Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Caroline Barstad
Hanna Torjusen

Privacy in Recommender Systems

Inferring User Personality Traits From Personalized Movie Recommendations

Master's thesis in Computer Science
Supervisor: Özlem Özgöbek
June 2023

NTNU
Norwegian University of
Science and Technology

Caroline Barstad
Hanna Torjusen

Privacy in Recommender Systems

Inferring User Personality Traits From Personalized Movie Recommendations

Master's thesis in Computer Science
Supervisor: Özlem Özgöbek
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Privacy in Recommender Systems
Inferring User Personality Traits From Personalized Movie Recommendations

Caroline Barstad and Hanna Torjusen

Supervised by Özlem Özgöbek

June, 2023

Abstract

Recommender systems are personalized systems that collect data about the users to recommend and tailor the content. This thesis investigates inferring personality traits from personalized top 10 movie recommendations to investigate the potential leakage from this user data. The research explores the impact on classification accuracy using different strategies, namely various personality trait splits, integrating personality traits into recommender systems, and resampling techniques. In the experiments, random, rating-based, and personality-based recommendations were generated, and six separate classifiers were used to infer personality traits from them.

Findings indicate potential information leakage of personality from personalized recommendations. Still, no consistent pattern was observed across all personality traits, as different experimental setups gave different results. Additionally, no significant difference in classification accuracy was observed after incorporating the personality traits in the recommender system. These findings contribute to understanding privacy concerns from user recommendations and offer insights for future research in recommender system privacy.

Sammendrag

Anbefalingssystemer er personaliserte systemer som samler inn brukerdata for å anbefale og skreddersy innhold. Denne masteroppgaven undersøker hvordan personlighetstrekk kan utledes fra personaliserte topp 10-filmanbefalinger for å undersøke potensiell lekkasje fra brukerdataen. Oppgaven utforsker effekten på treffsikkerheten av klassifikasjonen ved hjelp av ulike metoder, nærmere bestemt ulike inndelinger av personlighetstrekk, integrering av personlighetstrekk i et anbefalingssystem og resampling-teknikker. I eksperimentene ble det generert tilfeldige, vurderingsbaserte og personlighetsbaserte anbefalinger, og seks ulike klassifiseringsmetoder ble brukt til å utlede personlighetstrekk fra dem.

Funnene indikerer potensiell informasjonslekkasje av personlighet fra de personaliserte anbefalingene. Likevel ble det ikke observert noen konsekvente mønstre for personlighetstrekkene, ettersom ulike eksperimentelle oppsett ga forskjellige resultater. Videre ble det ikke observert noen signifikant forskjell i treffsikkerheten på klassifiseringen etter at personlighetstrekkene ble integrert i anbefalingssystemet. Disse funnene bidrar til å øke forståelsen av personvernhensyn i forbindelse med brukeranbefalinger og gir innsikt til fremtidig forskning på personvern i anbefalingssystemer.

Preface

This thesis was written to complete the five-year Master's degree program in Computer Science at the Norwegian University of Science and Technology (NTNU). The literature study and experiments were conducted under the supervision of Özlem Özgöbek, Associate Professor at the Department of Computer Science at NTNU.

First and foremost, we want to thank our supervisor, Özlem Özgöbek, for her guidance and weekly meetings during our master's thesis. Additionally, we are grateful to our families for their encouragement, support, and faith in us. Our friends, particularly those in study hall 303 at Gamle Fysikk, also deserve our thanks for being available for discussions, feedback, and moments of laughter during the master's thesis and throughout our time at NTNU. Their presence has made our final years at NTNU fun and unforgettable.

Finally, we want to thank each other for the collaborative effort in writing this thesis. Our combined strengths complement each other in a good partnership, which has been important to successfully complete this thesis during its ups and downs. Writing in each other's company has made this thesis an enjoyable and educational experience.

Contents

Figures

Tables

Acronyms

ARHR Average Reciprocal Hit Rate. 21

AUC Area Under Curve. 3, 23, 45, 46, 49, 51, 64, 65, 68, 76

BFI Big Five Inventory. 25

CA Classification Accuracy. 3, 23, 45, 49–51, 58, 62–65, 68, 73, 75, 76, 79

CART Classification and Regression Trees. 15

CBF Content-based Filtering. 6

CF Collaborative Filtering. 7–11

FFM Five-Factor Model. 24

FN False Negative. 18

FP False Positive. 18

FPR False Positive Rate. 21, 23

GDPR General Data Protection Regulation. 12

HR Hit Rate. 21, 30, 57

kNN k-Nearest Neighbors. 16, 30, 32, 62, 63, 65, 68

LR Logistic Regression. 15, 32, 58, 62, 63, 65, 68, 73

MAE Mean Average Error. 20

MLP Multi-Layer Perceptron. 49

MSE Mean Square Error. 20

NB Naive Bayes. 58, 62, 63, 65, 68

NCF Neural Collaborative Filtering. 11

NDCG Normalized Discounted Cumulative Gain. 22, 30, 57

NMF Non-Negative Matrix Factorization. 11

NN Neural Network. 11, 16, 34, 62, 63, 65, 68

OCEAN Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism. 24, 28, 32, 47, 49, 58, 65, 73

PBRS Personality-based Recommender System. 31, 32, 35, 45, 47, 50, 51, 57, 64, 65, 73, 74, 77, 79

RBRS Rating-based Recommender System. 45, 57, 64, 65, 70, 73

RF Random Forest. 58, 62, 63, 65, 68, 75

RMSE Root Mean Square Error. 20

ROC Receiver Operating Curve. 21, 23

RS Recommender System. 5, 45, 57

SVD Singular Value Decomposition. 10

SVM Support Vector Machine. 15, 31, 32, 34, 58, 62, 65, 68

TIPI Ten-Item Personality Inventory. 2, 28, 49

TN True Negative. 18

TP True Positive. 18

TPR True Positive Rate. 21, 23

Chapter 1

Introduction

This thesis investigates privacy concerns regarding inferring personality data in the context of recommender systems, specifically by inferring user personality traits from personalized movie recommendations. This chapter aims to introduce the motivation for our research, the project goal and research questions, as well as the thesis outline.

1.1 Motivation

Recommender systems are algorithms designed to provide tailored content to the users of a system and can be utilized in domains such as e-commerce, health, and entertainment. Personalizing the system can enhance the overall user experience, benefitting both users and businesses. These systems improve user engagement, interactivity, and recommendation quality [1], while businesses simultaneously experience increased revenue due to their implementation [2].

To make accurate, personalized recommendations, the system requires information such as user attributes or preferences to generate user profiles [3]. However, acquiring and storing these user profiles might be considered an intrusive process that raises concerns regarding user privacy [4]. When creating the user profiles used in the system, data about the users and their behaviors can be collected without the users' explicit consent. When the data is collected implicitly, this raises a privacy concern due to the fact that many users are unaware of what data is collected, how much, and for what purpose [3, 4]. Some argue that potential threats to a user's privacy are commonly underestimated in recommender systems [3]. The availability of this data in itself could pose a privacy risk. Additionally, manipulating the data can possibly reveal even more information about the user than what is directly accessible.

An adversary can rely on seemingly innocent user input, e.g., ratings of movies, to derive sensitive and private information [3]. Previous research has shown that it

is possible to infer gender based on movie ratings [5] and other attributes, including sexual orientation, political views, and personality traits, from likes on social media [6]. Milano *et al.* [7] argue that users might object to the inference of their personal data if they were better informed of these possibilities. This opens up a discussion about inference in personalized systems.

A newly emerging field within user-adaptive systems is using personality to augment user profiles. Personality is proven to greatly impact a person's decision-making process [8, 9], thus making it valuable in the context of a recommender system, which fundamentally is based around making choices among different options. Users with similar personalities generally have similar preferences [10]. Applying personality in a recommender system will therefore help enhance the quality of the recommendations [11] and additionally help alleviate the problems regarding insufficient data about new users or items, commonly called cold start problems [12, 13]. Personality remains consistent and predictable regardless of the situation or location, as noted in studies by Nguyen *et al.* [8], Hu and Pu [9], and Martijn *et al.* [14]. Additionally, personality is not limited to a specific domain and can be useful in various areas such as movies, music, and shopping.

Personality has historically been assessed using psychological questionnaires such as the Ten-Item Personality Inventory (TIPI) [8, 15]. However, these methods are not practical for everyday use as they require a lot of user effort to complete the questionnaires. Nguyen *et al.* [8] experienced that explicitly asking for the users' personality is not to be advised as it is strenuous and time-consuming. In an effort to overcome the limitations of explicit personality collection, researchers have explored alternative methods for identifying personality traits. For example, studies have been conducted to identify personality traits based on Facebook [6, 16, 17] or Twitter profiles [18], or from people's microblogs [19]. In 2014, Youyou *et al.* [20] even demonstrated that well-trained algorithms could outperform humans when it comes to assessing users' personality traits. Nguyen *et al.* [8] predict that commercial sites such as Amazon and Facebook will be able to obtain user personalities implicitly through their user's traces, which would open a privacy concern that needs to be researched and addressed.

As presented, the usefulness of personality in recommender systems makes it a valuable target of inference. In this thesis, we wish to explore the vulnerability of user privacy and inference from recommender systems, focusing on the sensitive state of personality. As mentioned, personality is a highly valuable attribute in the context of recommender systems. However, it is also a private and personal attribute that should be handled with confidentiality to prevent any unwanted leaks, especially when dealing with personalized systems or newly emerging personality-based systems.

1.2 Project Goal and Research Questions

The project goal of our master's thesis is to explore the vulnerability of personalized movie recommendations by inferring the users' personality traits from the recommendations provided by the recommender system.

Despite conducting a study of the existing literature, we have not come across any previous studies that specifically utilize top n recommendations for inferring personality traits, which opens up a gap for our novel contribution to the field.

To address this gap, we have formulated the following research questions:

RQ1 To what extent is it possible to infer users' personality traits from movie recommendations?

- **RQ1.1** How does the number of classes for each personality trait impact the inference accuracy?

RQ2 How does incorporating personality traits into a recommender system influence the ability to infer personality from the recommendations?

RQ3 How does the application of resampling affect the Classification Accuracy and Area Under Curve for inferring users' personality traits from movie recommendations?

1.3 Thesis Outline

Chapter 1: Introduction
The initial chapter presents the motivation behind the project, the project's goal, and the research questions we aim to address. Additionally, this section outlines the structure of the thesis.

Chapter 2: Background
Theoretical background and relevant terminology are introduced in Chapter 2 to establish the necessary groundwork for the thesis.

Chapter 3: Related Work
Chapter 3 provides an overview of the related work derived from the literature study conducted during the preliminary fall report. Further, we have enriched our related work by including newly discovered papers incorporating novel findings and recently published studies.

Chapter 4: Data and Technology
The data and technology utilized in the thesis are introduced and thoroughly explained in Chapter 4. The chapter includes an exploratory data analysis (EDA) and a description of the data processing and cleaning applied. Moreover, it provides an overview of the tools and technologies utilized in the experiments conducted throughout this thesis.

Chapter 5: Method and experiments
Chapter 5 presents the method employed in our research and describes the experiments in detail. The four experiments conducted to address the research questions are outlined.

Chapter 6: Results
In Chapter 6, the quantitative results obtained from the experiments, previously introduced in Chapter 5, are presented. The results are arranged and presented according to each of the four experiments.

Chapter 7: Discussion
In Chapter 7, an in-depth discussion of the results presented in Chapter 6 is provided. The findings are interpreted in relation to the research questions and goal of the thesis. The implications of the results are explored, and any limitations or weaknesses are discussed. Insights into the broader significance of the findings are provided. The goal of this chapter is to thoroughly evaluate the results and ensure a comprehensive discussion of their importance. Additionally, possible directions for future research are suggested.

Chapter 8: Conclusion
The final chapter of the thesis summarizes the main discoveries, contributions, and implications of the research. The strengths and limitations of our work are acknowledged, and the potential future research is discussed. This chapter provides a coherent conclusion to the thesis and the research questions.

Chapter 2

Background

The theoretical background introduces the foundational knowledge and concepts relevant to this thesis, including recommender systems, privacy aspects, inference techniques, and personality models. This chapter aims to provide a solid understanding of these key elements.

2.1 Recommender Systems

Using various techniques, Recommender System (RS) are software programs that produce and offer suggestions for items and content to the system user [21]. Recommender systems aim to predict how satisfactory content in a chosen domain is for a user. The system should present users with interesting products they might not otherwise find at first. The recommendations are most often based on the users' past interactions with the system to predict interests. For example, if a user has previously enjoyed a comedy movie, they are more inclined to enjoy that again in the future rather than a historical documentary [22]. Implementing a recommender system may significantly increase a company's clicks, the users' interaction with the system, ultimately increasing the business revenue [23].

