A PERFORMANT PREDICTIVE ANALYTICS APPROACH TO RECOMMENDER SYSTEMS USING DEEP LEARNING METHODS

A Thesis Presented to the Department of Computer Science

African University of Science and Technology, Abuja

In partial fulfillment of the requirements for the award

MASTER OF SCIENCE DEGREE IN COMPUTER SCIENCE

By

USMAN WILLIAMS OBAJE

Supervised by

Dr. Ekpe Okorafor

African University of Science and Technology

www.aust.edu.ng

P.M.B 681, Garki, Abuja F.C.T

Nigeria.

September 2022

A PERFORMANT PREDICTIVE ANALYTICS APPROACH TO RECOMMENDER SYSTEMS USING DEEP LEARNING METHODS

By

USMAN WILLIAMS OBAJE

A THESIS APPROVED BY THE DEPARTMENT OF COMPUTER SCIENCE

RECOMMENDED:

....

Supervisor, Dr. Ekpe Okorafor

....

Head, Department of Computer Science

APPROVED:

....

Chief Academic Officer

:

....

Date

ABSTRACT

Recently, Massive amounts of data have been generated as a result of the frequency at which the amount of information available digitally is advancing. To enable users to effectively utilise the huge amount of information, recommendation system has been implemented to effectively manipulate a large amount of data in other to communicate necessary output to the user. The reliability of the final recommendations is a common metric used to determine if a recommendation system is effective or not. The RMSE, MSE, and MAE were used as recommendation-based metrics. The recommender system's performance is typically measured using the metrics RMSE, MSE, and MAE. This statistic demonstrates how effectively the Recommender system performs. The performance of the recommendations improves with decreasing RMSE, MAE, and MSE. It offers an erroneous value that illustrates how far off from the real data our model was. It assesses how closely the projections supplied correlate to the quantities that were observed. The final result of evaluating RMSE, MAE, and MSE on 1M Movies Datasets. Taking the mean result of the output, MAE outperformed other matric because it has the lowest mean value of 0.5843. Also, evaluating results of algorithm SVD on the 100k movies dataset MAE outperformed other matric having the lowest output of 0.6593. Furthermore, evaluating the RMSE, MAE, and MSE of algorithm SVD on 5k movies data set, MAE still outperformed other matric having the lowest mean value of 2.8898. Ultimately, it was discovered that the movie data sets with the most customers and reviews performed better than the others with fewer datasets obtainable. Additionally, we suggest a deep learning approach for creating efficient and accurate deep learning collaborative filtering systems (DLCFS). The proposed method and the currently used methods were compared.

ACKNOWLEDGEMENT

I want to extend my thankfulness to God Almighty for giving me the strength, information, inspiration, and insight needed to complete my study and my research.

I want to convey my sincere gratitude to my supervisor, Dr. Ekpe Okorafor, for his amazing assistance as well as his commitment, inspiration, recommendations, and corrections. I appreciate your confidence in me.

I want to extend my sincere gratitude to Dr. Rajesh Prasad, Head of the Department of Computer Science, for mentoring us all along the way. I also want to appreciate all the lecturers who taught us, such as Prof. Suneeta Agarwal, Dr. Eddy Kwessi, and prof. Ben Abdallah, Prof. Lehel Csato, prof. Mohammed Hamada, and Amos David, among others. We appreciate you all for taking the time to teach us and for bringing out the best in us.

I owe my older brother, Dr. Usman Timothy O., an unreserved debt of gratitude for his encouragement and contribution to making this program a reality.

I want to take a moment to thank my entire family for their unwavering support throughout all of my academic endeavors. Thank you for everyone's generosity. My sincere gratitude also goes out to all of my friends, my colleagues of the AUST Class of 2021–2022, and the other scholars in the computer science department for their support, friendship, and cooperation.

Finally, a sincere thank you to everyone who helped make the programme successful in one way or another. I pray that God will lavishly bless you.

DEDICATION

In acknowledgment of all of God Almighty's unfathomable favours in my life, I dedicate this work to him.

TABLE OF CONTENTS

ABSTRACT ..... **iii**

ACKNOWLEDGEMENT ..... **iv**

DEDICATION ..... **v**

TABLE OF CONTENTS ..... **vi**

LIST OF FIGURES ..... **viii**

LIST OF TABLES ..... **ix**

LIST OF ABBREVIATIONS AND ACRONYMS ..... **x**

CHAPTER ONE **1**

INTRODUCTION **1**

1.1 Background of the Study 1

1.2 Problem Statement 4

1.3 Aim and objectives 5

CHAPTER TWO **6**

LITERATURE REVIEW **6**

2.1 Information Filtering (IF) and Information Retrieval (IR) system 6

2.2 Recommender System Types and Techniques 7

2.2.1 Recommendation System Entities 7

2.2.2 Collaborative Filtering (CF) 10

2.2.2.1 Similarity Measures in CF Algorithm 11

2.2.3 Content-Based Recommendation (CBR) 13

2.2.4. Hybrid-Based Recommendation (HBR) 14

2.3 Comparison of Recommender Systems Techniques 16

2.3.1 Various data mining techniques used with a recommender system. 17

2.4 Deep Learning for Recommendation 18

2.4.1 Deep Neural Network Models for Recommendation 18

2.5 NVIDIA Merlin 19

2.5.1 Components of NVIDIA Merlin 19

2.6 Related Work 20

CHAPTER THREE **25**

RESEARCH METHODOLOGY **25**

3.1 Introduction 25
3.2 The Major Concern in Collaborative Filtering.. 25
3.2.1 Recommender systems Generation 26
3.3 The Proposed Methodology 27
3.4 CF Using Nvidia Merlin.. 28
3.4.1 Nvtabular 29
3.4.2 Data Loading For RS.. 30
3.4.2.1 NVTabular data loader... 30
3.5 Performance Evaluation Criteria.. 30
3.6 Summary 31
CHAPTER FOUR.. 32
IMPLEMENTATION, RESULTS, AND DISCUSSION 32
4.1 Introduction 32
4.2 Performance Measure and Evaluation... 32
4.3 Experiment (A) – Implementation & Results 34
4.3.1 Movie Lens Rating Information 35
4.3.2 Performance Measure... 37
4.4 Experiment (B) - Implementation & Analysis 39
4.4.1 Pre-processing.. 39
4.5 Comparison between Traditional Collaborative filtering and Deep Learning Collaborative Filtering Systems.. 42
CHAPTER FIVE 43
Summary, Conclusions & Recommendations 43
5.1 Summary 43
5.2 Conclusion.. 43
5.3 Recommendations 43

LIST OF FIGURES

Figure 1: Structure of a Recommender System 2
Figure 2: Deep Learning Architecture 3
Figure 3: The example of the taxonomy of the recommender systems 9
Figure 4: Recommendation system Techniques 10
Figure 5: Collaborative Filtering Vs Content-Based Filtering 14
Figure 6: Example of a Hybrid System........ 15
Figure 7: Deep Learning for Recommendation Training 18
Figure 8: Recommender System Generations........ 26
Figure 9: NVIDIA Merlin Recommender System Framework 28
Figure 10: Criteo dataset processing ETL 29
Figure 11: MSE Formula 33
Figure 12: MAE Formula........ 33
Figure 13: RMSE Formula........ 34
Figure 14: Rating Distribution for 1M movies Datasets........ 36
Figure 15: Rating Distribution for 100k movies Datasets 36
Figure 16: Rating Distribution for 5K movies Datasets 37
Figure 17: Evaluating RMSE, MAE, MSE of Algorithm SVD on 1M Movies Datasets........ 38
Figure 18: Evaluating RMSE, MAE, MSE of Algorithm SVD on 100k Movies Datasets........ 38
Figure 19: Evaluating RMSE, MAE, and MSE of Algorithm SVD on 5K Movies Datasets 39
Figure 20 NVIDIA System Management Interface (Nvidia-smi) 40
Figure 21 preprocessing pipeline........ 41
Figure 22: Model Evaluation 41

LIST OF TABLES

Table 1: Lists of the benefits and drawbacks of recommender systems....................................... 16
Table 2: Comparing the benefits and solutions of several publications 20
Table 3: Extract of Rating Data 35

LIST OF ABBREVIATIONS AND ACRONYMS

The list of abbreviations and acronyms used in the duration for research purposes is as follows. We defined them as follows for clarity:

• CF	Collaborative Filtering
• CB	Content-Based
• IF	Information Filtering
• IR	Information Retrieval
• RS	Recommendation system
• DL	Deep Learning
• MF	Matrix Factorisation
• NLP	Natural Language Processing
• IF	Information Filtering
• IR	Information Retrieval
• IMDB	Internet Movies Database
• ALS	Alternating Least Square
• GPU	Graphical Processing Unit
• CPU	Central Processing Unit
• ETL	Extract Transform, and Load
• MAE	Mean Absolute Error
• MSE	Mean Squared Error
• RMSE	Root Mean Square Error
• HugeCTR	Huge Click-Through-Rate
• MF	Matrix Factorization
• DLCFS	Deep Learning Collaborative Filtering System

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The rapid expansion of content on the internet has resulted in large volumes of data and an increase in online subscribers. This massive data explosion has inundated people with massive amounts of data, posing a significant difficulty in terms of information overload. As a result, it is extremely difficult for humans to manually digest such information and even more difficult for them to find the correct information. Large internet corporations such as Amazon, Google, and Facebook have struggled to keep up with this avalanche of data. Recommendation systems have been used to intelligently transform this situation.

Big data is on the rise as a result of the enormous growth in online data and users. The Recommendation System has received the greatest interest in the Big Data industry. Big Data has increased our ability to make widespread suggestions. Due to its ability to forecast the correct information from a massive amount of information, the recommendation system has become more crucial for users. This system is a special sort of data screening that uses past user behaviour or the actions of those who are related to the user in consideration to construct a collection of data sources that is specifically tailored towards the end user's preferences.

Before the advent of e-commerce services, customers had to spend a lot of time browsing through lists of goods or services to locate what they were looking for. The users' adoption of the technology and the availability of recommender systems, however, led to an increase in the services' revenues. At the moment, recommender systems are employed by e-commerce businesses in many different industries, such as Netflix, Amazon, and YouTube consumers (Deuk et al, 2011). The best way to use them in each field is the subject of numerous ongoing studies. Other recommendation systems, like knowledge-based filtering and CBF (Adomavicius, G., & Tuzhilin, A., 2005). Were developed to address the issues of CF.

Several studies have demonstrated that by using MF techniques, the aforementioned challenge can be overcome. Even though the technique is probably going to suffer from a lack of some important signals due to the use of a low-ranked approximation as well as a lack of sparsity when singular vectors are denser. DL approaches have recently demonstrated their ability to learn accurate representations for tasks like picture classification and natural language processing.

This research will need a set of recommender systems to use as a case study to see how well they perform. That is, measuring their correctness in terms of recommendation. This will imply recording their MAE, MSE, and RMSE. This information would likely form our dataset which you will correlate with the goal and function of the recommender system in question. Then there is a need to investigate why the results as recorded are the way they are.

