Fatma Özdemir RECOMMENDER SYSTEM FOR EMPLOYEE ATTRITION PREDICTION AND MOVIE SUGGESTION AGU 2020

# RECOMMENDER SYSTEM FOR EMPLOYEE ATTRITION PREDICTION AND MOVIE SUGGESTION

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
M.Sc.

By
Fatma Özdemir
July 2020

# RECOMMENDER SYSTEM FOR EMPLOYEE ATTRITION PREDICTION AND MOVIE SUGGESTION

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
M.Sc.

By
Fatma Özdemir
July 2020

## SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name-Surname: Fatma Özdemir

Signature :

## REGULATORY COMPLIANCE

M.Sc. thesis titled Recommender System for Employee Attrition Prediction and Movie Suggestion has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

| Prepared By | Co-Advisor | Advisor |
|---|---|---|
| Fatma ÖZDEMİR | Dr. Mustafa COŞKUN | Prof. Dr. Vehbi Çağrı GÜNGÖR |

Head of the Electrical and Computer Engineering Program
Prof. Dr. Vehbi Çağrı GÜNGÖR

## ACCEPTANCE AND APPROVAL

M.Sc. thesis titled Recommender System for Employee Attrition Prediction and Movie Suggestion and prepared by Fatma Özdemir has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

17 / 07 / 2020

**JURY:**

| | |
|---|---|
| Advisor | : Prof. Dr. Vehbi Çağrı GÜNGÖR |
| Co-Advisor | : Dr. Mustafa COŞKUN |
| Member | : Dr. Ahmet SORAN |
| Member | : Dr. Fehim KÖYLÜ |
| Member | : Dr. Özkan Ufuk NALBANTOĞLU |

**APPROVAL:**

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science,
Executive Board dated ….. /….. / …….. and numbered ……..……. .

…….. /…….. / ……..

Graduate School Dean
Prof. Dr. İrfan ALAN

# ABSTRACT

# RECOMMENDER SYSTEM FOR EMPLOYEE ATTRITION PREDICTION AND MOVIE SUGGESTION

Fatma ÖZDEMİR

M.Sc. in Electrical and Computer Engineering Department

**Supervisor:** Prof. Dr. Vehbi Çağrı GÜNGÖR

**Co-Advisor:** Dr. Mustafa COŞKUN

July 2020

In this thesis, we focus on two problems raised in Machine Learning Community, namely, the recommender system and employee attrition problem. The recommender system is an information filtering system that predicts whether users would prefer a given item when purchasing a product. Recommender systems utilize information of users/items to predict. These systems, especially the collaborative filtering based ones, are used widely in E-commerce. In this work, we propose a hybrid model that combines collaborative filtering and side-information of users/items. In the proposed model, side-information of users/items is utilized to find correlated neighbors and cluster them. Then, collaborative filtering methods are applied to these clusters. The matrix factorization and random walk with restart are implemented to evaluate the performance of the proposed model. The proposed approach is systematically evaluated on MovieLens data. Experimental results show that the proposed model, which uses the side-information of the user/item, considerably improves the performance of traditional collaborative filtering methods.

In the second part of the thesis, we try to address the employee attrition prediction problem, which is trying to predict which persons will leave/continue a company for which they currently work. Nowadays, it is very critical for companies to predict that the employees will leave their jobs or not. Leaving employees, who are top performers, may cause financial or institutional knowledge losses in the organizations. To avoid such losses, companies have to predict employee attrition. However, the HR departments of companies are not advanced enough to make such a prediction. To this end, companies are using data mining methods to timely and accurately predict

employee attrition. In this study, the performance of different classification methods, such as Linear discriminant analysis (LDA), Naive Bayes, Bagging, AdaBoost, Logistic Regression, Support Vector Machine (SVM), Random Forest, J48, LogitBoost, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), XGBoost, Graph Convolutional Networks, have been presented to predict employee attrition based on two private company datasets, i.e., IBM and Adesso Human Resource datasets. Different from existing studies, we systematically evaluate our findings with various classification metrics, such as F-measure, Area Under Curve, accuracy, sensitivity, and specificity. Performance results show that data mining methods, such as LogitBoost and Logistic Regression algorithms, can be very useful for predicting employee attrition.

*Keywords: Receommender System, Hybrid Filtering, Matrix Factorization, Employee Attrition, Graph Convolutional Network*

# ÖZET

# ÇALIŞAN YIPRANMASI TAHMİNİ VE FİLM TAVSİYESİ İÇİN ÖNERİ SİSTEMİ

Fatma ÖZDEMİR

Elektrik ve Bilgisayar Mühendisliği Bölümü Yüksek Lisans

**Tez Yöneticisi:** Prof. Dr. Vehbi Çağrı GÜNGÖR

**Eş-Danışman:** Dr. Mustafa COŞKUN

Temmuz-2020

Bu tezde Makine Öğrenimi Topluluğunda ortaya atılan iki probleme odaklanıyoruz: tavsiye sistemi ve çalışanların yıpranma sorunu. Tavsiye sistemi, kullanıcıların bir ürün satın alırken belirli bir öğeyi tercih edip etmeyeceğini tahmin eden bir bilgi filtreleme sistemidir. Tavsiye sistemleri tahmin etmek için kullanıcı / öğe bilgilerini kullanır. Bu sistemler, özellikle işbirlikçi filtreleme tabanlı sistemler, E-ticarette yaygın olarak kullanılmaktadır. Bu çalışmada, ortak filtreleme ve kullanıcıların / öğelerin yan bilgilerini birleştiren karma bir model öneriyoruz. Önerilen modelde, ilişkili komşuları bulmak ve onları kümelemek için kullanıcıların / öğelerin yan bilgileri kullanılır. Daha sonra, bu kümelere ortak filtreleme yöntemleri uygulanır. Önerilen modelin performansını değerlendirmek için matris çarpanlara ayırma ve yeniden başlatma ile rastgele yürüme uygulanır. Önerilen yaklaşım MovieLens verileri üzerinde sistematik olarak değerlendirilir. Deneysel sonuçlar, kullanıcının / öğenin yan bilgisini kullanan önerilen modelin geleneksel ortak filtreleme yöntemlerinin performansını önemli ölçüde geliştirdiğini göstermektedir.

Tezin ikinci bölümünde, hangi kişilerin şu anda çalıştıkları bir şirketten ayrılacağını / devam edeceğini tahmin etmeye çalışan, çalışan yıpranması tahmini sorununu ele almaya çalışıyoruz. Günümüzde şirketler için çalışanların işlerini bırakıp bırakmayacaklarını tahmin etmeleri çok önemlidir. En iyi performans gösteren çalışanların işi bırakması, kuruluşlarda finansal veya kurumsal bilgi kaybına neden olabilir. Bu tür kayıplardan kaçınmak için şirketler, çalışanların yıpranmasını tahmin etmelidir. Bununla birlikte, şirketlerin İK departmanları bu tür tahminleri yapacak kadar gelişmiş değildir. Bu amaçla şirketler, çalışanların yıpranmasını zamanında ve doğru bir

şekilde tahmin etmek için veri madenciliği yöntemleri kullanmaktadır. Bu çalışmada, Doğrusal diskriminant analizi (LDA), Naive Bayes, Bagging, AdaBoost, Lojistik Regresyon, Destek Vektör Makinesi (SVM), Rastgele Orman, J48, LogitBoost, Çok Katmanlı Algılayıcı (MLP), K-En Yakın Komşular (KNN), XGBoost, Graph Convolutional Networks, iki özel şirket veri kümesinde (IBM ve Adesso İnsan Kaynakları veri kümelerine) çalışanların yıpranmasını tahmin etmek için uygulanmıştır. Mevcut çalışmalardan farklı olarak, bulgularımızı sistematik olarak F-ölçü, Eğri Altında Alan, doğruluk, duyarlılık ve özgüllük gibi çeşitli sınıflandırma metrikleri ile değerlendiriyoruz. Performans sonuçları, LogitBoost ve Lojistik Regresyon algoritmaları gibi veri madenciliği yöntemlerinin çalışanların yıpranmasını tahmin etmede çok yararlı olabileceğini göstermektedir.

*Anahtar Kelimeler:Öneri Sistemi, Melez Filtreleme, Matris Çarpanlarına Ayırma, Çalışanların Yıpranması, Grafik Konvolüsyon Ağı*

# Acknowledgements

I would like to thank Prof. Dr. Vehbi Çağrı GÜNGÖR for believing in me and decided to be my supervisor throughout my master's degree. I would like to thank also Dr. Mustafa COŞKUN for his support and being my co-advisor. Their guidance made it possible for me to conduct tireless and fruitful research on different Machine learning algorithms in various areas.

I am so grateful to my dear friends Mustafa Çağatay KOÇER, and Bengisu KOÇER who always support me and share our wonderful times.

I have to express my deep gratitude to my dear family for providing me with endless support.

Finally, a special thanks to my husband who has always supported me in all my decisions and always encouraged me to be the best version of myself. For being models of commitment and courage I dedicate this work to him.

# Table of Contents

# List of Figures

# List of Tables

*This thesis is dedicated to my husband*

# Chapter 1

# Introduction

The developing technology and rising the number of users increase of data on the internet. The storage of these data and access to information emerge as an important problem. Recommender System is a field of study that is emerged with these developments. Recommender systems try to estimate items that the users can choose using a database that contains users, items, ratings. Besides, companies examine not only the customer-product relationship but also the conditions of employees. Organizations have to calculate employee attrition to not reduce their profits. Therefore, it is very important to predict employee attrition. In this thesis, movie suggestion and employee attrition prediction are studied in detail.

## 1.1 Problems

### 1.1.1 Movie Suggestion

A recommender system aims to predict the rating (or the preference) that a user would give to an item and is primarily used in various commercial applications. Nowadays, online platforms and e-commerce sites offer different types of products and services to their users and the volume of information about these products or services has grown amazingly. In general, recommender systems are utilized in different online platforms and used as product recommenders for services, such as Amazon, or playlist recommenders for video and music services, such as Netflix and Spotify, or content recommenders for social media platforms, such as Facebook and Twitter. One of the most successful recommender systems is based on collaborative filtering approaches, in which a given item to a certain user is recommended by using collected ratings of items from many users [1-3].

Recently, researchers studied different recommender systems to improve classification accuracy [4-12]. All these studies are compared and summarized in Table 2.1.1. Although all these existing studies provide useful insights and valuable foundations about the recommender systems, there is no internationally accepted standard approach. Furthermore, none of them presents detailed performance evaluations of different recommender systems in terms of precision@k, Spearman's ρ, MAE, and RMSE. The aim of this study is to fulfill this gap and show that not only accuracy measure is critical, but also other performance metrics are critical for recommender systems. In addition, to improve the performance of collaborative filtering methods, in this study, we applied user-based and item-based collaborative filtering methods on clusters that are generated with the k-means algorithm by using the side information of users and items. More specifically, side information of users and items are utilized to find correlated neighbor clusters and collaborative filtering methods are applied to these clusters. To this end, two different collaborative filtering methods, the matrix factorization, and random walk with restart, are implemented to evaluate the performance of the proposed model. In general, the proposed approach is a hybrid system, which combines content-based and collaborative filters.

The proposed approach is systematically evaluated on MovieLens dataset [13], in which there are 943 users, 1682 items, and 100000 ratings. In addition, this dataset includes user-side information, such as age, gender occupation, and zip code as well as item-side information, such as genre and the year. Experimental results show that the proposed model, which uses the side-information of the users and items, significantly improves the performance of collaborative filtering methods.

### 1.1.2 Employee Attrition Prediction

Employee attrition today has been a challenging problem for both companies and employees. Working for long hours at an intense pace, short durations of holidays and low salaries might be some of the main reasons why employees leave their jobs. In general, employees leave their jobs when they come across better working conditions or would like to take a break. In this case, organizations face unexpected losses. In the global market, companies are also competing heavily and would like to keep the profit at the highest level and to sustain their business growth. Unexpected turnover can be

particularly challenging for talented employees and cause a huge drop in profits. It can even not only affect the financial profit but also disrupt the workflow in the organizations. To prevent such losses, the companies need to predict employee attrition so that they can take precautions timely, such as raising salaries or giving promotions. To this end, companies are looking for data mining methods to timely and accurately predict the employee attrition.

The existing studies [14-23] are compared and summarized in Table 2.2.1. Although all these existing studies provide valuable foundations to assess employee attrition, none of them presents a detailed performance evaluation of different classification methods in terms of accuracy, sensitivity, specificity, F-measure, Area Under Curve (AUC). In our earlier study, this gap is partially filled by showing that not only accuracy measure is critical, but also other performance metrics are critical for assessing employee attrition [24].

The objective of this study is to extend the study further by addressing the employee attrition problem using different classification algorithms and feature selection techniques. We use a real-world dataset and a synthetic dataset. To evaluate our findings, we utilize two private company datasets, i.e., IBM Human Resource (HR) dataset and Adesso (a private company in Turkey) HR dataset. In the IBM HR dataset, there are 35 features and 1470 samples, whereas there are 9 features and 532 samples in the Adesso HR dataset. Specifically, we applied various classification methods, such as Linear discriminant analysis (LDA), Naive Bayes, Bagging, AdaBoost, Logistic Regression, Support Vector Machine (SVM), Random Forest, J48, LogitBoost, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), XGBoost, Graph Convolutional Networks to predict the employee attrition. To the best of our knowledge, GCN has not been utilized for the attrition problem. Furthermore, we applied 4 different feature selection methods, such as chi-square, infogain, gainratio, and relief. Different from existing studies, we extensively evaluate the performance of state-of-the-art methods for various evaluation measures. Performance results show that data mining methods, such as LogitBoost and Logistic Regression algorithms, can be very useful for predicting employee attrition. To the best of our knowledge, this is the first study that evaluates the performance of classification and feature selection methods on both international company (IBM) and local company (Adesso) HR datasets. Upon

request, the complete HR datasets will be made available. This can help the research community develop novel prediction algorithms to assess employee attrition.