The recommender system can generate both personalized and non-personalized suggestions [24]. Non-personalized recommendations are less complicated to generate and may contain the same list of items for every user. That can, for example, be recommending the top 10 most-rated movies, the top 10 editor-selected movies, or the ten newest movies added. These non-personalized recommendations are typically used when there is insufficient information on the user's preferences.

The usage of recommender systems ranges from different domains such as social media, different types of entertainment, healthcare applications, and e-commerce platforms. In social media and entertainment, recommender systems suggest relevant content, posts, or connections to users based on their previous preferences,

browsing behavior, or social network. Moreover, recommender systems can also assist in domains like healthcare with personalized treatment recommendations, disease diagnosis, or medication suggestions based on a patient's symptoms and medical history. In each of these domains, recommender systems play a crucial role in enhancing user experiences by providing personalized and tailored recommendations.

Cross-domain recommendations are defined as when the recommender system recommends items to a user that cross into a different domain than the user's ratings are from [25–27]. Cross-domain recommendations are possible if the domains have similarities. Using movie recommendations as an illustration, it is likely that a user who enjoys a certain genre of movies will also enjoy related books and music. This can be used to solve when there is data sparsity in the domain where recommendations are made or the cold start problem. The "cold start problem" refers to the situation where the recommender system has issues inferring anything about the user because it has not yet gathered sufficient ratings or other relevant data [22]. However, recommender systems are usually limited to a single domain, where both the items and user ratings belong. An attribute is considered domain-independent if it can be used to generate recommendations in a different domain than the one it was obtained from.

2.2 Recommender Methods

Recommender methods encompass a diverse range of approaches and techniques employed by recommender systems, which are vital in generating recommendations. These methods involve algorithms and statistical models that analyze user preferences, historical behavior, and item attributes to deliver accurate and relevant recommendations. By leveraging collaborative filtering, content-based filtering, hybrid approaches, and other methodologies, recommender systems provide users with valuable recommendations.

2.2.1 Content-based Filtering

Content-based Filtering (CBF) use a combination of an item's description and the user's interests to recommend new items [28]. The system compares the similarity of the user's previously enjoyed items with items the user has not yet interacted with to determine which items to recommend. As the user interacts with new items and provides feedback through explicit ratings or interactions, the system updates the user's profile accordingly. This means that content-based recommender systems do not require data from other users to make recommendations. Overall, this system learns to recommend content similar to what the user has enjoyed in the past [24]. The diagram in Figure 2.1 provides a visual representation of how content-based filtering recommends items to users.

[1]Movie icons designed by macrovector_official at Freepik

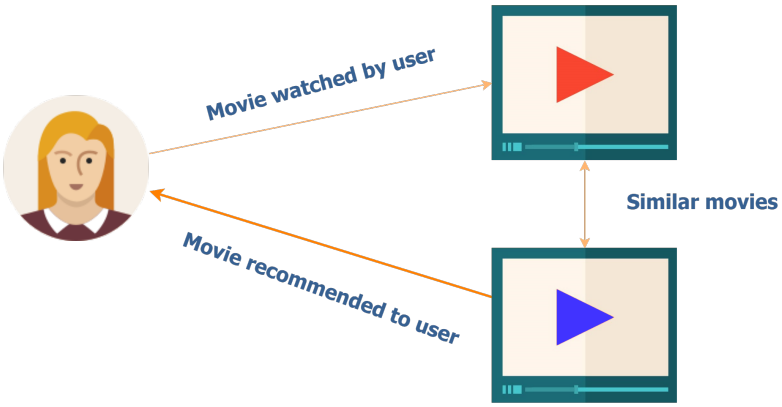


Figure 2.1: Illustration of content-based filtering[1].

Because the target user has rated items similar to a new item, content-based methods have the advantage that they can make recommendations for new items with no rating data. However, CBF has some disadvantages as well [29]. While CBF works well with new items, the recommendations are generated using user profiles and information about their interests. This makes it difficult for the system to handle new users. Moreover, content may be sorted or filtered such that a user is never presented with certain items, a phenomenon known as filter bubbles.

2.2.2 Collaborative Filtering

Collaborative Filtering (CF) is a method that identifies similarities between users and considers similar users' opinions in the recommendation process [24, 30]. It is based on the idea that users who enjoy similar content will likely enjoy the same things in the future. Suppose certain users in the same interconnected community who share the same preferences rate an item highly. In that case, it will be recommended to others in that community who are yet to engage with it. An illustration of how collaborative filtering recommends items is shown in Figure 2.2.

User-item, user-user, and item-item matrices are three types of user matrices frequently used to calculate similarities in CF. In a user-item matrix, the rows represent users, and the columns represent items, showing the interaction between a user and an item. In user-user and item-item matrices, on the other hand, the rows and columns both represent users or items, respectively. These matrices are used to calculate the similarities between users or items. Unfilled values in a matrix indicate data sparsity. For instance, an empty field in a user-item matrix might indicate that a user has not yet rated that item. Similarities are calculated based on the rows in the matrices.

The two most common collaborative filtering methods are memory- and model-based filtering.

[2]Movie icons designed by macrovector_official at Freepik

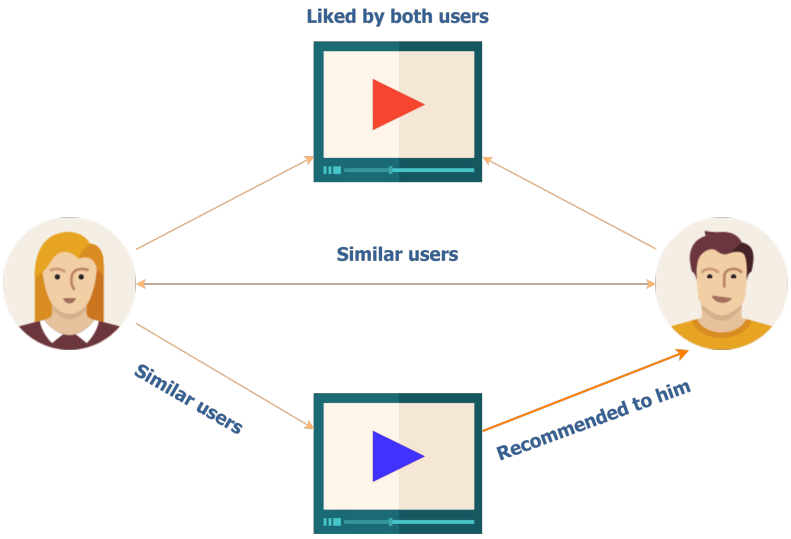


Figure 2.2: Illustration of collaborative filtering[2].

2.2.2.1 Memory-based

Being one of the first CF algorithms, memory-based methods are heuristic in nature but are the simplest to implement, as they operate on the entire database to make simple and intuitive recommendations [31]. Using the entire database consequently leads to some disadvantages of the method. The method can be computationally complex. Calculating the similarities between rows or columns in the matrix may require a significant amount of processing power and time, especially when dealing with large matrices. They do not work optimally with sparse rating matrices and lack full coverage of rating predictions. However, the lack of coverage is not an issue if one focuses only on the top k items in a set of produced recommendations, the lack of coverage is not an issue. Still, this method is quite slow as it needs to make predictions over all the data each time [29].

Memory-based methods are methods applied to raw data without any preprocessing [22]. They can also be referred to as neighborhood-based algorithms. Here user-item combinations are predicted in one of two neighborhoods: user- and item-based collaborative filtering.

- **User-based Collaborative Filtering**
  User-based Collaborative Filtering recommends an item to a target user if similar users similarly rate the item. Thus it is important to determine users who are similar to the target user. For instance, if two users, A and B, have rated movies similarly in the past, A's rating of movie M that B has not rated can be used to predict B's rating of movie M.
  Typically, the k-most similar users to the target user are used to make recommendations. The computation of user similarity is often based on the

Pearson correlation coefficient, which involves calculating the mean of all users' ratings for a given item. This calculation is performed by comparing the rows of the user-user matrix [22], shown in Equation (2.1).

$$Sim(u,v) = Pearson(u,v) = \frac{\sum_{k \in I_u \cap I_v} \cdot (R_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}} \tag{2.1}$$

where:

r_{uk} is the rating from user u on item k
μ_u is the average rating by user u
I_u is the item set for u

- **Item-based Collaborative Filtering (CF)**
  Item-based Collaborative Filtering is based on the similarity between items assuming that similar items are rated similarly by the same user. By identifying a set of items similar to a particular item of interest, the system can predict whether a user will enjoy that item by analyzing the user's ratings of the other items in the same set.
  Opposite to user-based, the k-most similar items to the target item are used to make recommendations to the target user. The similarity between items can be calculated using the rows of the item-item matrix. In item-based CF, adjusted cosine is often used to calculate this similarity [22]. As shown in Equation (2.2), it measures the similarity between the ratings of two items by considering the deviations from their average ratings.

$$Sim(i,j) = AdjustedCosine(i,j) = \frac{\sum_{u \in U_i \cap U_j} S_{ui} \cdot S_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} S_{ui}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} S_{uj}^2}} \tag{2.2}$$

  where:

  U_i is the set of indices for the users that have rated item i
  $S_{ui} = (r_{ui} - \mu_u)$

- **User-based Vs. Item-based Collaborative Filtering**
  To compare the two approaches, item-based CF often provides more accurate and reliable recommendations than user-based CF. This is because the similarity between items remains constant over time, whereas users may change their interests and preferences over time. Another advantage is that new users join more often than new items in real-case scenarios, and item-based handles new users better than user-based. Item-based CF is also more resilient to shilling attacks because it relies on item similarity rather than user ratings, making it harder for attackers to manipulate the system and produce incorrect recommendations.

2.2.2.2 Model-based

Model-based methods are developed using machine learning methods and data mining. In contrast to memory-based CF, not all data is used in the recommendation; thus, it is faster. Additionally, model-based models are scalable and work well with large datasets. Examples of model-based methods are decision trees, rule-based models, and Bayesian methods. Matrix factorization is a dimensional reduction method often used in model-based CF.

2.2.3 Hybrid Recommender Systems

Hybrid recommender systems are a combination of two or more recommendation techniques [21, 24]. Different types of recommender systems use different inputs, meaning hybrid recommender systems can only make use of the methods where the required input is available. By using multiple strategies, hybrid recommender systems benefit from the advantages each utilized approach offers. As the various recommender systems have different disadvantages, adding a complementary technique fixes the problems.

2.2.4 Personality-based Recommender Systems

Personality-based recommender systems leverage user personality traits to generate recommendations for the user [11]. Research has indicated that personality-based recommender systems outperform rating-based recommender systems for mean absolute error, recall, and specificity [32]. By considering personality factors, recommender systems can possibly better understand users' preferences to generate personalized recommendations that align with their individual tastes and preferences.

2.3 Recommender Algorithms

For the different types of recommender systems, there are also several recommender algorithms that can be employed. This section introduces the algorithms that we use later in our thesis.

2.3.1 SVD

Singular Value Decomposition (SVD) is a powerful technique for dimensionality reduction, widely used in recommender systems. It automatically identifies meaningful concepts in a lower-dimensional space, making it ideal as the basis for latent-semantic analysis. One notable advantage of SVD is its ability to handle incremental updates, allowing it to accept new users or ratings without recomputing the entire model. After its success in the Netflix Prize competition, SVD methods have been more commonly used in recommender systems [33].

2.3.2 kNN

The kNN algorithm is widely recognized as one of the preferred approaches for Collaborative Filtering (CF) recommenders [33]. By leveraging the preferences of similar users or items, the kNN algorithm predicts user preferences by computing the similarity between the target user and their neighbors. This similarity is typically calculated using a distance metric, such as cosine similarity or Euclidean distance. After identifying the nearest neighbors, the algorithm predicts the target user's preferences by aggregating the ratings or preferences of these neighbors. The use of kNN for other classification purposes is introduced in Section 2.5.

2.3.3 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is a matrix factorization technique used in recommender systems. It works by breaking down the user-item rating matrix into smaller matrices with lower ranks and requires that the factor matrices are non-negative. Unlike other matrix factorization techniques, NMF is specifically designed for implicit feedback datasets, where users only express their likes but not their dislikes. It can also handle unary ratings matrices or matrices that represent activity frequency. Missing information can be easily filled in by setting them to zero. NMF uses special multiplicative rules to update matrices U and V to continuously improve the factorization, which can be further enhanced by applying regularization. However, when dealing with large rating matrices, computational challenges must be addressed to ensure efficient processing and optimal utilization of resources [22].