Recommender systems (RS) are computational models that generate predictions or suggestions for users based on data acquired from multiple sources or user similarities. The computers then analyze the users' purchasing behaviors and preferences and provide recommendations based on their findings. A concern to be handled by methodical means in the field of RS research is how to continuously modify a suggestion mechanism to the subscriber's desired information available at a set period. Figure 1 shows and describes the structure of the typical structure of a recommender systems

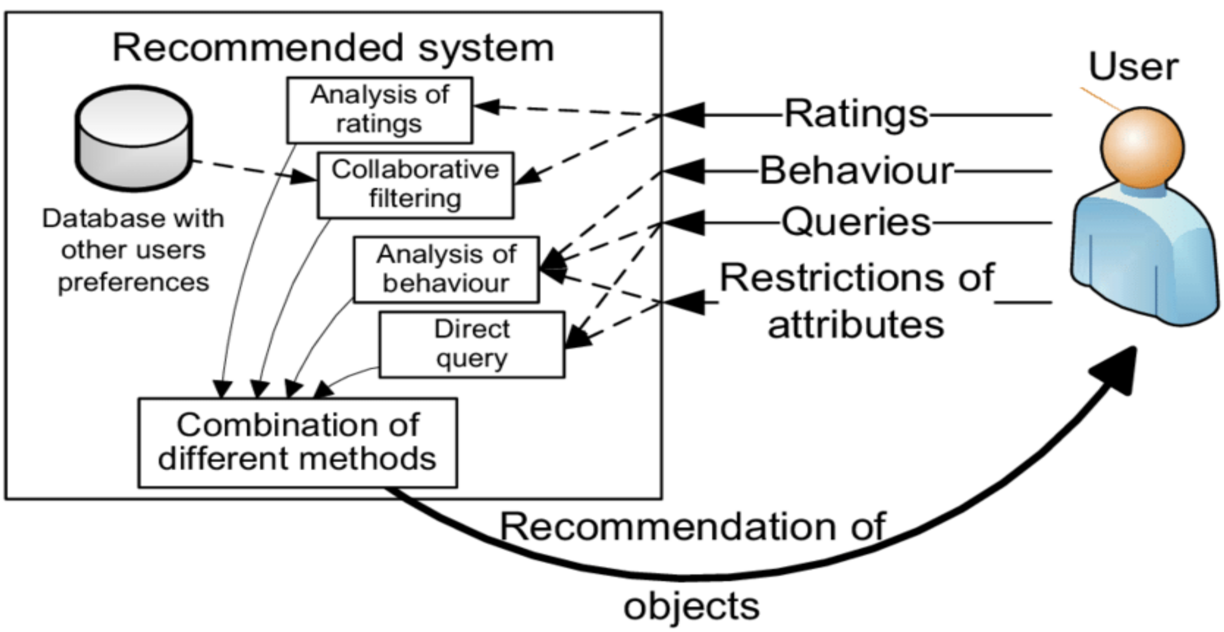

Figure 1: Structure of a Recommender System

Deep learning (DL) and its capacity to foretell the next obvious item or product for a visitor will have a significant impact on the future of previously stated methodologies, including CBF, CF, and HRS.

In this study, we will go over some of the key ideas underlying deep learning, the reasons why it's starting to catch on, and how it’s an incredibly successful application in natural language processing (NLP) being translated to Deep Learning-based recommendation engines.

The enormous success of DL approaches in the last ten years has thrilled the scientific community and contributed significantly to the recent growth of recommendation systems. DL in computer vision and NLP has enabled the most significant of these changes in NLP. Even though the great majority of conventional filtering techniques conduct linear analyses of the data.

The architecture of the deep learning system makes it possible to analyze data non-linearly. This enables deep learning algorithms to more effectively separate valuable information from the data, create a large number of new features, and generate further features automatically.

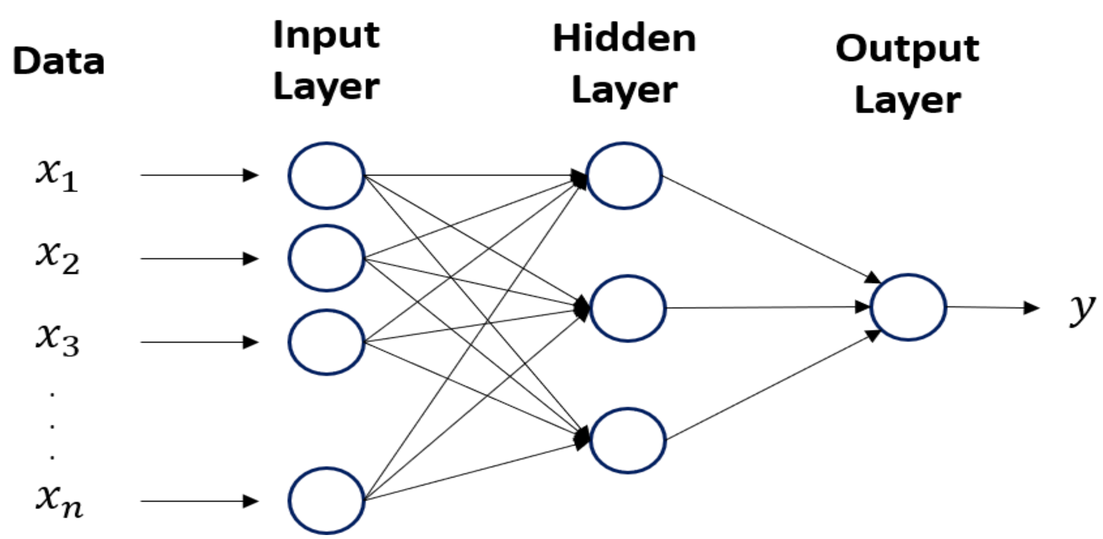

Figure 2: Deep Learning Architecture

1.2 Problem Statement

Most recently, the MAE or RMSE between the projected rating and the observed interest rating is the most widely accepted RS assessment measure. These measurements compute accuracy without making any assumptions about the RS's intent.

However, as McNee & Konstan, (2006) point out, there is a lot more to judging whether or not something should be recommended than just accuracy. Herlocker et al. (2004) give a complete assessment of computational approaches to the recommender's evaluation. They imply that some methods might be better suited for particular jobs. Furthermore, they are unable to validate the metrics when assessing the alternative strategies directly on a category of existing algorithms and a certain collection of data.

Recommender systems were mainly utilized on home computers before the increasing adoption of smartphones. Initially, recommender systems were modelled by concentrating just on people and things, and they worked based on fundamental data like users' purchase histories or assessment scores. Simply said, recommender systems are methods and software tools that provide suggestions for products that users would find beneficial. The suggestion covers a wide range of decision-making operations, such as choosing what to purchase, what entertainment to subscribe to, and what news content to consume. To assist users in finding information, goods, or applications, such as publications, films, songs, online content, webpages, and Television programs, recommender systems gather and evaluate suggestions from other.

To produce high-quality suggestions and address the issues with pure CF, which primarily includes accuracy, scalability, neighbour transitivity, sparsity, and cold start, several deep learning algorithms or approaches have been introduced recently. But other strategies, such as knowledge- and content-based filtering, experience nearly the same limitations.

1.3 Aim and objectives

In the area of recommenders, we want to look into the possibilities for performance recommendations in unique situations and obstacles. We also intend to establish a DL model that would enable high-performance recommendations.

The objectives are:

1) Employing conventional collaborative filtering techniques on existing movies dataset

2) To find the recommender system's most efficient performance recommendation methods.

3) Enhancing effectiveness by incorporating the primary component of current recommenders, CF, with DL

4) To obtain the most effective performance recommendation techniques for the recommender system.

5) To develop an enhanced movies recommendation system model using NVIDIA Merlin

1.4 Significance of the Study

CF has emerged as the most attractive and well-liked recommendation approach for recommenders after evaluating the many recently created recommender systems. Although CF has been successful in a variety of application contexts. The CF approach still has important limitations, such as the capacity to handle data sparsity, cold start difficulties, and complexity, to name just a few. Due to data scarcity, applicability and relevance are diminished. Data sparsity is a word used to describe a circumstance in which consumers typically only rate a few products.

This research seeks to improve the accuracy of the contemporary CF framework by using the DL approach.

CHAPTER TWO

LITERATURE REVIEW

2.1 Information Filtering (IF) and Information Retrieval (IR) system

The research domains of IR and IF have become very active. The challenges posed by the massive amount of information that is easily accessible online, much of it is fundamentally unstructured, have called attention to the need for effective mechanisms to separate the useful information from the irrelevant. Finding information within a document, within materials themselves, as well as within databases of texts, images, or sounds, and the metadata that defines those databases is known as information retrieval. While an information filtering system eliminates unnecessary or superfluous information from an information stream using semi-automated or computational processes before presenting it to a human user.

Information retrieval (IR), a well-established field of information science, addresses the problems that occur when users request certain documents be retrieved from collections. Information filtering (IF), a relatively new field of research in information science, has grown in prominence as a result of the increase in the volume of online transitory data. Information retrieval (IR), a well-established field of information science, addresses the problems that occur when users request certain documents be retrieved from collections. Information filtering (IF), a relatively new field of research in information science, has grown in prominence as a result of the increase in the volume of online volatile data Bellogin (2012). Top-N recommendations aim to suggest to each customer a small group of N products from a huge selection of products Cremonesi et. al., 2010.

Instead of concentrating on user interactions and comments, a content-based approach calls for a sizable amount of data about specific product features. Natural Language Processing, for example, can extract movie attributes such as genre, year, director, actor, and so on, or article textual content.

On the other hand, collaborative filtering only needs users' prior preferences or information about a group of objects. The fundamental presumption is that people who have previously agreed tend to agree in the future since it is based on verifiable evidence. Collaborative filtering offers suggestions based on similarities between individuals and things simultaneously, addressing some of the drawbacks of content-based filtering.

As a result, synchronistic recommendations are made possible, which take place when collaborative filtering algorithms suggest an item to user A based on the interests of user B who shares those interests. Furthermore, without the requirement for feature engineering, the retrieved features can be automatically learned.

2.2 Recommender System Types and Techniques

The idea of using computers to recommend the best product for a user has been around since the dawn of computing. The first implementation of the recommendation concept appeared in 1979, in the form of Grundy, a specialized comprehensive computer-based Liberian that suggested books to the user. A recommendation system, commonly referred to as a recommender system, is a division or class of information filtering system that makes predictions based on the user's "rating" or "preference" for the item.

Recommender systems have advanced in terms of user needs, allowing users to handle massive amounts of data or information on the internet. It was a huge success or a solution to the problem of information overload. Furthermore, it has emerged as an important platform for connecting people with similar interests (Terveen, L., & Hill, W. 2001). A system that uses models created using machine learning tools and algorithms is known as a recommender system. Collaborative filtering and content-based filtering are the two categories. Its main objective is to assist in the filtering and recommendation of information.

