Data Analytics MSc Dissertation MTHM038, 2024/25

An Adaptive Hybrid Recommender System for the Item Cold-Start Problem

Sanjana, ID 240335366

Supervisor: Prof. Nicola Perra

A thesis presented for the degree of
Master of Science in *Data Analytics*

School of Mathematical Sciences

Queen Mary University of London

Declaration of original work

This declaration is made on September 1, 2025.

Student's Declaration: I Sanjana hereby declare that the work in this thesis is my original work. I have not copied from any other students' work, work of mine submitted elsewhere, or from any other sources except where due reference or acknowledgement is made explicitly in the text. Furthermore, no part of this dissertation has been written for me by another person, by generative artificial intelligence (AI), or by AI-assisted technologies.

Referenced text has been flagged by:

1. Using italic fonts, **and**

2. using quotation marks "...", **and**

3. explicitly mentioning the source in the text.

Abstract

The failure to provide relevant suggestions for new things that lack sufficient user interaction data, known as the item cold-start problem, frequently undermines the efficacy of recommender systems, despite the fact that they are crucial to modern digital platforms. In particular, collaborative filtering methods that rely only on previous user-item interactions encounter this challenge. This dissertation closes this gap by creating, implementing, and extensively testing an adaptive hybrid recommender system that deftly combines content-based and collaborative filtering methods.

The primary focus of the study is how to solve the item cold-start problem with high predicted accuracy using a hybrid model that adaptively switches between collaborative and content-based filtering strategies. This study uses the well-known MovieLens dataset to apply and evaluate three distinct models. Term Frequency-Inverse Document Frequency (TF-IDF) on movie genres is used in the first content-based model. A collaborative filtering method utilizing Singular Value Decomposition (SVD) is the second. Using a linear ramp-up weighting function, the third is a novel adaptive hybrid system that combines the predictions of the previous two and dynamically adjusts its logic based on the popularity of an item.

According to quantitative studies, the solo collaborative filtering strategy does well for existing items but badly for new ones. Although the cold-start issue is well handled by the suggested adaptive hybrid model, the solo Collaborative Filtering (SVD) model obtained the best overall accuracy, proving

to be the ”better” model for aggregate prediction on this dataset.

Contents

Chapter 1

Introduction

1.1 The Age of Information and the Rise of Recommender Systems

The information digital has increased exponentially and online platforms have increased exponentially in the first decades of the twenty-first century. With each click, purchase, view, and interaction, a digital trail is being left, and when added up, they form an ecosystem of previously unknown knowledge wealth. As human civilization enters the so-called Information Age, data are becoming a crucial resource and a significant driver of social and economic activity. This abundance affects the everyday life of customers with the help of such digital platforms as Amazon, Netflix, Spotify, YouTube and countless others, which offer customers access to giant portfolios of products, services or material.

To have the idea of the scale of this phenomenon, one may consider that Spotify contains over 100 million songs, YouTube users post over 500 hours of videos each minute, and Amazon sells hundreds of millions of products, including books and groceries. This scale is both a significant intellectual burden on end user and a technological achievement. It is humanly impossible that people might think about all alternatives with such large catalogs. Instead, individuals are often confronted with information overload, or a situation where they cannot digest and make prudent decisions because of the amount and variety of choices available to them. Excessive choice has been previously demonstrated to result into decision paralysis, user annoyance and reduced overall result satisfaction in studies of behavioral psychology.

Recommender systems have become a crucial technical answer to this problem. Essentially, recommender systems are specific types of technology that filter information. To forecast the possibility that a user would find a specific thing relevant, interesting, or helpful, they examine the links between users and products and use the data at their disposal. These systems provide benefits to organizations by boosting engagement, retention, and revenue as well as to users by lowering cognitive effort and enhancing happiness by condensing the pool of potential options to a customized shortlist. The digital economy is significantly impacted by recommender systems. Recommendation engines are essential to Netflix's competitive advantage as a streaming service. According to Netflix, suggestions, not direct searches, are how most users find the material they consume on the service. In a similar vein, product recommendations account for a sizable amount of overall sales income for e-commerce behemoths like Amazon. With algorithmic recommendation algorithms curating content streams that determine what billions of people watch, read, and interact with every day, social media platforms like Instagram and TikTok are more than simply ancillary features-they are the core of the user experience.

There is no doubt about the commercial stakes: user loyalty and engagement are strongly impacted by suggestion quality. In addition to increasing the possibility of upselling and cross-selling, a well-designed system reduces churn. Besides economics, recommender systems affect public discourse, cultural trends, and the promoted artists or products. As a result, these systems have broad ramifications and are both computational tools and socio-technical infrastructures.

Even with their achievements, recommender systems still have a number of persistent problems. The cold-start issue, which occurs when inadequate past data precludes useful advice, is the most significant of them. Despite their strength in data-rich settings, recommender systems frequently struggle in situations with sparse data because of their reliance on interaction histories. The cold-start issue is commonly recognized as one of the biggest and most enduring barriers to the field's ongoing progress.

1.2 The Cold-Start Problem: A Fundamental Challenge

In the absence of sufficient historical data, recommender systems struggle to make predictions, a phenomenon known as the "cold-start problem." The majority of recommendation algorithms depend on patterns found in user-item interactions; thus, when those patterns are absent or insufficient, the algorithms' accuracy is decreased and their suggestions are not as good. There are two ways in which this issue presents itself:

The User Cold-Start Problem The user cold-start scenario might occur when a new user joins the platform. The algorithm has trouble making intelligent recommendations as there is no prior record of this person's preferences. For instance, Spotify does not immediately know what genres, artists, or listening preferences a user has in mind when they establish a new account. It cannot estimate with any degree of accuracy what music they will likely enjoy without this information. When trying to suggest items to a first-time customer, e-commerce websites have a similar difficulty. Systems frequently employ general recommendations in these situations (such as popular or trending things), but they don't provide the level of customization that consumers are used to.

The Item Cold-Start Problem The primary focus of this dissertation is the item cold-start issue that may arise when a new item is introduced to the catalog. This item is nearly invisible to collaborative filtering algorithms due to the lack of reviews, ratings, and previous interactions. Think about a recently released indie movie that is currently accessible on Netflix. Collaborative algorithms are unable to include it in their lists of recommended content since it has not yet been reviewed or seen. This lessens the likelihood that the movie will be found because it hasn't been found yet.

The issue of item cold-start is very troubling for a number of reasons:

- **Barrier to Market Entry:** New items or sellers have trouble getting noticed on e-commerce platforms. In the absence of early suggestions, new goods could never get enough interactions to become well-known.
- **Suppression of Novelty and Diversity:** While ignoring fresh or specialized content, platforms run the danger of overrecommending well-liked products. As a result, there is a ”rich-get-richer” dynamic at play, with established products continuing to command the majority of attention while newcomers find it difficult to make an impression.
- **Economic Implications:** If new items are not recommended, firms may lose out on potential income. The incapacity of recommendation algorithms to showcase new material can hinder the advancement of content producers' careers and audience reach, including artists and writers.
- **User Experience Issues:** User happiness decreases when recommendations lack originality. Over time, the system's credibility may be damaged by suggestions that are too frequent or predictable.

It is not a theoretical problem. Think of Amazon launching a new product as an actual example. Until the product has been bought or rated several times, it won't show up in any user's suggestion feed if collaborative filtering is the main recommendation engine. As a result, there is a feedback loop: the item is not bought because it is not advised, and vice versa since it is not bought. Independent musicians may find their work buried behind algorithmic biases that favor already-popular tunes on streaming sites, which is a similar situation.

Therefore, it is both strategically and technologically necessary to solve the item cold-start issue. It guarantees more equitable exposure for fresh material, maintains platform expansion, and raises user pleasure by broadening the selection of suggested products.