2.3.4 Neural Networks

Neural networks have gained popularity in recommender systems due to their ability to model complex patterns by processing user and item features through multiple hidden layers.

One way of using neural networks in a recommender system is with Neural Collaborative Filtering (NCF). NCF is a method that uses collaborative filtering models with a neural architecture. Instead of the user-item interaction in regular CF, NCF tries to learn and model these interactions through a Neural Network (NN). The MLP is a type of neural network that can learn any continuous function and has multiple layers that introduce high levels of nonlinearity. This makes the MLP well-suited to learn the complex user-item interaction function. In simpler terms, NCF enhances the capability to understand how users and items interact by using a neural network that captures intricate patterns and non-linear relationships [34].

2.4 Privacy

Privacy is an elemental aspect of our modern digital landscape and is recognized as a fundamental human right by the UN Declaration of Human Rights [35]. The right to privacy facilitates the individual's self-development, freedom of expression, and thought. Maintaining privacy is crucial in order to prevent personal information from being misused or manipulated. It empowers individuals to have control over their data and make educated choices about sharing it or not. As technology advances and systems such as recommender systems become more common, the preservation of privacy becomes increasingly important to ensure that individuals can freely navigate and engage with online platforms while retaining their independence and personal control. This section will serve as an introduction to privacy concepts that are relevant to the thesis.

2.4.1 Privacy Glossary

Personal data is defined as any information relating to an identifiable individual, either directly or indirectly [36].

Sensitive data defined by the General Data Protection Regulation (GDPR) refers to personal data revealing information about an individual. Health-related data, biometric data, and data regarding a person's sexual orientation and political or religious views are all considered sensitive data. The data is protected under EU laws and requires special handling. [37].

Anonymization is the process of making personal data anonymous. An anonymized dataset should make it impossible to identify an individual. A dataset can also be pseudo-anonymized, making the person identifiable by a pseudonym but not directly identified [38].

Adequate consent is a principle that requires the individual to be fully informed and have a clear understanding of what the data will be used for before giving consent. The consent should be freely provided: explicit, informed, and in writing. Adequate consent should be obtained prior to data collection or when the purpose of data re-use falls outside of the purpose for which consent was originally obtained. The individuals should be able to withdraw their consent at any moment [39].

Data collection can be done in two ways: explicitly or implicitly. Explicit data collection refers to data gathering that is provided directly by the user, such as answering a questionnaire or giving direct feedback. In other words, explicit data collection involves collecting data from the user by an action taken actively and voluntarily by the user. Implicit data collection, on the other hand, refers to the collection of data that is not provided directly by the user but is instead inferred from their behavior or actions. This can be done by, for example, analyzing data streams, how the user navigates the website, and so forth [40].

While explicit data might give the system more accurate data, implicit data will provide unbiased insight into user behavior that would otherwise be unavailable. If the user is unaware of the implicit data collection and what it may be used for and has not given the system adequate consent, it can raise privacy concerns.

An adversary in cyber security is a party that threatens the system. They may engage in malicious activity and see or alter information they are not authorized to [41].

Inference involves drawing a conclusion based on presented evidence [42]. Machine learning executes this by inputting data into an algorithm to generate a predicted outcome as the output of the model [43]. Various methods of machine learning and data mining are used to infer attributes.

Attribute inference attack is an attack where an adversary has partial knowledge about the training data and uses it to infer the missing attributes of the dataset [44]. Using, for example, available information about education, work, and location, it might be possible to infer an individual's age even if that data is not present in the dataset.

2.4.2 Privacy in Recommender System

As recommender systems have become increasingly widespread across many domains, there are growing concerns about the privacy of user data collected and analyzed. These systems rely on user information, including preferences, browsing history, and demographic data, to provide personalized recommendations. While the goal is to enhance user experiences, addressing the privacy implications associated with using such data is essential.

Recommender systems often use personal data such as gender, location, and personality to provide customized content. Privacy/personalization trade-off refers to the compromise between privacy and a more personalized system [3, 45]. However, preserving privacy involves limiting the amount of personal user data shared with the system. This approach results in less accurate recommendations that are more generalized. The system's performance is directly linked to the amount of information it has access to. Users may see their privacy as a commodity they may be willing to sacrifice to improve the system's performance.

2.5 Classifiers

Classification is the ability to categorize values into different labels. This can often be done by machine learning, however, machine learning can also be used for regression which is to categorize values as continuous. In classification, we analyze data points and categorize them into different labels based on a set of rules. Machine learning allows for constantly updating the model with new data, unlike

simple data mapping [46]. There are two techniques for building a classifier; supervised and unsupervised classification. A supervised classifier trains the model with a dataset including predetermined labels, while an unsupervised classifier categorizes data without using predetermined labels. Classification algorithms can be binary classifiers or multi-class classifiers.

In machine learning, a model aims to capture the true relationship of the dataset, but we often get a bias, which is the inability to do so. In other words, a model's bias is the measurement of error from the real value of the function [46]. To uncover this relationship, the model is trained with a subset of the dataset and then tested and evaluated on the remaining portion. The discrepancy in fit between datasets, often the train and test datasets, is called variance. Here the Bias-Variance tradeoff arises. When the bias is excessively low, it performs well on the specific training data. Still, it may not be generalized enough to identify patterns in new test data, resulting in high variance. This is known as an overfitted model.

On the other hand, an under-fitted model will have low variance performing similarly on test and train data but will not fit precisely on data points leading to high bias. An overfitted model typically has many parameters, while an underfitted is too simple and has fewer parameters. Figure 2.3 visually explains the relationship between bias, variance, and model fitting. The ideal model has low bias and variance, but as described, there is usually a trade-off between the two. The best model is the one that strikes a balance between bias and variance, which is often measured by minimizing the total error Equation (2.3) [47].

$$TotalError = Bias^2 + Variance + IrreductibleError \tag{2.3}$$

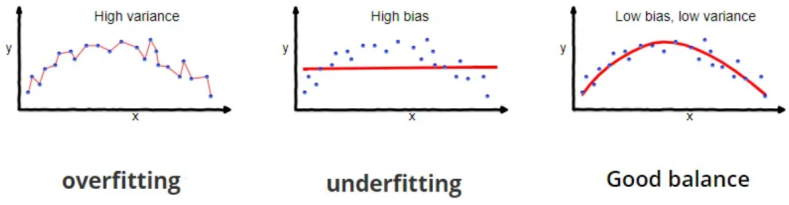


Figure 2.3: Visually explain the relationship between bias and variance and model overfitting, underfitting, or balanced fit. Figure from [47].

To prevent overfitting, many use regularization, a technique that reduces complexity in models by penalizing them. By decreasing the number of parameters, the model becomes simpler, can generalize better, and is less prone to overfitting. Not only that, but regularization also enhances performance when handling new input. Regularization techniques vary by classifier type; later in this chapter, some will be covered.

Different algorithms may be utilized to classify data; examples of supervised classification algorithms are discussed below.

2.5.1 Logistic Regression

Logistic Regression (LR) is a method used to predict the value of a dependent variable based on independent variables in a dataset. However, while linear regression predicts continuous variables, logistic regression predicts categorical values that can be true or false. This is known as binary logistic regression. An extension to logistic regression is multinomial logistic regression, used when the dependent variable has more than two possible values [48].

Regression analysis is a helpful tool but can sometimes lead to overfitting. Two adjustments can be made to address this issue: L1 regularization (also known as Lasso) and L2 regularization (known as Ridge). L1 regularization adds a penalty to the sum of absolute values of the weights, while L2 regularization adds a penalty to the sum of squares of the weights [49]. If you have many features, L1 regularization is often preferred because it results in a sparse solution and can avoid zero coefficients. On the other hand, L2 regularization is non-sparse and can handle highly correlated independent variables by constraining the coefficient and keeping all variables. Another difference is that L1 is robust to outliers, while L2 is not.

2.5.2 Support Vector Machines

Support Vector Machines (SVMs) classify data by creating a linear hyperplane that separates the data into two groups. The ideal hyperplane is the one that has the largest distance to the nearest data point for each group [50].

According to Tyagi [49], SVMs can produce models with a high variance but low bias. To address this issue, the cost parameter C can be increased, which reduces the variance and increases the bias by regulating the number of violations of the allowed margin in the training data.

2.5.3 Naive Bayes Classifiers

Naive Bayes classifiers are a family of classifiers that uses Bayes' theorem as a basis for the algorithm. Examples include Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes, which use Bayes' theorem as the foundation of their algorithm. These Naive Bayes classifiers operate under the assumption that each feature contributes equally and independently to the outcome [51].

2.5.4 Decision Trees

Decision trees are sometimes called Classification and Regression Trees (CART). To classify a variable, the binary tree structure starts at the root and moves down

to the leaf node, with internal nodes being decision nodes and edges representing the split. The final decision is made at the leaf node based on the condition set at the root [52]. To accurately split the nodes, a greedy search algorithm uses information gain or Gini index [53]. M5 rules is a technique based on decision trees that predicts continuous numerical values. Specifically, M5 employs binary decision trees with linear regression functions as leaf nodes [54].

2.5.5 Random Forest

Random forest is a classification algorithm that utilizes multiple decision trees, forming a "forest." These decision trees are generated through bootstrapping, where random subsets of data are selected to create the trees. The algorithm gathers the results from all decision trees to determine classification and calculates based on majority voting. This process is known as bagging [55]. Regularizing Random Forest by reducing tree depth and branches (new features) helps prevent overfitting [49]. This promotes diversity among the trees and leads to a balance between complexity and flexibility.

2.5.6 kNN

The k-Nearest Neighbors (kNN) algorithm is a classification algorithm that finds similar points near each other. [53]. For classification, the algorithm calculates the distance between a query and all the data samples, selects the specified number (K) of samples nearest to the query, and then votes for the most frequent label. While kNN is easy to understand and implement, it is considered a lazy learning method because it only stores a training dataset without undergoing training. This approach means that all computation occurs during classification. However, as data size increases, kNN becomes less efficient and does not scale well. It is also sensitive to irrelevant features and overfitting.

2.5.7 Neural Networks

A Neural Network (NN) classifier is a machine learning model that imitates the human brain to recognize patterns [53]. It consists of interconnected nodes in a layered structure, each a linear regression model connected by weights. During training, labeled datasets provide the correct answer, and the network gains knowledge about the dataset and adjusts its weights accordingly to minimize loss. Once trained, the network can classify new data accurately, making it useful for classification tasks. Neural networks learn from samples without being programmed with task-specific rules, making them different from traditional computing systems. Regularization in neural networks is done by limiting the complexity (weights) of the model, preventing overfitting, and promoting generalization [49].

2.6 Imbalanced Dataset

In an ideal scenario, datasets used for training should have evenly distributed class labels. However, in the real world, some classes have fewer samples than others, known as the minority class. This uneven distribution of labels is referred to as an imbalanced dataset. Training on an imbalanced dataset often results in a model incapable of learning the data's real patterns, leading to incorrect predictions. Models may also favor the majority class, which has more samples, and perform poorly on the minority class. To address this issue, techniques such as resampling, cost-sensitive learning, or ensemble methods can be used to balance the class distribution and improve model performance in underrepresented classes.

Resampling techniques are employed to even out the sample space when dealing with imbalanced datasets [56]. This helps to minimize the impact of the uneven class distribution during the learning process. In the trainset, either the majority class's samples can be undersampled, the minority class may be oversampled, or hybrid methods combine the two methods. This way, the model is able to train on a more evenly distributed dataset.

In many cases, all misclassifications have the same cost. However, there are instances when miscalculating one class has more severe consequences than the other. For instance, it could be riskier to classify a cancer patient as healthy than the reverse. Cost-sensitive learning addresses this by applying varying penalties for different misclassifications.

2.7 Data Splitting

Data splitting is the division of data into at least two subsets so the model can be trained, tested, and evaluated, and it is an important aspect of data science [57]. There are several ways to split data for different use cases; common to all is that the splitting aim to ensure accurate data models. In a two-part data split, the data is split into a train and test set. First, the training set trains the model by estimating the parameters and is used to compare different model performances. Then the model is tested on the test set to examine whether the model works as intended and check for overfitting. Depending on the situation, a two-part data split is often split 80/20 or 70/30, but other ratios may be used. It is also possible to have a three-part split with train, validation, and test sets. Here the additional subset is used to compare the different models and hyperparameters, meaning the train set is only for learning the data pattern.