## 1.2 Objectives

Firstly, the objectives of the movie suggestion system, which is the first of the studies conducted within the scope of the thesis, are explained. In this study, we consider the limitations of collaborative filtering. We present a clustering-based hybrid model. The proposed model cluster main-data by finding neighbors with demographics information. Then, collaborative filtering methods are applied to each cluster. In this study, the aim is to improve the performance of traditional collaborative filtering methods using demographic information.

Secondly, the objectives of employee attrition prediction, which is the second of the studies in the thesis, are explained. The objective of this study is to address the employee attrition problem using different classification algorithms and feature selection techniques. We systematically evaluate our findings with various classification metrics, such as F-measure, Area Under Curve, accuracy, sensitivity, and specificity.

## 1.3 Structure

In the second chapter of this thesis, the studies on recommender system algorithms and employee attrition predictions are examined. In the third Chapter, the movie suggestion which is the first of the studies conducted within the scope of the thesis is explained. Recommender systems and proposed hybrid model are described. Problems encountered in recommender systems and the factors that determine the quality of the recommender system algorithms are mentioned. In order to measure the performance of the recommender system algorithms, frequently used criteria are introduced in the literature. The properties of the data set used in the experimental studies are described. At the end of the third chapter, the results obtained in the experiments carried out in this section are explained. In the fourth Chapter, employee attrition prediction which is the second of the studies conducted within the scope of the thesis is explained. Used datasets, methods, and experimental results are explained. In the fifth chapter, which is the conclusion, the approaches developed in the thesis are interpreted and the contributions of the thesis are summarized.

# Chapter 2

# Related Work

## 2.1 Movie Suggestion

In this Chapter, various studies and results of the academy on recommender systems are included. Researchers studied different recommender systems methods to improve classification accuracy [4-12]. These studies are compared and summarized in Table 2.1.1.

Hadi Zare et al. propose a hybrid recommender system that combines Link Prediction and Diffusion techniques predict to recommend films [4]. Furthermore, in that study, They use three different datasets. These are Filmtrust, Epinion, and Flixster. They compare the accuracies of the methods with MAE and RMSE. Pierpaolo Basile et al.[5] implement a content-based method that exploits HoIE in a content-based recommender system. They utilize the only F1@K as a performance metric. Matthias Bogaert et al. propose multi-label classification techniques to recommend items [6].

Ruiping Yin et al. utilize Graph neural network-based collaborative filtering to recommend movies [7]. In this study, they use two different to test their approach. These datasets are Movielens and Taobao. The results are shown with two different metrics. These performance metrics are Hit ratio at K (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K).

Alexander AS Gunawan proposes CRNN (convolutional recurrent neural network) to recommend music [8]. They implement their methods on the Free Music Archive (FMA). They improve accuracy using this method. In this study, four different metrics are utilized to measure the performance of recommender methods. These are True Positive Rate, False Positive Rate, Roc Curve, and F1 Score.

Abinash Pujahari et al. use Movielens data set to recommend movies [9]. In the study, group recommendation with collaborative filtering is proposed. They use only precision to show their results. Urszula Kuzelewska et al. utilize a novel method that is a Multi Clustering Collaborative Recommender System [10]. In this study, the Movielens dataset is used also. This dataset is commonly used in recommender systems of studies. Furthermore, RMSE is a very popular metric in studies of recommender systems. They use RMSE as a metric.

| Study | Datasets | Methods | Performance Metrics |
|---|---|---|---|
| Hadi Zare et al.[4] | Filmtrust, Epinion and Flixster | Collaborative Filtering | MAE, RMSE |
| Pierpaolo Basile et al.[5] | Movielens 1M, Last.fm, Library-Thing | Content Based | F1@K |
| Matthias Bogaert et al.[6] | Dataset of a financial services provider Belgian | Multi- label classification techniques | Precision, recall, accuracy, *F* 1 measure , *G* -mean |
| Ruiping Yin et al. [7] | MovieLens, Taobao | Collaborative Filtering | Hit ratio at K (HR@K), Normalized Discounted Cumulative Gain (NDCG@K) |
| Alexander AS Gunawan et al.[8] | Free Music Archive (FMA) | Content Based | True Positive Rate, False Positive Rate, ROC Curve, F1 Score |
| Abinash Pujahari et al.[9] | Movielens | Collaborative Filtering | Precision |
| Urszula Kuzelewska et al.[10] | GroupLens | Collaborative Recommender Systems | RMSE |
| Sujoy Bag et al. [11] | MovieLens | Collaborative Filtering | MAE |
| Haekyu Park et al. [12] | Movielens, FilmTrust, Epinions, Lastfm, Audioscrobbler | Matrix Factorization and Random Walk with Restart in Recommender | Spearman's, precision@k |

**Table 2.1.1 Overview of recommender systems literature**

Sujoy Bag et al. implement methods that are combination similarity metrics and machine learning algorithms [11]. They used also the movielens dataset. Their performance metric is MAE. MAE is also a very popular metric in studies of recommender systems.

Haekyu Park et al. compare Random Walk with Restart and Matrix Factorization in different conditions [12]. They utilize 5 different datasets: Movielens, FilmTrust, Epinions, Lastfilm, and Audioscrobbler. They implement their methods on the explicit and implicit dataset. They evaluated separately data sets. In this study, Spearman's and precision@k are used as performance metrics. All these studies that used mostly Movielens.

Although all these existing studies provide valuable foundations about the recommendation, none of them presents detailed performance evaluations of different recommender systems methods in terms of precision@k, Spearman's ρ, MAE, and RMSE. In this study, our main aim is to evaluate methods not merely one or two measures is significant but also other performance measures, such as precision@k, Spearman's ρ, MAE, and RMSE.

## 2.2 Employee Attrition Prediction

To increase the prediction of employee attrition was studied on classification methods by researchers. [14-23]. These studies are summed up in Table 2.2.1.

Dilip Singh Sisodia et al proposed that using data mining techniques predicts the probability of attrition of each employee [14]. Furthermore, in that study, they applied KNN, LSVM, Naïve Bayes, Decision Tree, and Random Forest. Random Forest has the highest accuracy. Shankar et al [15] applied Logistic Regression, SVM classification methods to predict employee attrition. They also applied feature selection methods. Neil Brockett et al proposed a model for predicting employee attrition by using t CLARA [16]. They also applied Random Forest, XGBoost, SVM, K-means clustering for remediation attrition.

| Study | Method | FS | SN | SP | FM | AUC | ACC | Dataset |
|---|---|---|---|---|---|---|---|---|
| Dilip Singh Sisodia et al [14] | Random Forest | No | 98.8 % | 99.3% | 0.993 | - | 98.9% | HR Analytic Data set |
| Rohit Hebbar A et al [15] | SVM | Yes | 82.0% | 95.0% | - | - | 93.0% | IBM HR |
| Neil Brockett et al [16] | CLARA | Yes | - | 65.0% | - | - | - | IBM HR |
| İbrahim Onuralp Yiğit et al [17] | SVM | Yes | 37.0% | 98.0% | 0.530 | - | 89.7% | HR data |
| Rachna Jain et al[18] | XGBoost | No | - | - | - | - | 90.0% | IBM HR |
| Sandeep Yadav et al[19] | AdaBoost | Yes | 96.5% | 96.0% | 0.936 | - | 94.5% | Human Resource Attrition |
| Sarah S. Alduayj et al[20] | Gaussian SVM | No | 62.0% | 68.7% | 0.652 | - | 67.0% | IBM HR |
| Rahul Yedida et al[21] | KNN | No | - | - | 0.882 | 0.969 | 94.3% | HR |
| V. Vijaya Saradhi et al[22] | Random Forest | No | - | - | - | - | 97.5% | Dataset of a Large Organization |
| Rohit Punnose et al[23] | XGBoost | No | - | - | - | 0.880 | - | HRIS database of the organization and BLS |

*FS: Feature Selection, SN: Sensitivity, SP: Specificity, FM: F-Measure, AUC: Area Under Curve, ACC: Accuracy*

**Table 2.2.1 Overview of employee attrition prediction**

Ibrahim Onuralp Yiğit et al applied Logistic Regression, Decision Tree, SVM, KNN, Random Forest, and Naive Bayes methods on the HR data for prediction employee attrition [17]. They utilized feature selection methods. They represent some performance metrics such as precision, recall, f-measure, and accuracy. Rachna Jain et al proposed predicting employee attrition by using XGBoost[18]. They used IBM HR dataset. Sandeep Yadav et al used data set to predict employee turnover that is different

from IBM HR [19]. In the study, Human Resource Attrition is analyzed in detail and they applied Logistic Regression, SVM, Random Forest, Decision Tree, Adaboost. They evaluated classifications methods with accuracy, precision, recall, F1 Score.

Sarah S. Alduayj et al predict employee attrition by using machine learning [20]. They also used IBM HR dataset. They applied SVM, KNN, and Random Forest on an imbalanced dataset, ADASYN-balanced dataset, and under-sampling dataset. Rahul Yedida et al aims to predict whether an employee of a company will leave or not [21].In this study, KNN, Naïve Bayes, Logistic Regression, and MLP Classifier were used as machine learning techniques. They represent accuracy with some performance metrics. These metrics are AUC, accuracy, and F1 Score.

V. Vijaya Saradhi et al proposed that using data mining techniques for employee churn prediction. They compared SVM, Random Forest, Naive Bayes [22]. Rohit Punnoose and Pankaj Ajit used XGBoost for predicting employee turnover [23]. They used 2 different datasets. These are HRIS database of the organization and BLS (Bureau of Labor Statistics). Furthermore, they compared AUC of LDA, SVM, Random Forest, Logistic Regression, Naïve Bayes, KNN with XGBoost. They discussed the performance of these methods by looking only AUC.

All these studies that used IBM HR data or different datasets are summed up in Table 2.2.1. These studies used also different data set. Although these studies provide valuable insights, none of them presents a detailed performance evaluation in terms of accuracy, specificity, sensitivity, F-Measure, and AUC. The aim of this thesis is to fulfill this gap and indicate that not merely a few performances metric is important, but also other performance metrics are critical for assessing employee attrition. We show F-Measure, AUC, sensitivity, specificity, and accuracy. In this study, we use 2 different datasets. These are IBM HR and Adesso HR dataset.

# Chapter 3

# Movie Suggestion

## 3.1 Technical Background

### 3.1.1 Targeted Marketing

Targeted marketing is the process through environments where products and services are tailored to potential customers for their personal tastes. Targeted marketing is often limited. However, it is more efficient than wide marketing types since it is designed according to the personal preferences of the customers. Targeted marketing is a model of the ideal customer, derived from the demographic characteristics of customers, age, gender, preferred online platforms, blogs or movie channels, and other similar information. Organizations utilize information on products to promote their products and market them to the related people.

Target marketing finds customers who most closely match your product or service offerings for marketing. It is important to increase sales and make the business successful. The main advantage of target marketing is to direct your marketing efforts to specific consumer groups. It facilitates the promotion of your products or services. Marketing is done with more affordable cost. It allows you to focus more on your marketing activities.

Social media platforms, such as Facebook, LinkedIn, Twitter, and Instagram, which are widely used today, allow organizations to market to the right users. For example, a hotel business can target a married social media user with a romantic weekend escape pack ad.

Demographic grouping is based on measurable statistics such as Age, Gender, Income level, Marital status, and Education. Demographic information is often the most important user profiling benchmark to implement target markets. Therefore, the demographic information of users is very important for many businesses.

The success of marketing a good or service is to know who will ultimately take it. For this reason, organizations spend a lot of money to define their target market. This is because not all products and services are generally preferable to every consumer. Finding who is the target market can cause a company to spend a lot of money and time. A company can expand its target market internationally as its product sales increase. A company should expand the target market in different parts of the world to reach a wider international market.

### 3.1.2 Recommender Systems

Internet usage is becoming widespread with the increasing human population and developing technology in the world. As a result of the common use of the Internet, there is a huge increase in the amount of data. A large amount of data negatively affects effective internet usage. Big data makes it difficult to access the information requested. The importance of filtering this data and directing the information to the relevant people is also increased greatly. One of the popular examples of this is recommender systems. Recommender systems are systems that recommend suitable items to a user according to the characteristics of the user without the effort. Users sometimes don't specifically search for a product. They want to choose from those recommended to them. In such cases, the recommender systems match the features of a product with the tastes of the user. Thus, users find the products that they not know before but can prefer. It collects information in the first phase of the work of a recommender system. The system collects relevant data to create a profile that reflects their taste about the active user. Having a lot of information about the system user causes the system to create a better recommender list. Recommender systems receive two types of feedback that are seen in figure 3.1.2.1. The first one is implicit feedback. This is the data obtained from the data available in the system. The second is explicit feedback. This type of feedback is the most useful feedback. This type of feedback is the evaluation data received directly from the end-user.

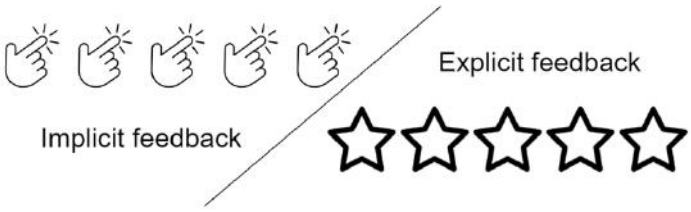


**Figure 3.1.2.1 Feedbacks types**

The next stage of the operation of the system is learning. At this stage, a learning algorithm is applied, the data is filtered. So the model is generated. In the last step, the recommender system suggests products to the user. Recommender systems should also provide users with items they may not have known before but may like. Recommender systems offer convenience both to the user and service providers. With this feature, today, recommender systems are used actively in many areas. Shopping products, movies, and music recommendations are the most popular. It is observed in the researches that shopping has increased with personalized recommenders. In recommender systems, recommenders are generally personal. Another method of recommender systems is group recommender systems. In group recommenders, a group is made taking into account the common characteristics of a user. In such systems, how the users are grouped, the characteristics of the groups, the number of individuals in the group are important factors.