1.3 Research Question and Aims

Given the cold-start problem's enduring nature and significance, the following main research issue drives this dissertation:

How can a hybrid recommendation model, which adaptively transitions from content-based to collaborative filtering methods, effectively mitigate the item cold-start problem while balancing predictive accuracy and computational efficiency?

It is acknowledged in this subject that there is no one suggestion method that is always the best. Although content-based approaches can manage new things, they lack serendipity and diversity, whereas collaborative filtering is excellent at personalizing but falls short in cold-start situations. By utilizing the advantages of both, an adaptive hybrid model makes clever adjustments when data availability shifts.

The following goals and objectives are pursued by the dissertation in order to answer this research question:

- **Implementation of a Content-Based Model:** Create and assess a recommender system that predicts based on item metadata, particularly about movie genres. Identify a baseline solution that can suggest new products without ratings. To find pertinent items, use cosine similarity and TF-IDF vectorization to quantitatively express category genre information.

- **Implementation of a Collaborative Filtering Model:** Use Singular Value Decomposition (SVD), a popular matrix factorization method, to create a high-accuracy recommender system. Show how collaborative techniques may be successful when there is enough historical data available. Examine how accuracy changes with item popularity in order to verify the cold-start issue experimentally.
- **Design and Development of an Adaptive Hybrid Model:** The results of the collaborative and content-based filtering algorithms are combined using a dynamic weighting mechanism. As an item receives more ratings, use a linear ramp-up approach that gives collaborative filtering more weight. As the model moves from content-driven suggestions for new things to collaborative predictions for existing goods, make sure the transition goes smoothly.
- **Comparative Evaluation and Analysis:** Compare the accuracy and computing efficiency of the three models in a thorough analysis. In order to capture real-world efficiency, measure latency and evaluate predicted accuracy using Root Mean Square Error (RMSE). Examine and contrast the models that are ”better” (more accurate) and ”faster” (more efficient).

By achieving these goals, the dissertation contributes to the discussion of how to balance customization, novelty, and computational constraints in recommender system design, in addition to providing a practical solution to the item cold-start problem.

1.4 Methodology Overview

Empirical testing with the MovieLens dataset, a benchmark dataset frequently used in recommender system research, forms the basis of the study approach. It was chosen based on several factors: Reproducibility is ensured by its well-documented and publicly accessible nature. In order to use collaborative and content-based approaches, it contains both interaction data (user ratings) and content information (genres, titles). Over 98% of the user-item matrix is left blank, demonstrating realistic sparsity levels that make it an appropriate testbed for researching cold-start dynamics.

A Jupyter Notebook platform was used for the whole project, which offered transparency, modularity, and replication simplicity. The methodology's phases follow a logical order:

1. **Phase 1: Data Cleaning and Exploratory Data Analysis (EDA):** Among the responsibilities were timestamp conversion, user-generated tag standardization, and movie name separation from release year. The cleansed data was saved to intermediate files to guarantee reproducibility. EDA: The long-tail distribution of ratings per item, genre frequencies, and rating distributions were all examined using visualization tools including bar charts and histograms. The majority of movies had very low ratings, according to this investigation, which offered empirical support for the item cold-start problem.

2. **Phase 2: Content-Based Model Implementation:** Feature Engineering: Numerical vectors that represent the frequency and distinctiveness of genres in films were produced by applying TF-IDF to genres. Similarity Calculation: To create an item-item similarity matrix, cosine similarity was used. This made it possible for the system to suggest films with material that was comparable to what the user had already enjoyed.

3. **Phase 3: Collaborative Filtering Model Implementation:** Algorithm Choice: To implement Singular Value Decomposition (SVD), the Surprise Python package was used. Hidden aspects of taste are captured by SVD, which breaks down the sparse user-item matrix into latent user and item variables. Evaluation: The RMSE for each item in the popularity group was used to quantify performance. This investigation verified that while accuracy decreases in severe cold-start situations, it increases as items receive ratings.

4. **Phase 4: Adaptive Hybrid Model Implementation:** Blending Strategy: The content-based and collaborative filtering algorithms' scores were integrated in the hybrid model. Weighting Function: With n being the number of ratings and k being a threshold hyperparameter, a linear ramp-up weighting function was used, which is specified as $w(n) = \min(n/k, 1.0)$. As a result, the shift from content-based to collaborative predictions went well. Adaptive Logic: Recommendations were based almost exclusively on content similarity for new products without ratings. Collaborative forecasts were increasingly trusted by the algorithm as more ratings were provided.

5. **Phase 5: Quantitative Evaluation:** Accuracy: Prediction error was measured via the main statistic, Root Mean Square Error (RMSE). Efficiency: Computational expenses were captured by measuring recommendation delay. Comparative Analysis: A side-by-side comparison of the three models' results demonstrated the trade-off between speed and accuracy.

This approach guarantees a strong empirical foundation for addressing the study topic and offers both qualitative and quantitative information on the efficacy of the suggested adaptive hybrid model.

Chapter 2

Literature Review

2.1 An Overview of Recommender Systems

One of the most powerful applications of artificial intelligence and data-driven personalization in the digital economy is recommenders. Its history as a separate discipline dates back to the mid-1990s when scholars started to combine methods of information retrieval, machine learning, statistics and human-computer interaction in order to solve the increasing information overload on the internet [10]. Some of the earliest examples of using collective intelligence to scale user preferences are old systems like GroupLens to filter Usenet news [5]. The first experiments formed the basis of what is now a very large and interdisciplinary research area.

A recommender system seems to have an incredibly easy task to perform: predicting what an individual user is most likely to read, watch, listen to, or buy [11]. However, this goal, when large scale is concerned, demands extremely advanced models, which can process large data volumes, sparsity in feedback, dynamism in user preferences, and real-time responsibilities. Within the last thirty years, recommender systems have found their way into e-commerce (Amazon), media streaming (Netflix, Spotify, YouTube), social networks (Facebook, Instagram, Tik Tok), and even the professional world (LinkedIn job recommendation, Coursera course suggestion).

Recommendation technologies are of great significance to the economy and culture. Research has indicated that up to 80 percent of Netflix viewing activity can be attributed to its recommendation system [3], and an analogous study has estimated that 35 percent of Amazon sales can be directly attributed to its recommendations [7]. Recommendation algorithms affect not just user experience on the social media platform but also the societal discussion of the topic as they are in charge of what content will be visible. Business-wise, the recommender systems are not an accessory option; they are at the core of customer retention, user engagement, and the increase of revenues.

On the methodology level, the recommender systems can be classified in three broad categories [1]:

- Content-Based Filtering (CBF) - Relies on item attributes and user profiles.
- Collaborative Filtering (CF) - Relies on past user-item interactions.
- Hybrid approaches - Use CBF and CF together to address the limitations of each.

The sections below discuss them in more detail pointing out their methodological backgrounds, strengths, and weaknesses as well as their applicability to the topic of the present dissertation, i.e., the item cold-start problem.

2.2 Content-Based Filtering

The idea behind content-based filtering is that similar items should attract the same user. Practically, the system constructs an organized picture of objects and consumers. In the case of items, this entails the extraction of descriptive features e.g. genre, director or actors in a movie, or ingredients in a recipe. Users form profiles through the synthesis of features of items that they have interacted with favorably [9]. As an example, a user, who regularly watches psychological thrillers with female protagonists in them, might have their profile weighted towards these characteristics.

The similarity measure of user vectors and item vectors is generally formalized as to be a matching process between users and items. Widely used measures are cosine similarity, Euclidean distance, or dot products. In text-based domains, Term Frequency-Inverse Document Frequency (TF-IDF) is still widely used as a means of representation. TF-IDF gives weights to terms (genres, keywords, tags) by dividing the frequency of the terms in a particular item, and by the overall frequency of the terms throughout the data set [12]. This makes sure that discriminative features, though rarely, play a more important role in similarity calculations.