A robust split is the cross-validation method, where the data is split into multiple subsets referred to as folds [58]. All but one fold is used to train the model, and the remaining is used to test. This process is repeated until all folds have been used as the test fold once. Evaluation metrics are calculated as the average across all folds. Thus this technique provides an accurate estimate of the model's performance.

Types of cross-validating iterators include k-fold, where the sample is divided into k groups of equal size. Common values for k are 5, 10, and n = samples in the full dataset. If k = n, it is called leave-one-out, meaning all but one sample is in the train set. This method does not waste much data, as only one sample is removed from the training set.

When splitting the data, there is no set rule on how to sample the data. It depends on the scenario at hand affected by variables such as data size or the number of variables used as predictors in the dataset. Common data sampling methods are random and stratified random sampling. The use of random sampling helps prevent bias toward various data characteristics. Conversely, total randomness may lead to an uneven distribution in the splits of datasets. In such cases, stratified random sampling may be applied. Here, data is chosen at random but within defined parameters. Stratified random sampling obtains a sample set that best represents the full dataset by aiming to maintain the same ratio between the groups of the entire dataset. Another option altogether is nonrandom sampling, where, for instance, the most current data is to be tested.

2.8 Evaluation

Assessing the performance, effectiveness, or quality of a system, process, or product is known as evaluation. It involves measuring how well it meets its intended goals or objectives using a set of predefined metrics or criteria and helps us identify areas where improvements can be made. Evaluation metrics vary depending on what is being evaluated, yet there are some similarities. This section presents a few of the relevant metrics for evaluating recommender systems and inference methods, respectively. Some metrics are used to evaluate both and will therefore be presented twice as the context of what the metric evaluates is relevant. Firstly, terminology that is relevant to both processes is explained.

Evaluation metrics in recommender systems and classifiers often use true positives, false positives, true negatives, and false negatives. These measurements are determined by the system's correct and incorrect predictions. A True Positive (TP) is when the system accurately predicts a positive instance, like suggesting a movie the user enjoys or correctly identifying a cancer diagnosis. A False Positive (FP) is when the system predicts a positive instance that is actually negative, such as recommending a movie the user dislikes or diagnosing cancer when there is none. A True Negative (TN) is when the system accurately predicts a negative instance, like not suggesting a movie the user dislikes or confirming a person does not have cancer. A False Negative (FN) is when the system predicts a negative instance that is actually positive, like failing to suggest a movie the user would enjoy or failing to diagnose cancer when it is present.

Figure 2.4 visually represents TP, FP, TN, and FN. These values are often presented in a confusion matrix as a summary of the predictions. The matrix has

two dimensions, actual and predicted, with identical sets of classes in both dimensions. We can set a threshold value to classify the values as true or false. The default threshold in a confusion matrix is 0.5, with values above 1 (True) or below 0 (False). This threshold value can be adjusted depending on if it is more crucial to classify all the true positives or avoid false positives.

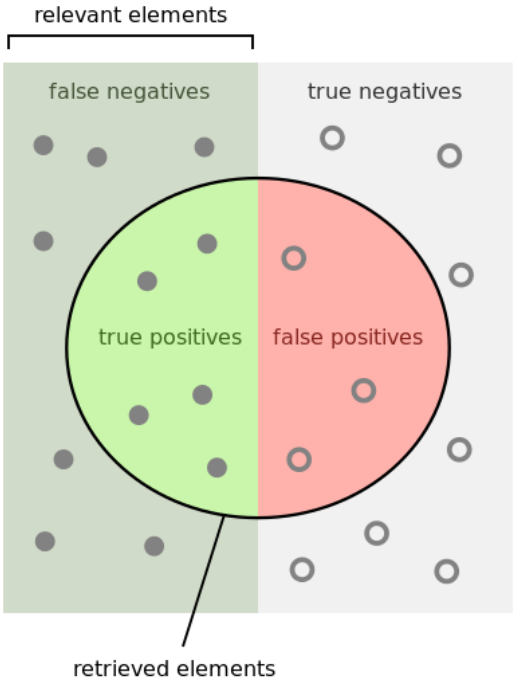


Figure 2.4: Illustration of true positives, false positives, true negatives, and false negatives. Figure from [59].

2.8.1 Evaluation of Recommender Systems

In order to improve the performance of recommender systems, it is important to evaluate and measure their effectiveness through the use of evaluation metrics. These metrics provide a quantitative measure of the accuracy of the system's recommendations and can be used to compare different models. This evaluation process is crucial in the development of recommender systems. A good recommender system should be both personalized and diverse, suggesting items that are actually available while avoiding repetition [29]. While accuracy is the primary factor when evaluating recommender systems [22], it may not always fully understand the system's performance [24]. Users also value additional metrics such as novelty, serendipity, coverage, and trust. Although these metrics are subjective, evaluation methods exist to measure them.

Evaluation Goals
Accuracy - How good is it?
Coverage - How many items and users do we serve?
Confidence and trust - Do we believe it?
Novelty - Any new content here?
Serendipity - Any new and surprising content?
Diversity - Is it all the same?
Robustness and stability - Can it deal with fake data?
Scalability - Is Big Data a problem?

2.8.1.1 Accuracy and Error Based Methods

In a recommender system, accuracy measures the error in estimated numeric ratings. Accuracy can be measured by Mean Square Error (MSE) and Root Mean Square Error (RMSE) (Equation (2.4)) [22]. RMSE largely penalizes large error values or outliers, meaning a few poorly predicted ratings can significantly impact its measure. Thus if the robustness of prediction across various ratings is important, RMSE may be a better option. One drawback of RMSE is that it does not truly reflect the average error and can produce misleading results. When evaluating accuracy where outliers are limited or unimportant, Mean Average Error (MAE) (Equation (2.5)), which does not disproportionately penalize large errors, is a better reflection than RMSE. Ultimately, deciding between MSE or RMSE should depend on the specific application being used.

$$MSE = \frac{\sum_{i=1}^{n}(\hat{r_i} - r_i)^2}{n}, \qquad RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{r_i} - r_i)^2}{n}} \tag{2.4}$$

$$MAE = \frac{\sum_{i=1}^{n}|\hat{r_i} - r_i|}{n} \tag{2.5}$$

where:

r_i is the rating for item i k
$\hat{r_i}$ is the predicted rating for item i k

2.8.1.2 Evaluation Ranking

A recommender system can recommend any number of items, typically referred to as the top n items, where n indicates the size of the recommended list. Recommending all unrated items to a user would cover all potentially relevant items and include irrelevant ones. Altering the value of n would affect the balance between the relevant and irrelevant recommended items. Different metrics are available that gauge this tradeoff and indicate the percentage of relevant items in the recommendations. We measure precision (Equation (2.6)) by calculating the percentage

of recommended items the user finds relevant. Meanwhile, recall (Equation (2.7)) refers to the percentage of true positive recommendations that were identified as such. In short, precision tells how many retrieved items are relevant, while recall tells how many relevant items are retrieved. Both precision and recall are most often represented by values ranging from 0 to 1, although percentages may be used. A value of 1 indicates the best performance in terms of precision or recall, reflecting perfect accuracy or completeness.

$$Precision@k = \frac{\#\text{of recommended items@}k\text{that are relevant}}{\#\text{of recommended items@}k} \tag{2.6}$$

$$Recall@k = \frac{\#\text{of recommended items @k that are relevant}}{\text{total \# of relevant items}} \tag{2.7}$$

Although there is a trade-off between precision and recall, it is not always a direct relationship. Simply increasing recall does not always mean a decrease in precision. To measure precision and recall combined, the F1-measure (Equation (2.8)) calculates their harmonic mean. Thus F1 provides a better quantification than precision and recall. Still, it depends on the number of recommendations (top n).

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{2.8}$$

The Receiver Operating Curve (ROC) curve visually represents how precision and recall are balanced. The x-axis indicates the False Positive Rate (FPR), which is the percentage of irrelevant items in the top n, meaning false positives. The y-axis shows the True Positive Rate (TPR), which is the percentage of true positives included in top n. TPR is also known as recall, while FPR can be considered the recall of negative items.

Hit Rate (HR) measures the fraction of users where the correct answer is included in the top n recommendations [22, 60–62]. When testing on ground truth, a hit occurs if one of the recommendations in the top n is actually rated by that user. HR is calculated by dividing the number of users for which the corrected answer is included in the top n by the total number of users (Equation (2.9)). HR does not take ranking within the top n recommendations into account; for this Average Reciprocal Hit Rate (ARHR) can be measured. ARHR accounts for where in the top n the hit occurs, giving more credit to the top slots. ARHR is calculated by summing up the reciprocal of each hit (Equation (2.9)).

$$HR = \frac{hits}{users}, \qquad ARHR = \frac{\sum_{i=1}^{n} \frac{1}{rank_i}}{users} \tag{2.9}$$

Normalized Discounted Cumulative Gain (NDCG) quantifies the relevance of the recommended items while taking into account the position of the relevant item in the ranked list and is best explained iteratively through its components [22, 60]. Gain is the relevancy score of an item, often represented as a numerical scale or binary implicit rating, and cumulative gain is the sum of all the gains in the first k recommendations associated with the item. Since cumulative gain does not consider the ordering of the recommendations, we can introduce discounted cumulative gain (DCG) that divides the gain by its rank. Lastly, since the discounted cumulative gain depends on k, the number of top recommendations, we cannot compare systems comparing different numbers of items. To adjust for this, we calculate DCG for the ideal ranking, IDCG. Finally, Normalized Discounted Cumulative Gain (Equation (2.10)) is the normalized DCG over the IDCG such that the value is independent of k and has values 0 to 1.

$$NDCG = \frac{DCG@k}{IDCG@k} = \frac{\sum_{i=1}^{k} \frac{G_i}{\log_2(i+1)}}{\sum_{i=1}^{|I(k)|} \frac{G_i}{\log_2(i+1)}} \tag{2.10}$$

where:

$I(k)$ represents the ideal list of items up to k
$|I(k)|$ is the length of $I(k)$ which is k
G_i is the gain for item i

2.8.1.3 Other Methods

The ability to recommend a portion of items or to a portion of users is called coverage [22]. Coverage can be measured for the users (Equation (2.11)), items (Equation (2.12)), or all the user-item pairs (Equation (2.13)) in the catalog. When ratings are sparse, there may be instances where it may not be able to recommend certain items or to certain users giving low coverage.

$$\text{User Coverage} = \frac{\#\text{ of users with recommendations}}{\text{Total number of users in the catalog}} * 100 \tag{2.11}$$

$$\text{Item Coverage} = \frac{\#\text{ of items that are recommended}}{\text{Total number of items in the catalog}} * 100 \tag{2.12}$$

$$\text{Catalog Coverage} = \frac{\#\text{ of recommended user-item pairs}}{\text{Total number of user-item pairs in the catalog}} * 100 \tag{2.13}$$

2.8.2 Evaluation of Classifiers

When developing machine learning models, evaluating how well the classifiers can predict the correct labels for new data is crucial. This evaluation process considers various factors, such as the class distribution of the dataset, the cost of false

positives and false negatives, and the interpretability of the model, in addition to traditional metrics. Evaluation is an essential step to ensure that the model performs well on unseen data and identifies areas for improvement.

A machine learning model's classification accuracy (Equation (2.14)) is defined as the percentage of correct predictions the model made. Like recommender systems, precision (Equation (2.14)) measures how many of the positive predictions were correct. Recall demonstrates how well the model predicts positives through sensitivity (Equation (2.16)) and negatives through specificity (Equation (2.17)). F1 score is the same as previously reported for recommendation evaluation (Equation (2.8)) when sensitivity is used as recall.

$$\text{Classification Accuracy} = \frac{\text{TP} + \text{TN}}{\text{n of observations}} \tag{2.14}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{Total predicted positive}} \tag{2.15}$$

$$\text{Sensitivity (recall of positives)} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{Total actual positive}} \tag{2.16}$$

$$\text{Specificity (recall of negatives)} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{TN}}{\text{Total actual pegative}} \tag{2.17}$$

When evaluating a binary classification model, a Receiver Operating Curve (ROC) curve can show its performance at different thresholds by plotting TPR and FPR [63]. By adjusting the threshold, more items can be classified as positive or negative. To avoid evaluating the ROC curve multiple times with different thresholds, the AUC measures the entire area under the curve. AUC represents the probability of the model ranking a random positive sample higher than a negative one. AUC values range from 0 to 1, where 0 indicates a model whose predictions are entirely incorrect, and 1 represents a model with perfect accuracy. If the AUC value is 0.5, it means that the model's predictive ability is no better than randomly guessing. However, if the value is above 0.5, the model performs better than random guessing.