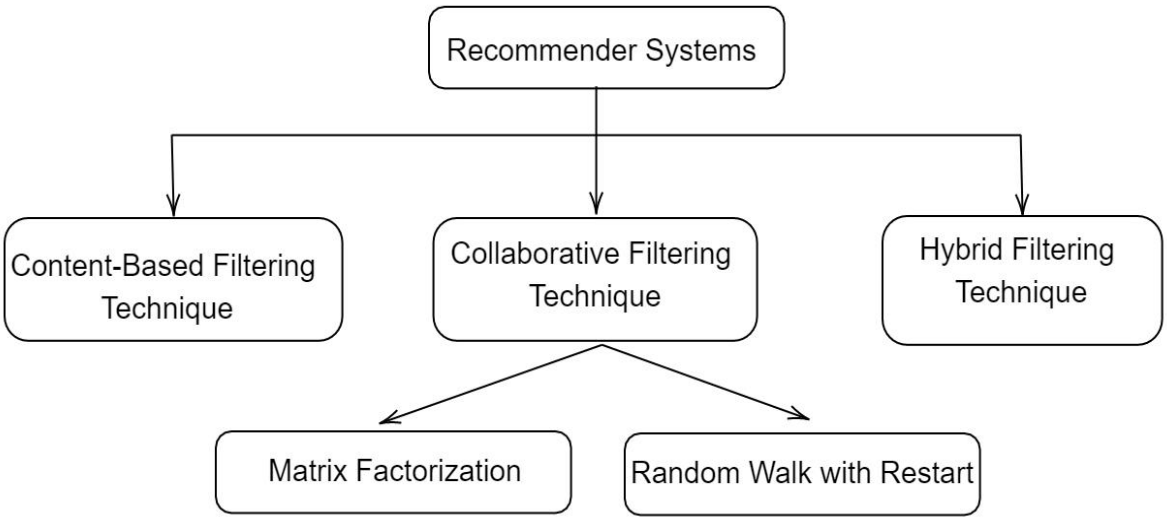


**Figure 3.1.2.2 Recommender Systems Types**

Recommender systems are generally examined in three main branches as in figure 3.1.2.2. Each of these methods has its advantages and disadvantages compared to the

others. These are content based filtering techniques, collaborative filtering techniques, and hybrid techniques.

#### 3.1.2.1 Content-Based Filtering

This method recommends based on the information found about the content. As with other methods, other users or items are not considered. Besides, no similarity is calculated between users. Content-based approaches use features of users and items. While recommending, it looks profiles of items that users have preferred in the past. Content that user likes and dislikes is determined to recommend other items.

A content-based recommender system creates user-profile by looking at the content of the items the user has rated in the past. The more users use the system, the more data is generated in the system. In this way, the recommender system starts to offer more accurate recommenders. Figure 3.1.2.1.1 shows the working principle of the CBF. There is no complicated calculation process in content-based systems. The contents may differ according to the systems. The content can be explicitly explicit, the genre of a movie, its main actors, its production year, its sub-genre, etc. an example of this. Content can also be text-based, for example, a movie's title, subject, synopsis, or comments. It is less affected by the cold start problem. Features about new items and users added to the system can be entered into the system, and users can be offered recommenders similar items. However, new users and items with features not previously seen may be affected by the cold start problem. It can work on more dense data with less computing power. Thanks to user profiles, items are recommended that attracts fewer users' attention can be recommended. The advantage of this method is that there is no dependence on other users or items, it can recommend items to users with a unique taste, and can recommend new or unpopular products. The disadvantages are that users only get recommenders similar to the items they liked in the past, the content has to have meaningful features, it can be quite difficult to create a model. In content-based filtering, new products added to the system can be recommended by the system using content information even if they have never been evaluated.

Content-based filtering is one of the widely utilized recommender systems algorithms. For example, in a movie recommender system, users' characteristics are age, gender, job, income, hobbies, etc. can. For a movie, there may be the category, lead, length, director, and other characteristic features. After, the step that the content-based recommender system should do is to match users with items. For example, "Young women love romantic-comedy movies more".

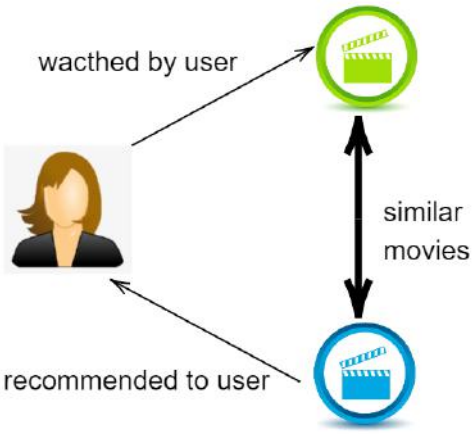


**Figure 3.1.2.1.1 Content Based Filtering Method**

### 3.1.2.2 Collaborative Filtering

One of the most utilized and utilized recommender systems is the Collaborative Filtering (CF) method. In this method, the data is filtered using the evaluations of other users. Basically, users who have similar tastes in the past are assumed to have similar tastes in the future. In this method, users first evaluate the items, then, it recommends items to the active user by looking at the evaluations of other similar users. In order to recommend the active user in the CF system, the preferences of other users who show similar behavior tendencies with the active user are checked. Similar users are matched by looking at the users' distinctive features. Thanks to these similarities, it offers recommenders to users. Figure 3.1.2.2.1 shows the working principle of the CF. It thinks that by finding people who made similar preferences in the past, he will make similar preferences in the future. CF techniques use the users' items evaluation based profiles instead of the content features of the items in the process of generating estimates. In a recommender system using the CF method, the rating matrix is created

with users-items. This matrix contains the results of users' evaluation of items. In practice, this method is used in intense data. However, since users will not evaluate every item, there are gaps in the matrix. In the recommender system using CF, there are similarities between user or item in the data on the rating matrix.

Thanks to these similarities, recommenders, and predictions about users and items are produced. In other words, suppose there are n users and m products in an office system. A user-item matrix of size [n x m] is created. In real life, these systems have many users and items. Therefore, it can be seen that the size of this matrix is quite high. In this system, the small number of evaluations of the items causes the matrix to be sparsity. The sparsity is a problem for the CF method. CF Recommender systems find other system users that are similar to the active user. Various similarity algorithms are used to calculate similarities between users. Neighborhoods are created by looking at the result of these similarity calculations. The created neighborhoods are processed with CF algorithms.

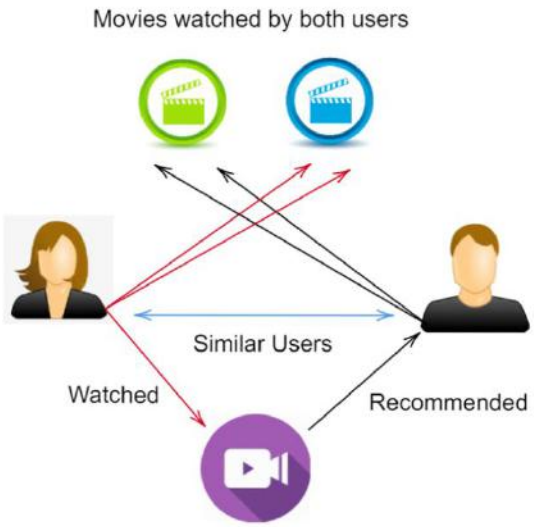


**Figure 3.1.2.2.1 Collaborative Filtering Method**

The recommended items are the target items. The target item is offered to the active user as a recommendation. CF cannot recommend items that are not rated. There are two different basic approaches in CF techniques: memory-based approaches and model-based approaches. Memory-based approaches use the user-item matrix to predict. In model-based approaches, various data mining and machine learning techniques are

utilized, and a model is created on the user-item matrix. Each collaborative filtering technique has its advantages and disadvantages. The biggest advantage is that collaborative filtering techniques do not need to know the domain of the system on which the filtering algorithm is working. In addition, the system using these methods does not need any information other than users, items, and ratings. For most common situations, these filtering methods produce good results. The biggest disadvantage of the system is that it requires a lot of data to start working. User and items data must be stored in a standard way, and users' past behavior must be kept constant. Problems of CF recommender systems are the cold start and sparsity. When a new user or item arrives in the cold start system, it cannot produce recommenders because there is no historical data. The sparsity problem occurs because the users in the recommender system cannot evaluate all items. Explicit data is obtained by users' item ratings. Implicit data are data obtained indirectly, such as the number of clicks. While explicit phases are easy to use for interpretation and recommender, implicit data are difficult to interpret. Obtaining explicit data can be difficult. Users may not want to take time for evaluation. Implicit data is easy to obtain. Users do not need extra time.

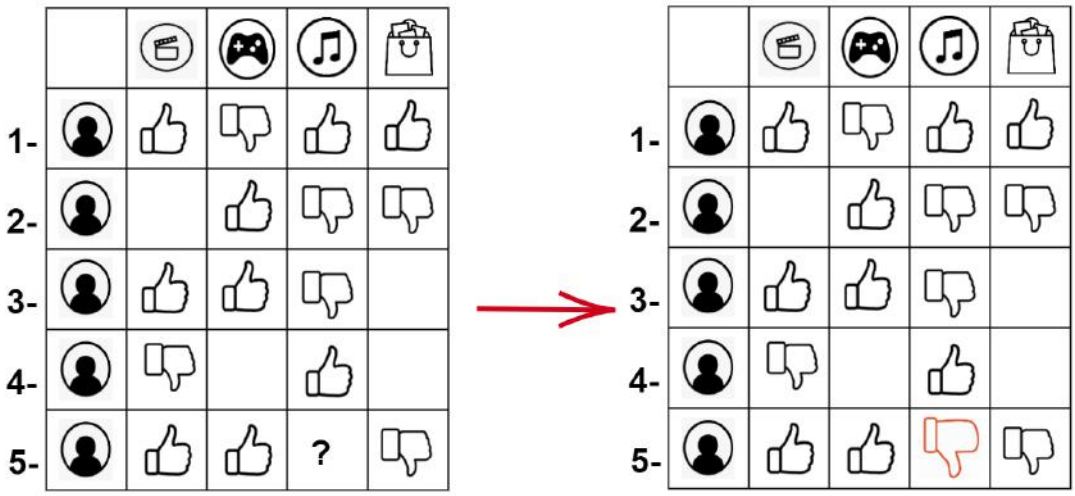


**Figure 3.1.2.2.2 Collaborative Filtering Example**

An example of a recommender system that used the Collaborative Filtering method is shown in Figure 3.1.2.2.2. There are 4 items and 5 users in this recommender system. These users evaluate four items. Users evaluate items as binary selection as likes and dislikes. It is expected that produce recommendations from the system about whether the 5th user will like music that is 3rd item. In this case, the 5th user is an

active user and the target item is the music. The CF algorithm creates recommendations to the active user about the target item by using similarities with other users. In this context, when the table is analyzed, it is observed the 5 amount of users

The most similar users are users 2, 3and 5. Since the 5th user and users 2nd and 3rd like/ not like similar items, the CF algorithm follows the approach that 5 will evaluate the same with 2 and 3 in music listening activity. For this reason, the evaluation of user 2 and 3 to the music listening activity is considered by the recommender system to recommend whether 5 like the music listening activity. Here, the evaluation criterion can be 0 (liked), 1 (disliked). As a result, user 5 dislike item 3 like user 3.

#### 3.1.2.2.1 Matrix Factorization

Matrix factorization (MF) is the most commonly utilized collaborative filtering method as a latent factor model. A user evaluates to a certain item. Evaluation can be rate from one to five. This collection of ratings can be indicated in the form of a matrix. Each row symbolizes each user, while each column symbolizes different items. Clearly, the matrix will be sparse. Because everyone will not evaluate every item. In figure 3.1.2.2.1.1 summarizes the main idea of matrix factorization. There is a user-item matrix with the dimensionality of (m,n). This matrix can be reduced as two matrices with each matrices having dimensions of (m,k) and (k,n) that are latent features.

MF decomposes a user-item rating matrix. It finds latent factors in relations between users and items. Matrix factorization (MF) predicts ratings by using given ratings. Ratings are calculated as in the seen equation (3.1.2.2.1.1),

$$\hat{\boldsymbol{r}}_{\boldsymbol{ui}} = \boldsymbol{x}_{\boldsymbol{u}}^{\boldsymbol{T}}\boldsymbol{y}_{\boldsymbol{i}} \qquad (3.1.2.2.1.1)$$

Rating of item $i$ given by user $u$ and $x_u$ represents $u$'s vector and $y_i$ is $i$'s vector.

$$\boldsymbol{L} = \frac{\boldsymbol{1}}{\boldsymbol{2}} \sum_{(\boldsymbol{u},\boldsymbol{i})\in\boldsymbol{\Omega}_{\boldsymbol{R}}} ((\boldsymbol{r}_{\boldsymbol{ui}} - \boldsymbol{x}_{\boldsymbol{u}}^{\boldsymbol{T}}\boldsymbol{y}_{\boldsymbol{i}}) + \boldsymbol{\lambda}(\|\boldsymbol{x}_{\boldsymbol{u}}\|^{\boldsymbol{2}} + \|\boldsymbol{y}_{\boldsymbol{i}}\|^{\boldsymbol{2}})) \qquad (3.1.2.2.1.2)$$

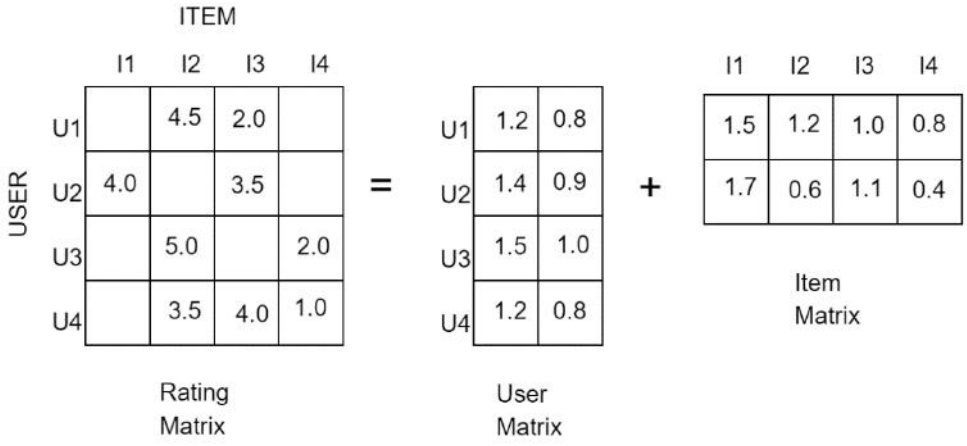


**Figure 3.1.2.2.1.1 Matrix Factorization Example**

The equation shows the objective function. $r_{ui}$ is a given rating. $\Omega_R$ represents ratings a set of (user, item) pairs. The term $\lambda(\|x_u\|^2 + \|y_i\|^2))$ controls overfitting. The hyperparameter $\lambda$ controls the degree of regularization.