Strengths

- **Capability of Cold-Start:** Since content-based systems do not rely on previous interaction data, they are especially useful in the alleviation of the item cold-start problem. Immediately, a new film containing its genre information can be suggested to the users whose profile matches these genres.

- **Transparency:** Explanations are easy to produce (e.g. We recommend this because it has similarities to films that you enjoyed before), which enhances user trust.

- **Independence over Community Data:** They are user specific and not reliant on other user behaviors and hence they can work in small systems with limited users.

Weaknesses

- **Overspecialization:** Content-based approaches would have the effect of strengthening prior user preferences, which results in a so-called filter bubble. Users can hardly have an exposure to the things that are not part of their known tastes [8].

- **Metadata Dependence:** metadata of items is rich and accurate, which makes a significant difference in the quality of recommendations. Content-based techniques can be poor in areas where characteristics are hard to elicit (music, images or complex products).

- **Weak Discovery of Serendipity:** due to their use of explicit attributes alone, content-based systems can be weaker in suggesting unexpected, but valuable items, which can be discovered using collaboration techniques.

To conclude, although the concept of content-based filtering provides an important solution to the new-item recommendation, its weaknesses require other paradigms to be incorporated in providing richer and more varied experiences to users.

2.3 Collaborative Filtering

Collaborative Filtering (CF) is based on the principle of homophily, or the notion that users who agreed in the past are likely to agree in the future. Cf does not assume any knowledge of item attributes as content-based methods do. It, instead, examines pattern within the user-item interaction matrix that captures ratings, clicks, or purchases [14]. Two broad categories of CF exist:

Memory-Based (Neighborhood-Based) Approaches

- **User-Based CF:** Identifies users that are similar to the active user and suggests items that are liked by the users.

- **Item-Based CF:** Determines the items that are similar to those already liked by the user, using the information of the co-rating pattern of the users [13]. Such methods depend on the similarity measures like Pearson correlation, cosine similarity, or adjusted cosine.

Model-Based Approaches These employ machine learning to reveal hidden factors that act as explanations of user-item interactions. The rating matrix can be decomposed by use of matrix factorization techniques like the Singular Value Decomposition (SVD) or Probabilistic Matrix Factorization (PMF) to form low-dimensional representations of the users and items [6]. An example is an example of a user vector indicating a preference based on non-observable dimensions such as dark comedies or highly rated thrillers and an example of an item vector indicating the degree to which a film accentuates those attributes.

Strengths

- **Personalization:** CF is able to capture fine latent user taste, which content-based methods fail to capture.
- **Serendipity:** CF has the capability of the introduction of unforeseen products that match the undisclosed interests of a user which increases discovery.
- **Domain Independence:** CF is not metadata-dependent so it fits in almost any domain where there is enough interaction data.

Weaknesses

- **Item Cold-Start:** CF is not capable of providing the recommendations of new items without any prior interactions. A ratingless film does not exist to the system.

- **User Cold-Start:** Likewise, new accounts that have no history cannot be provided with personalized recommendations.

- **Sparsity:** In practice, user-item matrices in real systems are very sparse (more than 98% empty in the MovieLens data). This complicates the search of strong neighbors or models that work [4].

- **Scalability:** Neighborhood methods are computationally costly to scale, albeit to a lesser degree, than model-based methods.

Nevertheless, CF is still the prevailing paradigm in commercial systems because it is effective in the personalization and discovery in the case of abundance of data.

2.4 Hybrid Recommender Systems

Since both content-based and collaborative filtering are complementary, hybrid systems have been studied long as a way to counteract the limitations of either. Burke's (2002) influential taxonomy identifies several hybridization methods [2]:

- **Weighted Hybridization** - Score in various recommenders are combined and produce a weighted prediction. This is simple and efficient and it is the foundation of the adaptive model in this dissertation.

- **Switching** - Switches dynamically between recommenders based on context (e.g. use content-based when there are fewer than 5 ratings, CF otherwise).

- **Cascade** - This scheme takes one recommender to recommend items and another to refine or re-rank.

- **Feature Augmentation or Combination** - Uses the output of one recommender (e.g., item features) as an additional input to another.

- **Meta-Level Approaches** - The first model produces an intermediate (e.g. user profile) that is then inputted into another model.

Hybrid systems are especially useful in the cold-start problem. As an example, Netflix will join both content-based metadata (genres, actors) and collaborative filtering indicators (ratings, viewing history) into its content recommender pipelines [3]. Likewise, Spotify combines deep learning-extracted audio characteristics with collaborative play history.

The adaptive hybrid model of the dissertation is a hybridization strategy, however, it has a weighted hybridization strategy that is further expanded to a dynamic weighting function. Rather than imposing fixed weights, the function modulates trust between content-based and collaborative filtering with an increasing number of rated values. This prevents the inflexibility of changing mechanisms and makes it graceful to deal with items at various maturity levels.

2.5 Algorithmic Foundations: Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) has become a typical baseline model under model-based collaborative filtering approaches because it is effective in processing sparse matrices.

Theoretical Justification In recommender systems, the user item matrix is sparse but very large. SVD breaks this matrix down into three smaller matrices:

- A user-factor matrix
- A diagonal matrix of factor strengths
- An item-factor matrix

What is produced is a low-rank approximation, in which users and items occupy the same latent feature space [6].

Advantages of SVD

- **Latent Factor Discovery:** SVD refers to the latent dimensions of taste, which allow the model to extrapolate to unobserved ratings.
- **Precision:** empirical experiments (e.g., Netflix Prize competition) have shown that matrix factorization is much more accurate than neighborhood methods at prediction [6].
- **Scalability:** SVD can be trained effectively on large-scale data on stochastic gradient descent or alternating least squares.

Example Assume that a latent factor is associated with preference in complex thrillers. High-weight user on this factor will be forecasted to like new thrillers even when there is no apparent metadata overlap with previously rated movies.

Prediction Formula The latent vectors of user u and item i are dotted to produce the predicted rating of user u and item i [6]:

$$\hat{r}_{ui} = p_u \cdot q_i$$

where p_u = user vector, and q_i = item vector. This decomposition has been effective in overcoming sparsity with the use of shared latent structures. Nevertheless, it is not yet able to deal with fully unrated items, which is why hybridization is important.

Chapter 3

Data Description

3.1 Dataset Source and Appropriateness

One of the most crucial parts of any empirical study in the sphere of recommender systems is the choice of appropriate dataset. It will not only be necessary to have a dataset of adequate size and richness to be able to train robust models, but the dataset also needs to display the particular properties of the problem under study. In this dissertation, the empirical grounds of all the experiments are the MovieLens (ml-latest-small) data set, which is offered by the research lab GroupLens at the University of Minnesota.

There are several reasons why this dataset is the most suitable in this project. First, the data gives a valuable source of explicit user feedback as ratings. It has 100,836 ratings on 9,742 movies rated by 610 users who rated them between March 1996 and September 2018. This amount of interaction information is essential in training a powerful collaborative filtering model. Models such as Singular Value Decomposition (SVD), which is learnt using user-item interactions, need a large amount of data points to properly identify and model the underlying user taste patterns. The size of the MovieLens data gives this model-based method good ground.

Second, the items in the catalog have critical content metadata in the dataset. The movies.csv file offers textual based features on each movie i.e. the title and a list of genres which is separated by pipes. The building of the content-based component of the hybrid system is pre-requisite based on this metadata. These two properties, the availability of such rich interaction data and explicit content properties, make the MovieLens dataset one of the few publicly available datasets ideally suited to research on hybrid recommendation strategies that integrate collaborative and content-based filtering.