When evaluating classifiers, classification accuracy is the most commonly used evaluation metric [56]. However, other metrics may be better suited depending on the situation. For example, in cases where positive or negative cases are rare, accuracy can be biased towards the majority class. Precision is useful when the cost of FP or TP is higher than FN or TN, respectively, while recall is more appropriate when the cost of negatives is higher than positives. F1 is a good option for imbalanced datasets as it weighs precision and recall equally, and considers class distribution. Despite alternatives, accuracy remains a widely used metric in research as it is a general and intuitive way to evaluate classification tasks. Yet, multiple metrics are used to measure simultaneously. F-measures and AUC are commonly used for imbalanced datasets, with AUC being the most proper performance measure for imbalanced datasets [64].

2.9 Personality

Personality is the consistency of differences between humans [65]. According to studies like [66] and [67], a user's personality can provide valuable insights. In fact, personality traits can greatly influence decision-making, as highlighted in [24]. Personality is context-independent, meaning it remains unchanged and predictable over time, location, or other contexts [8, 9, 14]. It is also domain-independent and easily generalized to other domains. This means it can be used cross-domain; personality may be just as useful in the movie, music, and shopping domain. These traits can be quantified and represented as vectors, which makes them a useful input for computer algorithms.

Five-Factor Model (FFM)

When it comes to discussing personality, there are various models to take into account. The Five-Factor Model (FFM) of personality is one of the most commonly used because it measures personality traits across five distinct factors, which are frequently referred to as OCEAN [24, 67, 68]. OCEAN assigns a numerical score to each factor, including openness, conscientiousness, extraversion, agreeableness, and neuroticism. This model is also known as the Big Five. In this thesis, we use the wording personality trait value and OCEAN value interchangeably.

Ricci's "The Recommender Systems Handbook" Ricci *et al.* [24] offers a clear explanation of the OCEAN personality traits, which were first introduced in McCrae and John's journal "An Introduction to the Five-Factor Model" in 1992 [68]. Below is a summary of how each trait is described in the book.

Openness (O)
Openness, often called openness to experience, refers to an individual's willingness to consider new ideas and perspectives. A high score in this trait suggests the person is creative, imaginative, and in touch with their emotions. Conversely, a low score indicates a preference for clear routines and practical thinking.

Conscientiousness (C)
Conscientiousness (C) is characterized by impulse control, self-discipline, and cautiousness. People with high C values tend to prefer making plans and sticking to them, while those with low C values tend to act spontaneously.

Extraversion/extroversion (E)
A person who is high in Extraversion tends to be friendly and assertive and enjoys seeking out excitement. The level of Extraversion is related to how much a person interacts with the external world. A high level of Extraversion suggests a lot of interaction, while a low level suggests a lack of external interaction. Essentially, Extraversion can indicate how sociable and outgoing a person is. Those with low levels of E tend to be quiet and reserved in social situations.

Table 2.1: The five factors of the FFM and the correlated adjectives

Factor	**Adjectives**
Openness (O)	Artistic, curious, imaginative, insightful, original, wide interest
Conscientiousness (C)	Efficient, organized, planful, reliable, responsible, thorough
Extraversion (E)	Active, assertive, energetic, enthusiastic, outgoing, talkative
Agreeableness (A)	Appreciative, forgiving, generous, kind, sympathetic, trusting
Neuroticism (N)	Anxious, self-pitying, tense, touchy, unstable, worrying

Agreeableness (A)
Individuals who score high in the Agreeableness values tend to be cooperative, warm, and friendly. Those with a strong A value are optimistic about human nature and have a desire to assist others. Conversely, a low A value may suggest that a person prioritizes their own needs over others, resulting in uncooperative and unfriendly behavior.

Neuroticism (N)
Neuroticism is a measure of a person's emotional reactivity. Those with high levels of Neuroticism are more likely to have strong emotional reactions regardless of the size of the trigger. In contrast, those with lower levels are generally more emotionally stable and balanced. People with high Neuroticism often experience negative emotions and feelings.

Additionally, the Recommender Systems Handbook [24] presents a list of adjectives describing each trait's characteristics. These adjectives are summarized in Table 2.1.

Traditionally, a comprehensive test is administered to measure the Big-Five personality dimensions. This may involve completing the self-report Big Five Inventory (BFI) [69], responding to hundreds of statements in an interface, or observing behavior in gamelike situations [12]. When time is limited, a shorter inventory comprising five or ten items may be used [70]. Alternatively, newer methods of automatic personality detection have been developed, which are less precise but more convenient [71]. For example, personality detection has been performed by analyzing social media posts containing the user's opinions.

Chapter 3

Related Work

This chapter provides an overview of research papers relevant to the project goal of inferring personality traits from personalized movie recommendations. Further, it identifies research gaps and opportunities for further exploration. The findings aim to advance the attempt to infer personality traits from movie recommendations. During a fall report conducted as preliminary work for this master's thesis, the authors undertook a literature study with the aim of understanding ethical aspects connected to recommender systems, particularly privacy issues. The findings of the fall report shaped the project goal and direction of this master's thesis. As a result, this chapter is heavily inspired by findings from the aforementioned literature study conducted in the fall report, with additions to related work discovered during this master's thesis. We utilized four distinct academic search engines in our literature study, namely Google Scholar, Scinapse, Scopus, and Semantic Scholar.

3.1 Privacy in Recommender Systems

The topic of privacy in recommender systems has been discussed in previous works, particularly seeing as implementing a personalized recommender system necessitates collecting user data. In 2020, Milano *et al.* [7] presented the first systematic literature review on the ethical challenges of recommender systems and stated that research on ethical issues from recommender systems was still in its early stages. They identified six main ethical concerns, with privacy being one of the primary. The authors stated that privacy breaches can occur in four stages: data collection, data storage, inference, and the inability to protect such inferences.

As mentioned in the introduction, Jeckmans *et al.* [3] claims that user privacy risks in recommender systems are often underestimated. According to Kobsa and Schreck [72], it is important to maintain an emphasis on privacy in recommender

systems as it helps mitigate the risks of data being exploited by unauthorized parties. Friedman *et al.* [45] adds that recommender systems are particularly vulnerable to privacy breaches and violating Fair Information Practices due to their large amounts of data that can potentially reveal a user's personal preferences. They mention three aspects of a recommender system at risk due to inference of new data: exposure of sensitive information, targeted advertising, and discrimination. Among these aspects, this thesis will specifically focus on the exposure of sensitive data.

In their study, Resheff *et al.* [73] pointed out that using recommendations or user representations in a recommender system could reveal private information about users, posing a risk to their privacy. Therefore, they recommend further research on the information in top n recommendations, as sharing such information could seriously affect the user's privacy and security. In this thesis, the top n recommendations will be the basis for our experiments.

Privacy is especially challenging in cross-domain recommender systems [27]. Utilizing knowledge about users from multiple domains, and thus also sharing this knowledge between the platforms, increases the risk of a privacy breach. One domain with inferior security can be used as the source domain to violate privacy in a more important and secure domain.

3.2 Personality in the Movie Domain

Previous work has indicated a significant relationship between personality traits and people's preferences and interests, as well as defining it as a trustworthy source to characterize habits and behavior. According to Tkalčič and Chen [32], personality is domain-independent, seeing that it does not change between domains such as movies, books, and music. It is shown that personality influences users' media preferences [74] and behavior [75]. Even further, work has specifically demonstrated a correlation between personality traits and preferred movie genres [10, 15, 76, 77]. The correlation between the five OCEAN personality traits and movie genre discovered in previous works is summarized in Table 3.1.

Nguyen *et al.* [8] have explored the connection between movie preference and personality traits. Their experiment determined users' personality traits using the Ten-Item Personality Inventory (TIPI) and asked them to rate their satisfaction with a personalized list of recommended movies. The lists varied in levels of diversity, popularity, and serendipity. Previous work assumes that all users have the same preference for the levels of these properties. However, their work points to a correlation between different personality traits and the preference for the three aforementioned recommendation properties. The study by Nguyen *et al.* [8] suggests that user satisfaction improves when personality traits are integrated into the process of generating recommendations, paving the way for personality-based recommender systems. The dataset from the user study is further explained

Table 3.1: Preferred movie categories by personality traits according to the literature studied in the related works.

Paper → **Traits ↓**	**Khan *et al.* [76]**	**Cantador *et al.* [10]**	**Karumur *et al.* [77]**	**Wu and Chen [15]**
Low Openness	-	action, comedy, romance, war	adventure, fantasy, thriller	documentary, romance
High Openness	comedy, sci-fi	comedy, cult, fantasy, foreign, independent, neo-noir, tragedy	drama, romance	animation, comedy, music
Low Conscientiousness	-	animation, cartoon, cult	action, thriller	animation, comedy
High Conscientiousness	action, adventure, thriller, sci-fi	action, adventure, independent, sci-fi	romance	sci-fi, war
Low Extraversion	-	animation, neo-noir, sci-fi, tragedy	romance	crime, mystery
High Extraversion	-	action, comedy, drama, romance	-	romance
Low Agreeableness	-	animation, cult, horror, neo-noir, parody	-	animation
High Agreeableness	-	adventure, comedy, drama, romance	-	sci-fi, war
Low Neuroticism	-	adventure, independent, war	action, adventure, fantasy, thriller	adventure, documentary, history
High Neuroticism	drama	animation, cult, romance, tragedy	comedy, romance	animation, drama, romance

in Section 4.1.1 and will be referred to as the Personality2018 dataset.

3.3 Personality-based Recommender Systems

As a result of research indicating a correlation between personality and effective recommendations, many are researching and developing personality-based recommender systems. For instance, Hu and Pu [78] incorporate personality information to enhance collaborative filtering systems. Their results support that the incorporated personality information effectively addresses the cold start system, with a cascade hybrid approach performing the best in terms of prediction and classification accuracy. Nalmpantis and Tjortjis [79] combine collaborative techniques with a personality test to provide more personalized movie recommendations. They conclude that users preferred a 50/50 method combining personality with kNN over just using a standard kNN-based method.

In their study published in March 2023, Lu and Kan [71] researched the potential benefits of incorporating personality traits into recommendation systems. The authors selected the neural collaborative filtering (NCF) model developed by He *et al.* [80] as the foundational model for their investigation. Lu and Kan designed a personality-enhanced version of the NCF and proposed three methods for integrating personality traits: most-salient personality, soft-labeled personality, and hard-coded personality. To assess the effectiveness of these recommender systems, they compared them against two baseline NCF models that did not utilize personality traits: random personality assignment and uniform assignment of the same trait for all users. The evaluation was conducted on three datasets: Amazon-beauty and Amazon-music, where personality traits were inferred from texts as the first part of Lu and Kan's study, and the Personality2018 dataset from Nguyen *et al.* [8]'s study presented previously. The performance of the recommender systems was measured using the HR@k and NDCG@k metrics. Across all three datasets, one of the three personality-based NCF models consistently outperformed the two baseline NCF models. Moreover, the NCF models that include all five personality traits, namely NCF+Hard-Coded and NCF+Soft-labeled, yielded better results in terms of NDCG compared to NCF+Most-salient, which only incorporates the personality trait with the highest value.

However, the improvements observed in the Personality2018 dataset were less significant compared to the Amazon-beauty dataset. The researchers hypothesize that this discrepancy may be attributed to the smaller size of the Amazon-beauty dataset and the potential for personality information to help the data sparsity problem. Nevertheless, the NCF+soft-labeled personality approach surpassed all other algorithms regarding HR and NDCG for all k values in the Personality2018 dataset, specifically achieving HR@10 = 0.805 and NDCG@10 = 0.511.

The researchers further examined how the inclusion of personality information influenced the performance for each trait by comparing the soft-labeled approach

to the baseline NCF+same. While the Amazon-beauty dataset exhibited noticeable improvements across all five personality traits, the Personality2018 dataset only displayed minor enhancements in Conscientiousness, Extraversion, and Agreeableness. This analysis shows that certain personality traits can benefit more from a Personality-based Recommender System (PBRS) and that the trait that benefits the most varies depending on the dataset.

In summary, Lu and Kan's empirical results indicate a 3-28% boost in recommendations when incorporating personality traits. However, further investigation is required to understand the underlying mechanism of how personality influences recommendations. Based on the findings presented in the paper, we will utilize the NCF+soft-labeled personality approach in our thesis, as it demonstrated the best performance on the Personality2018 dataset we will be using.