The main goal of a recommender system is to propose to help users with various decision-making processes. This method is used by the majority of technology conglomerates, like Amazon, to show a user a list of suggested products that might be of interest based on the user's prior preferences and actions.

2.2.1 Recommendation System Entities

An entity is something that exists apart from other things and has its existence. There are a few categories of entities in the recommender system that are highlighted and discussed as follows.

- User-user: The "people like you, like that" logic is a popular or widely used recommender system algorithm, also known as a "user-user" algorithm because it suggests or recommends a specific item to a user if similar users linked to this item previously.
- Item-item: This is a kind of recommendation system that bases recommendations on the similarity between items as indicated by user ratings.
- User-based: refers to a technique for figuring out which things a user will likely enjoy based on the ratings that other users who share the target user's tastes have given to those products.
- Memory-based collaborative filtering creates a prediction using all the information in the database. Both user-based and item-based components are present.
- Through the use of matrix factorization, the relationship between the entities of things and people is discovered cooperatively.
- Content-based filtering creates recommendations following user preferences for product qualities.
- The Demographic Filtering (DF) technique makes use of the user's demographic data to determine which products could be appropriate for the proposal.
- An explicit understanding of the item assortment, user preferences, and proposal criteria forms the basis of one type of recommender system in particular, known as a knowledge-based recommender system.
- When there is insufficient data to make recommendations to a new user or object, it is referred to as a "cold start."
- A problem known as sparsity occurs when there are not enough data points to identify similar users.
- When the performance and latency of the RS significantly decline with a growth in the number of users and system components, this is known as a scalability problem.
- Finding objects (or users, or user and item) that are similar is what a recommender system refers to as similarity. It depends on the sort of recommender one employs as to how to measure it.

Particularly, recommender systems include the following elements:

Background data, or the knowledge with which the system starts before making recommendations,

- Input data, the details the system needs from the user to provide a recommendation, and
- Finally, an algorithm generates suggestions by combining input and background information.

Burke, R. (2002) claimed to have identified five different recommendation approaches (Figure 2.1) depending on the backdrop and input data's nature, the algorithm used to produce the suggestions, and other factors.

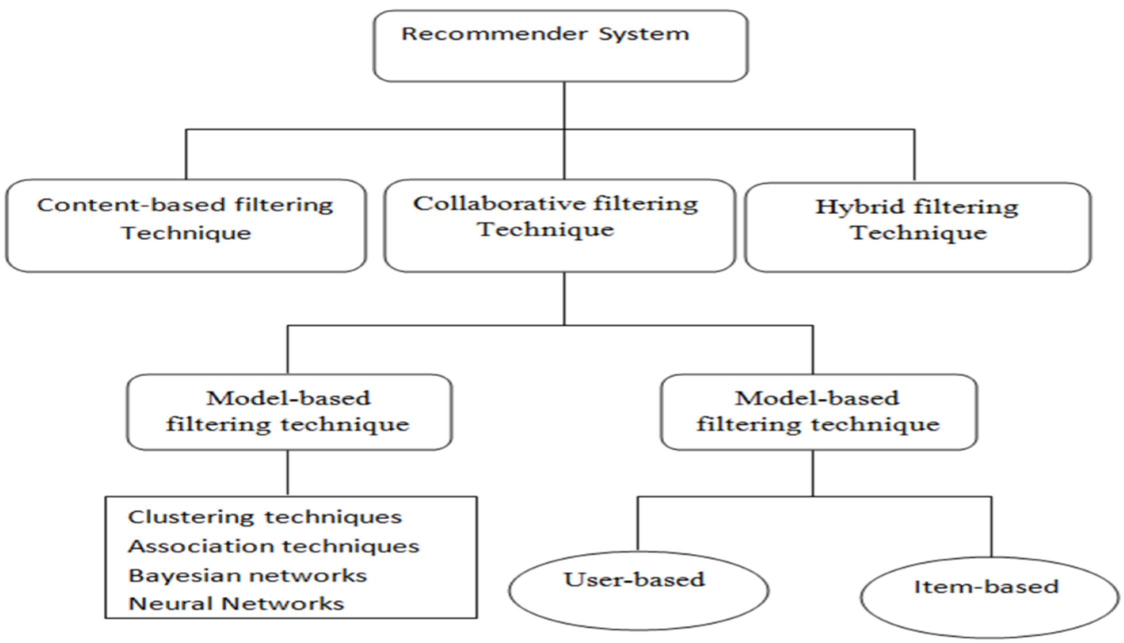

Figure 3: The example of the taxonomy of the RS

Recommender systems gather user data from multiple approaches and sources to determine which items a user will need and to recommend them based on the findings of the analytical process. (Priyanga, P., & Kamal, A. N. B., 2017). Another researcher makes the following distinctions between the three (3) basic types of RSs:

- Collaborative filtering
- Content-based filtering

- Hybrid methods

The core ideas of this thesis are collaborative filtering, content-based recommendation, and the hybrid approach. Fig. 1 depicts the recommender system's overall concept.

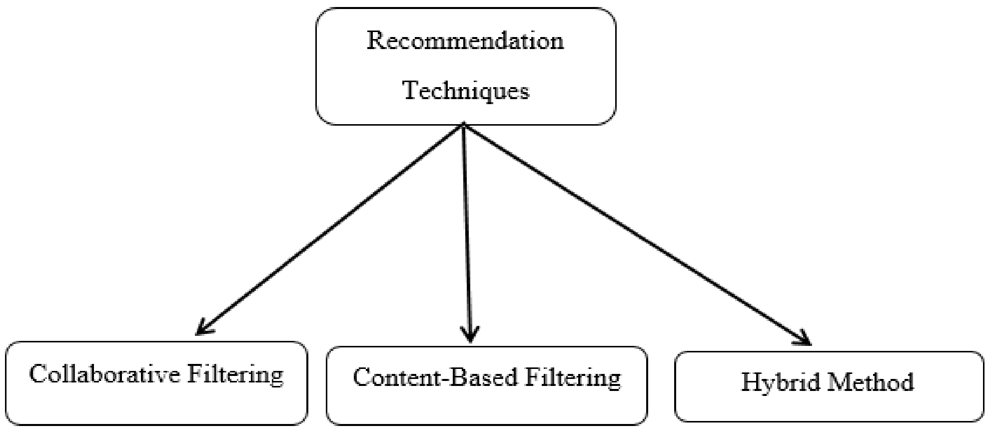

Figure 4: Recommendation system Techniques

2.2.2 Collaborative Filtering (CF)

At Xerox PARC in 1992, David Goldberg and a colleague developed the first-ever concept, which they eventually dubbed "collaborative filtering." Their main objective was to make product recommendations to individuals based on how they were similar to other users.

Collaborative filtering, one of the most extensively used or applied techniques in recommender systems, involves recommending to users who are now active things that other users with similar viewpoints have previously enjoyed (Schafer, J. B et al., 2007).

Based on the users' rating histories, the commonality, also known as similarity in taste, between two users is determined. The core of this filtering type is the user feedback loop. Person ratings, thumbs-ups, and downvotes, or simply how much a user interacts with a piece of material are all examples of feedback. There are two definitions of collaborative filtering: a specific one and a broad one. Collaborative filtering, in a more constrained sense, is a method for combining preferences or data from many users to automatically predict (filter) a user's interests (collaborating). The user feedback loops are the basis of this filtering type. Person ratings, thumbs-ups, and downvotes, or simply how much a user interacts with a piece of material are all examples of feedback.

There are two definitions of collaborative filtering: a specific one and a broad one. Collaborative filtering investigates a method for making product recommendations based on matching users with like-minded interests. The foundation of CF is the idea that goods are more likely to be liked by similar people (Anil, R. et al., 2010). Many algorithms in the collaborative filtering family can be used to find comparable people or things, and there are many methods for figuring out ratings based on the ratings of similar users.

2.2.2.1 Similarity Measures in CF Algorithm

In collaborative filtering algorithms, finding comparable users and objects is the most significant phase. Finding similar people and items makes it simple to analyze their similarities before selecting a set of individuals and items that most closely resembles the target user (Fard, K. B., 2013).

The popular similarity metrics that are utilized in collaborative filtering and that are also used for analysis in this work are listed below.

- Euclidean Distance:

  A line segment's length between two locations in Euclidean space equals the Euclidean distance between them. Euclidean distance serves as the foundation for many comparisons of similarity and dissimilarity. The following definition describes the distance between vectors X and Y:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

  Where n is the number of frequently rated items and qi and pi are the rating scores of the same item provided by two separate users.

- Pearson Correlation Coefficient:

The most popular technique for analyzing numerical variables is the Pearson correlation approach, which assigns a value between 0 and 1, with 1 denoting total positive correlation and 0 denoting total negative correlation.

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

- Cosine Similarity:

The cosine similarity measure calculates how similar two matrices in an inner product space are to one another. It establishes whether two variables are roughly pointing in the same direction by calculating the cosine of the angle between them. In text analysis, it is frequently used to measure document similarity.

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

- Jaccard Coefficient:

The intersection of the items divided by their union is how the Jaccard coefficient also known as the Tanimoto coefficient measures similarity. The Jaccard coefficient for text documents contrasts the sum weight of terms that are present in both papers but are not shared terms with the sum weight of terms that are present in only one of the two documents. The formal definition is:

$$EJ(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x} \cdot \mathbf{y}}$$

Four significant similarity measures are observed. The dataset of movie ratings, which varies in how each user rates various novels, serves as the system's input. Depending on several variables, including the information's qualities and context, any similarity measurements might yield similar or dissimilar answers.

2.2.3 Content-Based Recommendation (CBR)

The content-based technique is a domain-dependent algorithm that places more emphasis on the evaluation of item attributes to produce predictions. Content-based filtering is the most effective method for recommending materials like web pages, magazines, and news. With the use of features that are taken from the content of the items the user has previously evaluated, recommendations are created using the user profiles in the content-based filtering technique (Burke, R., 2002). An item's description and a user profile serve as the foundation for content-based filtering techniques (Das, D., Sahoo, L., & Datta, S., 2017).

Additionally, it makes recommendations based on similar items that a particular user has liked in the item rate list. The core purpose of a content-based system is to match a user's demographic data, such as age, race, locality, and the rated items on the website that are stored in his account, with comparable products that have a shared specification (Mohamed, M. et al., 2019).

The stages of the CB recommendation process:

- Content analyzer

There must be some sort of pre-processing step taken to extract structured relevant information from sources where there isn't one, like text. The component's primary responsibility is to present the content of things like documents, websites, media, product descriptions, etc. originating from sources of information in a manner suitable for the processing stages after it.

- Profile learner

This methodology builds a user profile by gathering data on user preferences and makes an effort to generalize the data. The generalization strategy is often implemented using machine learning techniques since they can grasp a decent example of consumer interests by beginning with previous preferences for particular products.

- Filtering component

The user gets shown similar but new products that suit the item list once this model examines the user's privacy profile for them. A suggested selection of articles that might be of interest. The matching is carried out by evaluating the cosine similarity between both the pattern and item vectors.