### 3.1.2.2.2 Random Walk with Restart

Random Walk with Restart (RWR) is one of the widely utilized in recommender systems. It is a graph-based collaborative filtering method. In figure 3.1.2.2.2.1, it is seen as a user-item bipartite graph $G$ that represents RWR. RWR predicts the rating of items that are given by the user $u$. In the graph, $V$ represents the set of nodes and $U$ represents the set of users and $I$ represents the set of items. Hence, the equation is $V = U \cup I$. Each edge (u, i, r_ui) ∈ E represents the rating. The rating is the weight of the edge. RWR utilizes a random surfer to calculate the rating of items for a specific user u by moving around on the user-item bipartite graph. In the graph, the weight of edges is ratings that are between users and items. The random surfer starts to move around the graph from $u$-th the user. $u$-th user is currently node $v$. After, the surfer walks random or restarts. Random walk demonstrates the surfer act to other nodes from the current node with probability $1 - c$. The probability of restart is represented with $c$. The node $v$ that is visited many times by surfer, is highly connected with node u. The node $v$ is rated high by $u$. Items that are rated highly by users are constantly visited by the random surfer. Similar users like probably the same item $i$. Hence, if a user likes an item, a similar user also likes the item. The probability of visited each item is ranking scores. This score is RWR scores for the query user $u$.

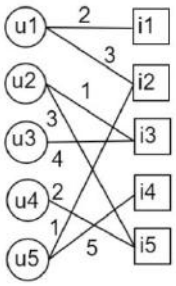


**Figure 3.1.2.2.2.1 Random Walk with Restart Bipartite Graph**

In the below recursive equation, RWR scores are described for a beginning node $u$,

$$\boldsymbol{r} = (\mathbf{1} - \boldsymbol{c})\tilde{\boldsymbol{A}}^{\boldsymbol{T}}\boldsymbol{r} + \boldsymbol{cq} \quad (3.1.2.2.2.1)$$

$r$ is the RWR score vector the starting node $u$. q is the starting vector whose $u$-th entry is 1 and all other entries are 0. The probability the restarting is $c$. $A$ is the weighted adjacency matrix of the graph $G$. $\tilde{A}$ is the row-normalized adjacency matrix. The RWR score vector is updated in the below equation:

$$\boldsymbol{r}^{(t)} \leftarrow (\mathbf{1} - \boldsymbol{c})\tilde{\boldsymbol{A}}^{\boldsymbol{T}}\boldsymbol{r}^{(t)} + \boldsymbol{cq} \quad (3.1.2.2.2.2)$$

where $r^{(t)}$ is the RWR score vector of $t$-th iteration.

### 3.1.2.3 Hybrid Filtering

Hybrid filtering techniques are combining multiple different recommender systems techniques. Thus, it tries to solve the problems of the systems that use a single method. In addition, hybrid approaches combining Collaborative Filtering and Content-Based Filtering methods are generally utilized to improve the performance of recommender systems. Figure 3.1.2.3.1 shows the working principle of the Hybrid Recommender Systems.

Whereas CF recommender systems are based on ratings, CBF recommender systems are based on textual explanations and the active user's personal ratings. The systems use different methods depending on the input types when recommending. Types of recommender systems have advantages and disadvantages. CF operates more effectively in systems where data is dense.

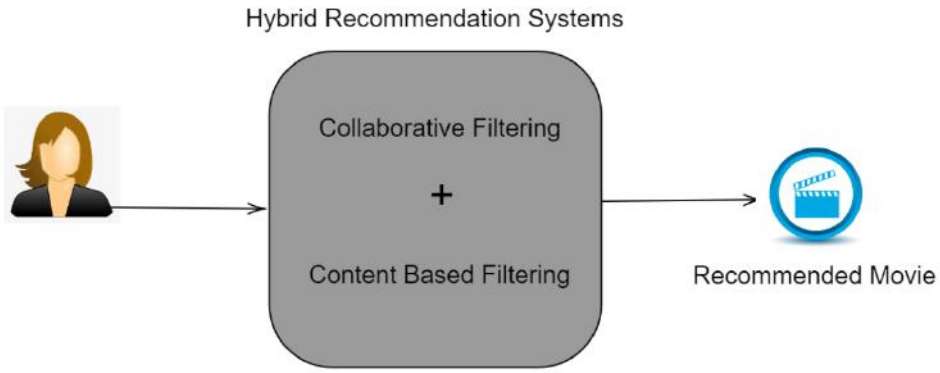


**Figure 3.1.2.3.1 Hybrid Model**

Hybrid approaches are tailored to needs. Content-based and collaborative filtering methods can be applied in different ways. In this approach, content-based and collaborative filtering methods are utilized together. The purpose of this approach is to get rid of the disadvantages of a single method and to combine the advantages of the methods to create a more successful method. Methods are used together to solve cold start, scalability, and sparsity problems, which are the main problems of recommender systems.

Studies conducted show that when compared to hybrid methods, CBF or CF methods used alone, hybrid methods increase performance. So the results of hybrid filtering techniques are more successful. The main reason for this is that in cases where a technique is not sufficient, a recommendation list can be obtained by referring to the other technique in the hybrid method. Using Netflix CF, it identifies similar users according to the tastes and makes recommenders in line with user preferences. In addition, by using CBF, users can look at the content they like (explicit or implicit) and suggest similar contents.

### 3.1.3. Recommender Systems Problems

#### 3.1.3.1 Scalability

Recommender systems have to work with large data sets. In addition, it has to recommend to users in real-time. Recommender systems should be able to serve millions of users simultaneously. The number of items recommended in many e-commerce sites reaches billions. An effective recommender system should be very fast when used in systems with a large amount of data. It is often difficult to suggest in real-time systems with millions of users and items. This is the case for popular systems such as e-commerce recommender systems, movies, and music recommender systems. No matter what type of recommender method is used, scalability is one of the biggest challenges for a recommender system. As the number of items and the number of users increases, the complexity of the nearest neighbor algorithm used on the basis of most recommender systems also increases. For this reason, scalability is a serious problem in systems with millions of users and billions of items. Various solutions are produced to overcome this problem. Dimensional reduction and clustering techniques can be given as examples, but the scalability problem of recommender systems still continues today.

#### 3.1.3.2 Sparsity

The sparse data problem occurs when users evaluate a small number of items. The sparsity problem is that there are not enough votes in the system. In this case, the matrix used for the collaborative filtering method is sparse. Matrices can be sparse for different reasons. One of these situations is that the number of users is low and the number of items is high. Users may not be able to rate all items. This causes the user-item matrix to be sparse. Similarly, having millions of users and items causes sparse data problems. The sparse matrix affects the performance of the recommender system badly. Recommender systems should receive necessary and sufficient information to produce good results. Therefore, a small amount of input data reduces the performance of the recommender systems. In practice, recommender systems usually have to work with missing data. Today, the sites that use these systems contain millions of products. Therefore, it is almost impossible for users to evaluate all of these products. Users can only see and evaluate a few of these products. For example, the Movielens (grouplens.org) dataset contains 943 users and 1682 movies. That means a 943x1682

matrix. This matrix is the user-product evaluation matrix. However, there are only 100,000 user reviews in this data set. 93.7% of this evaluation matrix is empty.

### 3.1.3.3 Cold Start Problem

Cold start problem is the problem of recommender systems using a collaborative filtering method. When a new item is added to the system, this makes it impossible to recommend the item. Figure 3.1.3.3.1 shows the Cold Start Problem.

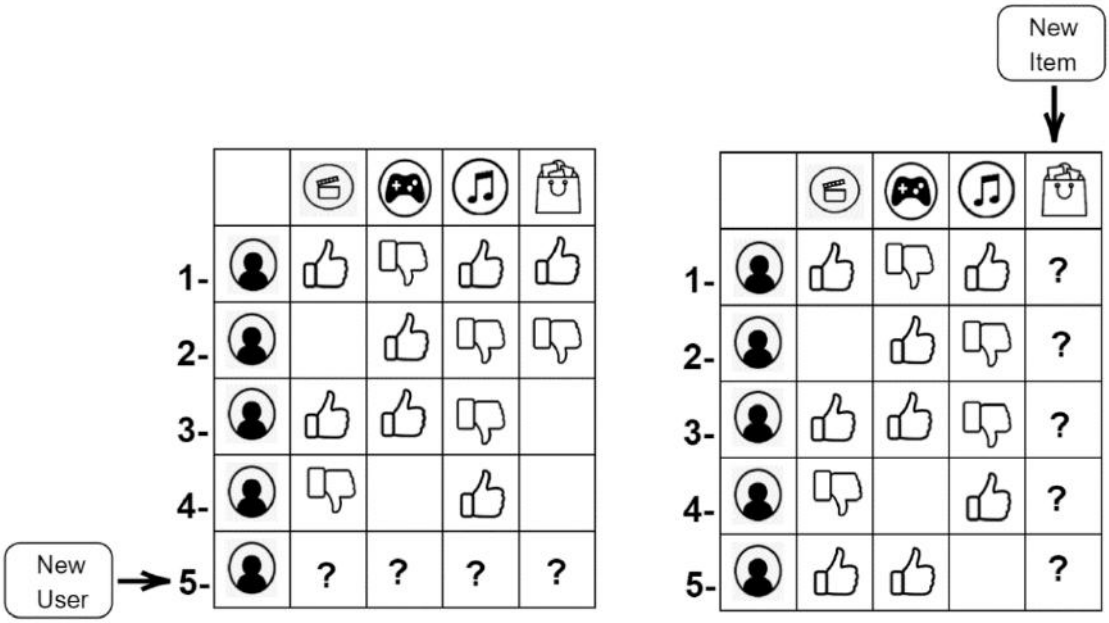


**Figure 3.1.3.3.1 Cold Start Problem**

Similarly, when a new user is added to the system, there will be no similarity between the new user and the users previously registered in the system. When a new user is registered in the system, there is no historical item evaluation information about this person. The system cannot find out user interest in which items. To solve this problem, some systems ask the user to evaluate a group of items while registering. In content-based filtering systems, there is no problem since the side features of the items or users are taken into consideration instead of the evaluations of the items. In these systems, item profiles and user profiles are used. It is one of the biggest problems of cold start recommender systems.

## 3.1.4 K-means Clustering

Creating groups of data with similar properties in a dataset is called clustering. There are many similarities between the samples in the same cluster. However, the

similarities between different clusters are small. K-Means is a clustering algorithm that is commonly used. The K-Means algorithm is an unsupervised learning clustering algorithm. K is the number of clusters. The algorithm takes the k value as a parameter. This can be seen as a disadvantage.

The K-means algorithm is simple. First, the K value is determined. Then, the algorithm randomly selects K center points. The distance of each sample to the center points is calculated. The data is included in the cluster with the closest center point. Then, for each cluster, new center points are selected and samples are clustered according to new center points. This process continues until the system becomes stable. Some problems may occur in the K-Means algorithm. This problem is randomly assigning the starting center points.

## 3.2 The Proposed Model

In this study, we apply four different methods. We separate the main data into clusters with k-means clustering using side information of the user/item. Then, we apply collaborative filtering methods to each cluster separately. These collaborative filtering methods are Matrix Factorization (MF) and Random Walk with Restart (RWR). After clustering users and items, we applied our hybrid approaches based on item-based collaborative filtering and user-based collaborative filtering. The proposed model is implemented based on four different methods, including User-based MF, Item-based MF, User-based RWR, and Item-based RWR.

In the user-based model, the first step is clustering with K-means using side information of users. Then, the model gets user_ids in each cluster. Next, the main data is clustered according to received user_ids. Finally, MF and RWR are applied to each cluster.

In the item-based model, the first step is clustering with K-means using side information of items. Then, the model gets item_ids in each cluster. Next, the main data is clustered according to received item_ids. Finally, MF and RWR are applied to each cluster.

The proposed model is realized using Python [12]. Used for experiments computer has Intel® Core™ i5-8300H CPU @ 2.30 GHz and 8.0 GB ram.

### 3.2.1 Proposed Model with Matrix Factorization

**User based MF.** First, the main dataset is clustered with k-means by using side information. The number of clusters used is 2,4,5,8, and 10. Users with similar side information are in the same cluster. This side information includes age, occupation, gender, and zip-code. Then we apply the Matrix Factorization method to each cluster separately. We compare the performance of the model with different numbers of clusters. Performance metrics are calculated with combined scores of clusters.

**Item based MF.** the main dataset is clustered with k-means clustering by using side-information of items. Then, we apply the Matrix Factorization to each cluster.

### 3.2.2 Proposed Model with Random Walk with Restart

**User based RWR.** First, the main dataset is clustered with k-means by using side information. The number of clusters used is 2,4,5,8, and 10. Users with similar side information are in the same cluster. This side information includes age, occupation, gender, and zip-code. Then we apply the Random Walk with Restart method to each cluster separately. We compare the performance of the model with different numbers of clusters. Performance metrics are calculated with combined scores of clusters.