3.4 Inference of User Attributes

Sensitive information can be derived through an inference attack by exploiting available data. Multiple features and inputs have been used to infer user attributes. Bi *et al.* [81] used search query history to infer age, gender, and political and religious views using logistic regression with L2 regularization. The research by Feng *et al.* [82] explains how users' viewing history can infer the gender of viewers of an online video system. Meanwhile, Jia *et al.* [83] introduced an inference model called AttriInfer, which uses social graphs to infer location and interests based on user networks, utilizing Loopy Belief propagation.

Another feature frequently used to infer user attributes is user ratings. A study by Weinsberg *et al.* [5] found that a user's gender can be inferred through their ratings, with up to 80% accuracy, using classifiers like Bayesian classifiers, logistic regression and SVM. To protect user privacy, the researchers obfuscated the ratings by adding 1% additional ratings, reducing inference accuracy by 80% while still providing accurate recommendations. Similarly to Weinsberg, others have used ratings of items to infer attributes like gender and political views. Salamatian *et al.* [84] gathered users' ratings of TV shows and their political affiliation and performed logistic regression on that data to infer political views based on the collected ratings. Utilizing ratings in the form of public Facebook likes, Kosinski *et al.* [6] managed to infer various attributes, including age, intelligence, gender, sexual orientation, or personality traits. They employed linear regression for numeric variables and logistic regression for dichotomous variables.

To generate data for the PBRS presented in Section 3.3, Lu and Kan [71] used the computational language psychology platform Receptiviti[1] to infer personality from public product review text in the Amazon-beauty and Amazon-music datasets. The Amazon datasets did not include personality traits to evaluate the inference, however after manually analyzing the users with the top 10 highest

[1] https://www.receptiviti.com/

scores for all OCEAN traits, inferred personality matched the review text 81% and 79% of the times for the two datasets respectively.

Wu and Chen [15] implicitly derived user personality based on behavior in the movie domain. They found correlations between multiple behaviors and users' personality traits, which they validated through user surveys. They employed three regression models, Gaussian Process, Pace Regression, and M5 Rules, to automatically infer personality based on these correlations. They found that the Gaussian model performed best when testing on data from the user surveys, indicating that it is possible to infer personality based on movie-watching behavior. The inference was only tested on the users from the user surveys. They further implemented the inference model on two real-life movie datasets (Yahoo! Movie and HetRec). However, as those datasets did not contain user personality data, they could not verify the accuracy of their personality estimates. The researchers aimed to use these derived personality traits in a Personality-based Recommender System to enhance movie recommendations. Therefore, their primary emphasis was on the practical application of the inferred personalities rather than the inference process itself. Thus they carried on with their research despite being unable to validate the inferred personalities.

In this thesis, user ratings will be used to make inferences comparable to the research mentioned earlier. However, these ratings will pass through a recommender system to produce top n recommendations, from which we will infer. Additionally, we will infer users' personality traits like Lu and Kan [71], but in the movie domain like Wu and Chen [15]. Our literature review on attribute inference revealed a gap in research regarding evaluating inferred personality with ground truth and the inference of personality traits using ratings. Table 3.2 presents a collection of papers with inferred user attributes and the inference models used in their research.

3.5 Resampling

Standard classifiers like Logistic Regression, SVM, kNN, and decision trees work well for evenly balanced training sets. However, these models might not provide the best classification results when dealing with imbalanced scenarios. They succeed at accurately classifying majority cases but frequently struggle with minority samples. In the past decade, various machine-learning techniques have been created to handle imbalanced data classification [56]. This thesis will focus on resampling techniques; however, other techniques include cost-sensitive learning and ensemble methods.

According to Haixiang *et al.* [56]'s literature study, where they review methods and applications to handle imbalanced data, resampling is a popular strategy to handle imbalanced datasets, indicated by the usage of resampling technics in 29.6% of all the papers they reviewed. Other, less popular basic strategies include

Table 3.2: Description of the inference models implemented in the different papers studied in the related works.

Paper	**Inference model**	**Inferred attribute(s)**
Bi *et al.* [81]	Logistic regression (L2)	Gender, age, political views, religious views
Bhagat *et al.* [85]	Logistic regression, multinomial Naive Bayes, SVM (RBF-kernel), FBC (factor-based classifier)	Gender, age, political views
Chen *et al.* [86]	Bernoulli Naive Bayes, logistic regression, random forest	Gender, sexual orientation
Feng *et al.* [82]	Class prior, expectation-maximization (EM), SVM, logistic regression	Gender
Gong and Liu [87]	VIAL	Location
Kosinski *et al.* [6]	Linear regression, logistic regression	Sexual orientation, ethnic origin, political views, religion, personality, intelligence, satisfaction with life, substance use, age, gender, relationship status
Jia *et al.* [83]	Loopy Belief Propagation	Location, interests
Lu and Kan [71]	Receptiviti API	User personality traits
Salamatian *et al.* [84]	Logistic regression	Political views (republican or democrat)
Slokom *et al.* [88]	Logistic regression, SVM	Gender
Weinsberg *et al.* [5]	Logistic regression, SVM, Bayesian	Gender

cost-sensitive learning and future selection and extraction. Popular methods for oversampling of the minority class were random oversampling with replication (ROWR) and synthetic minority oversampling technique (SMOTE). The most effective undersampling method was randomly undersampling (RUS) the majority-class samples. Oversampling was the most frequently used resampling method 84 times, with undersampling being present 39 times in the study and a hybrid solution 33 times.

A study on corporate bankruptcy was conducted by Zhou [64] to investigate the impact of various sampling methods on the performance of corporate bankruptcy prediction models. The study used highly imbalanced datasets to compare the

results in real-life situations. Measuring six different sampling methods' performance on five quantitative prediction models, where NN and SVM performed the best, Zhou [64] concluded with three insights about resampling, summarized in Haixiang *et al.* [56] and supported by [89] and [90]:

1. If there are numerous minority observations in the dataset, using an under-sampling method instead of an over-sampling method is recommended due to faster computational processing.
2. If there are only a few dozen minority instances, it is better to use the over-sampling method known as SMOTE.
3. In the case of a large training sample size, a combination of SMOTE and under-sampling is suggested as an alternative.

Cateni *et al.* [91] suggested a resampling method specifically designed for binary classification of imbalanced datasets. The proposed method plans to fix the issues of losing important info by undersampling and oversampling by not adding synthetic data. Cateni *et al.* [91] first split the data into a train-test split, maintaining the same ratio between the classes as the dataset. In their research, they chose a limit for the maximum resample of the minority class to be 50% of the dataset so as not to lose the importance of the majority class. Their experiments test ratios starting from the imbalanced dataset's ratio, working their way up to that limit. They propose a combination of over and undersampling called SUNDO to balance different example cases before classifying the data using the common classifiers SVM, decision trees, a labeled Self-Organizing Map (SOM), and a Bayesian decision tree. They conclude that this proposed approach outperforms the widely adopted combination of SMOTE oversampling and random undersampling.

We conclude that existing literature on resampling does not state an exact rule for the ratio for which to resample or the balance between over- and undersampling because it all depends on the specific case. In fact, Haixiang *et al.* [56] note that no resampling methods require an exact balance between majority and minority classes. Following, within the scope of this thesis and because the features we will infer are highly imbalanced, we will conduct two simple resampling experiments to improve personality inference attempts. In the two experiments, we will separately test over and undersampling to this ratio to test if either of the two options indicates promising results for our scenario. Like Cateni *et al.* [91], we will maintain the original class ratio by stratifying our samples in our train-test split before resampling.

3.6 Limitations of Related Work

Other researchers have inferred user attributes from user ratings, but to our knowledge, limited attention has been given to inferring attributes from recommendations based on these ratings. However, privacy concerns arise considering the

implications of top n recommendations, and it is necessary to investigate the potential information leakage associated with them and what can be inferred from them. Cross-domain recommender systems present a greater challenge for privacy, which research suggests that personality can be useful for. With Personality-based Recommender Systems on the rise, it is evident that research on possible information leakage of personality traits from recommender systems must be done. While some studies have explored other methods for inferring user personality than those investigated in this thesis, the scarcity of datasets containing both personality and ratings poses a significant challenge in validating the accuracy of personality inferences. This highlights the need for novel contributions to the field by addressing privacy concerns related to inferring personality from top n recommendations and determining the feasibility of such inferences.

Chapter 4

Data and Technology

This chapter encompasses the exploratory data analysis (EDA) of the data used in our thesis, as well as explaining the data processing and cleaning applied. Additionally, the technologies relevant to our experiments are introduced.

4.1 Data

Previous research has strongly preferred using the MovieLens and Flixster datasets to infer attributes, such as gender, through movie ratings. In fact, Weinsberg *et al.* [5], Slokom *et al.* [88], and Bhagat *et al.* [85] all employed both datasets to conduct inference attacks. Additionally, Feng *et al.* [82] relied on the MovieLens dataset to infer gender. More information about MovieLens can be found in Section 4.2. Other datasets, such as the Politics-and-TV (PTV) dataset, have also used movie ratings for attribute inference. This dataset is mostly used to infer political views [84, 85]. The now discontinued myPersonality dataset from Facebook was also used to infer private highly sensitive attributes based on digital records of behavior [6, 81].

4.1.1 Relevant Datasets

Our research will utilize the Personality2018 [8], and MovieLens datasets [92]. These datasets were created by GroupLens[1], a research lab located at the University of Minnesota's Computer Science and Engineering department. GroupLens is known for producing credible datasets widely used in academic and educational settings for recommender systems. They have been researching recommender systems since 1992 and currently operate multiple recommendation services, including the movie recommendation website, MovieLens[2] [93]. Overall, GroupLens aims to improve the field of recommendations through its services.

[1]https://grouplens.org/
[2]https://movielens.org/

The Personality2018 Dataset

A user experiment on MovieLens was conducted by Nguyen *et al.* [8] in 2015, from May 12th to October 14th. The participants were previous platform users who had already rated at least 15 movies. Users were recruited through an invitation when logging in to the website. In total, 1888 users participated. The participants received a personalized list of 12 movies chosen from their top 60 recommended movies, with random levels of serendipity, diversity, and popularity. Additionally, each participant received a set of 10 questions aimed at assessing their personality traits. The resulting distribution of personality traits is listed in 4.1.

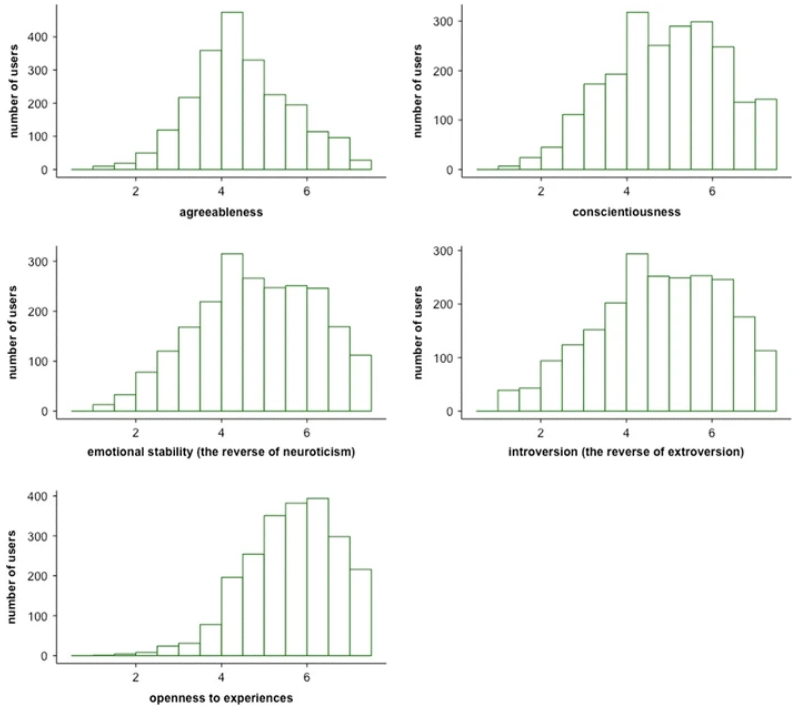


Figure 4.1: Distribution of the five personality traits of participants in Nguyen *et al.* [8]'s research. Figure from [8].