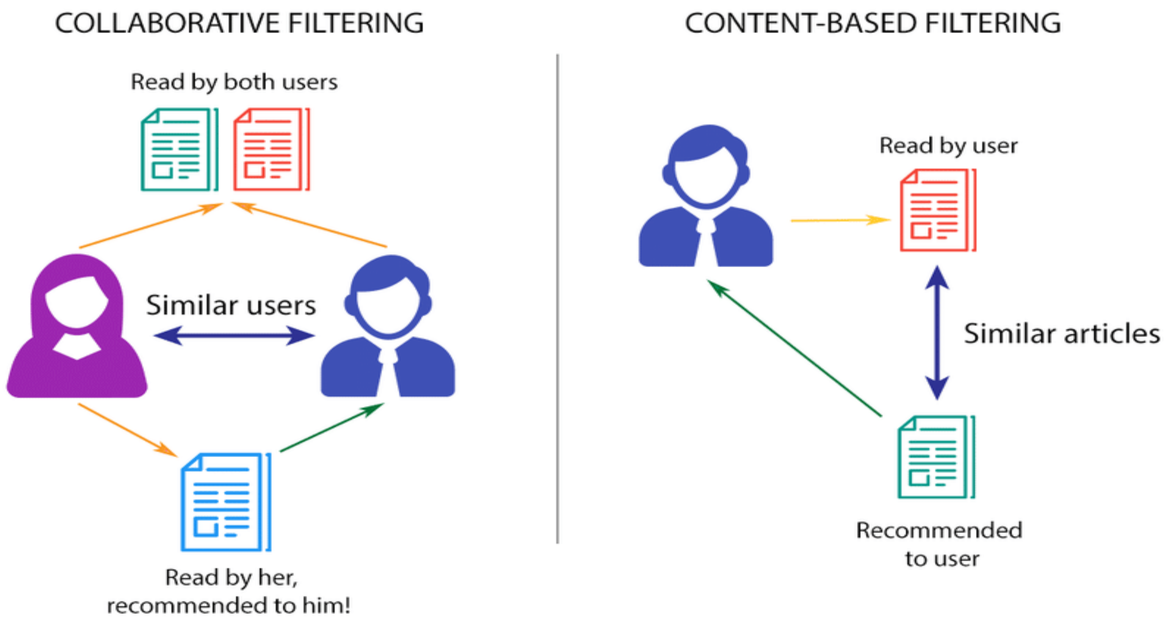

Figure 5: Collaborative Filtering Vs Content-Based Filtering

Figure 2 describes content-based filtering and collaborative filtering in detail.

2.2.4. Hybrid-Based Recommendation (HBR)

In a hybrid approach, we combine the two advised methods content-based and collaborative filtering to maximize benefits, improve outcomes, and lessen issues and difficulties associated with these applications (Khusro, S., Ali, Z., & Ullah, I., 2016). Multi-methods are used in the hybrid technique (Patel, Y. G., & Patel, V. P., 2015).

To maximize performance while minimizing the downsides of each recommendation strategy individually, hybrid recommender systems merge multiple or even more recommendation algorithms. Collaborative filtering is typically combined with another method to get around the ramp-up problem.

As suggested by Brusilovski, P et. al., (2007), collaborative filtering can be integrated with other recommendation algorithms in the following ways:

- Switching: The algorithm presents several recommendations to the user, choosing the best one depending on their preferences.
- Combining features: this is the process of creating recommendation system features by combining knowledge from several sources.
- Mixed: At the same time, recommendations are offered by a variety of recommenders.
- Cascade: The recommendations made by one recommender are improved by another.
- Meta-level: One of the input techniques used to build an algorithmic step model after the recommender system is this one. Combining these several strategies could result in excellent performance and lessen problems that can occur when using just collaborative or content-based filtering.
- Feature Augmentation: A technique's output is used as an input feature by another.

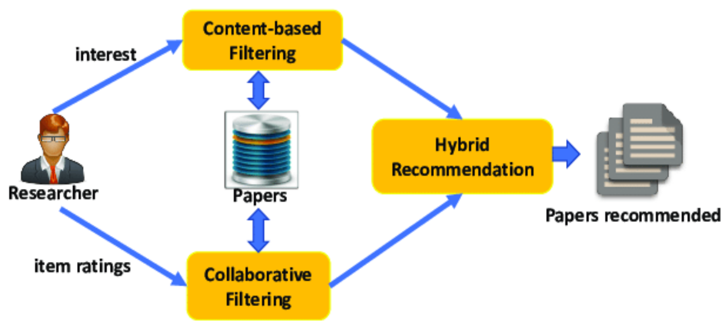

Figure 6: Example of a Hybrid System

2.3 Comparison of Recommender Systems Techniques

The advantages and disadvantages of recommender systems will be covered in this section in Table 1.

Table 1: Lists of the benefits and drawbacks of recommender systems

No	Techniques	Advantage	Disadvantage
1	Collaborative recommendation Filtering	• The system finds similar items among users. • The system can recommend to the user products that are outside of their tastes but that they might like.	• The system's quality is determined by the highest-rated item list. • The process of recommending products to new users has a problem (cold start problem).
2	Content-Based recommendation Filtering	• Using the similarity in the specs of the objects, the system can suggest new items to users. • Based on user data, the system did not make any recommendations for products.	• To create a recommendation list, we need to analyze and detect all item features. • An assessment of the product's quality was omitted as the algorithm did not rely on the user's rating of this item.
3	Hybrid Approaches	• It combines the benefits of collaborative filtering and content-based filtering. • Description and the user's evaluation.	• The Content Description and the user's assessment serve as the criteria. • Overspecialization should be resolved. • Increase the rate of customer satisfaction.

		• Overspecialization should be resolved. • Increase the rate of customer satisfaction.	

2.3.1 Various data mining techniques used with a recommender system.

Today, a variety of data mining techniques used to glean valuable business information from huge datasets like big data are essential due to the exponential expansion in data size.

The following strategies are used in data mining and recommender systems: We will talk about a few techniques used in recommender systems and data mining:

1) Regression analysis: is a technique for building models that examines the relationships between various independent variables and a dependent variable.
2) Classification: the process of grouping data using clustering techniques. To create matching from distinct clusters, we employ classification. A decision tree can accomplish this. We specify the initial node and the following objects in the item or user tree.
3) Association Analysis: To construct the inference rule, association analysis aims to establish the relationship between the data sets. We must discover which products were previously purchased in tandem to create a correlation.
4) Cluster Analysis: Using a sample of data, we perform a cluster analysis in which we create groups that are conceptually related to one another. The goal of this technique is to identify products with comparable specifications or customers who have purchased comparable products in the huge data structure. The user is then shown a new list of highly suggested items after these results have been matched.
5) Outlier detection: We choose values that differ from any sample of data, which is referred to as an outlier.

2.4 Deep Learning for Recommendation

Data analysts are increasingly turning away from more conventional machine learning approaches and toward highly expressive deep learning models to enhance the reliability of their recommendations as the amount of data available to fuel recommender systems increases quickly. The two steps of deep learning for recommendations are training and inference. By demonstrating examples of previous interactions between users and items, the model is trained to predict the likelihood of user-item interactions during the training phase (compute a preference score).

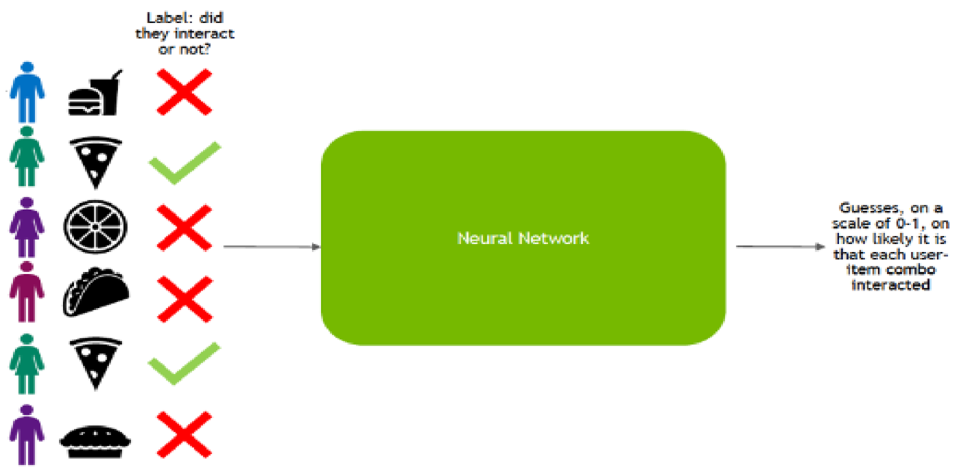

Figure 7: Deep Learning for Recommendation Training

The model is used to infer the likelihood of new interactions once it has mastered making predictions with a high enough level of accuracy.

2.4.1 Deep Neural Network Models for Recommendation

Factorization and encapsulation are two modern methods that are based on DL recommender models to represent the interactions among variables and handle categorical variables. Embedding is a learned vector of numbers storing entity attributes that allows related entities (such as persons or things) to have various distances in the vector space.

2.5 NVIDIA Merlin

To speed up recommender systems on NVIDIA GPUs, NVIDIA Merlin is an open-source library. The library makes it possible for researchers, machine learning engineers, and data scientists to quickly and efficiently create high-quality recommenders. The features, training, and inference challenges are addressed by the tools in Merlin. Every step of the Merlin pipeline is designed to handle data volumes of hundreds of gigabytes, and all of this data is available via simple APIs.

2.5.1 Components of NVIDIA Merlin

1) Merlin Nvtabular:

A feature engineering and pre-processing package for tabular data is called NVTabular. The library can quickly and easily handle Terabyte-sized datasets that are needed to train DL-based recommender systems. A high-level API provided by the library allows for the definition of intricate data transformation operations.

2) Merlin HugeCTR:

With the help of the GPU-accelerated training framework HugeCTR, massive deep-learning recommendation models may be scaled up. Training is split across various GPUs and nodes. HugeCTR includes solutions for scaling huge embedding tables beyond the amount of memory that is available as well as improved data loaders with GPU acceleration.

3) Merlin models

The Merlin Models library provides a standardized model for recommender systems that range from conventional ML models to extremely advanced DL models, with an emphasis on high-quality implementations.

4) Merlin systems

To create end-to-end recommendation pipelines that can be fed by Triton Inference Server, Merlin Systems offers tools for merging recommendation models with other production recommender system components including feature stores, nearest neighbor search, and exploration techniques.

5) Merlin core

The Merlin ecosystem makes use of the features provided by Merlin Core. Merlin Core allows you to:

1) Processing huge datasets across numerous GPUs and nodes, using a standard dataset abstraction.
2) Utilize a single API to build graphs of data transformation operators to streamline your code.

2.6 Related Work

In light of the current technology in use and the improvements made to the recommender system, we are probably not the first to see the potential. Concerning the work that already exists, we will discuss the work that has been done and the contributions that have been made.

