UNIVERSITY OF THE AEGEAN

DEPARTMENT OF INFORMATION AND COMMUNICATION SYSTEMS ENGINEERING

BACHELOR'S THESIS

Recommendation of safe drug combinations

Author:
Greg MANITARAS

Supervisor:
Prof. Panagiotis SYMEONIDIS

December 15, 2022

Contents

Chapter 1

Introduction

Synopsis: In this chapter, we will provide a small summary of our complete work. We also make an introduction to recommender systems and their importance in our lives. Lastly, we will give an in-depth overview of the algorithms that we used in our approach.

Recommendation engines nowadays are an essential part of many online systems working on different application domains. They are used in a variety of areas, from movie recommendations and media streaming to e-commerce and social media applications. Therefore, it is clear that the need for accurate personalized recommendations is growing bigger by the day since the aforementioned services are part of our daily lives and in many cases, unavoidable.

One area of interest for recommender systems is healthcare. The main purpose of our research work is to develop a framework for providing accurate and explainable personalized drugs recommendations for patients, making use of different algorithms while trying to avoid any possible side-effects from the recommended prescriptions. To achieve this we worked with the MIMIC-III database [39] and used some of Cornac library's built in Matrix Factorization algorithms for our experiments, as well as, Neural Networks and compared the results according to our evaluation protocol which is explained in detail in Chapter 4. From the database we made use of the relevant data about the patients, such as their diseases and the history of the user-drug relations. In addition, we had to make use of the

list of adversarial drug-drug interactions (DDIs), which contains all pairs of drugs that may cause different side effects if prescribed together, for our experiments to be meaningful. In the next Chapter of our work, we describe various models that have already been used and tested and showcase their results. Chapter 3 is an in depth theoretical overview of Neural Collaborative Filtering and Multi-Layer Perceptron which we also used for our experiments in the next Chapter. Chapter 4 contains the practical aspect of our research, as it presents the experiment runs for the various algorithms, that we compare in order to determine the best possible results for us to conclude which option is the safest overall.

1.1 Introduction to Recommendation Systems

According to [36] the Recommendation System (RS) should act as an information filtering system that urges to predict the preferences that the user may have for one item over another and to predict whether or not a particular item would be preferred by him. A precise definition of a recommendation system is given as (Figure 1.1): A recommendation system (which sometimes replaces the system with a synonym such as a platform or a machine) is a subcategory of the information filtering system that seeks to predict the rating or preference that a user would give to an item.

One of the lessons learned in recent years from recommendation systems research is that the scope has a significant influence on the types of methods that can be applied successfully. Particularly important are features of areas such as maintaining user utility. For example, users' tastes in music may change slowly, but their interest in celebrity news can fluctuate much more. Thus, the reliability of preferences accumulated in the past may vary. Similarly, some items, such as books, are available for recommendation and consumption for a long time - often for years. On the other hand, in a technology field such as cell phones or cameras, old products are quickly becoming obsolete and cannot be used in a useful way. This also applies to areas where current events are important, such as news and cultural events.

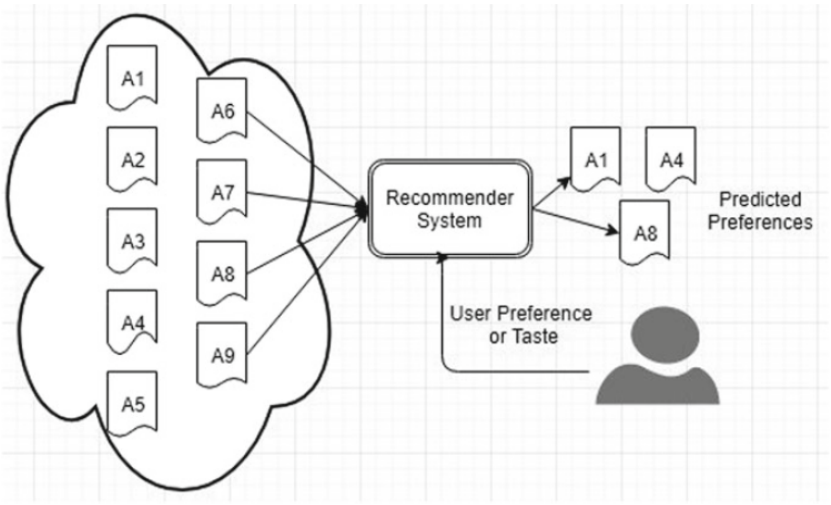

Figure 1.1: Recommender System.

It is therefore not surprising that there are many aspects of research into recommendation systems, as researchers are concerned with various areas of recommendation. To integrate these different approaches, it is useful to consider the aspects of artificial intelligence (AI) recommendation, and in particular the knowledge base on which a recommendation system is based.

Each artificial intelligence system relies on one or more sources of knowledge to do its job. For example, a supervised machine learning system would have a labelled data collection as its primary source of knowledge, but its algorithm and parameters can be considered another implied kind of knowledge used for classification work. Proposal algorithms can also be classified according to the knowledge sources they use. There are three basic types of knowledge [27]:

- social knowledge about the user base in general,
- personal knowledge of the specific user for whom recommendations are requested (and possibly knowledge of the specific requirements that these recommendations must meet) and, finally,
- content knowledge about recommended components, ranging from

simple feature lists to more complex ontological knowledge and knowledge that allows the system to explain how an item can meet a user's needs.

The different recommendation approaches come from different parts of this range of knowledge sources. The terms of the Netflix Award competition only contained opinions in the form of ratings, but there were no requirements or demographic information for users. Good knowledge of the field is notoriously difficult to gather in this field, due to the complexity of representation and reasoning for narrative content, directional style, etc. Thus, the problem was solved with a mathematical approach technique based solely on social and individual scores. In contrast, the problem of investing options mentioned in [27] can benefit from detailed knowledge of the client's income and financial condition, other elements of his portfolio, and his attitude towards risk. The opinions and choices of other users may be helpful, but they are not enough to make high quality recommendations in this area.

Various methods are proposed to develop an effective system of recommendations, two of which form the basis for the development of other approaches. These methods are content filtering, collaborative filtering [36]. In addition, these techniques are extended and in the present RS techniques are classified as (Figure 1.2):

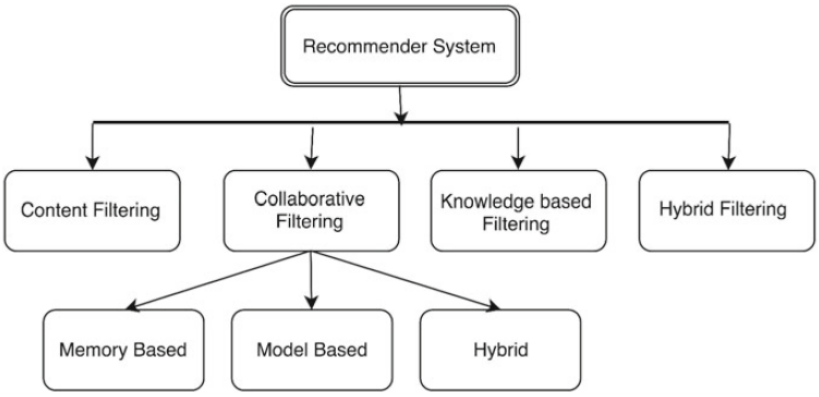

Figure 1.2: Recommender System Classification.

1.2 Item-KNN & User-KNN

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another. It is also one of the best-known classification algorithms. The principle is that known data are arranged in a space defined by the selected features. When a new data is supplied to the algorithm, the algorithm will compare the classes of the k closest data to determine the class of the new data. Analysis consists in the observation of the impact of parameters such as distance and classification rules on classification results. The major advantage of the KNN classification is its simplicity, it is also an efficient method. However, despite its efficiency, computation times can be long with large databases, the determination of the number of neighbors to use (k) requires trial and error and the algorithm is weak with outliers which can strongly impact its efficiency [19].

K-Nearest Neighbors method is based on analogous learning. That means that it compares a given set of controls with training sets that are similar to it. Each plural represents a point in an n-dimensional space. In this way, all training blocks are stored in a one-dimensional space template. When an unknown plural is given, a k - nearest neighbor classifier searches the space pattern for the k training plots that are closest to the unknown plural. These k training clusters are the k "nearest neighbors" of the unknown cluster.

The first step is to calculate the distance between the new point and each training point. The Euclidean distance between two points or multiples, that is, $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots, _{2n})$ is,

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

In this approach, similarities between pair of users and items respectively are computed using Pearson similarity metric.

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

where:

- n is sample size
- x_i,y_i are the individual sample points indexed with i
- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean); and analogously for $\bar{y}$

1.3 Non-negative matrix factorization (NMF)

In the Non Negative Matrix Factorization (NMF) algorithm, an array X is transformed into a product of two other arrays W and H, as shown in Equation 1.

$$nmf(X, k) = W \times H(1)$$

Besides table X, the algorithm takes as input a positive number k. The generated table W has k columns, while H has k rows. In general, the factorization of tables is not done in a unique way and therefore a number of different methods have been developed (e.g. principal component analysis, singular value decomposition) with each having its own limitations. In NMF there is a certain restriction that limits table X, allowing it to contain only elements greater than or equal to zero. Tables W and H are subject to the same constraint and are generated by the algorithm so that they do not have negative elements. Let V be the product of the tables W, H such that:

$$W \times H = V(2)$$

The multiplication of the arrays can be done by linearly combining the column vectors of W with the values of H. Thus, each column of array V is computed with the Equation 3.

$$v_i = \sum_{j=1}^{N} H_{ij} W_j (3)$$

where

- N is the number of columns of W
- v_i is the i-th column vector of the generated table V
- H_{ij} is the value of the table H in the row j and column i
- W_j is the j-th column of the table W

1.4 Multilayer Perceptron (MLP)

A multilayer network consists of two or more sensors (Perceptrons). Particularly, a MLP neural network with a hidden layer consists of the input level, the level of hidden neurons, which may be one or more, and the output level. A significant characteristic of these neural networks is that each node is powered by the output of the previous level, but also that it is a fully interconnected tree. That is, each neuron on one level is interconnected with all the neurons of the previous level.

The training process begins with the random initialization of the weights of each neuron. Then, the process proceeds with two network passages (one forward and one backward). In the forward process each input element goes through the hidden levels and then ends up at the network output giving a value to the dependent variable. In this passage the weights remain constant. After passing through the output level, all of the input elements are capable of giving a result. This process allows the calculation of the model error by subtracting the found values from the actual values using a cost function. Then, an extra calculation is needed in order to determine the amount of weight that affects the error. This can be done by calculating the few derivatives of the final node and then by applying the

derivative reduction algorithm (gradient descent). The form of the function used for weight renewal is as follows:

$$w = w - \eta \nabla_w J(w)$$

Afterwards, this information is transferred to the immediately previous level. There, the user will have to calculate some of the derivatives while using the previously acquired information about weights. Thus, their values can be determined with ease. This process continues until the user reaches the entry level, which will obviously remain the same.

Chapter 2

Bibliographic Review of Drug Recommendation Systems

Synopsis: In this chapter, we will provide a bibliographic review of recommendation systems in healthcare. We will showcase what the research on the field has achieved so far and describe the techniques for each work respectively, as well as their performances.

2.1 Recommendation Systems for Precision Medicine

The authors' study [56] briefly refers to biomarkers, which play an important role in the treatment of respiratory functions. It is also noted that more than 10% of patients treated in intensive care are affected by acute lung injury (ALI). The most severe type of ALI is the Acute respiratory distress syndrome (ARDS) in which a mortality rate of 40% is observed. The contribution of biomarkers to the pathophysiology of ARDS is much smaller to a significant degree. However, a major problem is that ARDS is such a diverse and multifactorial termination condition that the techniques for "clotting and disintegration" are considered to be very serious and intense.

However, the implementation of a biological network in ARDS is made possible by the sequence of the human genome and the provision of improved techniques to analyze mRNA transcription (gene expression) and also through the development of sensitive immunoassays. It is also used in the molecular phenotypic domain to identify patients at risk of de-

veloping ARDS through a panel of biomarkers. This disease is still serious and deadly despite the development of the treatment and control of ARDS. Monoclonal antibodies (anti-Tumor Necrosis Factor- anti-TNF) and TNFR fusion protein have given incompatible results. However, with advances in mechanical ventilation techniques, the increasingly frequent use of neuromuscular-blocking drugs has led to positive results and the development of somatic cell therapies. In the future, it is expected that this could provide reasonable corrective goals and ultimately improve clinical care, as the role of understanding biomarkers is integrated into the pathophysiology of ARDS and lung problems.

On the other hand, [46] briefly describes the approach that although there are many clinical benefits that clarify the importance of producing treatment for rare diseases and cancer, the impact on treating and controlling very complex diseases such as diabetes type 2, remains very low. It also primarily identifies the ways in which people are exposed to serious health problems through the application of diagnostic labels. According to his study, a different "pallet" model is proposed, based on a molecular classification that aims to categorize an individual according to the most pathophysiological processes, which involve the risk and development of diabetes.

2.2 Recommender System Using Collaborative Filtering

Over time, the health recommender system (HRS) is becoming more and more a very important platform for healthcare services. Against this background, intelligent health systems have become essential tools in healthcare decision-making processes. Their main goal is to ensure the availability of valuable information at the right time, ensuring the quality of information, reliability, authentication and confidentiality. Just as people use social media to understand their state of health, so is a system of health referrals important for delivering results such as diagnoses, health insurance, clinical trials, and alternative medicines, the patient's health profile. There has been a lot of research aimed at using a large volume of medical data while combining multimodal data from different sources. This reduces both the workload and the cost of health care. In the field

of healthcare, big data analytics using referral systems play an important role in decision-making processes related to patient health. In particular, the authors' research [60] includes a proposed, intelligent HRS using the deep learning method of the Restricted Boltzmann Machine (RBM) and the Convolutional Neural Network (CNN), which provides an overview of how big data analytics can be used to implement an effective health recommendation engine. In addition, it illustrates the opportunity given to a healthcare industry to move from a traditional situation to a more personalized model, within a tele-health environment. Finally, and taking into account the values of the Root Square Mean Error (RSME) and the Mean Absolute Error (MAE), the proposed Deep Learning Method (RBM-CNN) has fewer errors than other approaches.

2.3 Fast Data Processing in Recommendation Systems

The problem of analyzing a large volume of user data to determine their preferences and based on this data, to provide new product recommendations, is of paramount importance. Depending on the accuracy and timeliness of the recommendations, significant gains or losses may arise. The work of data analysis for users of services of different companies, is carried out in special referral systems. However, in cases where there is a large number of users, the data to be processed becomes too large, which complicates the operation of referral. The Singular Value Decomposition (SVD) method can perform intelligent information analysis, providing an efficient data analysis in trading systems. In addition, in the face of a large volume of processed information, the authors [54] proposed the use of distributed systems. This approach reduces data processing time and recommendations to users. For the experimental part of their study, they applied the distributed SVD method using Message Passing Interface (MPI) technologies, Hadoop and Spark, and obtained the results of reducing data processing time using distributed systems compared to non-distributed systems.

On the other hand, [18] proposed the algorithm based on SVD and the algorithm of multiple objective immunity (Multi-Objective Immune Algorithm - MOIA), or SVD-MOIA. In this research, MOIA was used to

optimize the conflicting goals of accuracy and diversity. However, the authors modified the algorithm only to determine the optimal number of parameters to provide recommendations, leaving the issue of applying the method that arises in cases of large volumes of data remains unresolved. The recommendation system is primarily concerned with the accuracy of the recommended result. For the target user, R is the list of items recommended for user u . Just Function f_1 is used to define the accuracy of the recommendation list as follows.

$$f_1 = \frac{\sum_{i \in R} S(u,i)}{|R|}$$

$S(u,i)$ denotes the similarity between the user u and item i . $|R|$ is the length of R. The accuracy is better if the function value is higher.