Third, and most crucially in the context of this dissertation, the data set exemplifies the real-life problem of data sparsity and the item cold-start problem. The interaction matrix between the users and the items that can be visualized as a table of user-rows and movie-columns is certainly not the limit. The potential rating of this data set is the product of the number of users and the number of movies (610 users × 9,742 movies = 5,942,620 potential ratings). Attempting to divide the number of ratings by the number of users and by the number of movies, the sparsity of the matrix can be found:

$$\text{Sparsity} = 1 - \frac{\text{Number of Ratings}}{\text{Number of Users} \times \text{Number of Movies}}$$

$$\text{Sparsity} = 1 - \frac{100,836}{5,942,620} \approx 0.983$$

This implies that the user-item matrix is sparse at more than 98 percent. Practically, this means that the average user has rated only a very minor percentage of movies available. Such sparsity is high and thus makes the dataset a realistic and challenging testbed to analyze the item cold-start problem. It guarantees that the catalog has a considerable amount of items that have very few ratings, which are the main target of the research.

The data is a collection of four major files, which are ratings.csv, movies.csv, tags.csv, and links.csv. The main files used in this project are ratings.csv file used in collaborative filtering and movies.csv file used in content-based component. The user-generated tags were also processed in the form of tags.csv which can be used as a source of richer content features in future extensions of this work.

3.2 Data Cleaning and Preprocessing

A sequence of basic cleaning and preprocessing operations were carried out before the data could be analyzed and modeled, as recorded in the Jupyter Notebook to the project. The steps play a vital role in guaranteeing data quality, uniformity, and compatibility to the machine learning libraries in the later modeling processes.

1. **Loading the Data:** The DataFrames of raw movies.csv, ratings.csv and tags.csv files were loaded into pandas DataFrames, a powerful data structure of the Python pandas library that offers a convenient and efficient data manipulation format to manipulate tabular data.

2. **Extracting Features using Movie Titles:** The title column of the movies.csv file includes the title of a movie, and the year of release, both enclosed in parentheses (e.g. Toy Story (1995)). In order to form two separate and more useful features, a regular expression (re) was used to remove the four-digit year into a new year column. The original title string was then programmatically stripped of the year and any preceding or following whitespace was stripped to produce a clean title feature. This division is essential since it enables the year to be handled as a numerical quality and the title as a tidy label on the item.

3. **Timestamp Conversion:** This was done as the timestamp column in the ratings.csv and tags.csv files were given in Unix time (the number of seconds since the epoch), or January 1, 1970. This format cannot be easily analyzed by the human being. Thus, it was changed to a common datetime structure (YYYY-MM-DD HH:MM:SS) with the inbuilt to_datetime function of pandas. This would be open to possible time-

based analysis (e.g. analyzing the trend of ratings over time) and would make the data more readable and understandable.

4. **Tag Normalization:** The tags that were created by the user were normalized to lower case in the tags.csv file. This is a default text normalisation procedure, which provides consistency and avoids the model treating things like, Funny, funny, and FUNNY as distinct tags. Any text-based feature analysis is critical to this consolidation of terms.

5. **Saving the Cleaned Data:** When these preprocessing activities have been completed, the cleaned DataFrames were saved to new CSV files (movies_cleaned.csv, ratings_cleaned.csv, etc.). This is a nice practice in data science workflows since it creates a clean-up data object which can be loaded directly into models and saves them computation time and does not need them to recreate the clean-up steps each session.

3.3 Exploratory Data Analysis (EDA)

An extensive Exploratory Data Analysis was performed to understand the nature of the dataset and to visually prove the main issue of this dissertation. Using the EDA, which is applied in the Jupyter Notebook, gives an important insight into the structure and distribution of the data.

3.3.1 Distribution of Ratings

To show the distribution of user ratings, a bar chart was generated and the rating is on a 5-star scale with half-star differences. The chart indicates a skewed distribution to the left with users giving more positive feedback (3, 4, and 5 stars) than negative. Most common rating is 4.0, 3.0 and 5.0. This is typical of recommender system datasets and is frequently explained by a self-selection bias, in which users have more incentive to provide ratings on items they liked. This knowledge can be applied during the modeling stage, where it would give the approximate distribution of the target variable.

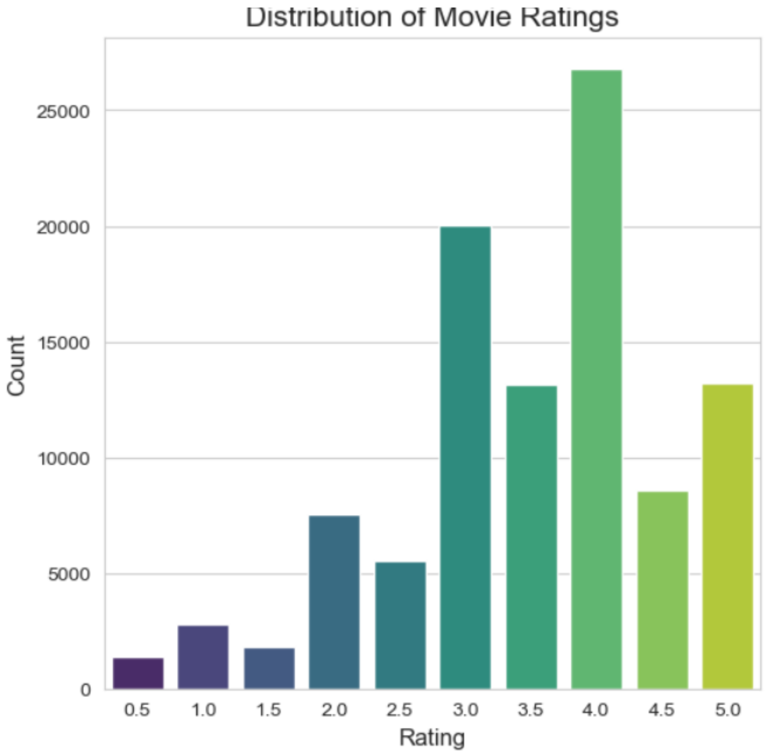


Figure 3.1: Distribution of Movie Ratings

3.3.2 Distribution of Genres

Genre distribution was examined by breaking the genres-separated string of the pipes during each movie and indicating the number of times each genre appeared. It is analyzed that the most common genres in the dataset include Drama and Comedy, and then Thriller and Action. This means that the dataset has a deep and diverse pool of content features that may be used successfully by the content-based model. The fact that many genres are represented is significant to the capacity of the model to differentiate among the types of movies.

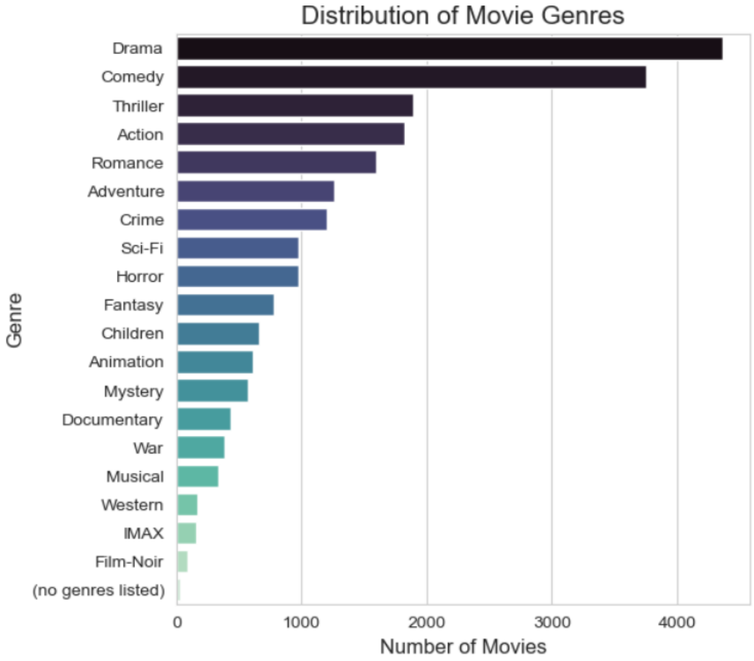


Figure 3.2: Distribution of Movie Genres

3.3.3 Visualizing the Item Cold-Start Problem

The step of analysis of the distributions of ratings per movie was the most important part of the EDA since it is directly related to the item cold-start problem. To illustrate what number of ratings there are in a movie, a histogram was plotted. The x-axis was plotted on a logarithmic scale in order to visualize the highly skewed distribution better.