Table 1 examines the advantages of each paper as well as the authors of other papers, their methods for resolving issues with the recommender system, and their names.

Table 2: Comparing the benefits and solutions of several publications

No	**Author Name**	**Solutions**	**Advantages**
1	Lee M.R et al. (2016)	They suggest developing a hybrid recommender system that uses data from Facebook Fan Pages and machine learning.	• Address the accuracy and cold start issues. • Boost client satisfaction.
2	Kim & Park (2013)	They suggested using method constructs that were modified from movie suggestions in online community systems.	• Improve efficiency and accuracy.
3	Wang, Q., Yuan, X., & Sun, M. (2010)	They employ a genetic algorithm in conjunction with the item's	• Better system scalability is the result.

		demographic data to find a group of nearby users who share their interests.	
4	Colombo et al. (2015)	This suggested system utilizes Semantic Web technology and falls under the free time domain, which can only be used for movie show times.	• The outcomes demonstrate the recommender system's efficacy.
5	Christakou et al. (2015)	They created a system for making movie recommendations that integrate collaborative and content-based data. On the Movie Lens DS, the offered system is tested.	• The suggested method produces results with excellent precision.
6	Ravi, L., & Vairavasundaram, S. (2016)	This study provides insights into social media data-based recommender systems by considering system functionality, interface modifications, a filtering approach, and artificial intelligence techniques, as well as how various recommendation algorithms are applied.	• It makes it easier for future research to go in the right direction and aids interested developers in creating trip recommendation systems.
7	Geetha, G., Safa, M., Fancy, C., & Saranya, D. (2018)	Employ collaborative filtering recommender systems to discover the preferences of new users.	Improve recommendation approach
8	Halder, S., Sarkar, A. J., & Lee, Y. K. (2012, November)	They put forth the idea of "movie swarm mining." This algorithm frequently mined items and employed two pruning rules.	• Solve the cold start problem

9	Madadipouya, K. (2015).	The work presented here introduces a novel collaborative filtering-based location-based movie recommender system.	• Improved accuracy and recommendation quality
10	Panigrahi, S., Lenka, R. K., & Stitipragyan, A. (2016).	They put out a brand-new hybrid method that makes use of Alternating Least squares (ALS) and K-means for dimension reduction.	• Fix the sparsity and scalability issue

To achieve accuracy, high consumer satisfaction when promoting the product, and to fix the cold start issue, Lee M.R et al. (2016) proposed creating a hybrid-filtering recommender system using data from Facebook Fan Pages and machine learning. This method for extracting from Yahoo used content-based filtering, and Facebook or Twitter pages were also considered. Additionally, contrast this algorithm's output results with those of other recommender systems like Netflix, YouTube, and Amazon.

A theoretical model and a system implementation via Kim & Park's (2013) rigorous work on an interactive recommender system were used to discuss the system. The suggested method generates personalized movie recommendations in online community systems. The method that is being described develops the tactic that is believed to be able to regulate the dynamics of socially mediated information transmitted in community networks.

Wang, Q., Yuan, X., & Sun, M. (2010) used genetic algorithms to increase the recommender system's accuracy and a demographic filtering technique to increase the system's scalability. They develop a hybrid user model by integrating item aggregate characteristics with screening for a group of nearby people who share your interests and utilizing an evolutionary algorithm to determine which characteristics should be given more importance in the user model.

RecomMetz, a recommender system proposed by Colombo et al. (2015), incorporates a mobile recommender system that relies on context-aware information. The suggested solution utilizes

Semantic Web technology and is in the free time domain, which can only be used for movie show times. The outcomes demonstrate the recommender system's efficacy.

A proposed clustering method that relies on semi-supervised learning was provided by Christakos et al. (2015). In this research, a method for creating a recommendation system for movies that integrates collaborative and content-based information is suggested. The proposed approach produces results with a high degree of precision when the offered system is tested on the Movie Lens DS.

In their 2015 study, Logesh & Subramaniyaswamy, discuss the use of multiple recommendation algorithms, system capabilities, various user interfaces, filtering strategies, and artificial intelligence approaches to present their opinions on social network data-based recommender systems. The study aids interested developers in the building of a trip recommendation system and supports future research direction after exploring the depths of the goal, methodology, and data sources of the existing models. In addition, a location recommendation system based on a socially pertinent trust walker (SPTW) was developed in this article, and comparisons were made with the outcomes of the current baseline random walk models.

In this study, Geetha et al. (2018) use collaborative filtering recommender systems to discover the preferences of new users. Among these techniques are information theory to select the items the recommender system will find most useful, aggregate strategies to select the products the customer is most likely to have a viewpoint about, and specialized techniques that predict which products a user will have an opinion about. This knowledge will help researchers develop the most advanced recommendation approach. Using MOVREC as an example, the CF-based system makes use of user-provided data, analyzes it, and then employs the k-means algorithm to recommend the movie that is most suitable for the user at that specific time.

Williams put forth the idea of swarm mining for movies. This was used to manage and extend watching hours for the newest and most well-known movies, as well as to address the cold start challenge of making recommendations to new customers. This algorithm frequently mined items and employed two pruning rules.

Madadipouya, K. (2015) improve the precision and quality of recommendations, a new location-based movie RS built on CF is presented in this work. The locations of the users have been used and taken into account throughout the processing of the suggestions and user-to-user selections in this method.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

As previously said, our goal is to develop an optimized movie recommendation model that is capable of reaching a greater recommender accuracy or precision than the conventional or widely used recommender system. Several well-known datasets, ranging from the smallest to the largest, were employed to accomplish the goals and aims of the study, including the 5K Internet Movies Dataset, 100k, 1M, and 25M movies Len, from which user ratings and content attributes were both extracted for the study's purposes. Despite only concentrating on movie data, the purpose of this study was to create a broad paradigm that could be applied in other domains.

In general, two significant areas where collaborative recommender systems shine are predicting how much a user would appreciate a specific item and providing a user with a list of goods. But in our study, we emphasize the recommendation.

Before making any choices regarding the architecture of our recommender models, we first evaluated all system restrictions. The section that follows gives detail on the suggested method and the system's features.

3.2 The Major Concern in Collaborative Filtering

In terms of conventional collaborative filtering techniques, the following are the main issues:

Scalability: Scalability refers to extensibility, or how well a system performs as the amount of data grows. While collaborative recommender systems perform admirably when dealing with tiny datasets, managing vastly expanding real-world datasets is a difficult issue, even though there are algorithms for handling large and dynamic data sets.

Data Sparsity: The term "sparse," which means "scattered," is the root of the word "sparsity." Sparsity in recommender systems refers to inconsistent, insufficient, or widely varied user ratings. One of the biggest problems with recommender systems is this. The fact that the vast majority of customers do not provide assessments and those who do are typically scanty is one of the primary causes of sparsity.

3.2.1 Recommender systems Generation

Figure 2 explains and illustrates the many generations of recommender systems, starting with the first generation and moving on to the second generation and third generations. In this study, we will deconstruct and concentrate on CF in the first generation before employing deep learning to advance and take into account CF in the third generation.

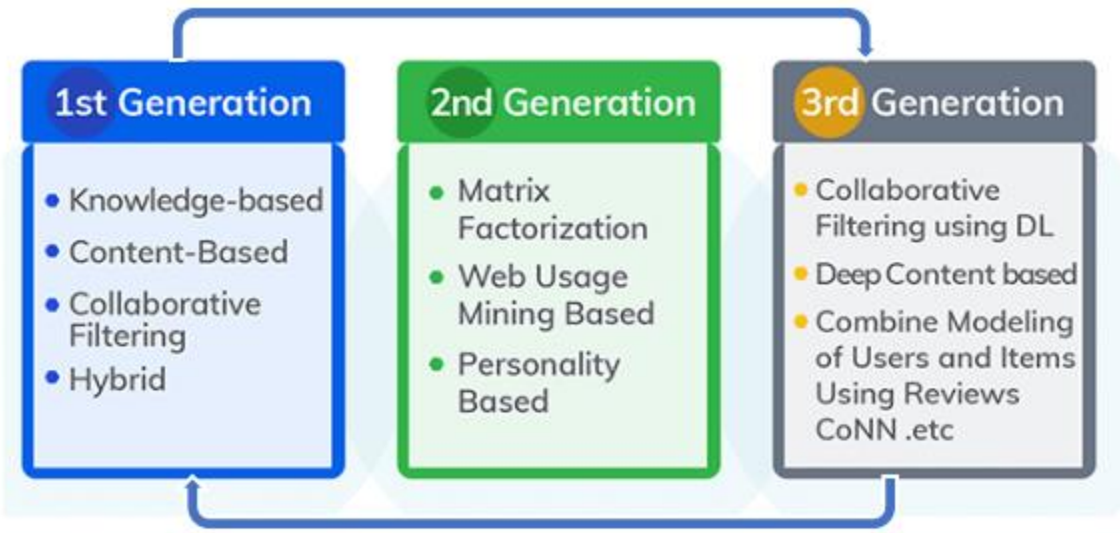

Figure 8: Recommender System Generations

EXPERIMENT 1: Conventional Collaborative Filtering Techniques

This experiment includes employing conventional collaborative filtering techniques on a movie dataset. In this research, as it was clearly stated in 1.2, a collection of recommender systems will be needed to assess how well they perform. Determining, specifically, how well they adhere to recommendations. Their MAE, MSE, and RMSE. These data will likely make up our dataset, which will align with the goals and purposes of the aforementioned recommender system. The next step is to investigate why the findings are recorded in this specific way.

The MAE, MSE, and RMSE were used to detect the most effective performance recommendation techniques for the recommender system.

3.3 The Proposed Methodology

The following infrastructure design decisions were taken to overcome the two main issues with conventional collaborative filtering in section 3.2.

Every online action is influenced by recommender systems, from the choice of web pages to more prominent examples like online buying. They are crucial in encouraging user interaction on online platforms and helping users choose the few most significant goods or services from a wide range of alternatives. A 1% increase in recommendation quality has the potential to generate billions of dollars in revenue.

The performance of DL recommender architectures, which utilize massive amounts of training data, has started to outperform that of more conventional methods such CF, CB, neighbourhood, latent factor method, and so on as a result of the exponential growth in the size of industry datasets.

Furthermore, given the combination of more complicated models and speedy data generation, the requirement for computational resources needed for training has skyrocketed. To accommodate the computational complexity for comprehensive DL recommender training and inference, NVIDIA released Merlins.

Merlin is an end-to-end framework for DL recommender models on GPUs that strives to provide quick feature engineering and high training throughput for quick experimentation and production retraining. Merlin also offers production inference with low latency and high throughput. The typical architecture is shown in figure 1.