In addition to the recommended accuracy, we also hope there are diverse recommended items in the list, which may bring a surprise to the user. Diversity of the recommended list can be formulated with function f_2 as follows.

$$f_2 = \frac{\sum_{i \in R} \sum_{j \in R, i \neq j} S(i,j)}{|R| \times (|R| - 1)}$$

The function f_2 is a measure of the similarity between the recommendation items. The smaller the similarity, the better the diversity. And so, the recommendation problem is converted into a multi-objective optimization problem as follows.

$$\min \quad \{-f_1, f_2\}$$

Therefore, [54] modified the SVD method in order to ensure the efficient operation of the algorithmic recommendations, even with increasing data. Furthermore, [37] proposed the normalization of the unique decomposition method based on the choice of parameters, while [3] considered the adaptation of the recommendation systems for data processing needs.The Rocchio algorithm is used to produce recommendations based on relevant and non-relevant documents previously voted by the user. The method can work with term frequency, tf-idf, or any term weighting schemes. Given a set of documents vectors that are found to be relevant by the user $r1, r2, \ldots, rn$ and another set of non-relevant documents $u1, u2, \ldots, um$, then the algorithm finds a document vector that combines

both sets of documents as follows.

$$q = q_0 + \frac{\alpha}{N}\sum_{i=1}^{N} r_i - \frac{\beta}{M}\sum_{j=1}^{M} u_j$$

where q_0 was originally proposed as the mean document vector of an initial query made into the system, such as a search for a particular keyword. The parameters α and β control how much the relevant and non-relevant documents affect the final query vector, respectively.

In addition, [45] proposed a combination of the Singular Decomposition method and statistical methods for the operation of the travel guide system, but without presenting an adaptation of the SVD method for use in distributed systems. [30] proposed a method of combining information about data characteristics with a historical scoreboard to predict potential users' preferences. This method combines features and time information in a decomposition table model. The method described is an effective way to solve the problem of slow start of new data. While, some methods of a data processing system that analyzes data for movies based on algorithms oriented to each customer, were formulated in their study [21].

Finally, the authors [1] proposed a new model by integrating three sources of information: a point-user table, explicit and implicit relationships. The basic strategy of this research is to use the method of Multistage Resource Allocation (MSRA) to identify hidden relationships in social information. First, explicit social information is used to identify similarities between each pair of users. Second, for each pair of users who are not friends with each other, the MSRA method is used to determine the likelihood of their relationship. Therefore, if the probability exceeds a threshold value, a new relationship is created. All sources are then included in the SVD method for calculating residual forecast values. Pearson Correlation Coefficient method is exploited to compute the similarity between explicit friends and implicit friends by measuring the common rated items for both users as follow.

$$sim(i,f) = \frac{\sum_{h=1}^{n}(r_{i,jh} - \bar{r}_i)(r_{f,jh} - \bar{r}_f)}{\sqrt{\sum_{h=1}^{n}(r_{i,jh} - \bar{r}_i)^2}\sqrt{\sum_{h=1}^{n}(r_{f,jh} - \bar{r}_f)^2}}$$

Where, $r_{i,j}$ is the rating of item j by user i, $r_{f,j}$ represents the rating of item j by user f and f indicates the friends of i which refers to the average

rating of the user. n is the number of the common items between users i and f. In this method, the similarity value between i and f is ranging from [–1,1], the higher value means that the two users are more similar. The similarity method is applied two times, first for the explicit friends and second for implicit friends. All these resources are employed in SVD to produce the new model.

2.4 Drug Recommendation System Based on User Profile

With the rapid growth of the Internet, the online medical platform has become an essential part of the drug trade. In order to help users find fast and satisfying products through a large number of products, the recommendation system has been proposed. The traditional recommendation algorithm usually only takes into account the user component score, which leads to low prediction accuracy. In their work [17] proposed a method of recommendation based on user profile, which uses in-depth learning to analyse user behaviour and create multidimensional features of the user. The profile can be constructed by analysing information about the drugs. By analysing historical information about user activity, including market, browsing, and collection, it is possible to make dynamic predictions of user evaluation of drugs with the help of a trained neural network. Experimental verification on the B2B medical platform shows that the prediction accuracy is higher than other algorithms. The proposed system can not only improve the user experience, but also increase sales of the platform.

The rapid development of technology has made people's lives more closely integrated into the Internet. Internet data is growing at an unprecedented rate. According to IDC, global data will increase from 33ZB in 2018 to 175ZB in 2025, where 1ZB equals 1 trillion GB. Big data has greatly facilitated human life, but it has brought new challenges, most notably information overload. One solution to this problem is the search engine although the problem can not be solved as long as users are not able to define the keywords of the correct search. Another solution is the recommendation system, which is based on user behaviour and can help them

quickly retrieve the desired information. The recommendation system has been widely used in both academia and industry.

The core of the recommendation system is the recommendation algorithm, which is responsible for finding the items that are potentially of interest to the user through user and item information. Traditional recommendation algorithms can be classified into trees: collaborative filtering [48], content-based algorithms [78] and hybrid recommendation algorithms [20]. Different recommendation algorithms are proposed for different scenarios and environments.

Providing high quality personalized services to users must be based on user understanding. In recent years, the user profile has been used to create multi-dimensional user-related tags. It depicts his background, his interests, his identity, his psychology, his needs, his personality, etc. and adequately describes the user information. In their work [18] proposed a recommendation algorithm based on the user profile, which can dynamically reflect the user preferences and purchase intention.

In recent years, deep learning has achieved excellent results in computer vision [66], speech recognition and comprehension of natural language [85]. Since deep learning can automatically learn features and the indirect relationship between them [77], recommendation algorithms based on deep learning can overcome the problem of data sparse and cold start that other existing algorithms have. recommendations. In their work [17], they proposed a dynamic user profile recommendation algorithm. Initially, the user profile is created dynamically and then the user-drug evaluation is based on a trained neural network. Experimental results show that the proposed algorithm performs much better than other recommendation algorithms.

The user profile is an essential part of any given recommendation. [67] devised a method of scalable user profiles from user profiles, which focuses on dynamic and static user information [32]. [50] attempted to classify user topics using the Bayesian algorithm, which was derived from user-generated text information [43]. [26] used user behaviour, recording, and history data to create a user profile that reflected the user's actual

psychological needs at the time [22]. [58] used textual and social characteristics to create user profiles and conducted research on age classification [83]. Multidimensional users began to be created through data from various types of features. Obviously, multidimensional data has proven to be a reliable source for creating stereoscopic user profiles, which can significantly improve the quality of the user profile.

User profile technology plays a vital role in many areas, such as advertising marketing and personalized recommendation services, mainly in the form of linking information and user goals. For the recommendation system, algorithms are responsible for finding the items that users are most likely interested in. Collaborative filtering algorithm is the most commonly used recommendation algorithm. The sort algorithm sorts and predicts whether a user buys a particular item type based on the data attributes and historical behaviour of the user and the item. Famous classification algorithms are the decision tree, the support carrier machine and the accounting regression. [13] proposed a random forest algorithm [10]. It is a complete learning method, and the GBDT algorithm uses the integration method, which has a strong generalization ability. In recent years, learning-based recommendation algorithms have been proposed to overcome the limitations of the traditional model that can achieve better performance. In-depth learning effectively captures non-linear and non-trivial user-drug relationships and supports the association of more complex abstractions in higher-level data representations [38].

User Profile Based Algorithm with Deep Neural Networks

In this section, a recommendation algorithm based on user profile is proposed. Features of users and drugs are first extracted and then the deep neural network (DNN) is trained to make predictions.

The characteristics of the user profile and the drug are shown in the following figures. The characteristics of the user profiles are developed by analyzing information from three dimensions, wealth index, activity index and interaction behavior. The wealth index is created by sales. The activity indicator describes information about users' activities, such as browser frequency and purchases. The interaction behavior shows the relationship

between user and different types of drugs. By analyzing the efficacy, the factory and the price of the drugs, it is possible to obtain characteristics of the drugs, as shown in the following figure.

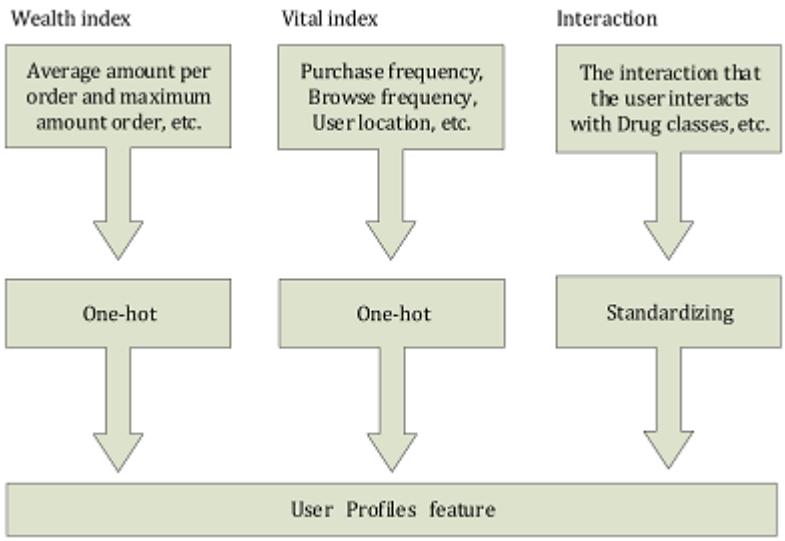

Figure 2.1: User Profiles.

Deep Neural Network Model

As a non-linear model, a deep neural network has a strong ability to adapt, can learn the indirect relationship between features, and improve the performance of the composition algorithm. The entry of the deep neural network developed by [17] is a combination of user and drug characteristics as shown in the figure below.

Experiments and Evaluation of the Model

The performance of the proposed algorithm on a server equipped with two GPUs (NVIDIA 1080TI) and one CPU (Intel (R) Xeon (R) e5-2670), 256G memory is investigated. The data set is collected by a medical B2B platform, includes user purchases, browsing, collection and other behavioural

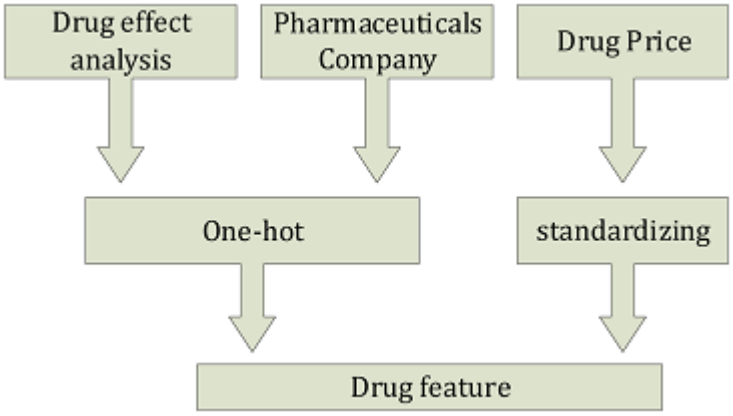

Figure 2.2: Characteristics of Drugs.

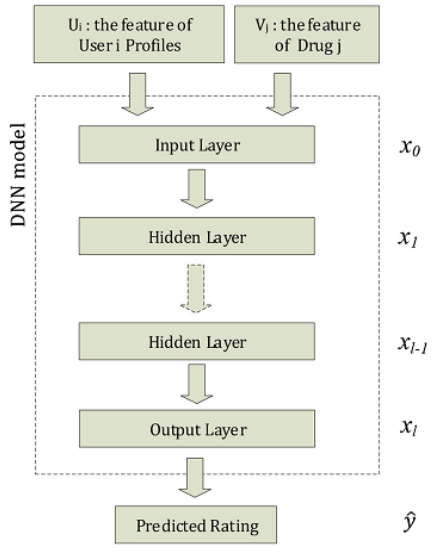

Figure 2.3: DNN Model.

data. It contains 1.83 million reviews, recording 103,799 users and 5,919 drugs. The statistics of the data set are shown in the table below.

#users	103799
#elements	5919
#ratings	1830000
#ratings/user	17,63
#ratings/element	309,17
Inability to evaluate	99,7%

Table 1. Data set statistics

There are many parameters for evaluating the performance of a recommendation algorithm, such as coverage, random discovery, and prediction accuracy. In his article [17], the Mean Absolute Error (MAE) and the Root Means Square Error (RMSE) are adopted for performance evaluation. They can demonstrate directly whether the model can capture the key features of educational data. The MAE value reflects the change in the actual error value and the RMSE cost indicates the difference in the amount of the maximum error. The expressions MAE and RMSE are given as follows:

$$MAE = \frac{1}{N}\sum_{i,j}|R_{ij} + \hat{R}_{ij}|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i,j}|R_{ij} + \hat{R}_{ij}|}$$

where N refers to the number of samples in the test set, Rij is the user's actual score value and s the predicted score value of the recommendation algorithm. The smaller these two values, the better the performance that the algorithm can achieve.

In the simulations performed, the proposed algorithm was compared with other algorithms, such as linear regression (LG), single decomposition matrix decomposition (SVD) [79], non-negative matrix factorization (NMF) [28] and accounting regression (LF). The following figures show that the proposed algorithm achieves the best performance. A hidden four-level network was used. The first level is the entrance level, which has 623 dimensions. From the second to the fifth level are the hidden levels. The number of hidden level neurons is 1000, 500, 200 and 100. ReLu was selected as the activation function. The output plane has five neurons and softmax was used as the activation function.

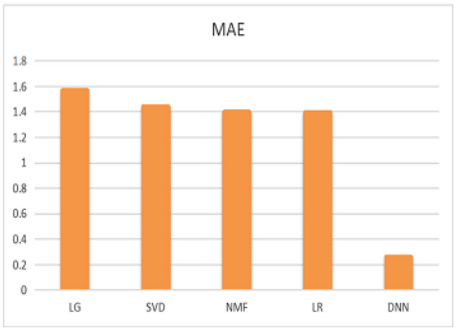

Figure 2.4: MAE of different methods of composition.

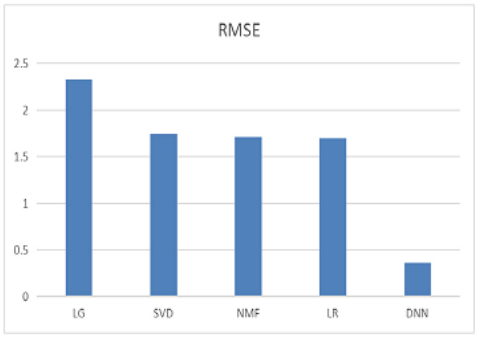

Figure 2.5: RMSE of different methods of composition

In their study [17], a user profile-based recommendation algorithm for an online medical platform was proposed and a deep neural network was used to calculate the user-drug rating prediction according to service and drug characteristics. The framework of the study was simple and flexible. The experimental results show that the proposed model achieves a very high prediction efficiency, which proves that the implementation of the deep learning model is a successful attempt at user profiles. In future work, other in-depth learning techniques will be used to improve system performance on user profiles, such as convergent neural networks, repetitive neural networks, etc.

2.5 Framework of Intelligent Drug Recommendation System

In their work [5], they designed and implemented a universal medicine recommendation system framework that applies data mining technologies to the recommendation system. The drug recommendation system consists of the database system unit, the data preparation unit, the recommendation unit, the model evaluation, and the data visualization unit. SVM (Support Vector Machine) drug recommendation algorithms, BP neural network algorithm and ID3 decision tree algorithm based on diagnostic data were analysed and tested. Experiments were performed to coordinate the parameters for each algorithm for the best possible performance. Finally, for this specific open data set, the SVM recommendation model for the drug recommendation unit was selected in order to achieve a good balance between the accuracy, efficiency and scalability of the model. In addition, an error checking mechanism was proposed to ensure the accuracy of the diagnosis and the quality of services. Experimental results have shown that this system can recommend drugs with excellent performance, accuracy and scalability.

The Hospital Information System (HIS) provides a huge amount of data. The big challenge is how one can extract possible and useful knowledge from the diagnostic case data. Data mining and recommendation technologies are a very promising direction to solve these demanding problems. Since the mid-1990s, various setup systems techniques have been proposed, and recently many types of setup system software have been developed for a variety of applications. Traditional recommendation technologies complement collaborative filtering (CF), content (CB), knowledge (KB) and hybrid proposal techniques, but these techniques are subject to certain limitations. CB provides highly specialized recommendations and CF has problems with weakness, escalation and cold start. To solve these problems, data mining technologies are applied to recommendation systems in order to design a new drug recommendation system framework.

In their work [5], they introduced a system of universal medicine recommendation systems designed and implemented to apply data mining technologies to recommendation systems that use the potential knowledge

contained in medical records in order to reduce medical mistakes. The drug recommendation system consists of a database system unit, a data preparation unit, a recommendation model unit, a model evaluation unit and a data visualization unit. Data visualization presents some valuable knowledge behind diagnostic case data. The drug recommendation algorithms of SVM (Support Vector Machine), BP neural network and ID3 decision tree based on diagnostic data were examined. The SVM was finally chosen for the drug recommendation model thanks to its high accuracy, good performance and scalability. Taking into account the safety of the patient, a faulty control mechanism was proposed that ensures the safety and quality of the service. Experimental results have shown that the system proposed by them [5] can recommend drugs with excellent efficacy, accuracy and scalability.