**Item based RWR.** the main dataset is clustered with k-means clustering by using side-information of items. Then we implement the RWR method to each cluster.

## 3.3 Materials

### 3.3.1 Dataset

In this study, we utilized the Movielens dataset [13] including 3 data sets. The first is the main dataset. It contains users, items, and ratings. The second is the side-information of the users. It contains user_id, age, gender, occupation, zip-code of users. The last one is the side information of the items. It contains item_id, type, and year of items.

Data sets must be real-life to accurately measure the performance of the proposed recommender algorithms. Measuring the performance of algorithms with artificially created datasets can be misleading and inaccurate. The data sets in this area can be created by asking the real user to enter the data (personal, demographic, interest-like information, etc.) containing their personal information and preferences. As with many systems, collecting real data can be laborious, long-lasting, and costly. Due to such difficulties, the number of data sets created by real users to use in this area is very low. The movielens dataset created by Grouplens Research (grouplens.org) is the most widely used of these datasets. The Movielens data set was created at the University of Minnesota under the Grouplens Research Project. It is the product of a 7-month study between 19 September 1997 and 22 April 1998. This dataset was collected via the MovieLens website. There are 2 types of data used within the scope of the thesis. The first is called the main data. In this dataset, users who use the system actively evaluate the movies in the system. Some features of the data set used are listed below.

- There are 100000 ratings in the range of 1-5.
- 1682 movies.
- 943 users.

In the experiments carried out within the scope of the thesis, cross-validation was applied to the data set. The second is called side information. There are two side information data. The first contains information about users. The side information of the users includes age, gender, occupation, and zipcode, respectively. This data set has 11 different age ranges for 943 users. There are also 18 different occupations and 5 different zipcode. The second one contains information about the items. Side information of the item includes type and year information, respectively. The IMDb keyword dataset was created based on the keywords describing the movies. The keywords of the films refer to the genre of the film. There are 14 types of 1682 films in this data set.

In the main-dataset, there are 100,000 ratings, 943 users, and 1682 items. This dataset is used commonly for the recommender systems studies. Side information datasets for users/items are summarized in Tables 3.3.1.1.1 and 3.3.1.1.2, respectively.

| *No* | *Attribute-Description* | *Value* |
|---|---|---|
| 1 | User_id | 0-942 |
| 2 | Age | 1~10, 11~15,16~20, 21~25, 26~30, 31~35, 36~40, 41~45, 46~50, 51~55, 56~60, 61~65, 66~70,71~75 |
| 3 | Occupation | Technician, Writer, Other, Administrator, Executive, Student, Lawyer, Scientist, Educator, Entertainment, Programmer, Homemaker, Librarian, Artist, Engineer, Marketing, None, Healthcare, Retired, Salesman, Doctor |
| 4 | Gender | M, F |
| 5 | Zipcode | 2642-2661 |

**Table 3.3.1.1.1 Side information of users**

| *No* | *Attribute-Description* | *Value* |
|---|---|---|
| 1 | Genre | Unknown, Adventure, Action, Children's, Animation, Comedy, Crime, Documentary, Drama, Film-Noir, Fantasy, Horror, Musical, Mystery, Sci-Fi, Romance, Thriller, War, Western |
| 2 | Year | 1922-1998 |
| 3 | Item_id | 943,2624 |

**Table 3.3.1.1.2 Side information of items**

### 3.3.2 Performance Metrics

Various performance metrics are used to measure the accuracy of the results of the recommender systems studies. In this study, precision@k, Spearman's ρ, MAE and RMSE are used as performance metrics. The quality of the proposed Recommender Systems is determined by the level of accuracy and customization. While any algorithm can be more successful in one metric than another, it can fail in another metric. The most important criterion that measures the success of the recommender systems is accuracy. The correct functioning of a recommender system and producing logically acceptable results are important. The most important is to ensure the trust of its customers. Considering all these reasons, one of the most important factors that determine the quality of the recommender system algorithms is accuracy.

Statistical accuracy measures that measure the success of predictions made by a recommender system are techniques that measure success mathematically. Briefly, these are the measures that calculate the numerical distance of the estimate to real values. Statistical consistency metrics are the most common metrics used to compare the success of recommender systems. These methods measure the success of the system by comparing the recommenders produced by the system with the actual voting of the users. First, the average absolute error (MAE) metric is the calculation of the average of the difference between the actual votes the user gives to the products and the votes produced by the system. In short, the average absolute error is the average of the absolute values of the errors, as can be understood from the name.

**MAE.** is the average of absolute errors. The absolute difference of all the estimates produced by the system from the real value is divided by the number of estimates produced. In the equation, $\hat{y}_j$ represents the estimated value, and $y_j$ represents the real value. $n$ represents the number of estimates. The MAE value is inversely proportional to the success of the systems. The success of the system increases as the MAE value decreases.

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}}\left|\boldsymbol{y_j} - \boldsymbol{\hat{y}_j}\right| \tag{3.3.2.1}$$

**RMSE.** Root mean square error, another statistical metric, is used to evaluate success recommender systems. The RMSE value is inversely proportional to the success of the systems. The success of the system increases as the RMSE value decreases. In the equation, $\hat{y}_j$ represents the estimated value, and $y_j$ represents the real value. $n$ represents the number of estimates.

$$\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}}(\boldsymbol{y_j} - \hat{\boldsymbol{y}}_{\boldsymbol{j}})^{\mathbf{2}}} \tag{3.3.2.2}$$

Here, to get rid of the sign in front of the error, first, the squares of the errors are taken and then the square of the average error is obtained. The results obtained in this study were compared by both with the average absolute error and by calculating the average error squares sum with the root.

**Spearman's ρ.** We utilize Spearman's ρ. It is a ranking performance metric. Spearman's ρ indicates the correlation between a ground-truth ranked list with a ranked list. ρ can be in [−1, 1]. High ρ means that the rank of the predict list is similar to the truth ranked list.

As seen in Equation 3.3.2.3, ρ is calculated with an average of $\rho_u$. Equation 3.3.2.4 shows $\rho_u$. $\Omega_R$ test$_{[u]}$ represents a set of items. $s_{ui}$ represents the rank of i in a sorted list of items with predicted ratings. $s^*_{ui}$ represents the rank of i in a ranked list of items with actual ratings in the test set. $\bar{s}_{ui}$ represents the average of $s_{ui}$ for all items in $\Omega_R$ test$_{[u]}$, and $\bar{s}^*_{ui}$ represents the average of $s^*_{ui}$ for all items in $\Omega_R$ test$_{[u]}$.

$$\boldsymbol{\rho} = \frac{1}{|U|}\sum_{u \in U}\boldsymbol{\rho}_u \tag{3.3.2.3}$$

$$\boldsymbol{\rho}_u = \frac{\sum_{i\in\Omega_R} test_{[u]}\,(s_{ui} - \bar{s}_{ui})(s^*_{ui} - \bar{s}^*_{ui})}{\sqrt{\sum_{i\in\Omega_R} test_{[u]}\,(s_{ui} - \bar{s}_{ui})}\sqrt{\sum_{i\in\Omega_R} test_{[u]}\,(s^*_{ui} - \bar{s}^*_{ui})}} \quad (3.3.2.4)$$

**Precision@k.** We use precision@k as a metric. In this metric, k represents the number of top items of interest. Precision@k is the ratio of the number of actual positive items among the first k items in the recommendation list predicted. It can be within[0, 1]. High precision@k means it has better performance. precision@k is the average of precision@$k_u$. For a user, precision@$k_u$ is shown in Equation. $Actual_u(k)$ is a set of top-k items. U sorts these items with observed ratings in the test set.

$$\text{precision@k} = \frac{1}{|U|}\sum_{u\in U} \text{precision@k}_u \quad (3.3.2.5)$$

$$\text{precision@k}_u = \frac{|Actual_u(k) \cap Predicted_u(k)|}{\boldsymbol{k}} \quad (3.3.2.6)$$

## 3.4 Performance Results

In this study, various experiments are realized to observe the performance of the proposed recommender system. In these experiments, user-based MF, item-based MF, user-based RWR, and item-based RWR are implemented to the proposed model. We compare the results of the proposed recommender system and traditional recommender systems in terms of RMSE, MAE, precision@k, Spearman's ρ. The results are summarized in Table 3.4.1.

| ***Method*** | ***K*** | ***Spearman's ρ*** | ***Precision@k*** | ***MAE*** | ***RMSE*** |
|---|---|---|---|---|---|
| MF- user based | 1 | 0.313 | 0.145 | 0.770 | 0.958 |
| MF- user based | 2 | 0.343 | 0.166 | 0.677 | 0.848 |
| MF- user based | 4 | 0.357 | 0.163 | 0.593 | 0.744 |
| MF- user based | 5 | 0.383 | 0.154 | 0.640 | 0.793 |
| MF- user based | 8 | 0.487 | 0.171 | 0.770 | 0.958 |
| MF- user based | 10 | 0.553 | 0.187 | 1.172 | 1.453 |
| RWR-user based | 1 | 0.254 | 0.113 | 1.897 | 2.245 |
| RWR-user based | 2 | 0.259 | 0.111 | 1.848 | 2.198 |
| RWR-user based | 4 | 0.310 | 0.115 | 1.821 | 2.179 |
| RWR-user based | 5 | 0.330 | 0.122 | 1.747 | 2.101 |
| RWR-user based | 8 | 0.444 | 0.139 | 1.697 | 2.066 |
| RWR-user based | 10 | 0.515 | 0.171 | 1.779 | 2.174 |
| MF-item based | 1 | 0.419 | 0.350 | 0.709 | 0.885 |
| MF-item based | 2 | 0.421 | 0.315 | 0.682 | 0.854 |
| MF-item based | 4 | 0.454 | 0.354 | 0.617 | 0.773 |
| MF-item based | 5 | 0.464 | 0.325 | 0.568 | 0.712 |
| MF-item based | 8 | 0.574 | 0.348 | 0.548 | 0.689 |
| MF-item based | 10 | 0.613 | 0.367 | 0.538 | 0.678 |
| RWR-item based | 1 | 0.162 | 0.253 | 2.041 | 2.408 |
| RWR-item based | 2 | 0.208 | 0.247 | 1.968 | 2.355 |
| RWR-item based | 4 | 0.294 | 0.277 | 2.005 | 2.382 |
| RWR-item based | 5 | 0.284 | 0.232 | 1.908 | 2.276 |
| RWR-item based | 8 | 0.452 | 0.229 | 1.907 | 2.282 |
| RWR-item based | 10 | 0.569 | 0.222 | 1.876 | 2.260 |

*K: Cluster Number, MAE: Mean Absolute Error, RMSE: Root Mean Square Error*

**Table 3.4.1 Results of recommender systems**

### 3.4.1 User-based Model

First, the main dataset is clustered according to user clusters that are created by utilizing K-means and side information of users. Then both MF and RWR methods are applied to each cluster. As a result of these experiments on the Movielens dataset, precision@k, Spearman's ρ, RMSE, and MAE values are calculated. These results are presented in Table 3.4.1. In table 3.4.1, k is the cluster number. When k is 1, the main data is not clustered. Therefore, MF and RWR are applied to all data when k is 1. The fact that k is one means traditional MF is applied. In Table 3.4.1 as seen, the proposed model improves the performance of traditional MF and RWR.

Performance results show that the user-based MF model performs better than the user-based RWR model in terms of precision@k, Spearman's ρ metrics. The user-based MF shows lower MAE and RMSE. User-based MF gives the best result of Spearman's ρ, when the number of clusters is 10. User-based RWR performs the best when the number of clusters is 10. For example, when the cluster number is 10, the Spearman's ρ of the user-based MF is 0.553, and when the cluster number is 10, the Spearman's ρ of user-based RWR is 0.515. In addition, for MF, precision@k, values are 0.187, for RWR value are 0.171. As a result, we observe that the proposed model improves the accuracy of MF and RWR.

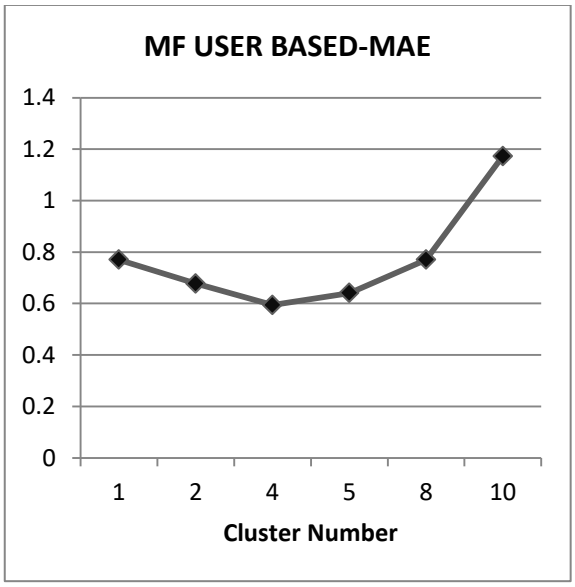


**Figure 3.4.1.1 MF-User based MAE**

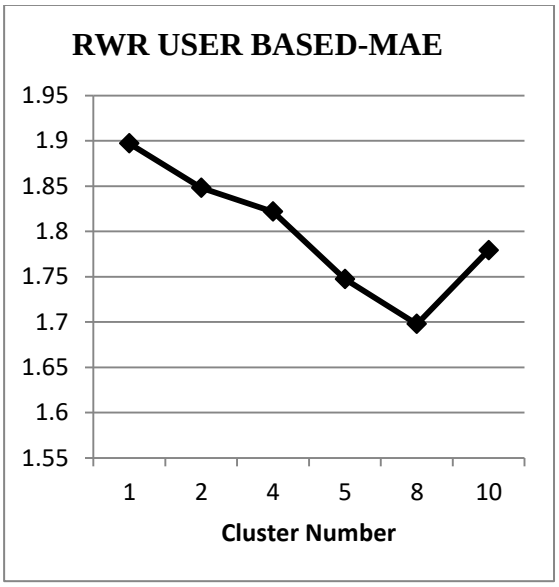


**Figure 3.4.1.2 RWR-User based MAE**

As seen in figure 3.4.1.1 and 3.4.1.2, proposed models are more successful than alone MF and RWR. Hence, using demographic information when clustering main data improves the performance of user-based collaborative filtering methods.