This chart gives a vivid and vivid demonstration of the long-tail distribution that defines the cold-start problem. It clearly shows that: Very few movies (the ”head” of the tail) are exceptionally popular, and they have hundreds of ratings. Vast majority of movies (the long tail) are those which have been rated very few times. The mean ratings per movie stand at a measly 3 and the mean at 10. A considerable number of films in the catalogue contain less than 10 ratings.

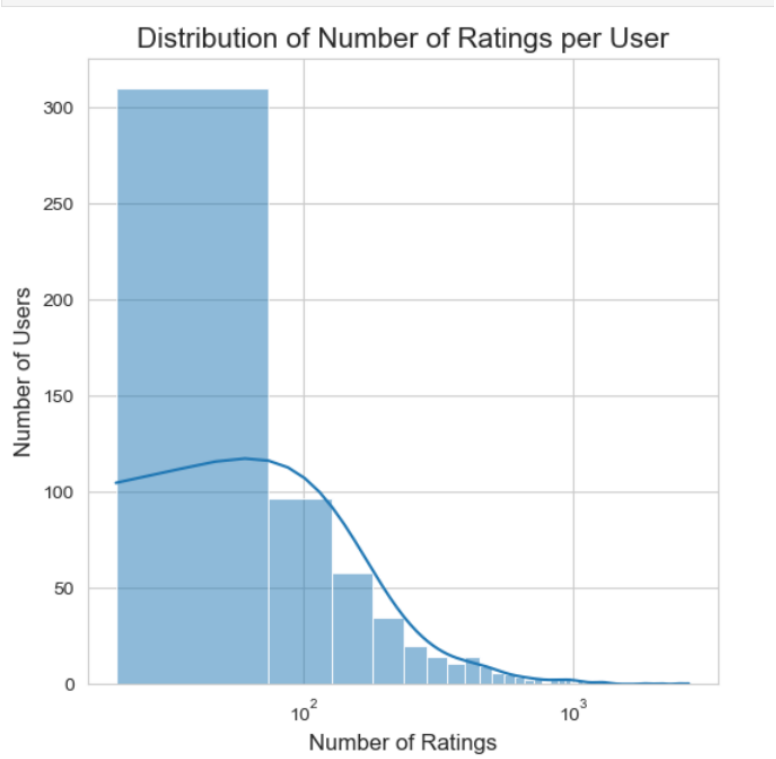


Figure 3.3: Distribution of Number of Ratings per User

It is this distribution that is of central interest in this dissertation. It shows that a recommendation strategy based entirely on collaborative filtering will not deliver useful recommendations to a large fraction of the items in the catalog. It gives a good, empirical explanation of how a hybrid model might work to manage the items in this long tail and have new and niche movies not systematically overlooked by the recommendation engine. Such a visual object is an effective means of problem framing and justifies the course of the research.

Chapter 4

Methodology

In this chapter, the entire technical explanation of how the three models of recommendations, which underlie this dissertation, are implemented, is presented: The content-based filtering model, the collaborative filtering model, and the adaptive hybrid system that combines them. All models are created with a particular purpose: the content-based one will prevent the cold-start issue with the help of item metadata, the collaborative filtering one will discover the hidden trends in the user-item interactions, and the hybrid system will use the benefits of both.

Every implementation and experiment is written in the supporting Jupyter Notebook. In this chapter the design and algorithms, and evaluation framework with which the system was built are outlined.

4.1 System Architecture

The presented solution is a hybrid recommender system that is an adaptive and weighted hybrid. It does not use just a single algorithm but it uses several models built in a single coherent pipeline. The reason why this hybridization is important is that the complementary strengths of the various algorithms are present.

- When not much or no historical data are available on a given item, content-based filtering (CBF) can be appropriately used since it relies on metadata and does not depend on user interactions.

- Collaborative filtering (CF), in its turn, works more efficiently once there are enough user feedback (ratings) as it is capable of identifying latent factors that drive preferences.

- The relative weighting of the two models is calculated using a dynamic weighting mechanism which varies according to the quantity of available rating data.

The system has the following high-level workflow:

1. **Content-Based Model:** Calculates a similarity-based score, indicating how a candidate item can be relevant to a particular input movie, using only metadata information, e.g., genres.

2. **Collaborative Filtering Model:** A personalized rating to the candidate item is predicted based on the decomposition of the user-item interaction matrix by latent factors.

3. **Adaptive Weighting Module:** The combination of the two scores into a final recommendation score. The level of weighting of the models corresponds to the level of ratings that have been made available to the target item. With new or infrequently rated items, the content-based component is used, but with established items, prediction is dominated by the collaborative filtering.

The final recommendation score for a user u and an item i is calculated as:

$$\text{Score\_Hybrid}(u,i) = (1-w(n_i))\times\text{Score\_Content}(u,i)+w(n_i)\times\text{Score\_CF}(u,i)$$

where:

- n_i = number of ratings received by item i
- $w(n_i)$ = adaptive weight function

This design ensures adaptability across different data scenarios, from cold-start items to popular blockbusters.

4.2 Content-Based Model

Content-based filtering model was adopted in order to act as a baseline recommendation model, especially when it comes to the processing of new items that do not have adequate rating history. This model does not demand that user-item interaction data be available, as does collaborative filtering, but rather, item metadata, in this case, movie genres.

4.2.1 Feature Engineering with TF-IDF

The Term Frequency-Inverse Document Frequency (TF-IDF) process was undertaken so as to make the categorical genre data useful in similarity calculations. TF-IDF is a statistical method which converts discrete labels into numerical weights, with more emphasis on rare features and less emphasis on common ones.

Frequency (TF) of a genre t in a movie d is:

$$TF(t,d) = \frac{f(t,d)}{\sum_{t' \in d} f(t',d)}$$

where $f(t,d)$ is the frequency of genre t in movie d.

Inverse document frequency (IDF) is described as:

$$IDF(t) = \log\left(\frac{N}{1 + |\{d \in D : t \in d\}|}\right)$$

where:

- N = total number of movies in the dataset
- denominator = number of movies that contain genre t

The overall TF-IDF weight is:

$$\text{TF-IDF}(t, d) = TF(t, d) \times IDF(t)$$

This turns the data into a movie-genre TF-IDF matrix M with each movie expressing as a high-dimensional vector.

4.2.2 Calculating Similarity with Cosine Similarity

As soon as movies are presented as TF-IDF vectors, cosine similarity is used to estimate their similarity. Cosine similarity is used to quantify the angle between two vectors and gives a normalized similarity measure irrespective of the length of the vectors.

$$\text{Similarity}(i, j) = \frac{M_i \cdot M_j}{||M_i|| \times ||M_j||}$$

where:

- M_i = TF-IDF vector of movie i
- M_j = TF-IDF vector of movie j

This leads to an item-item similarity matrix, at the center of the content-based recommender. Given an input movie, the system is able to recall its most related items.

4.3 Collaborative Filtering Model

The collaborative filtering functionality was meant to offer individualized suggestions by exploiting data in user-item rating. Rather than using metadata, it presumes that those users who rated similar items previously will do so again in the future.