Recommendation systems aim to provide users with personalized products and services to address the growing problem of information overload on the Internet. Various recommendation system techniques have been proposed since the mid-1990s and many types of recommendation system software have been recently developed for a variety of applications. Most of the recommendation technologies are applied in areas such as: e-government [31], e-commerce [44], e-commerce / e-marketing [82], tourism [74] and so on. However, the field of medicine rarely includes recommendation technologies and their work [5] focused on the design of the drug recommendation system and the knowledge of extracting from medical case data.

The most common recommendation techniques include collaborative filtering (CF) techniques [25] content-based (CB) techniques [50] and knowledge (KB) [47] as well as hybrid recommendation technologies [16]. Each recommendation technology has advantages and disadvantages: CB mainly produces recommendations using traditional recovery methods and machine learning methods, but on the other hand provides highly specialized recommendations. Collaborative filtering (CF) recommendation techniques help to make choices based on the views of other people with similar interests, but this technique presents problems of weakness, escalation, and cold start. The Knowledge Base (KB) proposition provides users with data based on knowledge about users, data and / or their relation-

ships. Typically, KB recommendations maintain a functional knowledge base that describes how a particular item meets the needs of a particular user. To achieve higher performance and overcome the disadvantages of traditional technical proposals, a hybrid composition technique has been proposed that combines the best features of two or more composition techniques. When it comes to a new application area, a new set of recommendations is needed to address these issues.

Smart medical diagnosis is a growing concern. Some selected techniques for data mining in medicine are discussed in [41]. Data mining technology has been used to predict heart disease and diagnose thyroid disease [70]. Health search research presents challenges and important issues related to medical search and discovery, suggesting that information retrieval, personalization, expert modelling, data mining and privacy protection are vital to make progress in health. These approaches solve some problems in medical diagnosis and application of new technologies based on data in the medical field. However, there is no one-size-fits-all model that works well for all data and conditions. When dealing with different data and application scenarios, it is necessary to create a different model and perform data analysis. In their work [5], they presented a framework of universal medicine recommendation system that has been designed and implemented for the application of data mining technologies in the recommendation system where the possible knowledge contained in medical records is used to reduce medical errors.

The recommendation system has become a valuable field of research as it is combined with the development of artificial intelligence technologies. Unlike most current recommendation systems that focus on e-business, book and movie suggestions, our system aims to provide a virtual experienced physician for inexperienced beginners and patients in the use of the right medications. As high accuracy and efficiency are critical to such a system of online drug delivery, it is advisable to evaluate some data mining approaches to achieve the best balance between accuracy, efficiency and scalability.

The Database System section provides data links to modules, which contain databases of diagnostic cases, drugs, and specialized knowledge. The

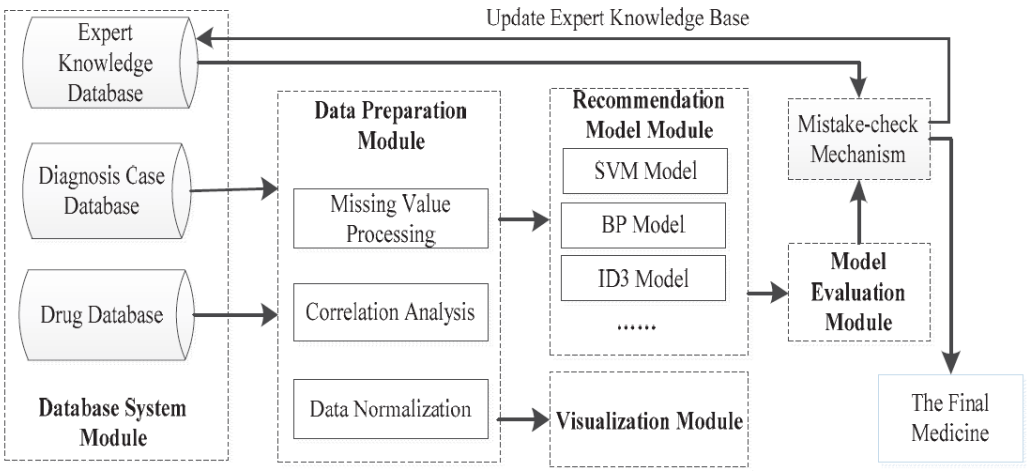

Figure 2.6: The context of a recommendation system

database communicates, stores the data of the hospital diagnostic cases and provides access to other units. The Drug Database collects all the drugs and indexes the largest ones to be found. The database of experts stores the knowledge of experts, which is obtained by collecting their experiences.

The data preparation unit acts as a data cleaner in the system. Actual data is raw data and therefore likely to be incomplete or contain noise. Therefore, the data preparation unit is intended to make clear data and consists of value sending processing, systems analysis and data normalization. The configuration model unit is the core of the system proposed by them [5], which is based on a combination of SVM algorithms, BP neural networks, ID3 decision tree and so on. Although this model is based on these three algorithms, it is of course possible to add new data mining algorithms. The visualization unit mainly provides the visualization technology in order to present valuable knowledge behind the diagnostic case data.

The model evaluation unit evaluates the different recommendation models in a specific data set. When dealing with different application scenarios, it becomes necessary to evaluate the models in order to achieve the

optimal compensation between the accuracy, efficiency and scalability of the model.

This universal medicine recommendation system applies data mining technologies to medical diagnostics, making full use of diagnostic case data and expert experience. Some models of recommendations are developed based on the case data of the diagnosis and suggest the appropriate drug for the patient through the experience of specialists. In addition, a faulty control mechanism is proposed to ensure the accuracy of the diagnosis and the quality of service. That is, if the drug recommended by the SVM model does not match the opinion of the specialists, the specialist doctor will examine the patient, suggest more appropriate drugs and update the specialist experience database, so that this strategy ensures the patient safety and improve the quality of service.

Vector Support Engines

In 1955, Cortes and Vapnik proposed the idea of the Support Vector Network, which was a new learning machine for two-group classification problems [24]. This is the first version of SVM. A carrier support machine constructs a super level or a set of super levels in a space of high or infinite dimensions, which can be used for sorting, regression or other tasks. Intuitively, a good separation is achieved by the layer that is farthest from the nearest training data point of any class (the so-called operating margin), as generally the larger the margin the smaller the classifier generalization error.

SVM is based on the basic idea that the input carrier can be divided by a linear decision surface when mapped to a very high-character feature space. SVM can only deal with the two classification problems initially, the model was extended to deal with multiple order problems and regression problems [20]. SVM offers a huge advantage for high-dimensional data and non-linear data. It overcomes the problem of overload and dimensional damage. The final optimized problem can be transformed into a limited quadratic programming (QP) problem [52]. When it comes to large-scale data, SVM does not perform fast. There is also no standard for

selecting kernel mode. The work in [65] applies the SVM model to identify complex patterns of human movement.

SVM has good scalability for multidimensional data and is suitable for pattern recognition and regression problems based on statistical considerations. SVM turns the request into an optimization problem. Let the set of training data be $(x_i y_i), i- = 1, 2, ...l, x \in R^n, y \in \{\pm 1\}$ where l is the number of training samples and n is the dimension of the data set. The final optimization problem is as follows:

$$maxQ(a) = \sum_{j=1}^{l} a_j - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} a_i a_j y_i y_j (x_i \cdot x_j)$$

$$st. \sum_{j=1}^{l} a_j y_j = 0, j = 1, 2, ..., l, a_j \geq 0, i = 1, 2, ..., l$$

So the final super level resulting from the double optimal problem is obtained. The classification function is as follows:

$$f(x) = sgn\left\{ \sum_{i=1}^{l} a_i * y_i k(\vec{x} \cdot \bar{x}_i) + b^* \right\}$$

Where $x \in R^n$ and a_i^* and b_i^* are optimization parameters.

$k(\vec{x} \cdot \vec{x_i}) = \phi(\vec{x}) \cdot \phi(\vec{x_i})$, is the definition of the kernel function. It is enough just to calculate the internal production of the feature space.The SVM enters the kernel function to compute the internal function. The above problem can therefore be treated as a non-linear classification problem.

Due to these features, the SVM has good scaling and performance. There are two important parameters that can greatly affect the performance of SVM. This is the gamma parameter denoted as g in the kernel function and the cost parameter denoted by c. And the K-aspect cross-validation method is applied in the experiments to obtain the appropriate g and c, in the K-aspect cross-validation, the K-1 aspects are used for training and the latter aspect is used for evaluation. This process is repeated K times,

so that each time the evaluation aspect is different. In their experiment [5] set K = 5 and the number of examples is 1200. Figure 4 shows the optimization process and the result of g and c. For simplicity of the design, the logarithm of g is treated as a y-coordinate and the logarithm of c is treated as a x-coordinate, the accuracy of the SVM is considered to be a z-coordinate. The colour of the line from blue to red indicates that the accuracy ranges from low to high. And we can understand it when g = 0.125 and c = 362.0387, the SVM has good 95% accuracy.

BP Neural Network

Backpropagation (BP) is a well-known method for training an artificial neural network with an optimization method such as cathodic inclination. The BP neural network is based on reverse propagation that minimizes the sum of square errors between the actual and desired output values. His work [53] demonstrates that artificial neural networks provide effective solutions to problems encountered in building systems that mimic a physician's experience and present a system of specific neural networks in medical diagnosis. However, due to the downward method, there are some disadvantages to the BP neural network, such as the long training time and the easy convergence to the local minimum. To address these issues, [71] some improved approaches are proposed.

Based on the diagnostic data, training of the BP model of the neural network is performed and specific experiments are performed with different number of hidden nodes. A variable backward learning rate training function is used to train the model. The following figure shows that the test error rate and the training error rate changed as a function of the number of hidden nodes. It has been observed that the learning error rate decreases with increasing number of hidden nodes, while the test error rate decreases first and then shows an increasing trend. For number of hidden nodes = 8 an accuracy of 97% was achieved.

ID3 Decision Tree

ID3 is an algorithm proposed by Ross Quinlan and used to create a data tree based on data [55] from which it is then possible to make decisions

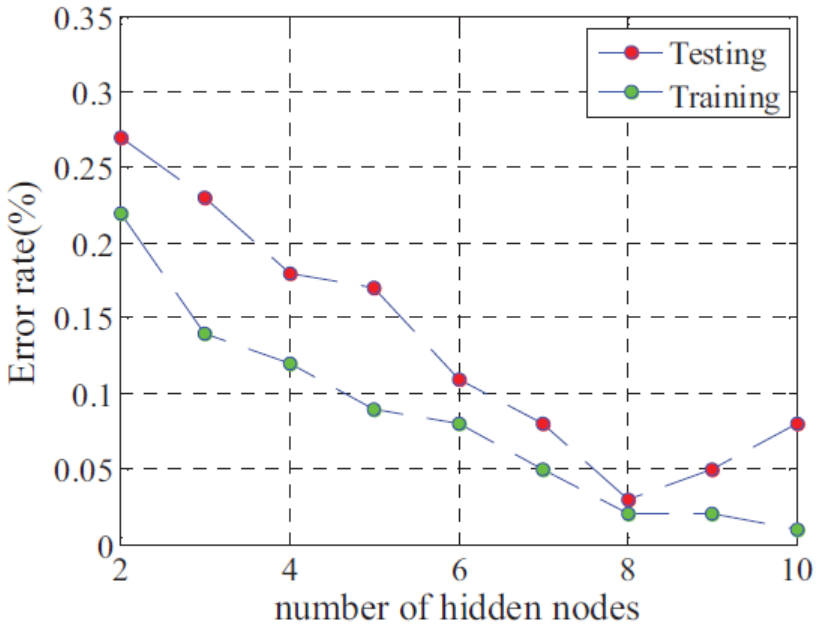

Figure 2.7: Evaluation of test and training error rate

based on this tree. The main challenge in ID3 is the choice of features while gaining information is of great value. In addition, ID3 can make decisions in an abstract and thoughtful way. Taking into account these features, in combination with good performance, the ID3 model emerges. The final decision tree is shown in Figure 6 and the numbers from one to five indicate the drug A, B, C, X, Y. After all the features are separated, a decision tree is created based on the information gain of each feature. So in the case of a new patient, the ID3 recommendation model will first check the K level and then find the output node of the tree.

Model Evaluation The diagnostic case data is divided into two parts, which consist of 70% training data and 30% test data, while for each model ten sets of experiments are performed in order to avoid accidental errors. The accuracy of the model and the execution time are the average of ten groups of experiments. The following figure illustrates the result of the experiment.

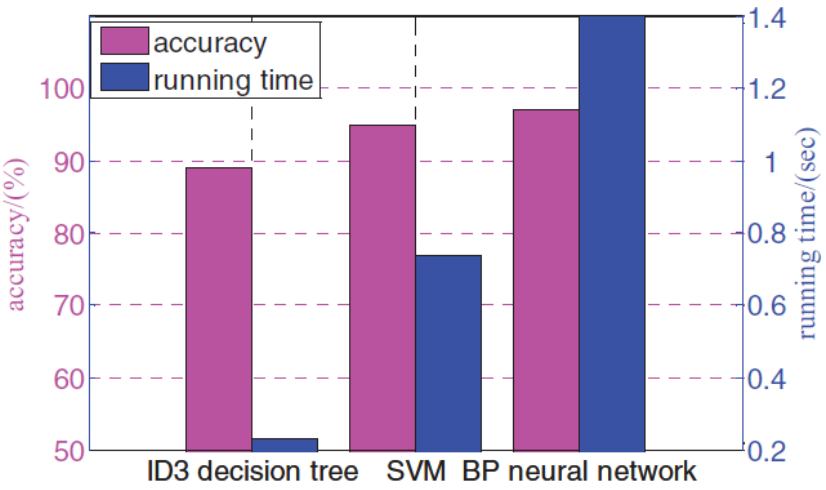

Figure 2.8: Accuracy and running time of the three models

The previous figure gives a visualization of the accuracy and operating time of the models, the length of the red bar indicates the accuracy of the model and the length of the blue bar indicates the operating time of the model. It seems that the accuracy is getting bigger while obviously the performance is changing.

When a drug recommendation system gives a diagnostic result and recommends the drug to a patient, the first thing to consider is accuracy. High accuracy leads to high quality services and high security. However, efficiency is also a very important factor for the electronic system, and the result of the diagnosis must be available as soon as possible. The ID3 decision tree model shows the shortest time, but its accuracy is only 89% and it has poor scalability. As the features get bigger, it's hard for ID3 to make a decision tree. The BP neural network on the other hand achieves the highest accuracy, but the operating time is 1.7 seconds, while requiring longer training time and poor understanding of the result. The SVM model has good 95% accuracy in 0.74 seconds. SVM is therefore selected as a model drug recommendation due to its high accuracy, good performance and scalability in this open data set.

2.6 Comparative Neural Network Graph for Drug Interchange System with Crossover Sales

Recent developments in deep neural networks for structured graph data have led to state-of-the-art performance in recommendation system evaluation criteria. Adapting these methods to drug cross-breeding work with one million products and hundreds of millions of sales remains a challenge, due to the complex medical and legal properties of pharmaceutical data. To meet this challenge, a graph convolutional network (GCN) algorithm called PharmaSage was developed, which uses chart convolutions to create drug integrations, which are then used in a downstream recommendation job. The underlying graph incorporates both cross-selling information from the sales transaction into the graph structure as well as product information as node characteristics. Through modifications to sampling involved in the network optimization process, a common phenomenon in recommendation systems, the so-called popularity bias, has been addressed: popular products are often recommended, while less popular ones are neglected and rarely or not recommended at all. PharmaSage was developed using real-world sales data and trained with 700,000 articles represented as nodes in a cross-node chart representing approximately 100 million sales transactions. By utilizing the properties of medicinal products, such as indications, ingredients and their side effects and combining them with long sales histories, better results have been achieved than with a purely statistical approach. This is the first application of deep embedding charts for intersection of pharmaceutical products on this scale to date.

His work [35] described the main challenges in implementing the idea of a system of recommendations based on convergent graph networks in cross-selling recommendations of pharmaceutical products, given the aforementioned limitations. The first challenge is to devise a system of recommendations based on product feature information and sales data that represents information about which products have been sold together while providing no information on product-customer relationships. To address this challenge, a graph of drug sales and product data is created and a neural graph is used for training on product integrations, which can be used for a downstream recommendation work. Describes how this infor-

mation was chosen to be encoded in a graph structure, representing drugs as nodes, their characteristics as node characteristics, and sales statistics as node edges.