### 3.4.2 Item-based Model

Similar to the user-based model, the main dataset is clustered according to items clusters that are created by K-means clustering and side information of items. Then both MF and RWR methods are applied to each cluster. As a result of these experiments on the Movielens dataset, precision@k, Spearman's RMSE, and MAE values are calculated. These results are presented in Table 3.4.1. In table 3.4.1, k is the cluster number. When k is 1, the main data is not clustered. Therefore, MF and RWR are applied to all data when k is 1. The fact that k is one means traditional MF is applied. In Table 3.4.1 as seen, the proposed model improves the performance of traditional MF and RWR.

Performance results show that the item-based MF model performs better than the item-based RWR model in terms of precision@k, Spearman's ρ. The item-based MF shows lower MAE and RMSE for all cluster number. Item-based MF gives the best result of Spearman's ρ when the number of clusters is 10. Item-based RWR performs the best when the number of clusters is 10.

For example, when the cluster number is 10, the Spearman's ρ of the item-based MF is 0.613, and when the cluster number is 10, the Spearman's ρ of item-based RWR is 0.569. In addition, for MF, precision@k value is 0.367, for RWR value is 0.222. As a result, observe that the proposed model improves the accuracy of MF and RWR.

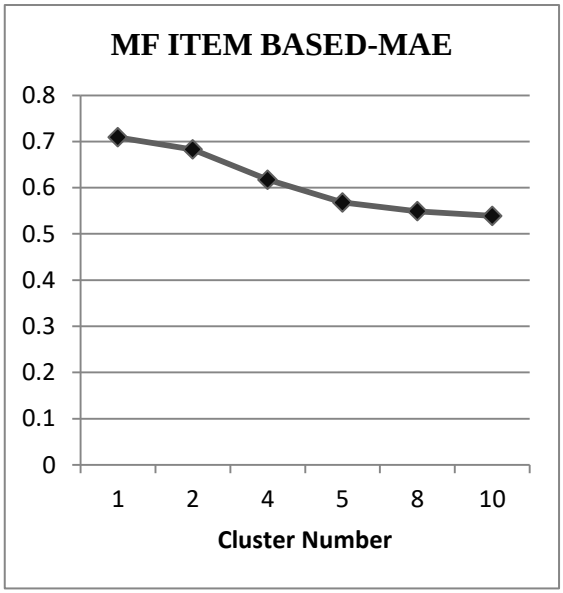


**Figure 3.4.2.1 MF-Item based MAE**

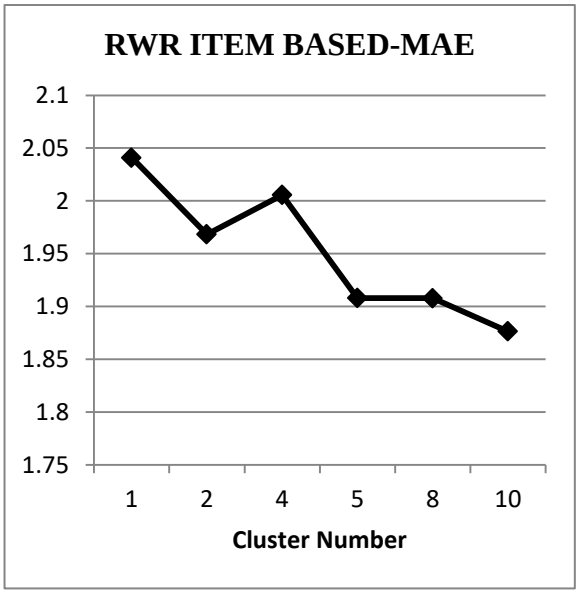


**Figure 3.4.2.2 RWR-Item based MAE**

As seen in figure 3.4.2.1 and 3.4.2.2, proposed models are more successful than alone MF and RWR. Hence, using demographic information when clustering the main data improves the performance of item-based collaborative filtering methods.

# Chapter 4

# Employee Attrition Prediction

## 4.1 Methods

In this thesis, we applied thirteen different classification algorithms and four different feature selection methods on two different HR datasets. These methods are Linear Discriminant Analysis (LDA), Naive Bayes, Bagging, AdaBoost, logistic regression, Support Vector Machine, J48, Random Forest, LogitBoost, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), XGBoost, Graph Convolutional Networks, GainRatio, Infogain, Relief, Chi-Square. In the following subsections, we briefly explain these methods.

The classification methods are realized using Weka and Python. XGboost and GCN are implemented using Python [27][28]. Other classification methods are applied using Weka. Used for experiments computer has Intel® Core™ i5-8300H CPU @ 2.30 GHz and 8.0 GB ram.

### 4.1.1 LogitBoost

Boosting is a general supervised learning method that generates a "strong" classifier from the "weak" classifiers. Logitboost immediately optimizes the possibility of the binomial log. LogitBoost considerably reduces the possibility of a positive loss function. LogitBoost is less sensitive to noisy data and changes linearly with an output error.

### 4.1.2 K-Nearest Neighbor (kNN)

The K-Nearest Neighbor (kNN) algorithm is one of the supervised machine learning algorithms. It is utilized in pattern recognition and data mining for

classification with a low error rate. The k samples that are most closely similar to a query is found by the algorithm. The category of the query point q is the same as the category of most of these examples.

### 4.1.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is an effective classification algorithm. The algorithm draws a vector between the two classes on the plane at the farthest distance from both classes to separate two classes. It is widely used for classification. Figure 4.1.3.1 shows the working principle of the SVM.

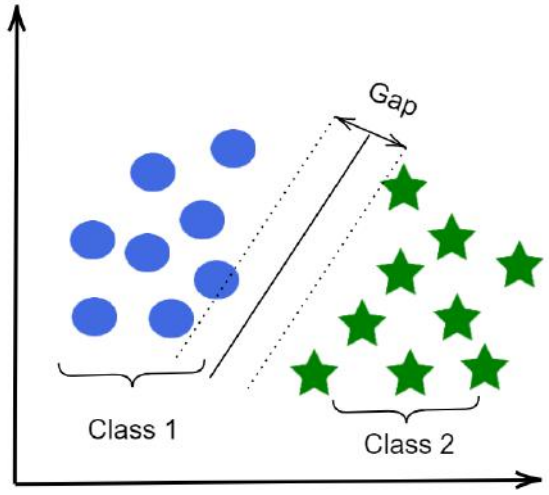


**Figure 4.1.3.1 SVM**

### 4.1.4 Bagging

Bagging predictor generates a predictor combined with multiple versions of a predictor. The algorithm deletes some examples or duplicates and changes the original training data each time.

### 4.1.5 J48

J48 has features like pruning decision trees, continuous feature value ranges, derivation of rules and it is an extension of ID3.

### 4.1.6 Random Forest (RF)

Random forest, a tree-based algorithm, is well known in machine learning problems. Random Forest (RF) is utilized for classification problems. The random

forest that works by producing multiple decision trees generates multiple random training subsets. Then it creates a tree with random training subsets.

### 4.1.7 AdaBoost

AdaBoost algorithm was generated by Robert Schapire and Yoav Freun. The algorithm is one of the important community methods. The AdaBoost has some advantages. These are robust theoretical foundations, very accurate predictions, great simplicity, and successful applications.

### 4.1.8 Logistic Regression

Logistic regression is a specific case of linear regression models. LR generates basic possibility classification formulas by using the maximum likelihood ratio when producing the equation. LR shows the statistical importance of the variables. LR is beneficial in status which the dependent variable is a dichotom. Figure 4.1.8.1 shows the working principle of Logistic Regression.

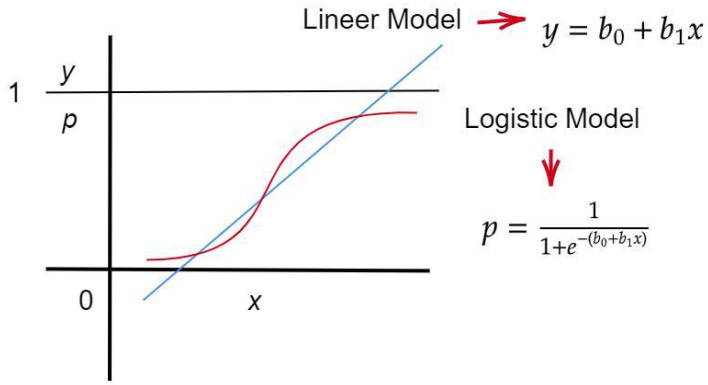


**Figure 4.1.8.1 Logistic Regression**

### 4.1.9 Naive Bayes (NB)

Naive Bayes classification technique is based on Bayes Theorem. The algorithm is a supervised learning algorithm. In a class, the features are not connected together according to Naive Bayes. Even if they are linked, it is assumed independently as probabilities. Naive Bayes is useful in unbalanced, large and small data sets. This algorithm can be better than other complex classification methods.

### 4.1.10 Linear Discriminant Analysis (LDA)

LDA is a supervised dimensionality reduction technique. This algorithm helps to separate the data different classes at the top level. LDA projects high-dimensional data to a lower-sized area and minimizes the simultaneous dispersion of data in the same class to obtain maximum class discrimination.

### 4.1.11 Multilayer Perceptron (MLP)

Multilayer perceptron (MLP) is an artificial neural network model. MLP generates multiple layers of nodes by matching the input data sets. MLP handles a supervised learning technique which named backpropagation for training the network.

### 4.1.12 XGBoost

XGBoost is an optimized distributed gradient boosting. It is a library intended to be highly efficient, flexible, and removable. XGBoost provides a parallel tree boosting to solve many data science problems [23].

XGBoost that adopts the principle of gradient boosting, which is a boosted tree algorithm. XGBoost utilizes a more regularized-model to control over-fitting then other gradient boosted machines. The function $f_t$ contains each the structure of the tree and the leaf scores. This is formalized as:

$$f_t(x) = w_{q(x)}\, w \in R^T\,, q: R^d \rightarrow \{1,2,\cdots,T\} \tag{4.1.12.1}$$

When ‘q’ is a function assigning each data point to the corresponding leaf, ‘w’ is the vector of scores on leaves. ‘T’ is the number of leaves. The model is formulated as:

$$\Omega(f) = \gamma T\ + \frac{1}{2}\,\lambda \sum_{j=1}^{T} w_j^2 \tag{4.1.12.2}$$

In the t-th iteration, the objective function at the is as:

$$Obj^{(t)}\ = \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T \tag{4.1.12.3}$$

In the above formula solving this quadratic, the best $w_j$ for a given $q(x)$ and the best objective reduction is:

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{4.1.12.4}$$

$$Obj^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{sqr(G_j)}{H_j + \lambda} + \gamma T \tag{4.1.12.5}$$

A leaf into 2 leaves is splitted and formed score gained is as seen:

$$Gain = \frac{1}{2}\left[\frac{sqr(GL)}{HL + \lambda} + \frac{sqr(GR)}{HR + \lambda} - \frac{sqr(GL + GR)}{HL + HR + \lambda}\right] - \gamma \tag{4.1.12.6}$$

In here, they are $G_j = \sum_{i \in Ij} g_i$ and $H_j = \sum_{i \in Ij} h_i$.

### 4.1.13 Graph Convolutional Network (GCN)

GCN that applies machine learning on graphs is so powerful neural network architecture. Actually, they are very strong because of even a randomly initiated 2-layer GCN can generate useful feature presentations of nodes in networks[25].

GCNs are generalizations of conventional convolutional neural networks (CNNs) on graphs. Furthermore, GCNs are simplified models of graph convolutional neural networks (GCNNs). CNNs are similarly, given the feature vectors of all nodes in the graph, GCNs learn a new feature representation for each node in the graph over multiple neural network layers which are then used as input to the final classifier. In the GCN, the input to the $\ell$th graph convolution layer is an activation matrix denoted $H^{(\ell-1)}$. The activation matrix denoted $H^{(\ell)}$ is the output of the layer. The input to the initial layer is, therefore, a feature matrix.

$H^{(0)} = X$ (4.1.13.1)

H is updated in three steps in each graph convolution layer. These steps are feature propagation, linear transformation, and the implementation of a nonlinear activation function.

**Feature Propagation.** propagates along with the graph. In each layer, the arriving features of each node $v_i \in V$ are collected with the arriving features of the nodes which are in the neighborhood of $v_i$ in $G$. In other words, describing the convolution matrix as $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ where $\tilde{A} = I + A$ **and** $\tilde{D} = I + D$ (i.e., adding self-loops for each node

in the adjacency matrix and the diagonal degree matrix), the update for all nodes transforms a single matrix multiplication:

$\widehat{H}^{(\ell)} = \hat{A}H^{(\ell-1)}$ (4.1.13.2)

Clearly, this step that causes incident nodes to have similar features is utilized to similar predictions for neighboring nodes.