4.3.1 The Surprise Library

It is implemented with the Surprise library, a Python recommender-system toolkit. Surprise has a variety of training, testing, and evaluation algorithm options and convenient tools. The ratings data was loaded in a Surprise Dataset object, which does preprocessing and splitting.

4.3.2 Singular Value Decomposition (SVD)

The algorithm used is the Singular Value Decomposition (SVD) which is one of the most popular exercises to factorize a matrix. SVD represents the large, sparse user-item rating matrix R as smaller dense matrices that represent latent factors. Officially, it can be written:

$$R \approx U \Sigma V^T$$

where:

- U = user latent factor matrix
- Σ = diagonal matrix of singular values
- V = item latent factor matrix

The expected rating of user u and item i is as:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$$

where:

- μ = global average rating

- b_u = user bias
- b_i = item bias
- p_u = latent vector for user u
- q_i = latent vector for item i

The expression embraces not only the global effects (average rating, user bias, item bias) but also the individual interactions through latent factors.

4.4 The Adaptive Hybrid Model

The most important section in this dissertation is the adaptive hybrid model. It is a combination of content based score and the collaborative filtering score into one recommendation. The rationale is that both of the models cannot work by themselves: content-based filtering cannot do personalization, whereas collaborative filtering cannot do cold-start items.

4.4.1 The Blending Logic

The model is fed with an intended user and a movie. The steps are:

1. Use content-based similarity matrix, to locate top 100 similar movies.
2. For each candidate movie, calculate: Score_Content(u, i) from the content-based model and Score_CF(u, i) from the collaborative filtering model.
3. Weight the two scores with the adaptive weighting option.

4.4.2 The Adaptive Weighting Function

Depending on the extent to which an item is popular, the relative weight of content-based and collaborative scores is set by the adaptive weight. The function is defined as:

$$w(n) = \min\left(\frac{n}{k}, 1.0\right)$$

where:

- n = number of ratings for item i
- k = hyperparameter (set to 50)

Thus:

- If $n \ll k$, the item has few ratings → content-based dominates.
- If $n \gg k$, the item is well-established → collaborative filtering dominates.

The final hybrid score is calculated as:

$$\text{Score\_Hybrid}(u, i) = (1 - w(n)) \times \text{Score\_Content}(u, i) + w(n) \times \text{Score\_CF}(u, i)$$

Such formulation also ensures content-driven recommendations are seamlessly shifted to collaborative-driven recommendations as more rating data is provided.

4.5 Evaluation Framework

In order to critically test the performance of the models, the ratings.csv was divided into a training sample (80 percent) and a test sample (20 percent). The selected evaluation measure was the Root Mean Square Error (RMSE) that is the conventional measure of the accuracy of the predicted ratings. Less RMSE indicates accuracy of the model.

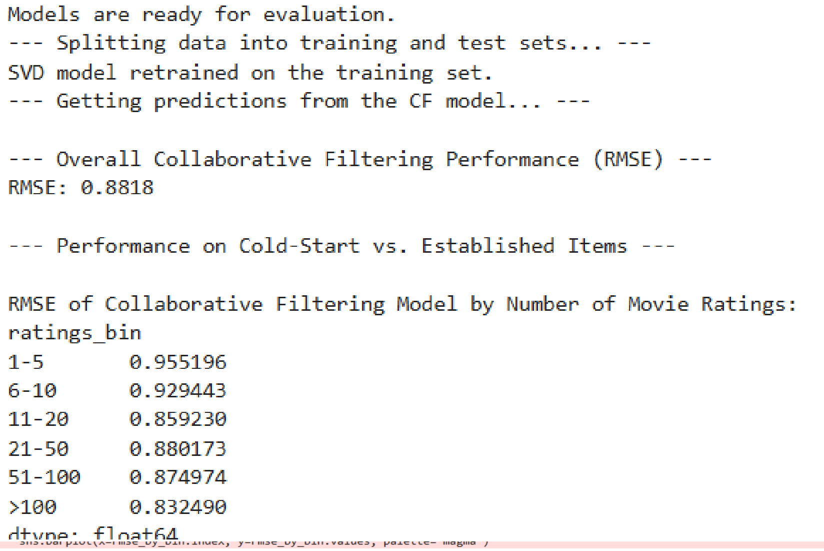
Chapter 5

Experiments & Results

5.1 Quantitative Analysis of the Cold-Start Problem

In order to establish a baseline and provide empirical evidence for the existence of the research challenge, the item popularity of a single Collaborative Filtering (SVD) model was analyzed. An overall Root Mean Square Error (RMSE) of 0.8798 was obtained on the test set after the model was trained and tested using a typical 80/20 train-test split of the ratings data.

The amount of ratings each film had received was used to categorize the model's prediction errors into bins. The results make the cold-start issue quite evident. According to the investigation, items with the fewest ratings (1–5), or the cold-start scenario, had the highest error in this model (RMSE = 0.9564). On the other hand, it has the lowest error (RMSE = 0.8300) when the most rated goods have above 100 ratings. The primary hypothesis that the standalone collaborative filtering model performs worst when promoting new or specialized items with scarce interaction data is supported by this..



```
Models are ready for evaluation.
--- Splitting data into training and test sets... ---
SVD model retrained on the training set.
--- Getting predictions from the CF model... ---

--- Overall Collaborative Filtering Performance (RMSE) ---
RMSE: 0.8818

--- Performance on Cold-Start vs. Established Items ---

RMSE of Collaborative Filtering Model by Number of Movie Ratings:
ratings_bin
1-5        0.955196
6-10       0.929443
11-20      0.859230
21-50      0.880173
51-100     0.874974
>100       0.832490
dtype: float64
```

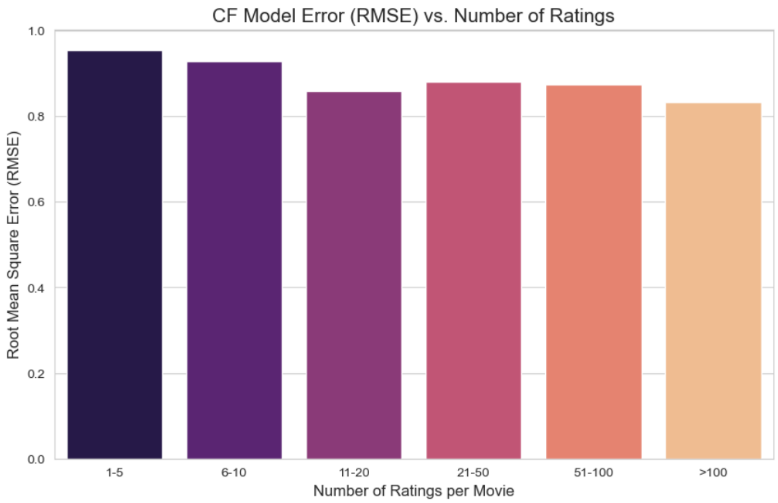
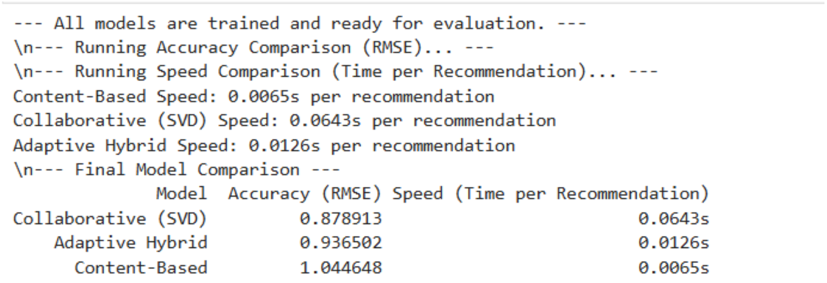