The Personality208 dataset[3] includes Rating data and User-Personality data. The Rating data stores the participants' movie ratings and personality traits. Of relevance to this thesis, the rating entity includes the user's Id, the rated movie's Id, and a score from 0 to 5, indicating how much the user enjoyed the movie. The User-Personality data stores each user's personality scores on a 7-point Likert scale for each OCEAN personality attribute. According to the dataset creator, Nguyen *et al.* [8], users with a score greater than four for a trait are considered high, while those with a score of four and below are considered low.

[3]https://grouplens.org/datasets/personality-2018/

The MovieLens Dataset

The MovieLens Datasets [4] are compilations of data from the MovieLens website[5], which suggests movies to its registered users based on their previous movie ratings [93]. The datasets available are of different sizes and are collected at different times for various purposes. We focus on the Movie data, which includes details such as a unique movieId for each movie, its title, and a genre column with a list of genres that describe it. Some parts of the thesis experiment include rating data as well. Although user information and tags are also available in the MovieLens datasets, we do not utilize them in our thesis.

4.1.2 Dataset for Experiments

For the experiments conducted in this thesis, we will utilize the Personality2018 dataset to obtain personality information about the users. The personality information in this dataset will be referred to as the User-Personality data from now on. The users in the User-Personality data are the users for which we generate the top 10 recommendations and infer personality traits. To generate the recommendations, we evaluate recommender systems training on two sets of data: the ratings provided by the users in the Personality2018 dataset and an extended dataset that combines these ratings with ratings from the Movielens datasets listed in Table 4.1. These datasets will be referred to as the Rating data and the Rating-extended data, respectively.

To create a comprehensive collection of movies that align with the ratings in Rating data, we constructed what we refer to as the Movie data by merging multiple MovieLens datasets. Since the Personality2018 dataset does not include any movie details or correspond to a single MovieLens dataset, this merging process was necessary. We began by incorporating all movies from the most recent MovieLens dataset that had ratings in the Rating Data, which consists of ratings by the users in the User-Personality data. Secondly, we added movies from the second most recent MovieLens dataset that were rated in the Rating Data and were not an exact duplicate of any movie already added. This procedure was repeated for all non-synthetic MovieLens datasets. The datasets that contributed with relevant movies are listed in Table 4.1 in the order they were incorporated.

Data Preprocessing and Cleaning

The possible genres in the Movie data were cleaned by choosing one spelling per genre, as some had different spellings. Similarly, movie titles were also cleaned. Movies that appeared in multiple datasets with different genres had their genres concatenated. Then, duplicates and 1269 movies with no genres listed were removed. The resulting genres can be found in Section 4.1.2. Finally, the Movie data

[4] https://grouplens.org/datasets/movielens/
[5] http://movielens.org

Table 4.1: Overview of the MovieLens datasets included in the Experiment Data, showing the release date, number of ratings, number of movies rated, and number of users who provided the ratings.

Name	Released/Updated	Ratings	Movies	Users
MovieLens 25 M	12/2019	25 million	62,000	162,000
MovieLens Latest Dataset (full)	9/2018	27,000,00	58,000	280,000
MovieLens 20 M	10/2016	20 million	27,000	188,000
MovieLens 1 M	2/2003	1 million	4000	6000

was restructured so that instead of the singular genre column, each genre has its own column, with values of 1 or 0, indicating if the movie had the specified genre listed in the original column.

- Action
- Adventure
- Animation
- Children
- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- IMAX
- Musical
- Mystical
- Romance
- Sci-Fi
- Thriller
- War and
- Western

Not all ratings in the Personality2018 dataset corresponded to a movie in the Movie data, as not all movies were found in any of the MovieLens datasets or because they did not have a listed genre. Consequently, ratings for those movies were deleted from the Rating data.

Ultimately, after all the data preprocessing and cleaning, the dataset used in this thesis's experiments consists of the *Rating data* with movie ratings from the Personality2018 dataset, the *Movie data* corresponding to all the rated movies in the *Rating data* with data from MovieLens and finally, the *User-Personality data* from the Personality2018 dataset consisting of the personality traits of all users with a rating in the *Rating data file*.

Entity Relationship Diagram

Figure 4.2 provides a simplified Entity-Relationship (ER) diagram of the Experiment data that emerged from the Personality2018 and the Movielens dataset that will be used in the experiments. The diagram shows the relationships and attributes of the entities involved, specifically User-Personality data, Rating data, and Movie Data. The connections between these entities indicate how they are related to each other and with what cardinality.

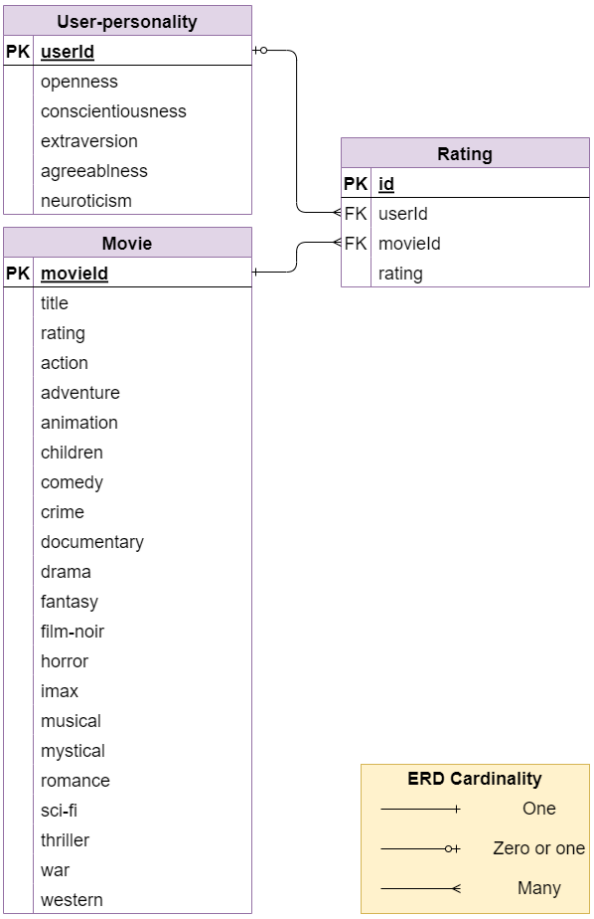


Figure 4.2: Entity-Relationship diagram of the data used in the thesis experiments.

Data Analysis

Merging and cleaning the Experiment dataset resulted in fewer users, ratings, and movies than in the Personality2018 dataset. Key numbers comparing the Personality2018 dataset to the Experimental dataset are listed in Table 4.2. The Experiment dataset refers to the pre-processed and cleaned User-Personality data, Rating data, and Movie data.

To access additional details and the code for data preprocessing and number calculation, please refer to the GitHub repository[6].

[6]https://github.com/hanntorj/masters-thesis.git

Table 4.2: Comparison of key numbers in the original Personality2018 dataset and the Experiment dataset after all pre-processing and cleaning described in Section 4.1.2 have been done.

	Personality2018 dataset	**Experiment dataset**	**Difference**
Number of users	1820	1799	-1.15%
Number of movies	35 196	33 610	-4,5%
Number of ratings	1 028 751	919 770	-10.59%
Average number of ratings per user	∼565	∼511	-54
Median number of ratings per user	∼335	∼328	-7
Average number of ratings per movie	∼29	∼31	+2
Median number of ratings per movie	∼3	∼3	0

4.2 Technology

This subsection overviews the various tools and technologies used throughout the research process.

4.2.1 Python

Python[7] is among the most popularly used programming languages for building recommender systems due to its compatibility with a wide range of data analysis and machine learning tools. Python also comes with several relevant packages for developing recommender systems. Followingly, all code for recommender systems used in this thesis is written in Python.

4.2.2 Conda

Conda[8] is an open-source package, dependency, and environment management for any programming language, including Python. In order to perform data analysis and run the recommender systems introduced in this thesis, a conda environment must be set up. All environment specifications are listed in the codebase README.

[7] https://www.python.org/
[8] https://docs.conda.io/en/latest/

Table 4.3: Recommender algorithms provided by the Lenskit package.

Lenskit
BiasedMF
ImplicitMF
FunkSVD
ItemItem
PopScore(quantile)
Popular
Random
UserUser

Table 4.4: Recommender algorithms provided by the Surprise package.

Surprise
BaselineOnly
CoClustering
KNNBaseline
KNNBasic
KNNWithMeans
NMF
NormalPredictor
SlopeOne
SVD

4.2.3 Lenskit

Lenskit[9] is a Python-based open-source toolkit used for creating recommender systems [94]. It is compatible with several other scientific Python libraries, including scikit-learn. The toolkit includes algorithms to recommend items and metrics to evaluate the recommendations. The Lenskit algorithms employed in this thesis are listed in Table 4.3.

4.2.4 Surprise

Surprise[10] is a Python sci-kit for implementing recommender systems with explicit ratings. It was designed with the purpose of making it easy to use, with an emphasis on good documentation. The toolkit includes algorithms such as SVD, NMF, and baseline algorithms for comparison reasons. The algorithms from Surprise are listed in Table 4.4.

4.2.5 Other Python Libraries

Pandas[11] is an open-source Python library useful for data science or machine learning. It offers a particular data structure known as DataFrames which allows for quick handling and manipulation of extensive data collections.

scikit-learn[12] is a Python library designed for machine learning, providing support for both supervised and unsupervised learning. It provides a wide range of established algorithms and efficient machine-learning tools, such as classification, regression, and clustering.

[9] https://lenskit.org/
[10] https://surpriselib.com/
[11] https://pandas.pydata.org/
[12] https://scikit-learn.org/stable/

PyTorch [13] is an open-source machine learning framework in Python. It supports using tensors that can live on CPU or GPU, making it useful when working with deep learning or a neural network. The package provides fast implementation, flexibility, and speed.

4.2.6 Orange Data Mining

Orange Data Mining[14] is a Python-based software tool utilized for machine learning and data mining purposes. The free software has a user-friendly interface allowing users to visualize data analysis workflows using widgets. Orange Widgets can be used for various functions such as classification, regression, or mining association rules.

[13] https://pypi.org/project/torch/
[14] https://orangedatamining.com/

Chapter 5

Method and Experiments

The goal of our master's thesis was to investigate the feasibility of employing machine learning techniques to extract insights into users' personality traits from movie recommendations, thereby posing potential threats to user privacy if done by an attacker. Specifically, we aimed to determine to what extent it is possible to infer users' personality traits from top n movie recommendations (RQ1) and how the number of classes for each personality trait impacts the inference accuracy (RQ1.1). We also aimed to explore how including personality traits in the recommender system influences the ability to infer personality (RQ2) and how resampling affects the Classification Accuracy and Area Under Curve (RQ3). Based on relevant theoretical background and related work, we have created a plaThe following section will presentriments to answer these research questions. In the following section, a comprehensive plan outlining the methodology to investigate these questions, providing a detailed explanation of each step. Prior to delving into the specifics, here is a concise preview of what you can anticipate:

- **Experiment 0: Generate and evaluate recommendations**
  The initial experiment was to implement a baseline RS, a Rating-based Recommender System, and a Personality-based Recommender System and evaluate their performance in terms of precision@10, recall@10, and F1. The primary objective of this experiment was to generate precise recommendations that are needed to effectively address the research questions at hand.
- **Experiment 1: Inference of User Personality from personalized movie recommendations**
  The fundamental experiment was to infer the personality traits based on the previously generated rating-based recommendations and evaluate the accuracy of the inference. The primary aim of this experiment was to address RQ1 and RQ1.1. Firstly, it aimed to investigate the possibility of inferring users' personality traits from movie recommendations, thereby assessing the extent to which such inference is feasible. Secondly, it aimed to examine how the number of classes for each personality trait impacts the inference

accuracy.

- **Experiment 2: Inference of User Personality Traits from Personality-based Recommendations**
  In this experiment, we compared inferring using recommendations generated with and without the influence of personality. The aim of this experiment was to answer RQ2 by exploring the impact of incorporating personality traits into the recommender system and investigating how it influences the ability to infer users' personalities from top n recommendations.
- **Experiment 3: Resampling**
  Lastly, we experimented with resampling training data aiming to improve inference AUC. The aim was to address RQ3 and examine if resampling could improve classification accuracy (CA) and area under the curve (AUC).

5.1 Detailed Plan for Experiments

In order to reach our overall thesis goal and address the research questions, we developed a detailed plan consisting of one preparation experiment for generating recommendations and three experiments. These four experiments exclude the data preprocessing and cleaning explained in Section 4.1.2. In the upcoming sections, we will thoroughly explain the process, including the procedures we used and the evaluations performed.