The goal of optimizing the neural network training process in relation to this data is further described. It was chosen to use a semi-supervised training program using the loss of trinity to optimize network parameters. Defining positive and negative samples is a key decision when using triple loss in optimization. The following is a description of how the positive and negative samples were selected based on cross-selling information and characteristics as shown in the graph.

Cross-selling Recommendations of Pharmaceutical Products

In their study, [59] interviewed sixteen pharmacists and found that pharmacists relied primarily on personal judgment augmented by feedback from patient feedback to make product recommendations. Another study that examined the factors that influence pharmacists' recommendations for complementary medicines also stated that the recommendations are based on personal experience and training and concluded that in order to encourage documented use of complementary medicines in pharmacies, there is a need to develop accessible quality resources. A study examining community pharmacists' recommendations for alternative natural products for stress in Melbourne found that out of 94 pharmacies, twenty-five provided the customer with an unsuitable product and concluded that there was a need to develop guidelines for pharmacists for taking evidence-based decisions recommending complementary and alternative medicines [23]. A study investigating a recently released drug recommendation system has developed a side effect labelling component that emits, for a given prescription, a set of drugs that should be avoided as they will cause side effects if taken at the same time as the prescription [8]. [84] proposed a system of indications for the use of cloud-based drugs for online pharmacies, in order to suggest to users the top N drugs according to the symptoms. In their system, they first grouped the drugs into several groups according to the functional description information and designed a basic individualized drug composition based on collaborative user filtering. In collaborative filtering based on their pipeline segment, the authors

leveraged drug user ratings. In their study [35] there was no access to historical data or user ratings. The modelling was based solely on raw sales data and product features. Therefore, a unique approach has been devised to solve the drug-product recommendation task based on node integrations created by a graphical neural network that can utilize sales data as well as product features without the need for customer history or ratings of users.

Graph of Neural Networks and Recommendation Systems

Conceptually, the approach of [35] relates to previous node integration algorithms and current developments in the application of convergent neural networks to graph-structured data. The basic idea behind node integration methods is to create useful small-dimensional vector representations of high-dimensional information for a node in the graph, including the node graph neighbourhood. The use of low-dimensional vector integrations as feature inputs for a wide variety of machine learning tasks, such as sorting, predicting, grouping, and setting tasks, has proven valuable [7]. His original work [15], which introduced a convergent graph variable based on spectral graph theory, was followed by several authors, proposing improvements and extensions [7]. The first approaches to creating node integrations, such as the GCN introduced by [40], were transferable and limited in their generalization to invisible nodes and their scalability, as these methods required to work in the whole of Laplace. of the graph [80]. Subsequent approaches worked inductively [51]. In contrast to inductive integration methods based on matrix factorization, newer approaches utilize node characteristics and the local neighbourhood spatial structure of each node, as well as the distribution of node characteristics in the neighbourhood [33]. Based on this, further improvements and new algorithmic capabilities that ensure performance, scalability and improved sampling have recently been explored [75]. These developments have led to new improved performance at benchmarks, such as node classification, link prediction, or web-scale proposal work, as well as the application of these methods to areas such as drug design [14].

Prejudice of Popularity

One obstacle to the effectiveness of recommendation systems is the problem of popularity bias [6]. Collaborative filtering recommendations usually focus on popular articles (e.g. those with more sales, views, or ratings) rather than articles [49] that may be popular with only a small group of customers or consumers. While popular articles can be a good recommendation, they are also likely to be well known and sometimes bad recommendations, especially in the face of cross-selling drug recommendations. In addition, delivering only popular articles will not accelerate the discovery of newly introduced articles and ignore good suggestions. The idea of the popularity of the article and its impact on the quality of the recommendations has often been the subject of investigation [81]. In these projects, the authors sought to improve the performance of the recommendation system in terms of accuracy, while others focused on reducing the bias of popularity through normalization [2] or reclassification. Significant research has also been published on the diversity of recommendations, with the aim of preventing the introduction of too many similar articles.

Representation of Data in Graph

In order to harness the power of convergent neural networks in order to learn product integration, pharmacy sales data and product information must be converted into a graph. Each unique pharmacy product is represented as a node, which also contains the descriptive information of the corresponding product coded into one, which in turn has approximately 15,000 entries, representing the medical indications, active ingredients and side effects of the products. The data was obtained from the main German commercial pharmaceutical database used in pharmaceutical and other software solutions for pharmacies, doctors, hospitals and other healthcare providers, provided by Pharmatechnik. The undirected weighted edges between two nodes represent the frequency of cross-selling for each product pair, where "cross-selling" is defined as the sale of two products sold in the same transaction. The total of approximately 100 million transactions with information about the products sold together was provided by Pharmatechnik, utilizing the sales transactions documented through the IXOS Pharmacy Management System, which is currently used in more than 5,000 pharmacies. Prior to further processing, transactions are restricted by those on which the cross-selling numbers are based on those with two

or three items sold, corresponding to approximately 25% of the total. The fewer products involved in a transaction, the greater the relationship between the products sold. These transactions involve about 700,000 different products, but also include many similar products that differ little (or not at all) from each other. For example, there are many Aspirin 100 product offerings from different manufacturers. For training, validation and testing, 60% of these default trades were randomly selected to create a training chart and 20% to generate validation and testing charts. These graphs serve as an entry into the model training and validation phase. However, the final model used for conclusions is then trained in all the sales transactions included. In model training, the highest-scoring cross-section articles (those with the highest acne weights with a specific query) were used as candidates for the concentration and optimization part of the modelling process. Unfortunately, popular articles dominate. To compensate for this so-called popularity bias, a probability theory-based approach is introduced that aims to update the weights of all the ends of the graphs.

Model Architecture

Suppose a model of a convergent graph neural network that uses local convergences in aggregate neighbourhood vectors to create product integrations represented by graph nodes, similar to that presented in [33]. The basic idea is to transform the representations of the neighbours of a given node through a dense neural network and then to implement an aggregation / concentration function (a weighted sum, presented in the form of blue boxes in the following figure, the "CONVOLVE" unit) in the resulting set of vectors. This assembly step provides a vector representation of the local neighbourhood of a node. This aggregate neighbourhood vector (dark grey frame) is then combined with the current representation of the nodes (light grey frame) and the connected vector is transformed through another dense neural network. The output of the algorithm is a representation of a node, called node integration, which integrates information about itself and the neighbourhood of the local graph. Details about the algorithm exist in their work [80]. The only change made is that the neighbourhood node information was collected only in the "top neighbours" of a given node. The upper bouts featured two cutaways, for easier access to the higher frets. The lower bouts featured two cutaways, for easier access

to the higher frets. Recommendations are then calculated using these final node integrations

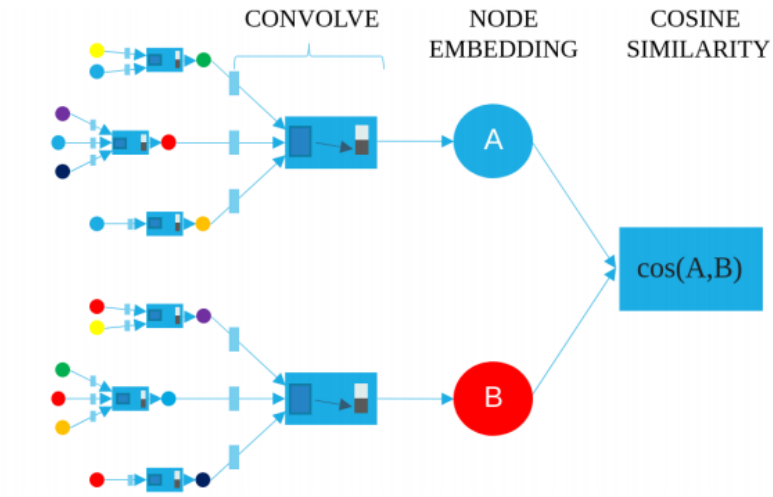

Figure 2.9: The convergent neural network graph uses local convergences in aggregate neighbourhood vectors to learn product integrations. Here, the two-level graph neural network is presented that calculates the final integrations of nodes A and B using the previous level representation of nodes A and B, respectively, and that of their respective neighbourhoods. Different colours indicate different nodes. The proposed score between two products A, B is then calculated by using the cosine similarity between the two final integration vectors of nodes A and B.

Model Training

To optimize the network parameters, the trinity loss is used, as shown in the equation $W_{residual}(B, A) = W_{raw}(B, A) - E(B|A)$, which is a distance-based loss function that operates on the final integration of three input nodes: the central A, the positive P, which are typically from the same order as A or related to some other measure, and the negative N, which is typically of a different order from A.

$$\min_{\theta} L(A, P, N)$$

$$L(A, P, N) = \max(0, sim(A, N) - sim(A, P) + \Delta)$$

The incorporations of pharmacy products represented by nodes in the graph were used as inputs for trinity loss. Each node can be a base node A, and each has positively correlated nodes P and negatives N. A given pair A, P must be associated with some measure, i.e., they are often purchased together or are similar in attribute coding. Pair A, N is believed to be related to some other measure, i.e., they are never bought together or bought together less than expected. The aim of the training phase is to optimize the PharmaSage parameters so that the sim (A, P) resemblance of the base-positive pair is higher than the residual sim (A, N) of the base-negative pair by some margin 0.5.

Sampling

Positive samples were selected for a given key node A by sampling nodes between the top neighbors of that node with equal probability in each iteration. The leading cross-sellers are those neighbors with the highest edge weight with the main, who represent the products that are sold more often than expected in relation to this product represented by the main node. Together, they form the positive sampling tank. To complete the positive sampling based on edge weight, nodes are used that share coding characteristics with the basic one as possible positive samples. In this hybrid positive sampling approach, 50% of the positive samples were selected based on the similarity of key node characteristics with respect to all other nodes and the other 50% based on edge weights representing reclassified cross-sales. Products are included in the additional positive sampling based on additional attributes for a given key node, if the product / node shares any attribute with the key.

Negative samples were selected between nodes that are not connected to the main one, which are never sold with it. Additional nodes with negative acne weights relative to the baseline were used as possible negative samples, as the expected amount of cross-selling with the base node is

higher than the actual cross-selling. In general, semi-hard negative potential mining was applied, as introduced on FaceNet [64], which has been widely used since then [69].

Recommendation

The final embodiments calculated from the model presented by them [35] after the training are used to calculate the scores of the proposal recommendations among all the products. We obtained a recommendation score defined within the interval [0, 1] using the cosine similarity of two vectors, sim (A, B), commonly used in the information retrieval and data mining technique (Singhal, 2001). Given the similarity scores between a search product and all other products (except prescription drugs, which can only be sold if they have been prescribed by a doctor and are therefore not legitimate cross-selling recommendations), the products with the highest similarity to search. as recommendations for the search article. A diagram illustrating the high-level architecture for how to calculate the recommendations between Articles A and B is shown in the previous figure.

System Setup and Execution Time

For the software, the Python framework, the networkx library were used to create the chart, and the PyTorch framework was used to implement the GCN. The network is trained in system with Intel Core i7 8750H, Nvidia GeForce GTX1070 GPU and 32 GB RAM. The training lasts about 48 hours and ends as soon as the loss of a trinity, which starts at the margin of 0.5, has reached a limit of 0.05 for the training chart. The validation loss is at 0.092 and the test loss at 0.095 at this point. It has been found empirically that the quality of results does not improve much beyond this point. The training of the model used for conclusions stopped at the same threshold.

Quality of Recommendation

In order to evaluate the performance of the algorithm, a collection of product reviews was provided by experts on a total of 25 product searches. For each recommendation generated by PharmaSage, the feedback given may have one of three values: 0 (no pharmaceutical relationship with the search product), 1 (there is a drug relationship and the product is a good recommendation for the search) or 2 (there is a drug relationship and the product is a very good recommendation to look for). The average of the scores for the top 15 recommendations for 25 products is then used as a performance quantifier.

Recommendations are calculated based on cross-selling data encoded in the graph, which can be thought of as a conventional propositional approach similar to collaborative filtering. The products that are most often sold with the search product are selected as recommendations. The recommendations are then calculated based on the PBR graph. This result provides an insight into what can be achieved without additional learning. The PBR graph is then used to train the PharmaSage model.

As shown in the following figure, the quality of the evaluated recommendations is comparatively lower for simple recommendations based on cross-sales statistics and the quality increases in the PBR approach. The PharmaSage model then trained for the PBR approach introduces another increase in recommendation quality. Compared to sales data-based approaches, this model is able to learn from cross-selling information encoded at the edges of the chart, as well as leverage characteristics information encoded as node characteristics.

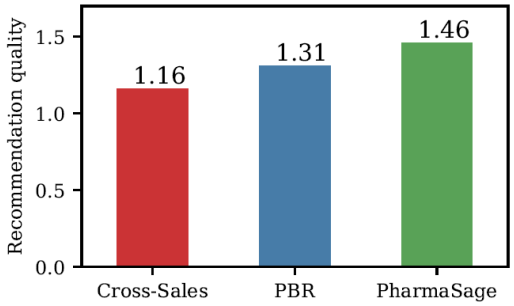

Figure 2.10: Average quality recommendations among the top 15 suggested articles for 25 rated products. Recommendations are calculated based on the graph including crude cross-selling statistics and the graph, where edge weights have been recalculated using the probability approximation (PBR). PharmaSage is optimized based on the PBR approach as input.

The following table shows examples of recommendations calculated by PharmaSage, their recommendation score, and expert feedback on prednisolone, a corticosteroid. Prednisolone is used to treat a wide range of health problems, such as allergies, skin conditions, infections and certain autoimmune disorders. Helps reduce inflammation and suppress the immune system. PharmaSage also recommends over-the-counter allergy medications (ranking 1, 5, Table 2) for which prednisolone and stomach acid inhibitors are often prescribed to reduce the negative effects of the active ingredient in the search article (ranking 2, 4, Table 2). Calcium and vitamin D3 can both help improve bone health, which can be affected by ongoing treatment with corticosteroids (rankings 6, 9, Table 2). The non-steroidal anti-inflammatory drug helps with pain and has additional anti-inflammatory action (ranking 8, Table 2). Another antipruritic and anti-inflammatory gel (ranking 10, Table 2) is recommended, which compensates for the symptoms associated with the main indications of prednisolone, as well as a nasal spray for additional treatment, which relieves a closed nose, reduces swelling and can cause swelling. to assist in the treatment of allergic airway reactions (ranking 3, Table 2). ASS100 (ranking 7, Table 2), an anticoagulant, should not be recommended without additional medical advice. Therefore, the expert feedback signal is 0.

Ranking	Recommended Product	Expert Comments
1	CETIRIZIN AL DIREKT	2
2	PANTOPRAZOL ABZ	2
3	PANTOPRAZOL ABZ	2
4	OMEPRADEX 20MG	2
5	LORANO AKUT	2
6	DEKRISTOL 400 IE	2
7	ASS AL 100 TAH	0
8	IBUPROFEN OPT 400MG	2
9	CALCIUM D3 RATIO	2
10	FENISTIL	2

Table 2. Examples of recommendations for prednisolone

In their work [35] they presented PharmaSage, a continuous network of charts for cross-selling recommendations for pharmaceutical products. PharmaSage is the first graphical synergistic neural network application for drug composition, utilizing sales statistics and drug characteristics such as indications, ingredients, and contraindications. In addition, a method based on probability theory presented that addresses the common problem of popularity bias was presented. It has been shown that there is a bias of popularity in pharmacy sales and cross-selling data sets and ways have been developed on how it can be successfully addressed in order to increase both the quality of the recommendations and the diversity. PharmaSage was developed on the basis of real pharmaceutical data and thoroughly evaluated the quality of learning incorporations for a cross-selling recommendation task, demonstrating a significant improvement in the quality of proposals compared to traditional cross-statistical approach-based approaches. This demonstrates the positive impact that graphical aggregation-based methods can have on pharmacy cross-selling recommendation systems, and reflects the belief that PharmaSage can be further expanded to address other learning problems of graph representation and retail in the retail industry. A future point of interest will be the further evaluation of the quality of PharmaSage recommendations by evaluating A / B tests against traditional recommendations by pharmacists and the way in which both cross-selling and customer feedback are affected.

Chapter 3

Matrix Factorizations with Neural Neworks (SVD, MLP and NCF algorithms)

Synopsis: In this chapter, we will focus on Multi-Layer Perceptron(MLP), Generalized Matrix Factorization(GMF) and Neural Collaborative Filtering(NCF) Framework for probabilistic models. We will introduce Neural Networks and describe the MLP and GMF models and their formulas, in order to show how they work with auxiliary data. Furthermore, after a brief introduction to NCF we will elaborate on the use of the NCF Framework with probabilistic models on a theoretical level, before actually implementing them in our experiments in Chapter 4. Finally, we will explain the Singular Value Decomposition(SVD) and Post Hoc Re-Ranking algorithms and present a Toy Example for a patient using the best performing algorithm (SVD) as shown in Chapter 4 and compare the recommended drugs with SVD Reranking to conclude which results are safer for actual use.

