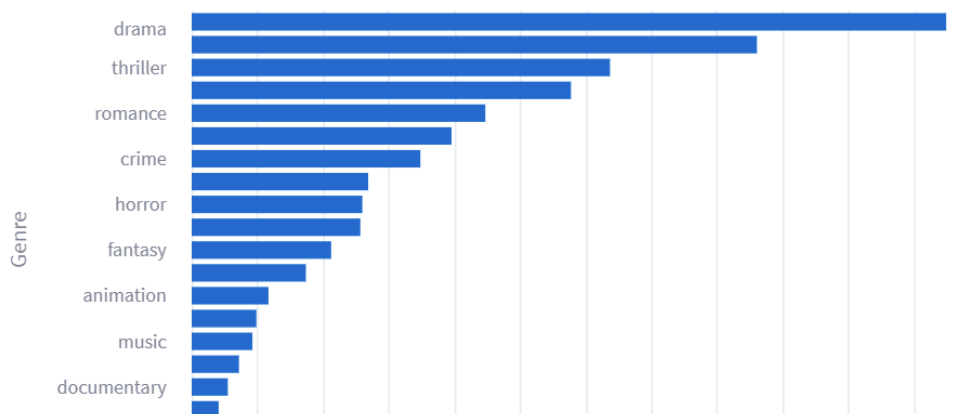
Explore the most popular movie genres based on recent data.

Select a genre

western ▼

There are 82 movies in the western genre.

Genre Popularity



Movie Genre Popularity

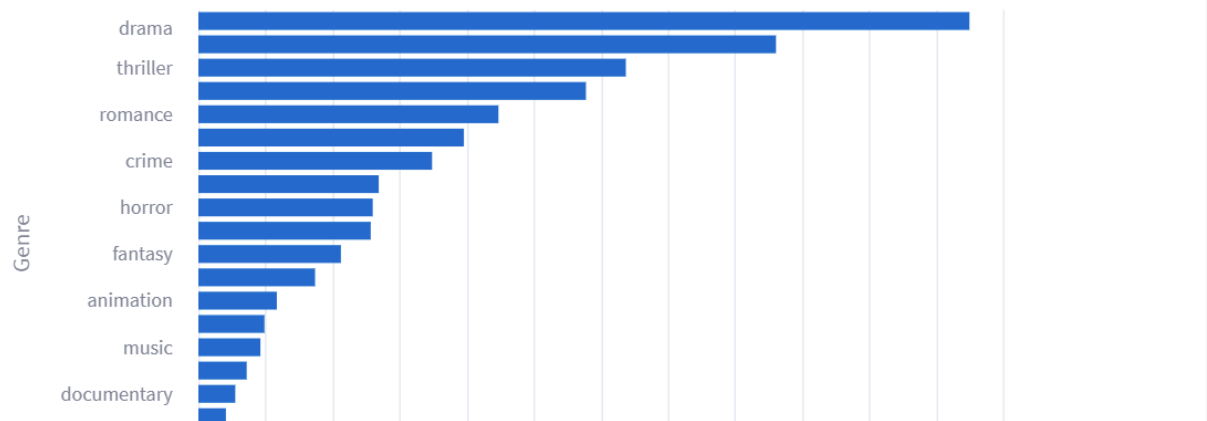
Explore the most popular movie genres based on recent data.

Select a genre

foreign

There are 34 movies in the foreign genre.

Genre Popularity



Appendix D-GitHub Repository

The code and resources used for this thesis project can be found in the GitHub repository.

To access the repository, click on the GitHub logo



Appendix E – My Thesis Reflection

This thesis provided a backdrop for deepening my knowledge in writing scientific reports. My supervisor's unalloyed support and guidance helped me structure the nooks and crannies of this report, which revolved around developing a machine learning-based application in the recommender system. I am grateful for what I have accomplished in this thesis. However, I am somewhat conflicted because I could not involve real users to test the system and see how similar movies can be recommended from the pool array of movies in the recommender system. Despite this drawback, I did develop a system that recommends movies using a slider bar, where prospective users can select movies they wish to watch and see similar movies. This development provided the backdrop that leverages the mitigation of challenges related to cold start and data sparsity, which have not been addressed in current research topics on developing movie recommender systems based on interest genres. In addition, I incorporated this development into the Streamlit web app, where I learned what it means to integrate and deploy machine learning applications to the web interface. Amidst Streamlit web app, other technologies of interest came in handy at this juncture. For instance, I employed cosine similarity in identifying and recognizing similar movies in the dataset. This bridged the gap between what I have been taught during lectures and the real-time practical application. In other words, I did accomplish this thesis work courtesy of these two technologies that are quite handy in ensuring that all my efforts were not futile.

On the heels of this accomplishment, the planning phase of this thesis went quite well. I must admit that I struggled to meet the deadline for the first two deliverables, as time was not on my side. I thought this struggle could have resulted from trying to write two theses simultaneously, but I have overcome it. This was made possible because my supervisor came up with a thesis schedule one month before the actual commencement of this thesis work. On the other hand, the thesis supervisor in the AI master's program also understood my situation and provided an alternative supervision time that did not overlap with this thesis schedule.

I think that my education in the computer systems and science program and AI master program at the prestigious Stockholm University furnished me with the requisite skills and knowledge needed for writing this thesis. Some courses in both master's programs that proved most useful are the DAMI course and the Machine Learning course. The DAMI course focuses on the Python program with data pre-processing, extraction, and model deployment. DAMI highlights more on the metrics that I used in this thesis. The Machine Learning course mainly delves deeper into model evaluations. A machine learning course would be highly relevant to understanding the use of k-nearest neighbours (KNN) in machine learning-based recommender systems. KNN is a popular algorithm used in recommender systems that utilize machine learning techniques to provide personalized recommendations to users. In KNN, the algorithm identifies the k-nearest neighbours to a particular user based on their preferences and past behaviour. The system then recommends items that these nearest neighbours have rated highly.

My future endeavour would definitely be in areas that employ machine learning and other technologies as enablers to improve users' experiences as they strive to find specific movies to watch. This thesis underpins the platform to fully explore all data science and machine learning positions that could be available in companies that are in high demand for my competence. Developing this recommender system is a great honour and something to be proud of. However, it would have been better to have someone to work with and bounce ideas off when approaching different scenarios for the machine learning recommender system.

Overall, this reflection provides a thoughtful reflection on the process of writing a thesis in machine learning, highlighting various aspects of the experience that could be useful for anyone writing a research paper in any field.