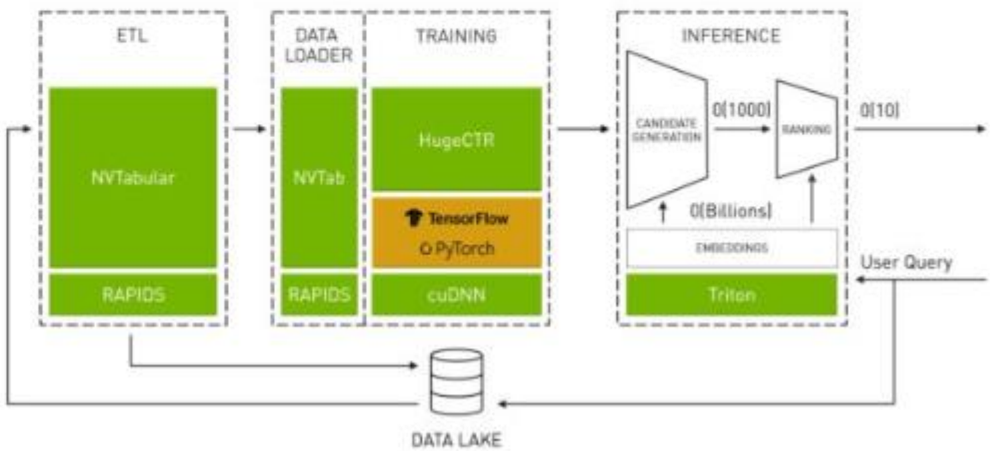

Figure 9: NVIDIA Merlin Recommender System Framework

3.4 CF Using Nvidia Merlin

To accelerate recommender system development across all stages, from experimental to production, NVIDIA Merlin was developed as an application framework and ecosystem. Figure 1 depicts the architectural layout of Merlin, which has three key parts:

- Merlin ETL

Merlin ETL contains a set of tools for Graphics card pre-processing and efficient feature engineering in recommender. Considering terabyte-scale table data, NVTabular provides excellent on-GPU data pre-processing and transformation possibilities.

- Merlin training

A collection of models and instructional materials for recommender for DL HugeCTR is written in C for the recommender system training mechanism that works incredibly well. It contains features for multi-GPU and multi-node training in addition to model- and data-parallel scaling.

- Merlin-inference

The Merlin-inference container allows users to deploy NVTabular processes, HugeCTR models, or Tensor Flow models to the Triton Inference server for production. Triton Inference Server will now enable Graphics card inference owing to NVTabular and HugeCTR.

3.4.1 Nvtabular

To overcome frequent data pipeline problems for recommender systems, the NVTabular ETL Merlin component was developed. Professional recommenders frequently use training sets that are multi-terabytes or even gigabytes in size and contain billions of user engagements. A common belief in the data science community is that since pre-processing or performing feature engineering for datasets of this size requires a significant amount of time, data analysts invest more time on ETL and data preparation than training the model and fine-tuning.

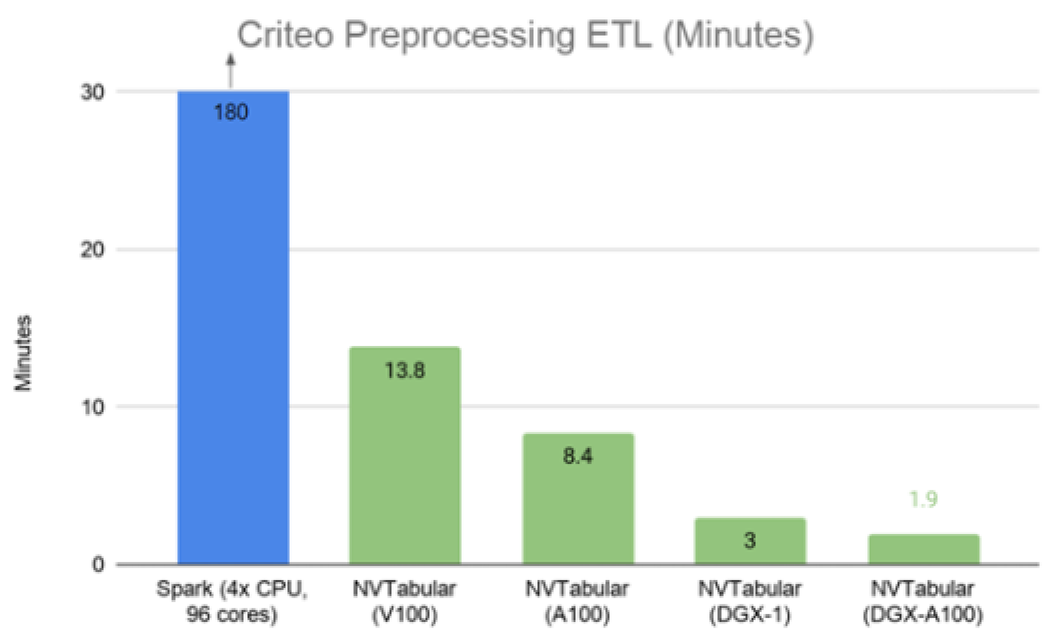

Figure 10: Criteo dataset processing ETL

Figure 3 demonstrates that the DGX A100's NVTabular multi-GPU gives a 95x speedup, translating to a 5.3x speedup.

3.4.2 Data Loading For RS

Originally, DL frameworks used photos as data loaders. Massive data sets' components were collected and combined into a batch before being sent to the GPU for training. Customers regularly read tabular data in recommender systems, however, there is very little data per sample.

We employed an alterable data loader, a concept from the PyTorch team, to challenge this paradigm. By treating the data loader as an iterator instead of constructing batches one item at a time, you may transfer entire batches to the training framework much more quickly. As a result, there are more restrictions on how the data can be mixed up. Data in a batch is twice-randomly shuffled using NVTabular data loaders for each epoch.

3.4.2.1 NVTabular data loader

The NVTabular data loader for Tensor Flow's purpose is to effectively deliver tabular data to Tensor Flow's DL model training. NVTabular offers a data loader that is similarly tailored for PyTorch.

3.5 Performance Evaluation Criteria

Any recommender system needs assessment since, without it, we can't tell whether the results are reliable or not.

Comparing two systems would only be conceivable after doing the evaluation. Evaluation can be used to improve the system. Technique, algorithm, and procedure quality are evaluated to provide an accurate recommendation

A statistical efficiency metric was used to assess a recommendation system's accuracy. The MAE statistic is widely used by the CF-based recommendation system to assess how far recommendations deviate from actual customer ratings. This metric is used to gauge how accurately the system forecasts. The capacity to predict what a user will like is prediction.

Important parameters to take into account are the prediction's accuracy and coverage. MAE, MAE, and RMSE are the categories of accuracy.

EXPERIMENT 2: Collaborative filtering using deep learning (Nvidia-Merlin Approach)

NVIDIA Merlin facilitates the development of high-performing recommenders at scale and offers a convenient platform for doing so. Merlin's libraries, methods, and tools address typical issues with pre-processing, feature engineering, training, inference, and deploying to production to accelerate the development of recommenders. Hundreds of terabytes of data can be retrieved, filtered, scored, and ranked using the components and capabilities of Merlin using simple-to-use APIs. With Merlin, you can accelerate production deployment, boost click-through rates, and make more accurate forecasts.

A few NVTabular TF additions, such as customized TensorFlow layers enabling multi-hot and the NVTabular TensorFlow data loader, were imported along with TensorFlow.

3.6 Summary

We propose a method that applies deep learning techniques to deliver a quick, fast, and reliable approach to recommender systems.

The implementation and trials utilizing the suggested method are thoroughly described in the next chapter, together with the findings on the efficiency or validity of the recommenders and a synopsis of the work's significant accomplishments.

CHAPTER FOUR

IMPLEMENTATION, RESULTS, AND DISCUSSION

4.1 Introduction

This chapter discusses the results of the two experiments that were carried out to create the best and most efficient advanced deep learning collaborative filtering systems, as described in chapter 1.1.

Experiment A: This experiment specifically evaluated the efficacy of serial recommender systems using a conventional collaborative filtering approach as a case study. Specifically, evaluating their recommendations' correctness and accuracy. Consequently, their MAE, MSE, and RMSE were also documented. This information would likely form our dataset, which you will correlate with the goal and function of the recommender system in question. Then there is a need to investigate why the results as recorded are the way they are. Refer to page [25] for more details.

Experiment B: the second experiment used the Nvidia merlin to build a collaborative deep-learning filtering system. Using an advanced deep learning collaborative filtering approach, the objective is to deliver high-performance recommendations at a large-scale data collection. The test was done to see if DLCFS outperformed traditional CF, provided a practical model that improves recommendation accuracy, and gave a long-term solution to the effectiveness and scaling issues present in processing enormous amounts of data for recommendations to customers. Page [26] has further information.

4.2 Performance Measure and Evaluation

The following standard measures were used to assess the recommender model:

1. MSE

MSE evaluates the level of accuracy in analytical models. Between observed and projected values, the average squared difference is calculated. When there is no error in a model, the MSE is zero. As model error increases, so does its value. The mean squared deviation is another term for the mean squared error (MSD). The purpose is to reduce MSE value as much as possible. MSE = 0 for a perfect model

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

Figure 11: MSE Formula

2. MAE

The real value is determined by dividing the sum of the absolute errors by the sample size, or MAE. On the same scale as the data, the MAE is measured.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error

y_i = prediction

x_i = true value

n = total number of data points

Figure 12: MAE Formula

3. RMSD

RMSE Calculate the difference between projection and fact for each data point, as well as the norm of residuals, mean of residuals, and the square root of that mean.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \hat{x}_i\right)^2}{N}}$$

RMSD = root-mean-square deviation

i = variable i

N = number of non-missing data points

x_i = actual observations time series

$\hat{x}_i$ = estimated time series

Figure 13: RMSE Formula

4. Logarithmic Loss or Log Loss:

In this, the model's recommendations are compared with the actual labels to determine how uncertain they are. Confident but incorrect predictions are harshly punished. The likelihood increases as log loss decrease. Consequently, diminishing the value is the objective.

5. Precision and Recall:

The proportion of appropriate results returned is known as precision. Recall, also known as sensitivity, is the percentage of significant events that are detected..

4.3 Experiment (A) – Implementation & Results

The initial experiment which is experiment A uses a dataset that contains movie ratings to test and evaluate how well typical collaborative filtering approaches perform and adhere to recommendations using their RMSE, MAE, and MSE. The following experiments and analyses were carried out to obtain this.

The Pareto Principle, popularly known as the 80/20 rule, was also taken into consideration. 20% of the data set was used to test the model, with the remaining 80% being used for training. In this Chapter, the test results have been used to compare models.

4.3.1 Movie Lens Rating Information

Even though the time factor for CF has been successfully applied in past research, here we primarily focus and are particularly interested in the first three fields: User ID, item ID, and rating. Examples of rating data were provided in text files. The fields are illustrated in Table 3.

Table 3: Extract of Rating Information

	userId	movieId	rating	timestamp
0	1	296	5.0	1147880044
1	1	306	3.5	1147868817
2	1	307	5.0	1147868828
3	1	665	5.0	1147878820
4	1	899	3.5	1147868510

The following results show the ratings and distribution of three different implementations of the conventional collaborative filtering techniques using singular value decomposition (SVD).