3.1 Problem Analysis

Our main focus is analysing the performances of the Multi-Layer Perceptron, Generalized Matrix Factorization and Singular Value Decomposition Learning models from the Cornac library. In particular, our goal is to check whether these algorithms can be applied to solve the intricate

challenges presented by the healthcare environment.

All the above-mentioned methods can deal with a main user-item interactions matrix and support a potential auxiliary network for the items. In our specific scenario, the drugs serve as our items and the patients serve as users. In addition, a list of adversarial drug-drug interactions will be used as side information. The results achieved by the models with and without auxiliary data will be presented in the next Chapter.

Other than simply examining the efficiency and effectiveness of the Cornac models, we have also implemented an extended version of the Singular Value Decomposition algorithm in order to see how the results would change by adding supplementary side information, by implementing the Post Hoc Re-Ranking algorithm. In particular, we tried to feed the latter method with the drug-drug interactions auxiliary networks. Different experiments and outcomes will also be shown and compared in the next Chapter.

3.2 Multi-Layer Perceptron MLP

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. In MLP-based prediction [73], the input data is the history observations, while the output is the prediction of the future states.

Since two paths are adopted for modeling users and objects, it is considered intuitive to combine the features of the two paths. However, a single vector concatenation cannot account for any interaction between user and object latent function. To address this issue, the authors [34] proposed adding hidden layers to the concatenated vector, using a standard MLP. In this way, the interaction between the user-object latent function can be established and, by extension, the model can be given a great level of flexibility and non-linearity. More specifically, the MLP model is defined, as:

$$z_1 = \phi_1(\boldsymbol{p}_u, \boldsymbol{q}_i) = \begin{bmatrix} \boldsymbol{p}_u \\ \boldsymbol{q}_i \end{bmatrix},$$

$$\phi_2(z_1) = \alpha_2(W_2^T z_1 + b_2),$$

.....

$$\phi_L(z_{L-1}) = \alpha_L(W_L^T z_{L-1} + b_L),$$

$$\hat{y}_{ui} = \sigma(h^T \phi_L(z_{L-1}))$$

where W_x, b_x and α_x denote the weight matrix, bias vector, and activation function for the x-th layer perception, respectively. Finally, regarding the activation functions for the MLP layers, the user can freely choose between the sigmoid, the hyperbolic tangent (tanh), and the rectifier (ReLU) (among others).

[4] in their textbook provided a simple example of neural networks being used as black-boxes. Neural networks simulate the human brain with the use of neurons, which are connected to one another via synaptic connections. In biological systems, learning is performed by changing the strength of synaptic connections in response to external stimuli. In artificial neural networks, the basic computation unit is also referred to as a neuron, and the strengths of the synaptic connections correspond to weights. These weights define the parameters used by the learning algorithm. The most basic architecture of the neural network is the perceptron, which contains a set of input nodes and an output node. An example of a perceptron is shown in Figure 3.1. For a data set containing d different dimensions, there are d different input units. The output node is associated with a set of weights W, which is used to compute a function $f(\cdot)$ of the d inputs. A typical example of such a function is the signed linear function, which would work well for binary output:

$$z_i = sign\{\bar{W} \cdot \bar{X}_i + b\}$$

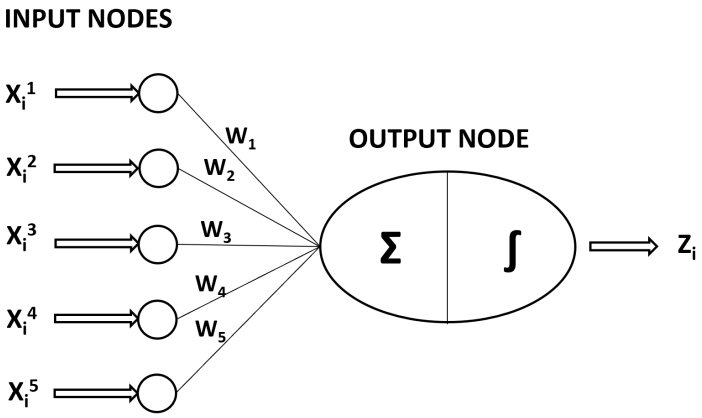

Figure 3.1: Perceptron.

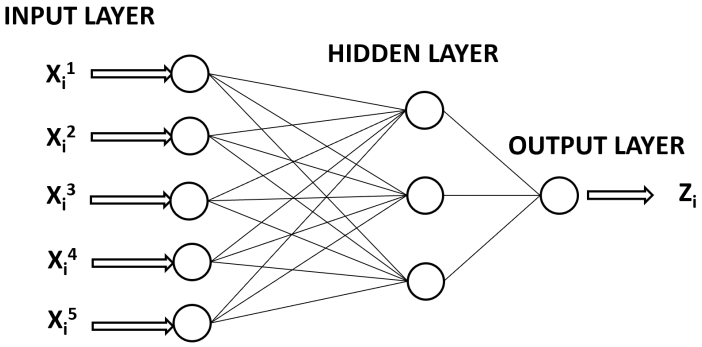

Figure 3.2: Multilayer Neural Network.

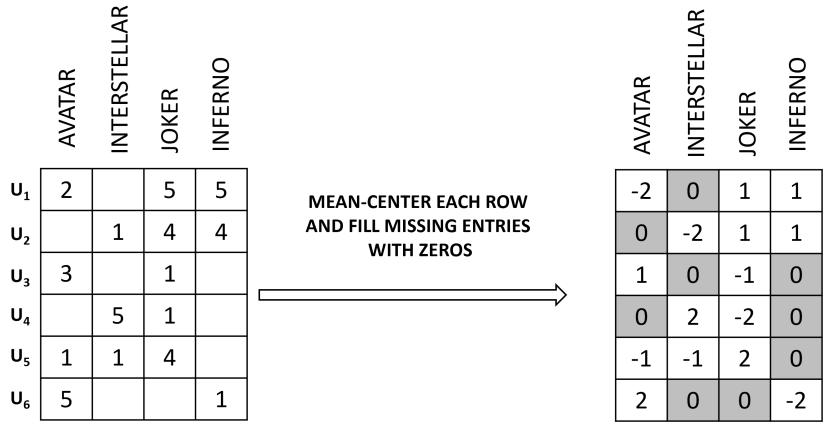

Figure 3.3: Pre-processing the ratings matrix. Shaded entries are iteratively updated.

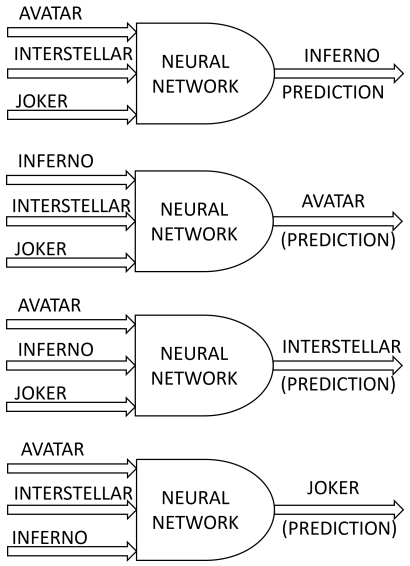

Figure 3.4: Neural networks for predicting and updating missing entries. Shaded entries of Figure 3.3 are iteratively updated by the neural networks.

Here, $\bar{X}_i$ is a d-dimensional row vector defining the d inputs of the ith training instance, and $\bar{W}$ is the coefficient vector. In the context of collab-

orative filtering, the d inputs correspond to the $(n-1)$ items, which are used to predict the rating of the remaining item. Assume that the label of the ith instance is y_i. In the context of collaborative filtering, y_i represents the observed ratings of the items being predicted. The parameter β denotes the bias. One can already notice the similarity of this approach with linear regression although the prediction function is slightly different. The value of z_i is the predicted output, and the error $(z_i y_i)^2$ of this predicted output is used to update the weights in $\bar{W}$ in a manner similar to linear regression. This update is similar to the updates in gradient descent, which are made for least-squares optimization. In the case of neural networks, the update function is as follows :

$$\bar{W}^{t+1} = \bar{W}^t + \alpha(y_i - z_i)\bar{X}_i$$

Here, $\alpha > 0$ denotes the learning rate and Wt is the value of the weight vector in the tth iteration. It is not difficult to show that the incremental update vector, is the negative gradient of $(y_i z_i)^2$ with respect to . We iterate through all the observed ratings in the item being predicted in order to make these updates. Since it was assumed that y_i is binary, this approach is designed for binary ratings matrices. One can also design neural networks in which the output need not be binary, and the prediction function need not be linear.

In general, a neural network can have multiple layers, and the intermediate nodes can compute nonlinear functions. An example of such a multilayer neural network is illustrated in Figure 3.2. Of course, such a network would have a larger number of learning parameters. The corresponding learning algorithm is referred to as the back-propagation algorithm [9]. The main advantage of neural networks is that the multi-layer architecture provides the ability to compute complex nonlinear functions that are not easily computable with other classification methods. Therefore, neural networks are also referred to as universal function approximators. For noisy data like ratings matrices, regularization can be used to reduce the impact of noise.

Consider a ratings matrix with four items, illustrated in Figure 3.3. In this example, the items correspond to movies. The first step is to mean-center each row, in order to remove user biases. The resulting mean-

centered matrix is shown on the right-hand side of Figure 3.3. Note that the missing values are replaced with the corresponding row average, which is 0 after mean-centering. Since there are four items, there are four possible neural network models, whereby each model is constructed by using the ratings input of the other three items as training columns, and the fourth as the test column. These four neural networks are shown in Figure 3.4. The completed matrix of Figure 3.3 is used to train each of these neural networks in the first iteration. For each column of the ratings matrix, the relevant neural network in Figure 3.4 is used for prediction purposes. The resulting predictions made by the neural networks are then used to create a new matrix in which the missing entries are updated with the predicted values. In other words, the neural networks are used only to update the values in the shaded entries of Figure 3.3 with the use of an off-the-shelf neural network training and prediction procedure. After the update, the shaded entries of Figure 3.3 will no longer be zeros. This matrix is now used to predict the entries for the next iteration. This approach is repeated iteratively until convergence. Note that each iteration requires the application of n training procedures, where n is the number of items. However, one does not need to learn the parameters of the neural networks from scratch in each iteration. The parameters from the previous iteration can be used as a good starting point. It is important to use regularization because of the high dimensionality of the underlying data [29].

This model can be considered an item-wise model, in which the inputs represent the ratings of various items. It is also possible to create a user-wise model [86], in which the inputs correspond to the ratings of various users. The main challenge with such an approach is that the number of inputs to the neural network becomes very large. Therefore, it is recommended in [86] that not all users should be used as input nodes. Rather, only users who have rated at least a minimum threshold number of items are used. Furthermore, the users should not all be highly similar to one another. Therefore, heuristics are proposed in [86] to preselect mutually diverse users in the initial phase. This approach can be considered a type of feature selection for neural networks, and it can also be used in the item-wise model.

3.3 Neural Collaborative Filtering

In today's information age, the proposed systems play a very decisive role in the 'relief' of information, from possible overloads. This has been helped by the adoption of a number of online services, including e-commerce, online news and social networking sites. According to the authors [63], the key to a personalized recommender system is modeling user preferences. Preferences can be identified based on previous user interactions (e.g. ratings and website browsing), and this system is known as Collaborative filtering. Among the various techniques encountered in collaborative filtering, matrix factorization (MF) is the most popular. This has the ability to project users and objects into a shared latent space using a latent feature vector. So the vector is meant to represent a user or an object. A user's interaction with an object is then modeled as the inner product of its latent vectors.

Having gained great popularity through the Netflix Prize competition, MF is considered the de facto approach for recommendations based on a latent factor model. Additionally, a great deal of research effort has been made to improve MF, such as through its integration with neighbor-based models, its combination with thematic content (object) models, but also through its extension in factoring machines (for a general modeling of features). However, despite its effectiveness, the performance of MF can be hampered by the simple choice of an interaction function (IFC). For example, in terms of rating prediction (in explicit feedback), the performance of the MF model can be improved by incorporating the user and item bias terms into the interaction function. Despite its apparent simplicity, this integration highlights the positive effect of designing a better, dedicated interaction function. Its goal is to model the interactions (the latent function) between users and objects. Also, the inner product, which linearly combines the multiplication of latent features, may not be sufficient to capture the complex structure of user interaction data.

The authors [34] carried out a thorough investigation, which focused on neural collaborative filtering. First, they presented the general NCF (Neural network-based Collaborative Filtering) framework, elaborating how the framework is learned, with a probabilistic model. In this case, the

model emphasizes the binary nature of implicit data. The authors then demonstrated how MF can be expressed and generalized with NCF. Then, in order to explore deep neural networks (DNNs) for collaborative filtering, they proposed a detailed representation of NCF, using a multi-layer perceptron (MLP). MLP was an important part of the research because it aimed to learn the user-object interaction function. Finally, [34] presented a new factorization model (neural matrix), which combines MF and MLP, in the context of NCF. In this way, the new model has the ability to unify both the advantages of linearity of MF and the advantages of nonlinearity of MLP. Thus, it is considered suitable for modeling the latent structures between user-object.

In order to enable a complete, neural processing of collaborative filtering, the authors [34] adopted a multi-layer representation. Their goal was to model a y_{ui} for user-object interaction, in which the output of one layer serves as the input of the next. The bottom layer of the input consists of two feature vectors (v_u^U and v_i^I), which respectively describe user u and object i. In addition, they can be adapted to provide support for a wide range of user and component modeling (e.g. context-aware), content-based, as well as neighborhood-based. Since this research focuses on setting up a 'pure' collaborative filtering, the authors used the identity of a single user (and an object) as input. Thus, it was possible to convert them into a binary, sparse vector (with single beam encoding). Also, by using such a generic feature representation (for inputs), the method in question can be easily adapted to deal with the cold start problem. It can achieve this by using content functions to represent users and objects.

Continuing with the framework description, above the input layer is the embedding layer. The embedding layer is a fully connected layer, which projects the sparse representation into a dense vector. In this, the obtained embedding of the user (object) can be considered as the latent vector for the user (object), in the context of the model. Both user and object integration are then fed into a multi-layered, neural architecture called "neural collaborative filtering layers". These are used to map the latent vectors to prediction scores. Also, each layer of the CF neural layers can be adapted to discover certain latent structures of user-object interactions. More specifically, the dimension of the last hidden layer X, determines the

model's ability, while the final output level is the predicted score $\hat{y}_{ui}$. In this case, training is performed by minimizing the point loss between $\hat{y}_{ui}$ and the target value y_{ui}. At this point, it is worth mentioning that there is another way of training the model and more specifically, performing pairwise learning. In it, Bayesian Personalized Ranking [57] and margin-based loss [68] are used. Returning to the proposed method and performing the aforementioned steps, the authors [34] formulated the prediction model of NCF as

$$\hat{y}_{ui} = f(P^T v_u^U, Q^t v_i^I | P, Q, \Theta_f)$$

where the $\boldsymbol{P} \in \mathbb{R}^{M \times K}$ and $\boldsymbol{Q} \in \mathbb{R}^{N \times K}$, denote the factor latent matrix for users and objects, respectively, while Θ_f denotes the model parameters for the interaction function f. Since the function f is defined as a multi-layer neural network, it can be expressed as

$$f(P^T v_u^U, Q^t v_i^I) = \phi_{out}\Big(\phi_x\Big(\ldots \phi_2\Big(\phi_1(P^T v_u^U, Q^t v_i^I)\Big)\ldots\Big)\Big)$$

where ϕ_{out} and ϕ_x denote the matching function for the output layer and the x layer of the neural, collaborative filtering (CF) respectively. Still, there are a total of X CF neuronal layers.

Additionally, in order to better understand the model parameters, the authors' existing point methods [76], perform (largely) a regression with the following squared loss

$$\mathbb{L}_{sqr} = \sum_{(u,i) \in \dagger \cup y^-} w_{ui}(y_{ui} - \hat{y}_{ui})^2$$

Where † denotes the set of observed interactions in $\mathbb{Y}$ and $\dagger^-$ denotes the set of negative interactions. The w_{ui} is a hyperparameter denoting the weight of the training instance (u,i). On the other hand, although quadratic loss can be explained according to a Gaussian distribution [62], the authors [34] point out that it may not adequately correspond to implicit data. This is because for the indirect data, the value of the target y_{ui} is a binary 1 or 0, indicating whether u has interacted with i. Then, they present a probabilistic approach to learning the point NCF, which pays special attention to the binary property of the implicit data.