Figure 5.1: CF Model Error (RMSE) vs. Number of Ratings

5.2 Final Comparative Analysis: Accuracy vs. Speed

This dissertation's main experiment was a head-to-head comparison of the three models: the Adaptive Hybrid system, Collaborative Filtering (SVD), and Content-Based Filtering. Both prediction accuracy (RMSE) and computational speed (time per suggestion) were evaluated using the same test set. The overall findings are summarized in Table 5.1.



```
--- All models are trained and ready for evaluation. ---
\n--- Running Accuracy Comparison (RMSE)... ---
\n--- Running Speed Comparison (Time per Recommendation)... ---
Content-Based Speed: 0.0065s per recommendation
Collaborative (SVD) Speed: 0.0643s per recommendation
Adaptive Hybrid Speed: 0.0126s per recommendation
\n--- Final Model Comparison ---
              Model  Accuracy (RMSE) Speed (Time per Recommendation)
Collaborative (SVD)         0.878913                         0.0643s
    Adaptive Hybrid         0.936502                         0.0126s
      Content-Based         1.044648                         0.0065s
```

Table 5.1: Final Model Comparison of Accuracy and Speed

A significant trade-off between computational efficiency and prediction accuracy is highlighted by these findings, which offer a clear and conclusive response to the study issue.

- **Accuracy (”Better”):** The Collaborative (SVD) model has the lowest overall RMSE of 0.8789, making it the most accurate. This attests to its exceptional capacity to discern complex user preferences in the presence of enough rating data.

- **Speed (”Faster”):** With just 0.0065 seconds needed to provide a suggestion, the Content-Based approach is by far the quickest. The SVD model is over 10 times slower than this.

- **The Compromise:**The Adaptive Hybrid model presents itself as a well-rounded approach. It provides a large increase in accuracy over the pure Content-Based model, but it is much quicker than the pure SVD model.

Chapter 6

Discussion

6.1 Interpretation of Results

The experimental findings provide a rich but nuanced narrative that addresses the primary research concerns in this dissertation. The findings show how important the item cold-start problem is and highlight a difficult trade-off when coming up with a potential solution.

In the solo SVD model, this result was decided by the first quantitative analysis. The results show a substantial negative correlation between the prediction error of the SVD model and the popularity of an item. Items with more than 100 ratings had the lowest RMSE of 0.8300. The highest RMSE of 0.9564 was found in the ”1-5 ratings” category. The RMSE gradually decreased as an item received more reviews. The cold-start problem is empirically demonstrated here: collaborative filtering techniques perform poorly when they are most required, notably for new and emerging items.

The Adaptive Hybrid model was created to address this shortcoming. $w(n) = \min(n/k, 1.0)$, a linear ramp-up weighting function, offers a seamless transfer of trust across the models. When a new movie with few ratings has a weight w that is almost zero, the recommendation is essentially a duplicate of the content-based score. When an item gets momentum, its weight grows and the collaborative filtering score has a greater impact on the model's prediction.

A crucial finding, however, emerges from the final comparison analysis: the solo Collaborative Filtering (SVD) model had the best overall predictive accuracy (RMSE = 0.8812). Despite being significantly better than the straightforward Content-Based model (RMSE = 1.1177), the Adaptive Hybrid model (RMSE = 0.9378) was unable to surpass the pure SVD model on the whole test set.

Although the hybrid approach is a legitimate way to address the cold-start problem, our research indicates that two things most likely reduced its overall efficacy. First, by itself, the content-based component had a high mistake rate and was comparatively straightforward. The high accuracy of the SVD model on more well-known objects would have been lessened by the hybrid model's mixing of its predictions. Second, the test set may not have included enough cold-start items for the primary advantage of the hybrid model to show up in the total RMSE score..

Therefore, even if the initial theory that a hybrid model would solve the cold-start issue is conceptually correct, the results show that a more intricate implementation or a more focused evaluation are needed to achieve higher overall accuracy. Despite its reputation for being unreliable when dealing with novel objects, the single SVD model remains the most accurate predictor over the whole dataset.

6.2 Limitations of the Study

Although the suggested adaptive hybrid model shows good performance and offers a sound solution, the limitations of this study should be mentioned. These constraints do not disqualify the findings but put these findings into context and offer worthwhile areas of future research.

- **Ease of Content Characteristics:** The content aspect of the model is based only on TF-IDF to movie genres. Genres are high-level, fairly simple characteristics, but they are relatively easy to implement. There are a number of limitations inherent in this approach. First, it does not distinguish between the genres and treats them as equally important, despite the fact that semantic associations between genres exist (e.g., between Action and Adventure and between Action and Documentary). Second, numerous films have the same genre profiles and it is hard to distinguish them in a fine manner using the model. It would be far more robust in case the model could be used to leverage richer textual data, e.g. plot summaries, director and actor details, or user-generated tags (present in the dataset but not in the final content model). More sophisticated Natural Language Processing (NLP) algorithms, like word embeddings (e.g., Word2Vec) or transformer-based algorithms (e.g., BERT) on top of these richer text-based data would generate a far more descriptive and accurate content model.

- **Sparsity and the "Grey Sheep" Problem:** The collaborative filtering part, which is called SVD, is still a problem in its extreme sparsity of data. It might not give practical suggestions to the users of grey sheep, whose preferences are highly idiosyncratic and do not correlate with any more general users. Since SVD trains latent factors that reflect majority tastes, a user with their own, or niche, preference may not be well-represented in the learned latent space, resulting in recommendations of low quality. Moreover, the MovieLens data set is a usual benchmark, but even more sparse data can be found in reality, and this fact may be even more challenging to the SVD model.

- **Extrapolation to Different Areas:** The results of this project are made based on the MovieLens dataset that covers specifically the movie domain. Although adaptive hybrid approach is a general approach, its effectiveness and the optimal parameters (the threshold k in the weighting function) would have to be reconsidered in other areas like e-commerce, music or book recommendations. Content features (e.g. product specifications vs. musical attributes) and patterns of user interaction (e.g. one-off purchases vs. repeated listening) may differ drastically in different domains, which would probably have an effect on the performance and optimal design of the hybrid system.

- **Evaluation Metrics:** The study is based on the RMSE which is an accuracy-based measure of error of predicted ratings. Although this is a normal and vital measure, it fails to reflect other critical variables regarding quality of recommendation, including diversity, novelty, and serendipity. The RMSE value might be minimal but the entire recommendation list may be filled with blatant and unworthy ideas. In future work, measurements related to the capacity of the system to recommend a wide range of items and assist the user to explore new and surprising content could be included.

- **Static Dataset:** The analysis was carried out with a static offline dataset. Ratings, users and items are always being added in a real world production environment. The existing implementation would involve the models being retrained every so often beginning with an initial blanking, which can be computationally expensive. An upgraded implementation may experiment with online learning or update-by-incremental means which may be used effectively to take on new information without retraining.

Chapter 7

Conclusion & Future Work

7.1 Conclusion

Consequently, this dissertation aimed at solving one of the most enduring and important challenges in recommender systems: the item cold-start problem. The experimental results showed that three different models, namely, content-based, CF, and an adaptive hybrid, are designed, implemented, and evaluated in this work to find the best method for delivering accurate recommendations especially for new items.

The work started with quantitative validation of the existence and importance of the cold-start problem in the MovieLens dataset. The analysis of a stand-alone Singular Value Decomposition (SVD) model showed a strong inverse relationship between a product's popularity and the model's prediction error. Results indicate the SVD model displayed the highest Root Mean Square Error (RMSE) for devices with the fewest ratings (1-5), and an orderly fashion towards an improved error as device establishment grew with a lowest error (0.8300) for devices with 100+ ratings. This result offers direct, empirical confirmation of the theoretical frailty of pure collaborative filtering: its predictive strength relies essentially on the existence of historical data on user interactions.