**Linear Transformation and Point-wise Nonlinear Activation.** In each step of the GCN, the feature matrix is smoothed along with the graph. This feature matrix is put thought to linear transformation utilizing a trainable weight matrix $\Theta^{(l)}$. A nonlinear function, such as $ReLU = max(x,.)$, is utilized to generate the output activation matrix for that step:

$H^{(l+1)} = ReLU(\widehat{H}^{(l)}\Theta^{(l)})$ (4.1.13.3)

**Node Classification.** The final GCN layer predicts the unknown labels of nodes. Let $\hat{Y} \in R^{n\times k}$ indicate the class prediction matrix, where $\hat{y}_{ij}$ demonstrates the probability that the node $v_i \in V$ be in class j for $1 \leq j \leq k$. k shows the number of classes. In the final step, the class prediction matrix is calculated as:

$\hat{Y} = softmax(\hat{A}H^{(L-1)}\Theta^{(L)})$ (4.1.13.4)

## 4.1.14 Chi-Square

Chi-square is utilized to rank categorical attributes in a dataset. In this method, Chi-square is calculated between the target and each feature. The request number of features is selected with the best Chi-square scores. Chi-square score is shown equation:

$$x^2 = \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$ (4.1.13.4)

## 4.1.15 Information Gain

Information Gain that is commonly utilized in machine learning is an entropy-based feature selection method. Information Gain is utilized in feature selection, it is

introduced as the amount of information that is generated by the feature of items for the text category.

### 4.1.16 Gain Ratio

The gain ratio is used commonly to rank the attributes of the datasets as a filter feature subset approach namely. Gain ratio (GR) reduces the bias of the information gain. Gain ratio considers the number and size of branches when choosing a feature.

### 4.1.17 Relief

Relief is a feature selection algorithm. It was generated for the implementation of binary classification problems. Relief computes a feature score for each feature. It ranks features for feature selection. Relief calculates feature scoring by defining of feature value differences.

## 4.2 Materials

### 4.2.1. Dataset

In this study, we have studied with two different HR datasets of different companies, i.e. IBM and Adesso, which is a private company in Turkey. In Table 4.2.1.1, the HR data set of Adesso feature descriptions are given. This dataset is a real-world dataset. Overall, there are 9 features and 532 samples. 9 of the features are numeric. The largest and the smallest values of each feature are also shown. Attrition value of 296 samples is “no” while attrition value of 236 samples is “yes”. EmployeeNumber does not affect employee attrition prediction and hence, we did not use this attribute.

In Table 4.2.1.2, the IBM HR data set feature descriptions are given [26]. Overall, there are 35 features and 1470 samples in IBM HR data set. 26 of the features are numeric and the others are categorical. The largest and the smallest values of each feature are also shown. The attrition value of 1233 samples is “no” while the attrition value of 237 samples is “yes”. EmployeeCount, Over18, and StandartHours are the same values for each sample. EmployeeNumber does not affect employee attrition prediction and we did not use this attribute.

| *No* | *Attribute-Description* | *Value* | *No* | *Attribute-Description* | *Value* |
|---|---|---|---|---|---|
| 1 | Age | 18-60 | 19 | MonthlyIncome | 1009-19999 |
| 2 | Attrition | Yes,No | 20 | MonthlyRate | 2094-26999 |
| 3 | BusinessTravel | Travel_Rarely,Travel_Frequently,Non-Travel | 21 | NumCompaniesWorked | 0-9 |
| 4 | DailyRate | 102-1499 | 22 | Over18 | Y |
| 5 | Department | Sales,Research &Development, Human Resources | 23 | OverTime | Yes, No |
| 6 | DistanceFromHome | 1-29 | 24 | PercentSalaryHike | 11-25 |
| 7 | Education | 1-5 | 25 | PerformanceRating | 3-4 |
| 8 | EducationField | Life Sciences,Other, Medical, Marketing, Technical Degree, Human Resources | 26 | RelationshipSatisfaction | 1-4 |
| 9 | EmployeeCount | 1 | 27 | StandardHours | 80 |
| 10 | EmployeeNumber | 1-2068 | 28 | StockOptionLevel | 0-3 |
| 11 | EnvironmentSatisfaction | 1-4 | 29 | TotalWorkingYears | 0-40 |
| 12 | Gender | Female,Male | 30 | TrainingTimesLastYear | 0-6 |
| 13 | HourlyRate | 30-100 | 31 | WorkLifeBalance | 1-4 |
| 14 | JobInvolvement | 1-4 | 32 | YearsAtCompany | 0-40 |
| 15 | JobLevel | 1-5 | 33 | YearsCurrentRole | 0-18 |
| 16 | JobRole | Sales Executive, Research Scientst,Laboratory Technician, Manufacturing Director, Healthcare Representative, Manager, Sales Representative, Research Director, Human Resources | 34 | YearsSinceLastPromotion | 0-15 |
| 17 | JobSatisfaction | 1-4 | 35 | YearsWithCurrManager | 0,17 |
| 18 | MaritalStatus | Single, Married, Divorced | | | |

**Table 4.2.1.1 IBM HR dataset description**

| *No* | *Attribute-Description* | *Value* |
|---|---|---|
| 1 | Age | 20-53 |
| 2 | Attrition | Yes, No |
| 3 | Role | 1-142 |
| 4 | Department | 0-9 |
| 5 | Working_days | 0-2453 |
| 6 | Location | 0-8 |
| 7 | MilitaryStatus | 0-2 |
| 8 | EducationStatus | 0-4 |
| 9 | EmployeeNumber | 103-640 |

**Table 4.2.1.2 Hr dataset of ADESSO description**

| **ChiSquare** | |
|---|---|
| Rank | Attribute |
| 1 | Role |
| 2 | Department |
| 3 | EducationStatus |
| 4 | MilitaryStatus |
| 5 | Age |
| 6 | Working_days |
| 7 | Location |

| **Infogain** | |
|---|---|
| Rank | Attribute |
| 1 | Role |
| 2 | Department |
| 3 | EducationStatus |
| 4 | MilitaryStatus |
| 5 | Age |
| 6 | Working_days |
| 7 | Location |

| **GainRatio** | |
|---|---|
| Rank | Attribute |
| 1 | Role |
| 2 | Department |
| 3 | EducationStatus |
| 4 | Age |
| 5 | MilitaryStatus |
| 6 | Working_days |
| 7 | Location |

| **RelieF** | |
|---|---|
| Rank | Attribute |
| 1 | Role |
| 2 | Department |
| 3 | Working_days |
| 4 | EducationStatus |
| 5 | MilitaryStatus |
| 6 | Location |
| 7 | Age |

**Table 4.2.2.1. Feature Selection Methods Rank for Adesso HR Dataset**

### 4.2.2 Feature Selection

In this study, we have applied four different feature selection methods, i.e., chi-square, info gain, gain ratio, relief. They are shown with their rank of attributes in tables 4.2.2.1 and 4.2.2.2. For Adesso HR dataset, the most important features are role, department, educationStatus. For IBM Dataset, the most important features are overtime, totalworkingyears, age, joblevel, monthlyincome, yearsatcompany. Although the best ones are not exactly similar, the department and jobrole are the most important attributes in both datasets. For IBM dataset, Info gain and Chi-Square perform better. For Adesso dataset, Info Gain, Chi-Square and Gain ratio give the same rank and perform better.

| **Infogain** | | **ChiSquare** | | **GainRatio** | | **RelieF** | |
|---|---|---|---|---|---|---|---|
| R | Attribute | R | Attribute | R | Attribute | R | Attribute |
| 1 | OverTime | 1 | TotalWorkingYears | 1 | OverTime | 1 | OverTime |
| 2 | TotalWorkingYears | 2 | OverTime | 2 | JobRole | 2 | Gender |
| 3 | Age | 3 | Age | 3 | YearsWithCurrManager | 3 | MaritalStatus |
| 4 | JobLevel | 4 | MonthlyIncome | 4 | MonthlyIncome | 4 | JobLevel |
| 5 | MonthlyIncome | 5 | JobLevel | 5 | YearsAtCompany | 5 | JobRole |
| 6 | StockOptionLevel | 6 | YearsWithCurrManager | 6 | JobLevel | 6 | EnvironmentSatisfaction |
| 7 | YearsAtCompany | 7 | YearsAtCompany | 7 | Age | 7 | JobSatisfaction |
| 8 | YearsWithCurrManager | 8 | StockOptionLevel | 8 | TotalWorkingYears | 8 | StockOptionLevel |
| 9 | YearsInCurrentRole | 9 | YearsInCurrentRole | 9 | StockOptionLevel | 9 | MonthlyIncome |
| 10 | MaritalStatus | 10 | MaritalStatus | 10 | JobInvolvement | 10 | RelationshipSatisfaction |
| 11 | JobRole | 11 | JobRole | 11 | MaritalStatus | 11 | BusinessTravel |
| 12 | BusinessTravel | 12 | BusinessTravel | 12 | WorkLifeBalance | 12 | Age |
| 13 | EnvironmentSatisfaction | 13 | EnvironmentSatisfaction | 13 | YearsInCurrentRole | 13 | Department |
| 14 | JobInvolvement | 14 | JobInvolvement | 14 | EnvironmentSatisfaction | 14 | WorkLifeBalance |
| 15 | WorkLifeBalance | 15 | WorkLifeBalance | 15 | BusinessTravel | 15 | PerformanceRating |
| 16 | JobSatisfaction | 16 | JobSatisfaction | 16 | JobSatisfaction | 16 | TotalWorkingYears |
| 17 | Department | 17 | Department | 17 | Department | 17 | NumCompaniesWorked |

*R: Rank*

**Table 4.2.2.2. Feature Selection Methods Rank for IBM HR Dataset**

### 4.2.3 Performance Metrics

**Classification Accuracy** is the ratio of number of correct predictions to the total number of input samples. The accuracy can be defined as the percentage of correctly classified instances (TP + TN) / (TP + TN + FP + FN). where TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively.

**Sensitivity and Specificity.** are two of the other commonly used assessment criteria for recommender systems. In addition, Specificity and Sensitivity provide more information, especially in the classifications made on unbalanced distributed data sets. In order to calculate Specificity and Sensitivity, items must be classified into relevant and unrelated. Specificity is a classification metric that measures the ratio of negative patterns that are correctly classified. Sensitivity is a classification metric that measure the ratio of positive patterns that are correctly classified. In the equations, sensitivity and specificity are seen:

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{True\ Positive}}{\boldsymbol{True\ Positive + False\ Negative}} \quad (4.2.3.1)$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{True\ Negative}}{\boldsymbol{True\ Negative + False\ Positive}} \quad (4.2.3.2)$$

**F-measure.** Precision and recall need to be considered together to compare different recommender systems algorithms. The F-measure converts precision and recall into a single value. The F-measure is often used in the statistical analysis of the classification of data. In the equation, P represents precision and R represents recall.

$$\boldsymbol{F - measure} = \boldsymbol{2PR/(P + R)} \quad (4.2.3.3)$$

The f-measure value, which is calculated by using the above-described “called” and “completeness” values, is calculated by taking the harmonic average of the called and completeness values. The higher these metrics, the higher the accuracy of the recommenders made. Using these evaluation criteria, the accuracy of the recommenders produced with the proposed models was observed.

## 4.3 Performance Results

This section presents the performance results of employee attrition prediction based on two different HR datasets of different companies, i.e. IBM and Adesso, which is a private company in Turkey. In Tables 4.2.3.1 and 4.2.3.2, we present all the classification results. In our experiments, we applied four different feature selection methods on both datasets. Feature selection methods generally increased the accuracy of the classification methods. For IBM dataset, Info gain and Chi-Square perform better.

For Adesso dataset, Info Gain, Chi-Square, and Gain ratio give the same rank and perform better. We have also observed that the Logistic Regression achieves the highest accuracy of 87.34% for IBM HR Dataset, the accuracy of Random Forest, LogitBoost (bl: Random Forest), Bagging achieve the highest accuracy of 83.27% for Adesso HR Dataset.

Importantly, GNN generates acceptable results in terms of identifying regular and irregular patterns, and hence achieves a substantial improvement from existing methods and the XGBoost classifier outperforms the other classifiers in terms of accuracy. In the Adesso HR dataset, Random Forests achieves the highest accuracy with its property that trusts its stages of randomization to help it achieve better generalization. In the IBM dataset, Logistic Regression (LR) achieves the highest accuracy because when the AUC of the best model is below 0.8, and the LR outperformed compared to other algorithms.

Although the algorithms with the highest accuracy and lowest accuracy for both datasets are not the same, Adaboost and Random Forest algorithms show high accuracies for both datasets. For IBM HR Dataset, the sensitivity of Logistic Regression, which achieves the highest sensitivity, is 87.3%. The F-measure of Logistic Regression, that achieves the highest F-measure, is 0.856. For Adesso HR Dataset, the

sensitivity of Bagging, which achieves the highest sensitivity, is 83.3%. The F-measure of Bagging, that achieves the highest F-measure, is 0.828.

| Method | FS | SN | SP | F-Measure | AUC | Accuracy |
|---|---|---|---|---|---|---|
| SVM | No | 85.9% | 30.9% | 0.816 | 0.579 | 85.64% |
| SVM | Relief | 83.9% | 21.0% | 0.785 | 0.527 | 83.87 % |
| Random Forest | No | 85.9% | 30.1% | 0.818 | 0.788 | 85.91 % |
| Random Forest | Info Gain | 86.3% | 37.0% | 0.835 | 0.794 | 86.32 % |
| LogitBoost(bl: Random Forest) | No | 85.9% | 29.5% | 0.817 | 0.812 | 85.91 % |
| LogitBoost(bl: Random Forest) | Info Gain | 86.3% | 39.1% | 0.837 | 0.877 | 86.25 % |
| MLP | No | 84.3% | 50.6% | 0.837 | 0.778 | 84.28 % |
| MLP | Relief | 82.7% | 43.8% | 0.817 | 0.748 | 82.65 % |
| KNN(k=3) | No | 83.5% | 33.8% | 0.809 | 0.655 | 83.53 % |
| KNN(k=3) | Chi-square | 84.7% | 40.5% | 0.828 | 0.704 | 84.69 % |
| KNN(k=5) | No | 84.1% | 27.1% | 0.799 | 0.685 | 84.08% |
| KNN(k=5) | Chi-square | 85.2% | 36.5% | 0.825 | 0.734 | 85.17% |
| LDA | No | 86.7% | 44.3% | 0.848 | 0.814 | 86.66% |
| LDA | Chi-square | 86.1% | 41.1% | 0.839 | 0.791 | 86.12% |
| J48 | No | 82.9% | 40.5% | 0.814 | 0.581 | 82.85 % |
| J48 | Info Gain | 83.9% | 42.6% | 0.825 | 0.607 | 83.95 % |
| Naive Bayes | No | 79.1% | 63.9% | 0.807 | 0.758 | 79.11 % |
| Naive Bayes | Chi-square | 79.8% | 63.7% | 0.812 | 0.752 | 79.79 % |
| Bagging | No | 85.3% | 32.8% | 0.816 | 0.570 | 82.99 % |
| Bagging | Gain Ratio | 85.8% | 38.3% | 0.833 | 0.778 | 85.78 % |
| AdaBoost | No | 86.7% | 38.5% | 0.840 | 0.782 | 86.73 % |
| AdaBoost | Gain Ratio | 85.7% | 34.6% | 0.829 | 0.777 | 85.71 % |
| Logistic Regression | No | 87.3% | 46.4% | 0.856 | 0.817 | 87.34 % |
| Logistic Regression | Gain Ratio | 86.4% | 39.8% | 0.840 | 0.791 | 86.39% |
| XGBoost | No | 31.6% | 96.3% | 0.474 | 0.639 | 85.80% |
| XGBoost | Info Gain | 32.9% | 95.0% | 0.488 | 0.639 | 84.98% |
| GCN | No | 50.4% | 70.0% | 0.656 | 0.645 | 85.80% |