Furthermore, considering the nature of a category, the included feedback, the value of y_{ui} is set to label -1. This means that item i is differently related to u and 0. In this case, the prediction score $\hat{y}_{ui}$ represents the probability of the association. Still, to be able to give such a probabilistic explanation to the NCF, the authors first had to limit the output $\hat{y}_{ui}$, to the target of [0,1]. This can be easily achieved by using a probabilistic function (e.g. the Logistic or Probit function) as an activation function, for the output level ϕ_{out}. Therefore, according to the above settings as well, [34] defined the likelihood function as

$$p(y, y^{-}|\boldsymbol{P}, \boldsymbol{Q}, \Theta_f) = \prod_{(u,i)\in y} \hat{y}_{ui} \prod_{(u,i)\in \cup y^{-}} (1 - \hat{y}_{uj})$$

Subtracting the negative logarithm of the probability gives the following equation

$$\mathbb{L} = - \sum_{(u,i)\in \dagger \cup y^{-}} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})$$

This is the objective function to be minimized, for NCF methods. Optimizing this function can be done by performing Stochastic gradient descent (SGD) and is the same as binary cross-entropy loss (also known as log loss). Finally, as for the negative y^{-}, cases, they are uniformly sampled from unobserved interactions (in each iteration) and the proportion of sampling w.r.t., for the number of observed interactions is checked.

3.4 Generalized Matrix Factorization - GMF

Here, MF can be interpreted as a special case of the NCF framework. Because MF is the most popular model for factorization, its retrievability allows NCF to emulate a large family of factorization models. Also, due to the unique encoding of the user ID (for the input layer), the resulting embedding vector can be considered as the latent vector of the user (object). That is, let the user latent vector $\boldsymbol{p}_u$ be $\boldsymbol{P}^T \boldsymbol{v}_u^U$ and the object latent vector $\boldsymbol{q}_i$ be $\boldsymbol{Q}^T \boldsymbol{v}_i^I$. In this case, the mapping function of the first CF neural layer is defined as

$$\phi_1(\boldsymbol{p}_u, \boldsymbol{q}_i) = \boldsymbol{p}_u \odot \boldsymbol{q}_i$$

where $\odot$ denotes the elementary product of the vectors. The vector is then projected onto the output layer, as

$$\hat{y}_{ui} = \alpha_{out}(\boldsymbol{h}^T(\boldsymbol{p}_u \odot \boldsymbol{q}_i))$$

where α_{out} and $\boldsymbol{h}$ denote respectively the activation function and edge weights of the output layer. Intuitively, if an identity function is used for α_{out} and h ϵ is imposed as a uniform vector of 1, the MF model can be accurately recovered.

3.5 Singular Value Decomposition (SVD)

According to linear algebra, a register of X dimensions $n \times m$ is decoupled through the relation : $X = U\Lambda V^T$, where U is a register of dimensions $n \times m$ with orthogonal columns $(U^T U = I)$, V is an orthonormal table $(V^T V = I)$ of dimensions $m \times m$ and Λ is a diagonal array of dimensions $m \times m$ with positive or zero elements called eigenvalues. These values are the non-zero eigenvalues of table XX^T and are the same as those of table $X^T X$.

Firstly and based on the above, the SVD method calculates the λ values of the covariance array $C = \frac{1}{n}X^T X$ and then creates the diagonal array S. The s_i elements of the square roots of the non-zero eigenvalues λ are arranged in a descending order. Having calculated table S, the method creates table V which contains the orthogonal eigenvectors corresponding to the e values of S, which are also sorted in a descending order. After that, it calculates the table U. The first N columns are given by the relation $u_i = s_i^{-1} Xvi$ and the remaining $M - N$ by using the Gram-Schmidt orthogonalization method.

In this way, the initial table of input data X is analyzed in the form: $X = USV^T$ and the corresponding covariance table becomes $C = \frac{1}{n}X^T X = \frac{1}{n}US^2V^T$. According to the SVD method it is known that the first columns of table U correspond to the ordered non-zero eigenvalues of C. Thus, the principal components are now given by the transformation : $Y = U^T X = UUSV^T$ with U indicating a simple array of aces on the diagonal and zero values anywhere else.

The SVD algorithm accepts an array M of dimensions $m \times m$ as input and finds arrays $U(m \times m), S(m \times k), V(k \times m)$, such that $= USV^T$. U array is a unitary, i.e. $U - 1 = U$, S is an orthogonal diagonal array and V is also orthomonical.

So based on the theory of SVD, in this thesis we use the algorithm as follows:

$$X = USV^T, \text{ where:}$$

- U: is the left singular matrix, (Patients)
- S: is a diagonal matrix containing singular eigenvalues
- V: is right singular matrix (Drugs)

The element $X(i, j)$ in a commuting matrix X denotes the number of path instances from the patient p_i to drug d_j, then row i in the commuting matrix can be used as features of patient p_i, and column j can be used as features of drug d_j.

Our proposed Post Hoc Re-Ranking Method

Until now, algorithms were trained on data with patient-drug relationships. So the algorithms were learning to recommend new drugs to a patient who is already taking certain drugs based on other patients with a similar profile. However, in this context, the additional information that some drugs interact with other drugs (side effects) is not taken into account. Therefore, to address this problem, we will use the Post Hoc Rerank algorithm to incorporate drug interaction information into the recommendation scores to improve safety. This algorithm is defined as:

$$drugScore = \begin{cases} modelScore/DF, & \text{if } modelScore > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

where *modelScore* is the score given by the model to a specific drug. It is the value that predicts the possible patient-drug interaction. In particular, when the SVD method is applied, the empty values of the two-dimensional matrix that holds patient and drug interactions are filled. DF is the frequency with which a drug occurs in the set of unwanted drug-drug interactions.

3.6 A Toy Example to well-motivate our Proposed Method

Finally, we will show an example for better understanding.
User 1 is taking 12 medicines:
[690, 106, 753, 131, 137, 71, 656, 589, 192, 1000, 135, 543]
Using the SVD algorithm, we find that based on the patient-drug table, the 5 best drugs are :
[460, 690, 192, 52, 574]
The suggested scores of these drugs are :
[0.548, 0.478, 0.348, 0.261, 0.224]
However, there is a side effect between drugs 52 and 574. If we use the Post Hoc Re-Rank, then the top 5 drugs are:
[690, 102, 460, 574, 131]
There is no interaction between these drugs. The suggested scores of these drugs are:
[155.88, 9.02, 0.69, 0.52 0.43]

From the example above we can conclude that, the addition of the re-ranking mechanism allows SVD to drop its DDI Rate to a minimum, meaning that the recommendations produced by our model include mostly drugs that can be combined to generate safe treatments and nurse various diseases without generating side effects. So, in terms of accuracy, the two recommendations are equivalent, as they include two drugs from the list (690, 192 from the first list and 690, 131 from the second list). We observe that there is no interaction between the drugs suggested after re-ranking. Therefore, the second recommendation after re-ranking is more suitable for actual use.

Chapter 4

Evaluation Protocol

The link for the code in GitHub is https://github.com/OathKeeperGreg/RecSys.

4.1 Data Set Description

Synopsis: In this chapter, we will showcase our experiments for all the models (SVD, Item-KNN, User-KNN, NMF, MLP, GMF) and their results. First, we present an overview of the MIMIC-III database to describe the relationships between patients and drugs. Before the experiments, we used Grid Search to optimise the parameters for each model. We then present the evaluation metrics we used to choose the model with the best balance between performance and safety in drug recommendations. Finally, we will show why the Post Hoc Re-Ranking algorithm for our best performing model and the precision tradeoff is important to achieve much safer results for actual use.

Patient-Drug Relationships

From the MIMIC-III database file, we select the table, which lists the patient-drug relationship (Dbo_PATIENT-DRUGS_COUNT). It also has a third column that gives the frequency of use of the drug, which will be used as an evaluation by the referral system. In total we have 155661 relationships between 2351 users and 2987 drugs. Then we divide the data into 80% training set and 20% test set. The models will be created based on

the training data, but they will be evaluated on the test data, which have not been used in the construction of the model, so the results will be more objective.

4.2 Prerequisites

We used Python 3.7 programming language and the Cornac library [61] [72], which provides various recommendation algorithms. Also, in order to be able to execute algorithms with neural networks, we installed the package Tensorflow 1.15 and tensorflow-estimator 1.15. In more detail, we installed anaconda which provides many ready-made packages and created a virtual environment as follows:

- Conda create -n py37 python=3.7
- Conda activate py37
- Conda install cornac -c cornac-forge
- Conda install sklearn
- Conda install tensforflow==1.15
- Pip install tensorflow-estimator==1.15

Algorithms

We tested the following recommendation algorithms:

- **SVD** (Singular Value Decomposition)[11]
- **ItemKNN** (Item K-Nearest Neighbour) [12]
- **UserKNN** (User K-Nearest Neighbour) [63]
- **NMF** (Non-negative Matrix Factorization) [42]
- **MLP** (Multi-layer Perceptron neural network) [34]
- **GMF** (Generalized Matrix Factorization) [34]

4.3 Grid Search for parameters optimization

To find the best values of the parameters, we do Grid Search, that is, we try all possible combinations.

SVD

For the SVD algorithm we tested for the parameter k the values (10, 15, 20, 25, 30), for the learning rate the values (0.1, 0.01, 0.001, 0.0001) and for the parameter normalization (regularization) λ the values (0.1, 0.01, 0.001, 0.0001). The values with the maximum performance with the metric RMSE (root mean squared error) are k = 25, learning rate 0.001, and λ = 0.001.

UserKNN and ItemKNN

For the classification of k-nearest neighbors, the unknown plural is assigned the most common class among its k nearest neighbors. When $k = 1$, the unknown plural is assigned the class of the training plural that is closest to it in the template space.

The ItemKNN algorithm tries to find objects similar to those already associated with the user. This is done based on a similarity metric and the K-nearest neighbours algorithm. We used Pearson's correlation coefficient as a similarity metric. With Grid Search we tested values of K from 5 to 50 with step 5 and the optimal was K = 40.

The UserKNN algorithm tries to find users with similar characteristics for each object. Like ItemKNN, it is based on a similarity metric (we used Pearson's correlation coefficient) and the K algorithm closest neighbours. With Grid Search we tested values of K from 5 to 50 with step 5 and the optimal was K = 35.

NMF

In array multiplication, arrays that are the factors of the product can be significantly lower than the array produced. This is the property that forms the basis of the NMF. If a table can be factorized into factors of lesser degree than the original, then the column vectors of the first factor are regarded as spanning vectors in the vector space.

The NMF algorithm accepts as input an array $M(m \times n)$ and tries to find two arrays $W(m \times k), H(k \times n)$, such that $M = WH$, with the restriction that no array contains any negative elements. This method has gained widespread recognition in the Netflix Award Recommendation System for movie recommendations. In the NMF algorithm the parameters we searched with Grid Search are k: (10, 15, 20, 25, 30), learning rate: (0.1, 0.01, 0.001, 0.0001), normalization parameter λ: (0.1, 0.01, 0.001, 0.0001). Optimal performance was achieved with the values k = 30, learning rate 0.1 and λ = 0.0001.

MLP

The MLP algorithm is based on neural networks. For activation function we used ReLU (Rectified Linear Unit), while with Grid Search we tested two architectures, with 4 hidden levels with size 64, 32, 16, 8 and with 3 hidden levels with size 32, 16, 8, while for the learning pace we tested the values (0.1, 0.01, 0.001, 0.0001). Optimal performance comes from the model with 3 hidden levels and a learning rate of 0.0001.

4.4 Evaluation Metrics

To evaluate the efficacy and accuracy of the recommendations we used 3 different metrics. Firstly, we used Precision and Recall that are defined by the following equations :

$$Precision@k = \frac{\text{\# of recommended drugs @k that are relevant}}{\text{\# of recommended drugs @k}}$$

$$Recall@k = \frac{\text{\# of recommended drugs @k that are relevant}}{\text{total \# of relevant drugs}}$$

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

$$\text{True Positive Rate} : TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

$$\text{False Positive Rate} : FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - TNR$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

We then calculated the toxicity score, which is based on the data from side effects that exist for some drug pairs. So, for the combination of drugs that the algorithm suggests to a patient, we look at how many pairs have side effects in terms of the total of all possible pairs. So the score ranges in the interval [0,1] and a lower score indicates a safer result.

4.5 Models Comparison

The results are shown in Table 3. As evaluation metrics we have used the area under the curves (Area Under Curve AUC) Receiver Operator Curve (ROC) and Precision-Recall (PR). We also give the training time of each model in seconds. Receiver Operator Curve (ROC) and Precision-Recall (PR) curves are shown in Figures 4.1 and 4.2 for all models, in the next Section.

Algorithm	AUC-ROC	AUC-PR	training time (sec)	testing time (sec)
SVD	0.71	0.69	0.33	6.41
UserKNN	0.72	0.70	0.93	45.14
ItemKNN	0.64	0.61	0.66	51.77
NMF	0.69	0.67	2.24	6.43
MLP	0.60	0.60	336.43	25.87
GMF	0.60	0.60	327.06	24.11

Table 3. Evaluation Metrics

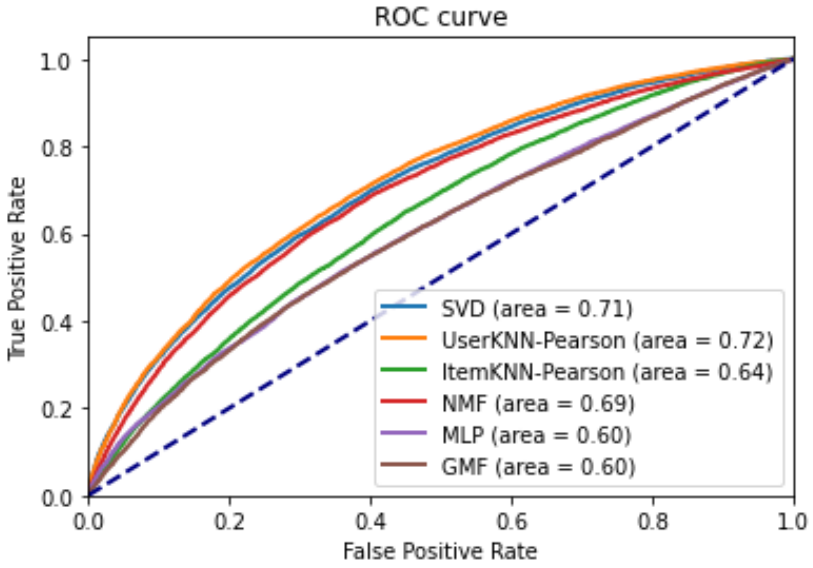

Figure 4.1: The Receiver-Operator-Curve curves for machine learning models.

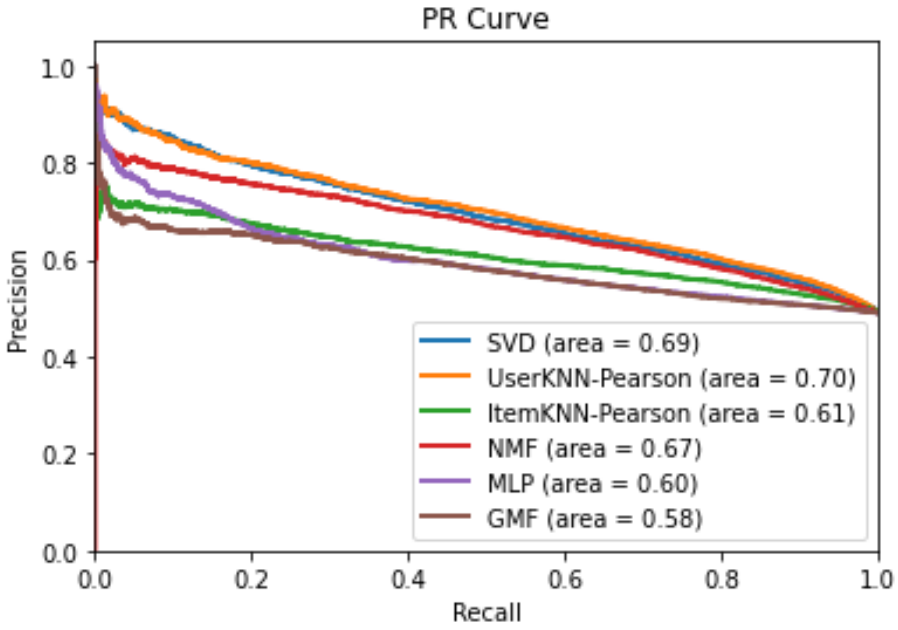

Figure 4.2: Precision-Recall curves for machine learning models.