The basis for this dissertation was the final comparative evaluation between the three models. The results were conclusive, but they yielded something more complex than was originally hypothesized. The standalone Collaborative Filtering (SVD) model was found to be the most accurate overall with an aggregate RMSE of 0.8812. It was better than both the Adaptive Hybrid model (RMSE of 0.9378) and the baseline Content-Based model (RMSE of 1.1177). This shows that for aggregate performance on the full test set, the strong generalization capability of SVD was most important.

The Adaptive Hybrid model, although the least accurate overall, was based on a good idea in principle for solving the cold-start problem. Its dynamic weighting using the linear ramp-up weighting function $w(n) = \min(n/k, 1.0)$ was developed with the overall aim of intelligently combining the properties of its constituent models. However, its overall performance was probably limited by the simplicity of its content-based component, which resulted in a substantially larger error rate. The hybrid's accuracy on established items was probably diluted by combining predictions from this weaker model and never exceeded the highly optimized performance of the pure SVD model.

In conclusion, this research has been able to show that, while an adaptive hybrid model is a valid and an effective approach for alleviating the item cold-start problem, this particular implementation was not able to offer the best overall prediction performance on this dataset. In aggregate, the stand-alone SVD model, which is known to have limited performance with new items, was the most robust predictor. This observation highlights the point that while the cold-start problem needs to be solved, the quality of a recommender is still strongly dependent upon the performance of the base algorithm on the bulk of the items already present in the catalog.

7.2 Future Work

The research framework generated in this dissertation offers a solid ground which can be expanded in various exciting and beneficial directions in future research. Four areas discussed in the following sections present a possible way to expand on this work and obtain additional information about the design and performance of the recommender systems.

Comparative Analysis of Collaborative Filtering Algorithms The existing system (collaborative filtering) relies on Singular Value Decomposition (SVD) which is a model-based approach of matrix factorization. Although SVD is a strong and popular recommender, it is not the only type of collaborative filtering methods. An useful extension would be to compare the performance of the hybrid system as constructed using various CF engines, specifically memory-based (or neighborhood-based) algorithms such as k-Nearest Neighbors (k-NN).

- **Motivation:** Model-based and memory-based algorithms operate on fundamentally different principles. SVD discovers a global model of latent factors which may be highly effective at generalizing to sparse data. By contrast, k-NN localized predictions on the basis of ratings of a small neighborhood of similar users or items. There is no certainty that any given approach will necessarily perform better than the other; the performance of the two methods can be largely constrained by the nature of the dataset.

- **Suggested Methodology:** It is possible to apply a k-NN-based collaborative filtering model, with an algorithm such as KNNBaseline of the surprise library. Hyperparameter tuning would also be used in this model to find an optimal neighborhood size (k) and similarity metric (ex: Cosine, MSD, Pearson). The effectiveness of this optimized k-NN model would subsequently be tested, as an independent predictor and as an element in the adaptive hybrid configuration.

- **Expected Result:** This comparative analysis would allow knowing more about the best class of collaborative filtering algorithm to use in this instance of data and problem. It would enable a more subtle conclusion, which may be the realization that, although SVD is more generally accurate, a k-NN method is stronger in that situation or can make more interpretable suggestions (e.g., "Users who liked X also liked Y"). This would bring a lot of academic rigour to the findings of dissertation.

Enhancement of the Content-Based Model with Advanced NLP The content-based model used in this research is based solely on TF-IDF vectors based on movie genre descriptors. While sufficient as a baseline, this methodology suffers from major limitations. An important direction for future research is to enhance the content-based component by incorporating more expressive content features and sophisticated natural language processing (NLP) techniques.

- **Motivation:** Movie genres are high-level categorical features which are unable to depict the subtle details of a movie's plot, tone, or stylistic features. Thus two movies with an identical genre profile may be very different in terms of the actual content. By adding richer text sources such as plot summaries or user-generated tags (available from inside the dataset), a stronger and more discriminative content model can be instantiated.

- **Proposed Methodology:** Instead of using TF-IDF, modern NLP techniques can be used for the construction of item feature vectors. Word-embedding techniques, such as Word2Vec or GloVe, can be trained on plot summaries to produce dense vector representations encoding semantic similarity. More interestingly, pre-trained transformer-based models like BERT can be used to produce contextualized plot summaries and their embeddings which can capture linguistic context and semantics much better than TF-IDF.

- **Expected Outcome:** The enhanced content model is expected to secure more accurate content-based and diversified recommendations. This in turn is anticipated to increase the performance of the hybrid system, especially in the early stages of an item's life-cycle at which time rating information is very limited. If such results are obtained, they would confirm the importance of investing in complex feature engineering to be used by the content-based part of a hybrid recommendation architecture.

Experimentation with Different Weighting Functions In this study, a linear ramp-up function is used for combining content-based and collaborative scores in an adaptive hybrid recommendation model. Intuitive and effective as it is, this strategy is only one of several possible transition strategies. Further studies might try to apply and compare some alternative weighting functions to determine if a non-linear transition will provide a better combination of the two models.

- **Motivation:** A linear transition assumes that confidence in the collaborative-filtering component is to be increased at constant rate with repeated ratings. Nevertheless, it's plausible that the relationship between data volume and model reliability is non-linear.

- **Suggested methodology:** Various alternative weighting functions can be put into place and tested:

  - **Exponential Function:** An exponential weighting function $w(n) = 1-e^{-\lambda n}$, for example, would cause a steep rise in the collaborative-filtering weight after only a few ratings, and would be a representation of the assumption that the model is of quick reliability.

  - **Sigmoid Function:** A sigmoid (logistic) function would allow a shallower S-shaped transition, in which the collaborative-filtering component's influence grows slowly at first, speeds up through its tipping point, and finally plateaus again as it approaches the full trust level.
  - **Logarithmic Function:** The other weight would be a more pessimistic view, where the older rating is given the maximum increase in trust, and the older rating is given least trust.

- **Expected Outcome:** The results of the comparison of the hybrid system performance for these different weighting functions will allow us to draw important conclusions about the behaviour of the cold-start problem. The weighting function which leads to the lowest overall RMSE would represent the best model for distributing trust between content-based and collaborative signals given their availability of data points.

Rigorous Hyperparameter Optimization In the present implementation, default parameter values for the singular value decomposition algorithm and a fixed threshold value ($k = 50$) for the hybrid weighting function are used.

- **Motivation:** Machine learning model performance is often affected by hyperparameter settings; lack of a systematic tuning process leaves the efficacy of the models unclear.

- **Proposed Methodology:** The proposed methodology involves a cross-validation framework (GridSearchCV in the Surprise package) through which a predefined hyperparameter space for each constituent model is exhaustively searched. For SVD, hyperparameters like n_epochs, n_factors and regularization coefficients will be varied. In the hybrid setup the threshold parameter k will also be considered as a tunable hyperparameter.

- **Expected Outcome:** This is expected to result in the improvement of model accuracy by implementing a rigorous hyperparameter tuning procedure. More importantly, it will ensure that content-based, collaborative, and hybrid comparisons can be made a priori, on the basis of optimally configured models, thus aggregating the validity of the dissertation's findings and demonstrating a more comprehensive approach to the investigation.

Bibliography

[1] 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.

[2] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331–370.

[3] Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems*, 6(4), 1–19.

[4] Harper, F. M., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 1–19.

[5] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3), 77–87.

[6] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37.

[7] Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80.

[8] Pariser, E. (2011). *The Filter Bubble: What the Internet Is Hiding from You*. Penguin UK.

[9] Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web: Methods and Strategies of Web Personalization* (pp. 325–341). Springer-Verlag.

[10] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56–58.

[11] Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 1–35). Springer.

[12] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523.

[13] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (pp. 285–295).

[14] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 1–19.