*FS: Feature Selection, SN: Sensitivity, SP: Specificity, AUC: Area Under Curve*

**Table 4.2.3.1 Results of classification algorithms on IBM dataset**

| Method | FS | SN | SP | F-Measure | AUC | Accuracy |
|---|---|---|---|---|---|---|
| SVM | No | 55.1% | 51.4% | 0.539 | 0.532 | 55.07% |
| SVM | Yes | 54.5% | 48.4% | 0.509 | 0.515 | 54.51 % |
| Random Forest | No | 82.0% | 80.0% | 0.816 | 0.887 | 81.76 % |
| Random Forest | Yes | 82.9% | 80.8% | 0.827 | 0.889 | 83.27 % |
| LogitBoost(bl:Random Forest) | No | 82.3% | 80.8% | 0.822 | 0.885 | 82.33 % |
| LogitBoost(bl:Random Forest) | Yes | 82.3% | 80.7% | 0.822 | 0.889 | 83.27 % |
| MLP | No | 73.5% | 70.7% | 0.730 | 0.783 | 73.49 % |
| MLP | Yes | 74.8% | 71.0% | 0.739 | 0.802 | 74.81 % |
| KNN(k=3) | No | 75.2% | 74.1% | 0.751 | 0.796 | 75.18 % |
| KNN(k=3) | Yes | 77.4% | 75.8% | 0.773 | 0.796 | 77.44 % |
| KNN(k=5) | No | 74.8% | 73.8% | 0.748 | 0.798 | 74.81 % |
| KNN(k=5) | Yes | 78.8% | 77.0% | 0.786 | 0.801 | 78.75% |
| LDA | No | 57.1% | 54.4% | 0.566 | 0.637 | 57.14% |
| LDA | Yes | 57.9% | 54.9% | 0.572 | 0.638 | 57.89 % |
| J48 | No | 82.3% | 79.7% | 0.820 | 0.817 | 82.33 % |
| J48 | Yes | 82.3% | 79.7% | 0.820 | 0.812 | 82.33 % |
| Naive Bayes | No | 66.2% | 67.4% | 0.662 | 0.767 | 66.16 % |
| Naive Bayes | Yes | 66.5% | 68.8% | 0.664 | 0.730 | 66.54 % |
| Bagging | No | 83.3% | 80.3% | 0.828 | 0.885 | 83.27 % |
| Bagging | Yes | 82.5% | 79.7% | 0.821 | 0.882 | 82.51 % |
| AdaBoost | No | 81.2% | 79.0% | 0.809 | 0.862 | 81.20 % |
| AdaBoost | Yes | 81.2% | 79.0% | 0.809 | 0.864 | 81.20 % |
| Logistic Regression | No | 57.5% | 55.0% | 0.570 | 0.637 | 57.51 % |
| Logistic Regression | Yes | 58.1% | 55.2% | 0.574 | 0.638 | 58.08 % |
| XGBoost | No | 60.0% | 85.7% | 0.705 | 0.728 | 64.86% |
| XGBoost | Yes | 63.3% | 57.1% | 0.600 | 0.602 | 62.16% |
| GCN | No | 50.0% | 90.0% | 0.637 | 0.724 | 75.90% |

*FS: Feature Selection, SN: Sensitivity, SP: Specificity, AUC: Area Under Curve*

**Table 4.2.3.2 Results of classification algorithms on Adesso HR dataset**

# Chapter 5

# Conclusions and Future Prospects

## 5.1 Conclusions

This thesis includes combining two research studies. In the first part of the study includes the movie suggestion. In this part, a hybrid recommender system is proposed and it is aimed to increase the performance of recommender systems. In order to evaluate the proposed model, matrix factorization and random walk with restart are utilized. These collaborative filtering methods are implemented to the hybrid model, both the user-based and item-based. In the proposed model, users/items are clustered with k-means clustering using the side-information of users/items. MF and RWR are applied to each cluster. To compare with the proposed model, traditional matrix factorization and random walk with restart are tested. In this study, the MovieLens dataset is used. The performance of different recommender systems is analyzed by utilizing Mean absolute error (MAE) and Root Mean Square Error (RMSE) in many studies. Both methods are used in the performance analysis of the recommender systems within the scope of the study. Besides, precision@k, Spearman's ρ metrics are used to compare recommender systems. MF performs better than RWR on both user-based/item-based models. In the user-based proposed model, MF achieved 0.553 Spearman's ρ. In the item-based proposed model, MF achieved 0.613 Spearman's ρ. The results of the experiments indicate the proposed hybrid model is more successful than traditional methods. Also, it is observed that MAE and RMSE of the proposed model are lower than traditional methods and the precision@k, Spearman's ρ are higher. The proposed hybrid recommender system created in this study can be applied to any recommender system. Datasets of applications using recommender systems are very big compared to the dataset used in this study. The method developed in this study can be applied to large datasets.

In the second part of the study includes the employee attrition prediction. In this study, different classification methods are applied, such as Linear discriminant analysis (LDA), Naive Bayes, Bagging, AdaBoost, logistic regression, Support Vector Machine (SVM), Random Forest, J48, LogitBoost, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), XGBoost, Graph Convolutional Networks, to address the employee attrition prediction problem. Two different datasets are utilized to evaluate performance. One of the datasets is a real-world dataset. Besides, four different feature selection methods are used to improve performance results, such as chi-square, info gain, gain ratio, and relief. Different from existing studies, we extensively evaluate the performance of state-of-the-art methods for various evaluation measures. To the best of our knowledge, GCN has not been utilized for the attrition problem. Although Bagging, Random Forest and LogitBoost give the highest accuracy of 83.27% on the Adesso dataset, Logistic Regression gives the highest accuracy of 87.34% on the IBM HR dataset. Feature selection increases the accuracy of most of the classification methods for employee attrition on both datasets. Performance results show that data mining methods, such as LogitBoost and Logistic Regression algorithms, can be very useful for predicting employee attrition. Further, GCN is also a successful method to predict employee attrition.

## 5.2 Contribution to Global Sustainability

In the first part of this thesis, we study movie suggestion. Recommender systems recommend related items to the user from billions of items. Accessing related items is important to users, it would be beneficial to advise the user. In this study, a user/item clustering-based model is proposed to improve the performance of traditional collaborative filtering. As a result of our experiments, the proposed model improves performance for both user-based and item-based methods. Therefore, this study helps both users and online-platforms.

In the second part of this thesis, we study employee attrition prediction. The methods that can be successful in predicting employee attrition are examined in detail. The most successful methods are determined in order to predict employee attrition. We expand performance metrics to compare classification methods. To the best of our

knowledge, GCN has not been utilized for the attrition problem. Therefore, we use GCN to analyze its prediction performance for employee attrition.

## 5.3 Future Prospects

As future work for movie suggestions, this method can be reconsidered in recommender systems without Matrix Factorization and Random Walk with Restart. It would be beneficial to see the performance of this method on other recommender systems. Therefore, we can have an enhanced algorithm, which can apply different recommender systems. Second, during our experiments, we utilized some side-information of users/items. There are many distinct side-information. We plan to include new attributes as side-information. If we add more attributes, we are able to model user's preferences more efficiently. Third, the performance of the proposed model can be analyzed in other recommender systems databases (i.e Lastfm). Thus, we can verify the feasibility of the results. Fourth, we plan to optimize the hyperparameters of methods. Finally, we want to mention the clustering method we used. Clustering makes groups of users/items. So cluster analysis is a critical step in the proposed model. So we also plan to try other clustering algorithms.

As a future direction for employee attrition prediction problems studied in this thesis, deep learning methods and ensemble methods developed recently can be implemented and combined with the existing method for prediction. This study can be expanded with new Deep Learning Methods and ensemble methods. It would be beneficial to see the performance of employee attrition prediction on other methods. Therefore, we can have an enhanced study, which can better predict employee attrition. We plan to optimize the hyperparameters of classification methods. Finally, we can also study with new big data sets, to analyze which method is best.

# BIBLIOGRAPHY


[1] J. Leino, User Factors in Recommender Systems: Case Studies in e-Commerce, News Recommending, and e-Learning, Tampere, 2014.

[2] S.-M. Choi, S.-K. Ko, Y.-S. Han, A movie recommendation algorithm based on genre correlations, Expert Syst. Appl. 39 (2012) 8079–8085,http://dx.doi.org/10.1016/j.eswa.2012.01.132.

[3] Q. Li, S.H. Myaeng, B.M. Kim, A probabilistic music recommender considering user opinions and audio features, Inf. Process. Manage. 43 (2007) 473–487,http://dx.doi.org/10.1016/j.ipm.2006.07.005.

[4] Zare, Hadi, Mina Abd Nikooie Pour, and Parham Moradi. "Enhanced recommender system using predictive network approach." *Physica A: Statistical Mechanics and its Applications* 520 (2019): 322-337.

[5] Basile, Pierpaolo, et al. "Bridging the gap between linked open data-based recommender systems and distributed representations." *Information Systems* 86 (2019): 1-8.

[6] Bogaert, Matthias, et al. "Evaluating multi-label classifiers and recommender systems in the financial service sector." *European Journal of Operational Research* 279.2 (2019): 620-634.

[7] Yin, Ruiping, et al. "A deeper graph neural network for recommender systems." *Knowledge-Based Systems* 185 (2019): 105020.

[8] Gunawan, Alexander AS, and Derwin Suhartono. "Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks." *Procedia Computer Science* 157 (2019): 99-109.

[9] Pujahari, Abinash, and Vineet Padmanabhan. "Group recommender systems: Combining user-user and item-item collaborative filtering techniques." *2015 International Conference on Information Technology (ICIT)*. IEEE, 2015.

[10] Kuzelewska, Urszula, and Arkadiusz Kuryłowicz. "Multi-Clustering Applied to Collaborative Recommender Systems." *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. IEEE, 2018.

[11] Bag, Sujoy, Abhijeet Ghadge, and Manoj Kumar Tiwari. "An integrated recommender system for improved accuracy and aggregate diversity." *Computers & Industrial Engineering* 130

[12] Park, Haekyu, Jinhong Jung, and U. Kang. "A comparative study of matrix factorization and random walk with restart in recommender systems." *2017.*

[13] Movielens 100k dataset | Grouplens. Retrieved June 6, 2020 from:https://grouplens.org/datasets/movielens/100k/

[14] Sisodia, Dilip Singh, Somdutta Vishwakarma, and Abinash Pujahari. "Evaluation of machine learning models for employee churn prediction." *2017 International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 2017.

[15] Hebbar, A. Rohit, et al. "Comparison of Machine Learning Techniques to Predict the Attrition Rate of the Employees." *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2018.

[16] Brockett, Neil, et al. "A System for Analysis and Remediation of Attrition." *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019.

[17] Yiğit, İbrahim Onuralp, and Hamed Shourabizadeh. "An approach for predicting employee churn by using data mining." *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2017.

[18] Jain, Rachna, and Anand Nayyar. "Predicting Employee Attrition using XGBoost Machine Learning Approach." *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE, 2018.

[19] Yadav, Sandeep, Aman Jain, and Deepti Singh. "Early Prediction of Employee Attrition using Data Mining Techniques." *2018 IEEE 8th International Advance Computing Conference (IACC)*. IEEE, 2018.

[20] Alduayj, Sarah S., and Kashif Rajpoot. "Predicting Employee Attrition using Machine Learning." *2018 International Conference on Innovations in Information Technology (IIT)*. IEEE, 2018.

[21] Yedida, Rahul, et al. "Employee Attrition Prediction." *arXiv preprint arXiv:1806.10480* (2018).

[22] Saradhi, V. Vijaya, and Girish Keshav Palshikar. "Employee churn prediction." *Expert Systems with Applications* 38.3 (2011): 1999-2006.

[23] Ajit, Pankaj. "Prediction of employee turnover in organizations using machine learning algorithms." *algorithms* 4.5 (2016): C5.

[24] Ozdemir, Fatma, et al. "Assessing Employee Attrition Using Classification" ICISMD 2020.

[25] Coskun, Mustafa, Burcu Bakir Gungor, and Mehmet Koyuturk. "Expanding Label Sets for Graph Convolutional Networks." *arXiv preprint arXiv:1912.09575* (2019).

[26] IBM HR dataset | IBM. Retrieved June 6, 2020 from: https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

[27] Klicpera, Johannes, Aleksandar Bojchevski, and Stephan Günnemann. "Predict then propagate: Graph neural networks meet personalized pagerank." *arXiv preprint arXiv:1810.05997* (2018).

[28] Browlee, Jason (2020) XGBoost source code [Source code]. http:// machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/