The precision-recall curve shows the tradeoff between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

So based on Recall-Precision metrics, the best is UserKNN. Then, SVD follows, NMF third, ItemKNN fourth and MLP,GMF last. However, they all had a result much higher than 50% which is a random prediction.

In terms of time, neural networks (MLP, GMF) are quite late in training but also in testing. Of the other algorithms, UserKNN, ItemKNN are quite slow to test. SVD and NMF are the fastest in both training and testing. In general, training occurs once the system is set up, so perhaps more important is test time, although in this case we have no real-time constraints.

Here we will mention that MLP, GMF algorithms are quite more complex and modern than UserKNN, ItemKNN and SVD. However, their performance is lower than the rest of the algorithms. One possible reason for this is that precisely because these algorithms are complex and need to specify a large number of parameters, they require very large datasets. For example, the MLP model has been used with the Movie Lens movie recommendation dataset. While on these data the model performs well, on our data it performed quite poorly. However, Movie Lens contains 27,000,000 ratings on 58,000 movies from 280,000 users, i.e. it is orders of magnitude larger than our dataset and for this the neural network can be trained efficiently.

The results for the toxicity scores are shown in Table 4 and Figure 4.3. UserKNN has a big difference from the rest (43.9%), followed by NMF (38.5%), SVD (31.8%), MLP (22.3%), GMF (22,2%) and ItemKNN (21.7%). So taking this parameter into account, UserKNN cannot be chosen as the best, although it has a high performance, since it recommends drugs whose combination often has side effects. Considering both factors, performance and security, SVD probably has a better balance between the two desired outcomes.

Algorithm	**Toxicity Score**
SVD	0.318
UserKNN	0.439
ItemKNN	0.217
NMF	0.385
MLP	0.223
GMF	0.222

Table 4. Toxicity Metrics

4.6 Our Post Hoc Re-Ranking Method

For the best performing SVD algorithm, we tested the Post Hoc Re-Rank algorithm. The result is shown below in figures 4.5 and 4.6. We notice that we have a worse performance compared to the curves in figures 1 and 2.

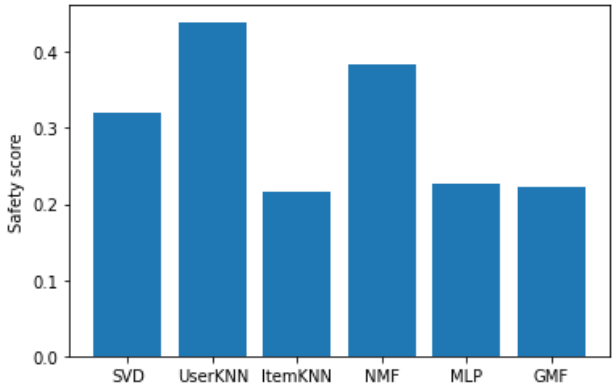

Figure 4.3: The performance of the algorithms in terms of toxicity. This metric quantifies side effects between drugs.

Specifically, for the ROC curve we have 0.64 compared to 0.71 before and for the PR curve 0.61 compared to 0.69. However, with this ranking, the safety score is reduced to 0.249, while previously it had a value of 0.318 (as shown in table 1). In terms of the side effects, we have an average of 2.3 in the top 5 recommendations in both cases (i.e. we range in the average, as we analyzed earlier). In the top 2 sentences, the average was 0.25 and after re-ranking it drops slightly to 0.246.

This is to be expected, as the original scores are optimized for performance, while the adjusted scores are aimed at increasing security. Therefore, with the Post Hoc Re-ranking technique we have a relatively small reduction in performance (about 10%), but the result is considerably safer (22% reduction in conflicts between drugs). So we have a trade-off between accuracy and security. But we have found that we can sacrifice accuracy by a small percentage while increasing security by a larger percentage. So, it is recommended to use the Post Hoc Re-ranking algorithm, to have a list of recommended drugs with greater safety.

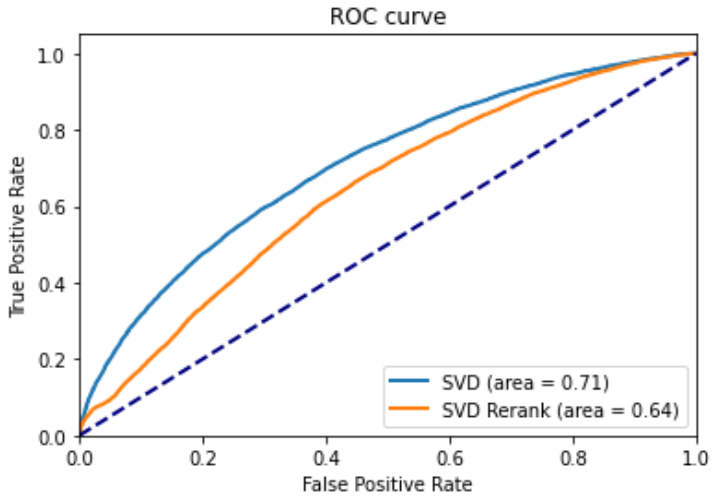

Figure 4.4: ROC curve for the SVD algorithm before and after re-ranking.

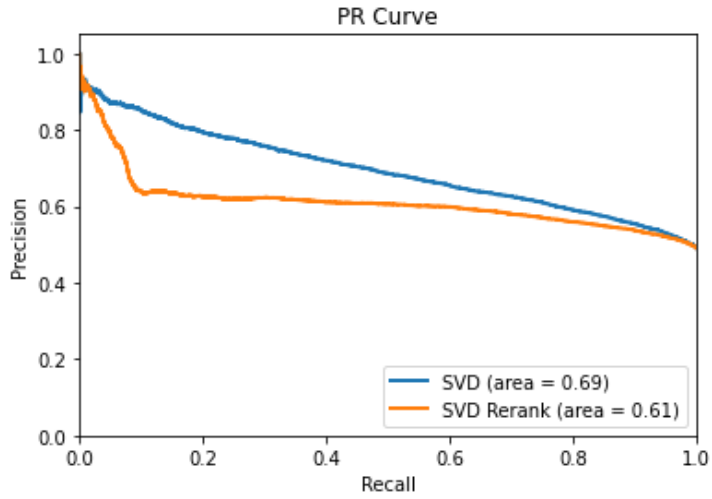

Figure 4.5: The PR curve for the SVD algorithm before and after re-ranking.

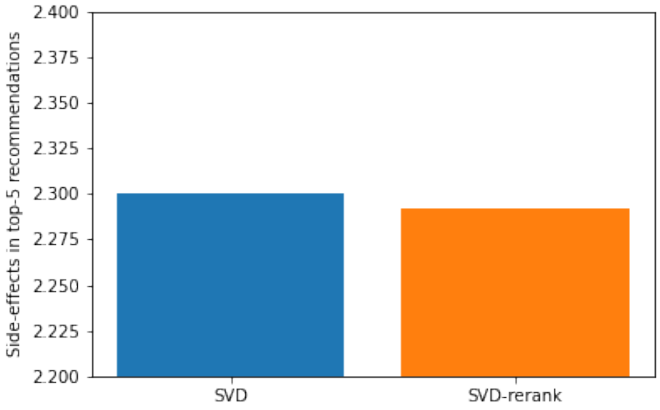

Figure 4.6: Side-effects for top 5 recommendations before and after re-ranking.

As can be seen in Figure 4.7, in terms of side effects, we have an average of 2.3 in the top 5 recommendations before re-ranking, which we manage to drop to 2.29 after the re-ranking.

4.7 Conclusion

In this thesis we proposed an approach to address the issues introduced by the healthcare environment, namely the application of Probabilistic Matrix Factorization models as well as Neural Network models to support the doctors in prescribing effective and safe treatments to patients affected by complex diseases. In particular, we focused on the SVD algorithm, that we have built to deal with additional auxiliary data and to evaluate it under different circumstances and tried to measure its efficiency and safety in the proposed prescriptions.

First of all, we conducted a Grid Search to ensure that the parameters of each algorithm are ideal and can provide with the best possible results. Followed by an ablation study to check how the results change by varying the amount of auxiliary data passed to each model. To compare results between models we used Precision-Recall and Receiver-Operator-Curve

to calculate the performance of each of the models. Then we took into account the toxicity score for the drug pairs for each model. While UserKNN has the best performance it also had the highest toxicity score so we chose to further improve and work with SVD which is close to the performance levels of UserKNN but also gave us a lower toxicity score.

Next, we used the Post Hoc Re-rank algorithm for SVD to incorporate information on drug interactions in the recommendation scores and the results were pretty promising. We counted the occurrences of each drug in the list of adversarial drug-drug interactions and the side effects that each drug combination could potentially cause. We observed a slight but expected loss in performance of the model but at the same time the toxicity score for recommendations dropped which guarantees us better results.

4.8 Our contributions

As can be seen in Section 4.6 we chose to value safety over performance. This precision - side effects tradeoff is the most important aspect of this thesis, in order to ensure that we can reach the lowest possible DDI Rate and Toxicity, which is the most crucial metric for our work. So even if the model tends to produce less accurate recommendations with respect to their executions without re-ranking, its slight drop is essential for our research if we want to achieve better and more acceptable results that can lead to safe treatment for the patients, leading to an overall better performance as far as every metric is concerned.

Post Hoc Re-Ranking Algorithm takes as input the recommendation list produced by SVD, updates the score of each drug based on its recurrence among the adversarial drug-drug interactions and delivers a re-ranked version of the initial input. From the Toy Example that we presented in Section 3.7, we can clearly observe that certain drugs that did not cause side-effects were strongly recommended by SVD before the re-ranking process, namely 690. This might be because the drug is recommended to nurse one of the diseases that affects this specific patient and does not cause any severe side effects if taken with any other recommended medications. At the same time, drug 102 was not considered by our model at

first, but after the re-ranking process it is prescribed as it does not appear in many adversarial drug-drug interactions compared to the previously prescribed drugs.

4.9 Future Work

Since our work covers delicate issues of the healthcare environment, it is necessary for medical experts to review the quality and integrity of our experimental results, to ensure that our research is safe for use.

With that in mind, in the future we will try to find and analyse other Probabilistic Matrix Factorization models which rely on different probability distributions. As we've seen from the use of Cornac algorithms, distinct implementations can give us different results. Furthermore, we have to review the Neural Networks Models, which might be more up to date and complex but they didn't give us the results we expected them to.

Another aspect in need of further attention is the analysis of the auxiliary data. For our Post Hoc Re-Ranking algorithm we counted the occurrences of each drug in the list of adversarial drug-drug interactions and also took into account the side effects that each drug combination may cause. This lead to a slightly worse performance of the model, which we have to analyze further, but reduced toxicity making the recommendations safer.

Finally, testing our proposed models by using different datasets would be appropriate to understand if it is able to provide consistent or even better results for some of the models. As we've mentioned above, we have used the MIMIC-III database that worked quite well with Matrix Factorization Algorithms but it didn't provide us with the best results as far as Neural Networks are concerned. In order to get better results as a whole we would need to make use of a larger dataset and compare the results with the current work to ensure the best possible outcome.

Bibliography

[1] Al-Sabaawi A. M. A., H. Karacan, and Y. E. Yenice. "A Novel Evidence-Based Bayesian Similarity Measure for Recommender Systems." In: *PloS one* 15.4 (2020). DOI: 10.1371/journal.pone.0231457.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. "Controlling Popularity Bias in Learning-to-Rank Recommendation". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys '17. Como, Italy: Association for Computing Machinery, 2017, pp. 42–46. ISBN: 9781450346528. DOI: 10.1145/3109859.3109912. URL: https://doi.org/10.1145/3109859.3109912.

[3] T. Achakulvisut, D. E. Acuna, T. Ruangrong, and K. Kording. "Science Concierge: A fast content-based recommendation system for scientific publications." In: *PloS one* 11.7 (2016), pp. 6061–6070. DOI: https://doi.org/10.1371/journal.pone.0158423.

[4] Charu C. Aggarwal. *Recommender Systems - The Textbook*. Springer, 2016, pp. 1–498. ISBN: 978-3-319-29659-3.

[5] Youjun Bao and Xiaohong Jiang. "An intelligent medicine recommender system framework". In: *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*. 2016, pp. 1383–1388. DOI: 10.1109/ICIEA.2016.7603801.

[6] Alejandro Bellogín, Pablo Castells, and Iván Cantador. "Statistical biases in Information Retrieval metrics for recommender systems". In: *Information Retrieval Journal* 20 (2017), pp. 606–634.

[7] Rianne van den Berg, Thomas N. Kipf, and Max Welling. *Graph Convolutional Matrix Completion*. 2017. DOI: 10.48550/ARXIV.1706.02263. URL: https://arxiv.org/abs/1706.02263.

[8] Shruthi Bhat and K. Aishwarya. "Item-based Hybrid Recommender System for newly marketed pharmaceutical drugs". In: *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2013, pp. 2107–2111. DOI: 10.1109/ICACCI.2013.6637506.

[9] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc., 1995. ISBN: 0198538642.

[10] Matias Bjørling, Jesper Madsen, Philippe Bonnet, Aviad Zuck, Z. Bandić, and Qingbo Wang. "LightNVM: Lightning Fast Evaluation Platform for Non-Volatile Memories". In: (Jan. 2014).

[11] John S. Breese, David Heckerman, and Carl Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52. ISBN: 155860555X.

[12] John S. Breese, David Heckerman, and Carl Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52. ISBN: 155860555X.

[13] L Breiman. "Random Forests". In: *Machine Learning* 45 (Oct. 2001), pp. 5–32. DOI: 10.1023/A:1010950718922.

[14] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (July 2017), pp. 18–42. DOI: 10.1109/msp.2017.2693418. URL: https://doi.org/10.1109%2Fmsp.2017.2693418.

[15] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. *Spectral Networks and Locally Connected Networks on Graphs*. 2013. DOI: 10.48550/ARXIV.1312.6203. URL: https://arxiv.org/abs/1312.6203.

[16] Robin Burke and Robin. "Hybrid Web Recommender Systems". In: vol. 4321. Jan. 2007. ISBN: 978-3-540-72078-2. DOI: 10.1007/978-3-540-72079-9_12.

[17] Jun Cai, Xuebin Hong, Qingyun Dai, Huimin Zhao, Yan Liu, Jianzhen Luo, and Zhijie Wu. "A User Profile Based Medical Recommendation System". In: *Advances in Brain Inspired Cognitive Systems*. Ed. by Jinchang Ren, Amir Hussain, Huimin Zhao, Kaizhu Huang, Jiangbin Zheng, Jun Cai, Rongjun Chen, and Yinyin Xiao. Cham: Springer International Publishing, 2020, pp. 293–301. ISBN: 978-3-030-39431-8.

[18] Z. Y. Chai, Li, Y. L., Y. M. Han, and S. F. Zhu. "Recommendation system based on singular value decomposition and multi-objective immune optimization." In: *IEEE Access* 7 (2018), pp. 6061–6070. DOI: 10.1109/ACCESS.2018.2842257.

[19] Damien Chanal, Nadia Yousfi Steiner, Raffaele Petrone, Didier Chamagne, and Marie-Cécile Péra. "Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering". In: *Encyclopedia of Energy Storage*. Ed. by Luisa F. Cabeza. Oxford: Elsevier, 2022, pp. 359–393. ISBN: 978-0-12-819730-1. DOI: https://doi.org/10.1016/B978-0-12-819723-3.00099-8. URL: https://www.sciencedirect.com/science/article/pii/B9780128197233000998.

[20] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A Library for Support Vector Machines". In: *ACM Trans. Intell. Syst. Technol.* 2.3 (May 2011). ISSN: 2157-6904. DOI: 10.1145/1961189.1961199. URL: https://doi.org/10.1145/1961189.1961199.

[21] V. X. Chen and T. Y. Tang, eds. *Regularized singular value decomposition in news recommendation system.* PRAI '19. 11th International Conference on Computer Science & Education (ICCSE). New York, NY, USA: Association for Computing Machinery, Aug. 2016.