Figure 3.0: The figure shows the rating distribution for the one million (1M) movies dataset, Figure 4.0 shows the rating distribution for the 100k movies data set while Figure 5.0 also rating distribution for the 5k movies data sets.

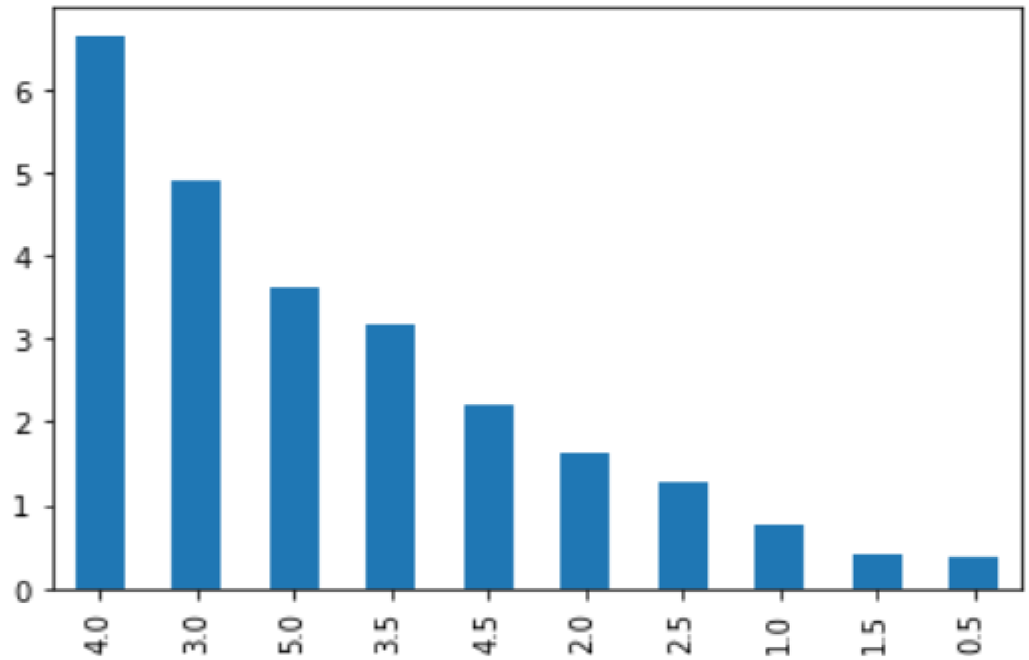

Figure 14: Rating Distribution for 1M movies Datasets

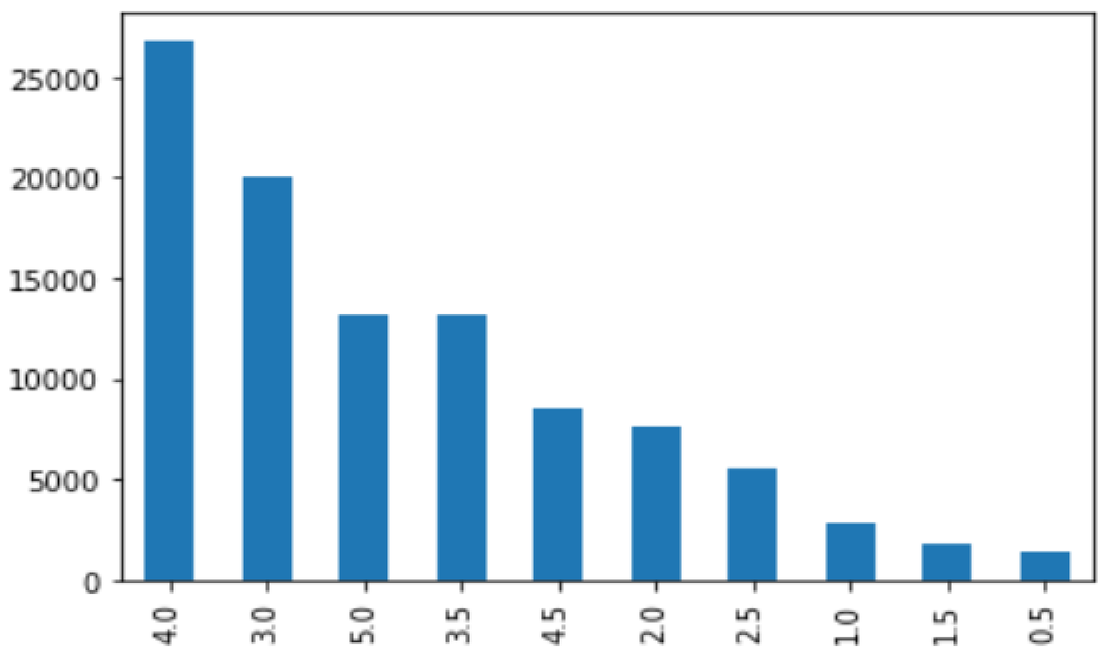

Figure 15: Rating Distribution for 100k movies Datasets

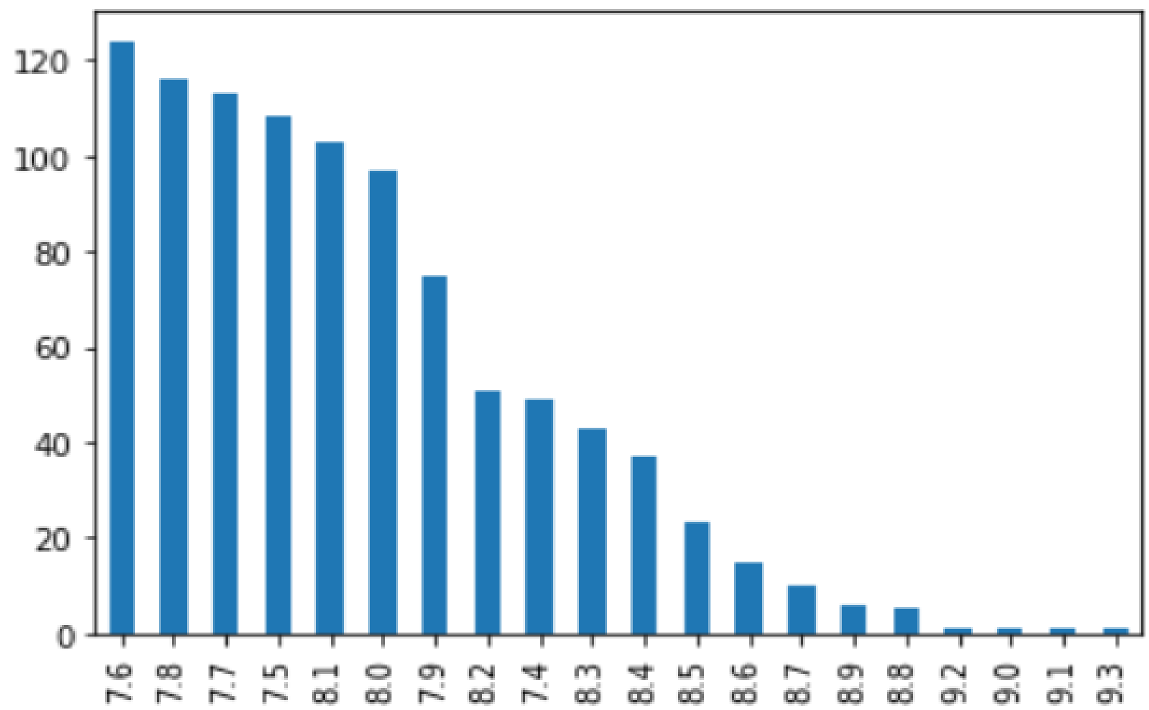

Figure 16: Rating Distribution for 5K movies Datasets

4.3.2 Performance Measure

The reliability of the final recommendations is a common metric used to determine if a recommendation system is effective or not. The RMSE, MSE, and MAE were used as recommendation-based metrics. A recommender system's performance is typically measured using the metrics RMSE, MSE, and MAE. This statistic demonstrates how effectively a Recommender performs. The performance of the recommendations improves with decreasing RMSE, MAE, and MSE. It offers an erroneous value that illustrates how far off from the real data our model is. It assesses how closely the projections supplied correlate to the quantities that were observed.

Figure 6 shows the final result of evaluating RMSE, MAE, and MSE on 1M Movies Datasets. Taking the mean result of the output, MAE outperformed another matric because it has the lowest mean value of 0.5843

Likewise, Figure 7 shows evaluating results of algorithm SVD on the 100k movies dataset. Also, MAE outperformed other matric having the lowest output of 0.6593

However, figure 8 displays the result of evaluating RMSE, MAE, and MSE of algorithm SVD on 5k movies data set. MAE still outperformed other matric having the lowest mean value of 2.8898

Ultimately, it was discovered that the movie data set with the most customers and reviews performed better than the others with fewer datasets obtainable.

```
Evaluating RMSE, MAE, MSE of algorithm SVD on 5 split(s).
```

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.7750	0.7752	0.7748	0.7753	0.7750	0.7751	0.0002
MAE (testset)	0.5841	0.5843	0.5842	0.5845	0.5845	0.5843	0.0002
MSE (testset)	0.6006	0.6009	0.6003	0.6012	0.6006	0.6007	0.0003
Fit time	614.63	386.24	300.81	290.15	338.71	386.11	119.12
Test time	60086.23	2823.37	2445.97	3193.58	3146.98	14339.23	22875.07

Figure 17: Evaluating RMSE, MAE, MSE of Algorithm SVD on 1M Movies Datasets

```
Evaluating RMSE, MAE, MSE of algorithm SVD on 5 split(s).
```

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	0.8605	0.8517	0.8556	0.8659	0.8683	0.8604	0.0062
MAE (testset)	0.6576	0.6553	0.6556	0.6636	0.6646	0.6593	0.0040
MSE (testset)	0.7405	0.7253	0.7321	0.7498	0.7539	0.7403	0.0106
Fit time	4.34	4.37	4.40	4.38	4.40	4.38	0.02
Test time	0.21	0.10	0.10	0.10	0.10	0.12	0.04

Figure 18: Evaluating RMSE, MAE, and MSE of Algorithm SVD on 100k Movies Datasets

Evaluating RMSE, MAE, MSE of algorithm SVD on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	2.9321	2.8893	2.8982	2.9151	2.9114	2.9092	0.0147
MAE (testset)	2.9087	2.8699	2.8806	2.8990	2.8908	2.8898	0.0136
MSE (testset)	8.5972	8.3478	8.3995	8.4979	8.4762	8.4637	0.0857
Fit time	0.01	0.01	0.00	0.00	0.03	0.01	0.01
Test time	0.00	0.00	0.02	0.00	0.00	0.00	0.01

Figure 19: Evaluating RMSE, MAE, and MSE of Algorithm SVD on 5K Movies Datasets

As mentioned Recommender systems are a means to propose or find ideas and products that are related to a user's particular manner of thinking. Here, we suggest movies that are the most popular to watch and that the viewer may object to.

```
1 [318, 109487, 475]
2 [1178, 2324, 306]
3 [858, 1198, 1732]
```

(A) 1M Movies Lens Datasets

```
1 [318, 48516]
2 [2959, 858]
3 [3578, 57669]
```

(B) 100k Movies Lens Datasets

```
1 [318]
2 [1178]
3 [858]
```

(C) 5k Movies Lens Datasets

4.4 Experiment (B) - Implementation & Analysis

The second experiment leverages Nvidia merlin to train the model using a DL technique. As earlier stated NVIDIA Merlin is an open-source library that speeds up recommender systems. We created effective recommenders at a large scale owing to the library. Common feature engineering, training, and inference problems are addressed by the tools in Merlin.