[22] Yen-Ting Chen, Ming-Chang Yang, Yuan-Hao Chang, Tseng-Yi Chen, Hsin-Wen Wei, and Wei-Kuan Shih. "KVFTL: Optimization of storage space utilization for key-value-specific flash storage devices". In: *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2017, pp. 584–590. DOI: 10.1109/ASPDAC.2017.7858387.

[23] Kevin Clayton, Yoni Luxford, Joshua Colaci, Meral Hasan, Rebecca Miltiadou, Daria Novikova, Dean Vlahopoulos, and Ieva Stupans. "Community pharmacists' recommendations for natural products for stress in Melbourne, Australia: A simulated patient study". In: *Pharmacy Practice* 18 (Mar. 2020), p. 1660. DOI: 10.18549/PharmPract.2020.1.1660.

[24] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411.

[25] Mukund Deshpande and George Karypis. "Item-based top- N recommendation algorithms". In: *ACM Transactions on Information Systems - TOIS* 22 (Jan. 2004), pp. 143–177. DOI: 10.1145/963770.963776.

[26] Angela J. Fawcett and Roderick I. Nicolson. "Performance of Dyslexic Children on Cerebellar and Cognitive Tests". In: *Journal of Motor Behavior* 31.1 (1999). PMID: 11177620, pp. 68–78. DOI: 10.1080/00222899909601892. eprint: https://doi.org/10.1080/00222899909601892. URL: https://doi.org/10.1080/00222899909601892.

[27] Alexander Felfernig and Robin Burke. "Constraint-based recommender systems: Technologies and research issues". In: *ACM International Conference Proceeding Series* (Jan. 2008), p. 3. DOI: 10.1145/1409540.1409544.

[28] Mingchen Feng, Jiangbin Zheng, Jinchang Ren, Amir Hussain, Xiuxiu Li, Yue Xi, and Qiaoyuan Liu. "Big Data Analytics and Mining for Effective Visualization and Trends Forecasting of Crime Data". In: *IEEE Access* 7 (2019), pp. 106111–106123. DOI: 10.1109/ACCESS.2019.2930410.

[29] Federico Girosi, Michael Jones, and Tomaso Poggio. "Regularization Theory and Neural Networks Architectures". In: *Neural Comput* 7 (Oct. 1998). DOI: 10.1162/neco.1995.7.2.219.

[30] G. Guo, J. Zhang, and N. Yorke-Smith. "A Novel Evidence-Based Bayesian Similarity Measure for Recommender Systems." In: *ACM Transactions on the Web (TWEB)* 10.2 (2016). DOI: https://doi.org/10.1155/2019/2072375.

[31] Xuetao Guo and Jie Lu. "Intelligent e-government services with personalized recommendation techniques". In: *International Journal of Intelligent Systems* 22.5 (2007), pp. 401–417. DOI: https://doi.org/10.1002/int.20206. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/int.20206. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.20206.

[32] S. S. Hahn, S. Lee, and J. Kim. "To collect or not to collect: Just-in-time garbage collection for high-performance SSDs with long lifetimes". In: *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 2015, pp. 1–6. DOI: 10.1145/2744769.2744918.

[33] William L. Hamilton, Rex Ying, and Jure Leskovec. *Inductive Representation Learning on Large Graphs*. 2017. DOI: 10.48550/ARXIV.1706.02216. URL: https://arxiv.org/abs/1706.02216.

[34] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. "Neural Collaborative Filtering". In: *Proceedings of the 26th International Conference on World Wide Web* (2017).

[35] Franz Hell, Yasser Taha, Gereon Hinz, Sabine Heibei, Harald Müller, and Alois Knoll. "Graph Convolutional Neural Network for a Pharmacy Cross-Selling Recommender System". In: *Information* 11.11 (2020). ISSN: 2078-2489. DOI: 10.3390/info11110525. URL: https://www.mdpi.com/2078-2489/11/11/525.

[36] Folasade Isinkaye, Yetunde Folajimi, and Bolanle Ojokoh. "Recommendation systems: Principles, methods and evaluation". In: *Egyptian Informatics Journal* 16 (Aug. 2015). DOI: 10.1016/j.eij.2015.06.005.

[37] Y. Ji, W. Hong, Y. Shangguan, H. Wang, and J. Ma, eds. *Regularized singular value decomposition in news recommendation system.* 11th International Conference on Computer Science & Education (ICCSE). Nagoya, Japan: IEEE, Aug. 2016.

[38] Song Jiang, Lei Zhang, XinHao Yuan, Hao Hu, and Yu Chen. "S-FTL: An efficient address translation for flash memory by exploiting spatial locality". In: *2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST)*. 2011, pp. 1–12. DOI: 10.1109/MSST.2011.5937215.

[39] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. *MIMIC-IV*. 2021. DOI: 10.13026/s6n6-xd98. URL: https://physionet.org/content/mimiciv/1.0/.

[40] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1609.02907. URL: https://arxiv.org/abs/1609.02907.

[41] Nada Lavrač. “Selected techniques for data mining in medicine”. In: *Artificial Intelligence in Medicine* 16.1 (1999). Data Mining Techniques and Applications in Medicine, pp. 3–23. ISSN: 0933-3657. DOI: https://doi.org/10.1016/S0933-3657(98)00062-1. URL: https://www.sciencedirect.com/science/article/pii/S0933365798000621.

[42] Daniel Lee and H. Seung. “Learning the Parts of Objects by Non-Negative Matrix Factorization”. In: *Nature* 401 (Nov. 1999), pp. 788–91. DOI: 10.1038/44565.

[43] Sungjin Lee, Ming Liu, Sangwoo Jun, Shuotao Xu, Jihong Kim, and Arvind. “Application-Managed Flash”. In: *14th USENIX Conference on File and Storage Technologies (FAST 16)*. Santa Clara, CA: USENIX Association, Feb. 2016, pp. 339–353. ISBN: 978-1-931971-28-7. URL: https://www.usenix.org/conference/fast16/technical-sessions/presentation/lee.

[44] Taehee Lee, Jonghoon Chun, Junho Shim, and Sang-goo Lee. “An Ontology-Based Product Recommender System for B2B Marketplaces”. In: *International Journal of Electronic Commerce / Winter* 7 (Dec. 2006), pp. 125–155. DOI: 10.2753/JEC1086-4415110206.

[45] G. Li, J. Hua, T. Yuan, J. Wu, Z. Jiang, H. Zhang, and T. Li. “Novel Recommendation System for Tourist Spots Based on Hierarchical Sampling Statistics and SVD++.” In: *Mathematical Problems in Engineering* 2019 (2019). DOI: https://doi.org/10.1155/2019/2072375.

[46] M. I. McCarthy. “Painting a new picture of personalised medicine for diabetes.” In: *Diabetologia* 60.5 (2017), pp. 793–799. DOI: 10.1007/s00125-017-4210-x.

[47] Stuart Middleton, David De Roure, and Nigel Shadbolt. “Ontology-Based Recommender Systems”. In: vol. 32. May 2009, pp. 779–796. DOI: 10.1007/978-3-540-92673-3_35.

[48] L. Nagel, T. Süß, K. Kremer, M. U. Hameed, L. Zeng, and A. Brinkmann. “Time-efficient Garbage Collection in SSDs.” In: *arXiv* (2018). DOI: abs/1807.09313.

[49] Yoon-Joo Park and Alexander Tuzhilin. “The Long Tail of Recommender Systems and How to Leverage It”. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. RecSys ’08. Lausanne, Switzerland: Association for Computing Machinery, 2008, pp. 11–18. ISBN:

9781605580937. DOI: 10.1145/1454008.1454012. URL: https://doi.org/10.1145/1454008.1454012.

[50] Michael J. Pazzani, Christopher J. Merz, Patrick M. Murphy, Kamal M. Ali, Timothy Hume, and Clifford Brunk. "Reducing Misclassification Costs". In: *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*. ICML'94. New Brunswick, NJ, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 217–225. ISBN: 1558603352.

[51] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "DeepWalk: Online Learning of Social Representations". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: Association for Computing Machinery, 2014, pp. 701–710. ISBN: 9781450329569. DOI: 10.1145/2623330.2623732. URL: https://doi.org/10.1145/2623330.2623732.

[52] John Platt. "Using Analytic QP and Sparseness to Speed Training of Support Vector Machines". In: *Advances in Neural Information Processing Systems* 11 (Feb. 1999).

[53] Riccardo Poli, Stefano Cagnoni, Roberta Livi, Giuseppe Coppini, and Guido Valli. "A neural network expert system for diagnosing and treating hypertension". In: *Computer* 24 (Apr. 1991), pp. 64–71. DOI: 10.1109/2.73514.

[54] K. Przystupa, M. Beshley, O. Hordiichuk-Bublivska, M. Kyryk, H. Beshley, J. Pyrih, and J. Selech. "Distributed Singular Value Decomposition Method for Fast Data Processing in Recommendation Systems." In: *Energies* 14.8 (2019). DOI: https://doi.org/10.3390/computation7020025.

[55] J. R. Quinlan. "Induction of Decision Trees". In: *Mach. Learn.* 1.1 (Mar. 1986), pp. 81–106. ISSN: 0885-6125. DOI: 10.1023/A:1022643204877. URL: https://doi.org/10.1023/A:1022643204877.

[56] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G. Z. Yang. "Deep learning for health informatics." In: *IEEE journal of biomedical and health informatics* 21.1 (2016). DOI: 10.1109/JBHI.2016.2636665.

[57] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. "BPR: Bayesian Personalized Ranking from Implicit Feedback". In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI '09. Montreal, Quebec, Canada: AUAI Press, 2009, pp. 452–461. ISBN: 9780974903958.

[58] Sara Rosenthal and Kathleen McKeown. "Age Prediction in Blogs: A Study of Style, Content, and Online Behavior in Pre- and Post-Social Media Generations". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 763–772. URL: https://aclanthology.org/P11-1077.

[59] Paul Rutter and Edna Wadesango. "Does evidence drive pharmacist over-the-counter product recommendations?" In: *Journal of evaluation in clinical practice* 20 (May 2014). DOI: 10.1111/jep.12157.

[60] A. K. Sahoo, C. Pradhan, R. K. Barik, and H. Dubey. "DeepReco: deep learning based health recommender system using collaborative filtering." In: *Computation* 7.2 (2019), p. 25. DOI: https://doi.org/10.3390/computation7020025.

[61] Aghiles Salah, Quoc-Tuan Truong, and Hady W. Lauw. "Cornac: A Comparative Framework for Multimodal Recommender Systems". In: *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435.

[62] Ruslan Salakhutdinov and Andriy Mnih. "Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 880–887. ISBN: 9781605582054. DOI: 10.1145/1390156.1390267. URL: https://doi.org/10.1145/1390156.1390267.

[63] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. "Item-based Collaborative Filtering Recommendation Algorithms". In: *Proceedings of ACM World Wide Web Conference* 1 (Aug. 2001). DOI: 10.1145/371920.372071.

[64] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[65] C. Schuldt, I. Laptev, and B. Caputo. "Recognizing human actions: a local SVM approach". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 3. 2004, 32–36 Vol.3. DOI: 10.1109/ICPR.2004.1334462.

[66] Xin Shi, Fei Wu, Shunzhuo Wang, Changsheng Xie, and Zhonghai Lu. "Program error rate-based wear leveling for NAND flash memory". In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2018, pp. 1241–1246. DOI: 10.23919/DATE.2018.8342205.

[67] KL. Skillen, C. Nugent, M. Donnelly, L. Chen, and W. Burns. "Using Ontologies for Managing User Profiles in Personalised Mobile Service Delivery". In: Springer International Publishing, 2015.

[68] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. "Reasoning with Neural Tensor Networks for Knowledge Base Completion". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 926–934.

[69] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. "Deep Metric Learning via Lifted Structured Feature Embedding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4004–4012. DOI: 10.1109/CVPR.2016.434.

[70] Jyoti Soni, Ujma Ansari, Dipesh Sharma, and Sunita Soni. "Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction". In: *International Journal of Computer Applications* 17 (Mar. 2011), pp. 43–48. DOI: 10.5120/2237-2860.

[71] Yan-jing SUN, Shen ZHANG, Chang-xin MIAO, and Jing-meng LI. "Improved BP Neural Network for Transformer Fault Diagnosis". In: *Journal of China University of Mining and Technology* 17.1 (2007), pp. 138–142. ISSN: 1006-1266. DOI: https://doi.org/10.1016/S1006-1266(07)60029-7. URL: https://www.sciencedirect.com/science/article/pii/S1006126607600297.

[72] Quoc-Tuan Truong, Aghiles Salah, Thanh-Binh Tran, Jingyao Guo, and Hady W. Lauw. "Exploring Cross-Modality Utilization in Recommender Systems". In: *IEEE Internet Computing* 25.4 (2021), pp. 50–57. DOI: 10.1109/MIC.2021.3059027.

[73] Vamsi Krishna Tumuluru, Ping Wang, and Dusit Niyato. "Channel Status Prediction for Cognitive Radio Networks". In: *Wirel. Commun. Mob. Comput.* 12.10 (July 2012), pp. 862–874. ISSN: 1530-8669. DOI: 10.1002/wcm.1017. URL: https://doi.org/10.1002/wcm.1017.

[74] Hung-Wen Tung and Von-Wun Soo. "A personalized restaurant recommender agent for mobile e-service". In: *IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004. EEE '04. 2004.* 2004, pp. 259–262. DOI: 10.1109/EEE.2004.1287319.

[75] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. *Graph Attention Networks*. 2017. DOI: 10.48550/ARXIV.1710.10903. URL: https://arxiv.org/abs/1710.10903.

[76] Meng Wang, Weijie Fu, Shijie Hao, Dacheng Tao, and Xindong Wu. "Scalable Semi-Supervised Learning by Efficient Anchor Graph Regularization". In: *IEEE Transactions on Knowledge and Data Engineering* 28.7 (2016), pp. 1864–1877. DOI: 10.1109/TKDE.2016.2535367.

[77] Mingbang Wang, Youguang Zhang, and Wang Kang. "ZFTL: A Zone-based Flash Translation Layer with a two-tier selective caching mechanism". In: *2012 IEEE 14th International Conference on Communication Technology*. 2012, pp. 578–588. DOI: 10.1109/ICCT.2012.6511426.

[78] W. Xie and Y. Chen. "An Adaptive Separation-Aware FTL for Improving the Efficiency of Garbage Collection in SSDs". In: *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Association for Computing Machinery, 2014, pp. 552–553. DOI: 10.1109/CCGrid.2014.52.

[79] Keun Soo Yim, Hyokyung Bahn, and Kern Koh. "A flash compression layer for SmartMedia card systems". In: *IEEE Transactions on Consumer Electronics* 50.1 (2004), pp. 192–197. DOI: 10.1109/TCE.2004.1277861.

[80] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. "Graph Convolutional Neural Networks for Web-Scale Recommender Systems". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*. ACM, July 2018. DOI: 10.1145/3219819.3219890. URL: https://doi.org/10.1145%5C%2F3219819.3219890.

[81] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. *GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models*. 2018. DOI: 10.48550/ARXIV.1802.08773. URL: https://arxiv.org/abs/1802.08773.

[82] O.R. Zaiane. "Building a recommender agent for e-learning systems". In: *International Conference on Computers in Education, 2002. Proceedings.* 2002, 55–59 vol.1. DOI: 10.1109/CIE.2002.1185862.

[83] Jiacheng Zhang, Jiwu Shu, and Youyou Lu. "ParaFS: A Log-Structured File System to Exploit the Internal Parallelism of Flash Devices". In: *2016 USENIX Annual Technical Conference (USENIX ATC 16)*. Denver, CO: USENIX Association, June 2016, pp. 87–100. ISBN: 978-1-931971-30-0. URL: https://www.usenix.org/conference/atc16/technical-sessions/presentation/zhang.

[84] Yin Zhang, Daqiang Zhang, Mohammad Mehedi Hassan, Atif Alamri, and Limei Peng. "CADRE: Cloud-Assisted Drug REcommendation Service for Online Pharmacies". In: *Mob. Netw. Appl.* 20.3 (June 2015), pp. 348–355. ISSN: 1383-469X. DOI: 10.1007/s11036-014-0537-4. URL: https://doi.org/10.1007/s11036-014-0537-4.

[85] You Zhou, Fei Wu, Ping Huang, Xubin He, Changsheng Xie, and Jian Zhou. "Understanding and Alleviating the Impact of the Flash Address Translation on Solid State Devices". In: *ACM Trans. Storage* 13.2 (May 2017). ISSN: 1553-3077. DOI: 10.1145/3051123. URL: https://doi.org/10.1145/3051123.

[86] C. Ziegler. "Applying feed-forward neural networks to collaborative filtering". MA thesis.