4.4.1 Pre-processing

A prominent dataset for recommender systems and one that is referenced in scholarly articles is MovieLens25M. The collection includes 25M user-submitted ratings for 62,000 movies from

162,000 people. Additionally, MovieLens25M is utilized for analysis, and the NVTabular tool is a feature engineering strategy and data pre-processing framework for data tables that is extensively used for data extraction, data transformation, and data loading. Figure 9 displays the availability and presence of a GPU for smooth program execution.

```
!nvidia-smi
```

```
Sat May 28 15:00:30 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8     8W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Figure 20 NVIDIA System Management Interface (Nvidia-smi)

The pipeline for feature engineering and preprocessing is depicted in its initial stage in Figure 10.

Multiple calculations for NVTabular have already been implemented and are referred to as “OPS”. An overloaded “>>” operator can apply an operation to a “ColumnGroup” and return a new “ColumnGroup” as a result. A “ColumnGroup” is a list of text-based column names.

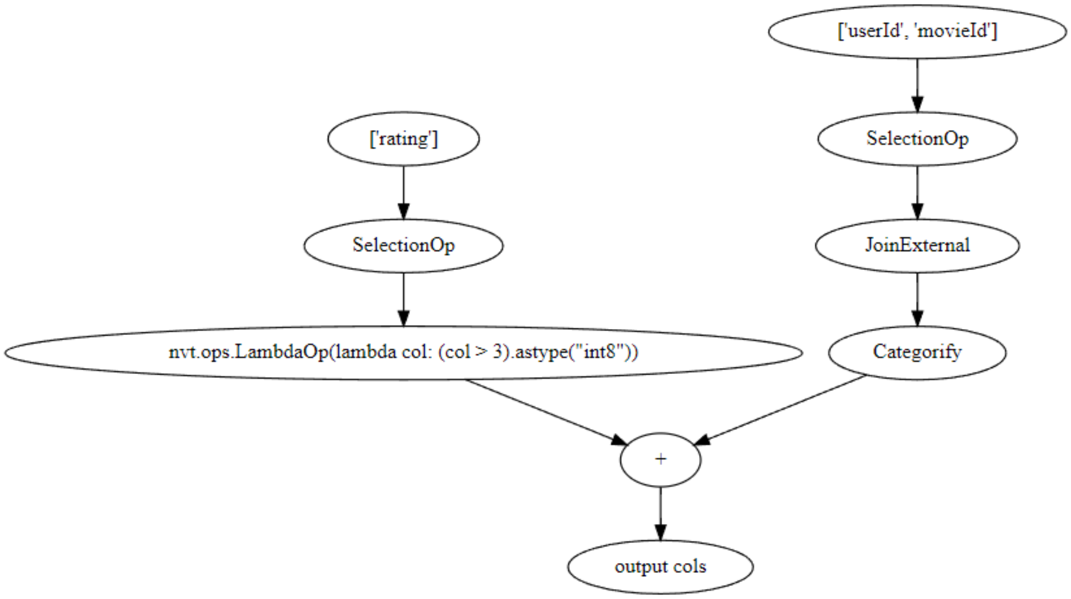

Figure 21 preprocessing pipeline

To analyze the model's performance, as well as its strengths and flaws, various evaluation measures were used. Model evaluation is crucial for determining a model's effectiveness throughout the early stages of research. It also helps with model monitoring. Figure 11 shows how the model performs using different categories.

```
[ ]  metrics = model.evaluate(valid_processed, batch_size=4096, return_dict=True)
     metrics

     1221/1221 [==============================] - 46s 36ms/step - recall_at_10: 0.0483 - mrr_at_10: 0.0180 -
     {'loss': 7.025489807128906,
      'map_at_10': 0.017951054498553276,
      'mrr_at_10': 0.017951054498553276,
      'ndcg_10': 0.02496352046728134,
      'precision_at_10': 0.004833637736737728,
      'recall_at_10': 0.048336416482925415,
      'regularization_loss': 0.0,
      'total_loss': 7.025489807128906}
```

Figure 22: Model Evaluation

4.5 Comparison between Traditional Collaborative filtering and Deep Learning Collaborative Filtering Systems

Traditional recommender systems include collaborative filtering (CF) systems, which base recommendations on previous interactions and user/item characteristics. The primary sources of collaborative filtering suggestions are the user's item and profile information, and CF searches for comparable audience preferences. These technologies do have some drawbacks, though. For instance, the "cold start problem" refers to a difficulty with irrelevant recommendations for a new user who has just started using the system. Data sparsity can also be a challenge. Consider about the thousands of products available on Amazon and the scant real interactions that a typical consumer has with each one

While deep learning-based recommender systems can manage non-linear data processing, its perform better than traditional ones. Non-linear transformation, representation learning, sequence modelling, and flexibility are the major benefits of DL for recommendations.

CHAPTER FIVE

Summary, Conclusions & Recommendations

5.1 Summary

All through the research, we thoroughly reviewed and worked on existing conventional collaborative filtering methodologies to assess and understand the effectiveness of some of the recommender systems employing at least four movie lens datasets systematically obtained from Kaggle. However, we also offered a novel technique to enhance recommendation efficiency and accuracy using a more modern and powerful algorithm. To accommodate a large volume of data and to take advantage of the various valuable tools and platforms available to generate better recommendation outcomes.

5.2 Conclusion

Through the DL recommendation system technique, the proposed methodology has made a valuable contribution. Although relatively new and intricate, this could be a step toward a more accurate and successful recommendation system process. This is significant since it improves suggestion accuracy and can exploit huge data.

DL recommendation systems are effective and outperform conventional recommender systems. This strategy is effective when there is a large amount of data available. Deep learning is widely acknowledged as the most hopeful machine learning strategy for big data analytics, and both big data and deep learning are fields that are expanding rapidly. As a result, the suggested improved deep learning recommendation technique has the potential to be a very effective approach.

5.3 Recommendations

Extensive studies may look at a successful strategy that combines the two techniques described. This could be a successful integrated approach that improves the validity of suggestions. The system can start by training the model with a deep learning methodology or another way, and then it can be upgraded with more data.

Reference

Anil, R., Owen, S., Dunning, T., & Friedman, E. (2010). Mahout in action

Bellogin, A.: Recommender System Performance Evaluation and Prediction: an Information Retrieval Perspective. Ph.D. thesis, Universidad Autonoma de Madrid, Spain (November 2012).

Brusilovski, P., Kobsa, A., & Nejdl, W. (Eds.). (2007). *The adaptive web: methods and strategies of web personalization* (Vol. 4321). Springer Science & Business Media.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), 331-370.

Christakou, C., Lefakis, L., Vrettos, S., & Stafylopatis, A. (2015, November). A movie Recommender system based on semi-supervised clustering. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)* (Vol. 2, pp. 897-903). IEEE.

Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., & Samper-Zapater, J. J. (2015). RecomMetz: A context-aware knowledge-based mobile Recommender system for movie showtimes. *Expert Systems with Applications*, *42*(3), 1202-1222.

Cremonesi, P., Koren, Y., & Turrin, R. (2010, September). Performance of recommender Algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference On Recommender systems* (pp. 39-46).

Das, D., Sahoo, L., & Datta, S. (2017). A survey on the recommendation system. *International Journal of Computer Applications*, *160*(7).

Fard, K. B., Nilashi, M., Rahmani, M., & Ibrahim, O. (2013). Recommender System Based on Semantic Similarity. *International Journal of Electrical & Computer Engineering (2088-8708)*, *3*(6).

Geetha, G., Safa, M., Fancy, C., & Saranya, D. (2018, April). A hybrid approach using Collaborative filtering and content-based filtering for the recommender system. In *Journal Of Physics: Conference Series* (Vol. 1000, No. 1, p. 012101). IOP Publishing.

Halder, S., Sarkar, A. J., & Lee, Y. K. (2012, November). Movie recommendation system based

On the movie swarm. In *2012 Second international conference on cloud and green Computing* (pp. 804-809). IEEE.

Khusro, S., Ali, Z., & Ullah, I. (2016). Recommender systems: issues, challenges, and research Opportunities. In *Information science and applications (ICISA) 2016* (pp. 1179-1189). Springer, Singapore.

Kim, S., & Park, H. (2013). Effects of various characteristics of social commerce (s-commerce) on consumers' trust and trust performance. *International Journal of Information Management*, *33*(2), 318-332.

Lee, M. R., Chen, T. T., & Cai, Y. S. (2016, August). Amalgamating social media data and movie Recommendations. In *Pacific Rim Knowledge Acquisition Workshop* (pp. 141-152). Springer, Cham.

Madadipouya, K. (2015). A Location-based recommender system framework to improve accuracy in user-based collaborative filtering.

Mohamed, M. H., Khafagy, M. H., & Ibrahim, M. H. (2019). Recommender systems challenges And solutions survey. In *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)* (pp. 149-155). IEEE.

Panigrahi, S., Lenka, R. K., & Stitipragyan, A. (2016). A hybrid distributed collaborative filtering Recommender engine using apache spark. *Procedia Computer Science*, *83*, 1000-1006.

Patel, Y. G., & Patel, V. P. (2015). A survey on various techniques of recommendation systems in Web mining. *International Journal of Engineering Development and Research*, *3*(4), 696-700.

Priyanga, P., & Kamal, A. N. B. (2017). Methods of mining the data from big data and social Networks based on the recommender System. *International Journal of Advanced Networking & Applications (IJANA)*, *8*(5), 55-60.

Ravi, L., & Vairavasundaram, S. (2016). A collaborative location-based travel recommendation System through enhanced rating prediction for the group of users. *Computational Intelligence and neuroscience*, *2016*.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering

Recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.

Terveen, L., & Hill, W. (2001). Beyond recommender systems: Helping people help each other. *HCI in the New Millennium*, *1*(2001), 487-509.

Wang, Q., Yuan, X., & Sun, M. (2010, August). Collaborative filtering recommendation algorithm Based on the hybrid user model. In *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery* (Vol. 4, pp. 1985-1990). IEEE.

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: how accuracy Metrics have hurt recommender systems. In CHI'06 extended abstracts on Human factors In computing systems (pp. 1097-1101).

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative Filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53.

Deuk, P., Hyea, K., Young, C., & Jae, K. (2011). A literature review and classification of Recommender systems. Academic journals of International Conference on Social Science.

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.