Experiment 0: Generate and Evaluate Recommendations

The initial experiment 0 involved generating movie recommendations for users and assessing the quality of these recommendations. This was a necessary preparation experiment before beginning the actual experiments.

First, we applied various collaborative filtering recommender algorithms to create personalized recommendations based on users' past movie preferences through their explicit ratings to find the most suited algorithm for the rating-based recommendation. The algorithms tested were state-of-the-art recommender algorithms from the Lenskit and the Surprise packages. We first trained the recommendation algorithms using the Rating data, as defined in Section 4.1.2. Furthermore, we trained the algorithms using the Rating-extended data, which is described in Section 4.1.2 as the expanded rating data to include the MovieLens ratings. Training and evaluating the algorithms on the different rating data helped us determine whether a more extensive dataset could result in more precise recommendations or if we should continue using only the ratings in the Personality2018 dataset.

To evaluate the performance of the rating-based recommender algorithms, we employed the established metrics precision, recall, and F1-score using a five-fold split. We chose the recommender algorithm with the best performance in terms of precision to generate the top 10 recommendations for all the users in the User-

Personality data. Precision was the most relevant metric because it was more important that the top 10 recommendations were true since we were basing our inference on these ten movies. In this scenario, we did not care if many relevant items were not recommended (recall) as long as the recommended ones were relevant.

Additionally, to establish a baseline for experiment 1, we generated non-personalized randomized recommendations. These recommendations should be generated using either Lenskit or Surprise's random algorithm, depending on which algorithm was selected for the rating-based recommendations. The recommendations should be generated from the same ratings that were used to train the final rating-based recommendations. This provides the most accurate baseline for comparison.

In addition to the personalized recommendations based solely on user ratings, we employed the NCF+soft-labeled Personality-based Recommender System[1] created by Lu and Kan [71] to generate recommendations based on personality types derived from the users' OCEAN attributes in the User-Personality data. This recommender system was trained on ratings from the Personality2018 dataset.

To summarize, we generated three versions of the top 10 recommendations: rating-based recommendations, randomly generated recommendations and personality-based recommendations.

In preparation for the experiments, we modified the format of the top 10 recommendations for the users to make it suitable as inference input. The recommender systems gave us ten rows for each user, with each row corresponding to a recommended movie ranked 1 through 10. Instead, we wanted one row per user with information about the 10 recommended movies. If we keep only the original information in the top 10 recommendations, each movie would be a separate feature, which might give too specific patterns to infer based on. Therefore, to transform the individual movie recommendations into a more generalized input for the classifiers, we aggregated the information in the top 10 recommendations by utilizing the predefined genres associated with each movie. This allowed us to capture patterns in the recommendations while ensuring a level of generalization that facilitates personality inference. Using information about the genres of the top 10 recommendations, we proposed three different versions: binary-class, count, and weighted. Recall from Section 4.1.2, each movie in the Movie data had a column per genre with values of 0 or 1 indicating if the movie was classified within that genre. Here we found the information to create the three versions. In the binary-class version, each genre would have a binary-class value indicating if the user had a movie belonging to that genre in their top 10 recommendations. The second version counted all instances of the specific genre in the user's top 10 recommendations. The weighted version was similar to count but weighted the ten values by dividing each value on the movie's recommended rank before summing. Figure 5.1 shows the data structure for these three suggested versions.

[1] Presented in Section 3.3

Out of these three versions, the binary-class version had the least amount of information about the top 10 recommendations. The weighted version had the most information; however, this led to a risk of inaccurate information if the ranking within the top ten movies was inaccurate regarding ranking-based metrics. As a result, we concluded that the count version was the most suitable to proceed with. This new data will be called the Recommended genres data: randomly generated, rating-based, and personality-based.

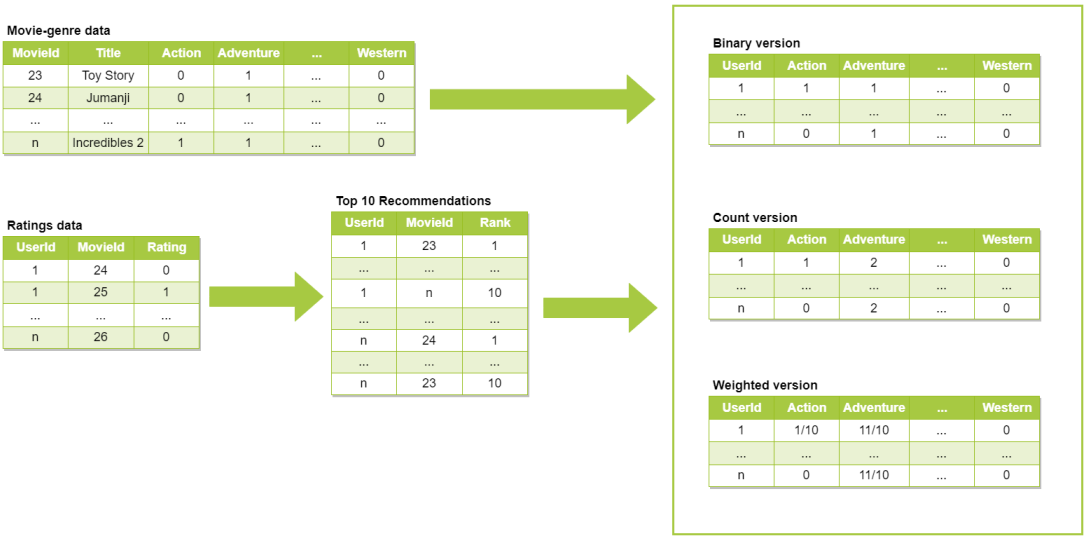


Figure 5.1: This figure demonstrates how to generate the three proposed suggestions for formatting the genres in the top 10 recommendations, which serve as input for inference. Ratings in the illustration are fictionary.

Experiment 1: Inference of User Personality from Personalized Movie Recommendations

Experiment 1 focused on inferring users' personality traits for users from the top 10 rating-based movie recommendations. We inferred the personality traits for the users in the User-Personality data, defined in Section 4.1.2. We used the randomly generated recommendations from Experiment 0 as a baseline. In addition, this experiment tested three different splits for the classes within the personality traits. The different splits allowed us to get different perspectives for the analysis, ensuring that our classification is neither overly complex nor overly simplified. Therefore, we will repeat the following process in this experiment once for each personality trait split for the rating-based Recommended genres and randomly Recommended genres, resulting in six iterations. Each iteration is done for each of the five OCEAN attributes.

The three personality splits are the seven-class split, the three-class split, and the binary-class split. We created the three new versions from the original 7-point Likert scale User-Personality data. The seven-class split round all personality trait

values down to the nearest integer. The three-class split reassigns the personality trait values into three categories Low, Medium, and High. Values under three are classified as Low, three to five as Medium, and above five as High. The third is a binary-class split categorizing users as having low or high OCEAN values. In the binary-class split, personality trait values of four and below on the original scale are considered Low, while values above four are considered High. We used the same criteria for the binary-class version as Nguyen *et al.* [8], who created the numeric dataset. For the three-class split, we used numbers from the online version[2] of the TIPI created by Gosling *et al.* [70].

The selected User-Personality data was merged on userId with the count version of the Recommended Genres data (Figure 5.1) in the Orange Data Mining application. This resulted in a data table with one row per user denoted by their userId. The subsequent columns correspond to personality trait values, representing the OCEAN traits of each user. The table comprises columns for each genre listed in Section 4.1.2 with values for that user's weighted genre score.

Then the data was split into train and test data, using the *Data Sampler* widget in Orange Data Mining, with ten-fold cross-validation with stratified samples to maintain the original class ratios. Separately, we used the *Select Columns* widget on the train data and the test data to select all 19 genres as features, the userId as Metas, and the allotted OCEAN attribute as the target value.

The train data was then sent into the classifiers widgets for the classifiers listed below. All classifiers were utilized without parameter adjustments. This approach allowed for a fair comparison among the different classifiers. By evaluating their performance in their out-of-the-box form, we can leave identifying the most promising classifier(s) for future investigation and refinement. This strategy systematically assesses the classifiers' initial capabilities before considering potential modifications or optimizations.

List of Classifiers:

- *kNN*
- *Logistic Regression*
- *Naive Bayes*
- *Neural Network* (MLP)
- *Random Forest*
- *SVM with kernel RBF*

To evaluate the accuracy of personality inference, we passed the learner output from the classifiers, the train data, and the test data from before into the *Test and Score* widget. Selection of the option to *Test on test data* for this widget compared the predicted traits with ground truth data for the unseen test data for the different classifier models. Followingly, the widget presented AUC, Classification Accuracy

[2]https://psytests.org/big5/tipien.html

(CA), precision, recall, and F1 for each of the classifiers' ability to infer the selected personality trait.

Experiment 2: Inference of User Personality Traits from Personality-based Recommendations

During our literature review, we discovered that the inference of user attributes poses a significant privacy threat and that the adoption of Personality-based Recommender System (PBRS) has witnessed a notable surge. Motivated by these findings, we developed a hypothesis before conducting Experiment 2. We hypothesized that inferring personality traits from personalized recommendations would be more feasible and accurate if the recommendations incorporated the user's personality as an additional factor. We speculated that including personality traits in the recommendations might inadvertently result in information leakage, facilitating a link between the ability to infer attributes and the personalized recommendations themselves. The primary objective of this experiment was to test this hypothesis and explore potential correlations between personality inference accuracy and the integration of personality in Personality-based Recommender Systems.

In Experiment 2, the same inference as in Experiment 1 was conducted, but with two alterations. First, this experiment aimed to investigate how including personality traits when generating recommendations affects the inference of those traits. Therefore, the inference process in Experiment 1 was conducted on the Recommended genre data based on the personality-based recommendations and compared to the inference from the rating-based Recommended genres data. Secondly, this experiment only used the binary-class User-Personality data to keep simplicity and clarity in the experiment as well as resource efficiency, as the focus was on the effects of adding personality to the recommendations. This will make the analysis and interpretation of results more straightforward, however, we must be aware of the oversimplification of personality. Yet, it will serve as a suitable initial exploratory phase for testing the hypothesis that inferring personality traits based on personalized recommendations, derived from both personality and ratings, is expected to yield more accurate results than recommendations solely based on ratings.

Experiment 3: Resampling

During the preliminary data analysis in Chapter 4 we observed an imbalance within each personality trait in the user dataset. This prompted us to investigate the potential benefits of resampling the training data during classification. To address this issue, we conducted a resampling experiment, focusing specifically on the binary-class User-Personality data, as it had the fewest amount of classes among the available versions of the User-Personality data. Additionally, we only conducted the resampling experiment using recommendations from the

Personality-based Recommender System because, in Experiment 2, we hypothesized that personality inference would be more accurate with the personality-based recommendations than the rating-based recommendations by including the attribute that would be inferred. We aimed to determine if an approach with resampling could yield improved inference results compared to the original results based on PBRS with no user resampling. We hypothesized that resampling might result in an increase in AUC.

To achieve a balanced representation of the minority and majority classes, we aimed for a 1:1 ratio. To explore this, we conducted two versions of the experiment; one with only oversampling and one with only undersampling, both with random resampling. The experiment followed the classification process in Orange Data mining from Experiment 2 as a starting point.

To perform the resampling experiments, we added the *Select row* widget to the train data to identify and select all rows where the specified personality attribute was labeled as "High". Subsequently, we divided the data into two paths; one path directed the data labeled "High" through a *Resample* widget, while the other path directed the labeled "Low" through another. This segregation allowed us to separately apply the resampling technique to the two classes.

In the first version, we employed undersampling, decreasing the number of instances in the majority class to match that of the minority class. This approach ensured that both classes were equally represented in the dataset. In the second version, we implemented oversampling, increasing the instances of the minority class to align with the number of instances in the majority class. By applying both undersampling and oversampling techniques, we sought to evaluate the impact of balancing the dataset on the performance and accuracy of our analysis.

This gave updated results in *Test and Score* widget. First, for undersampling to ratio 1:1. Secondly, by oversampling to ratio 1:1. These two sub-experiments were repeated with all five OCEAN attributes. Next, we compared the inference results from the two resampling experiments to previous results for the PBRS from Experiment 2 without resampling on the binary-class User-Personality data for the PBRS.

For this experiment, we compared both Classification Accuracy (CA) and Area Under Curve (AUC) with no resampling, oversampling, and undersampling. As we aim to improve classification on an imbalanced dataset, we wish to increase the AUC; the background theory is explained in Section 2.8.2.

5.1.1 Experimental Setup

Figure 5.2 visually represents the overall process of the thesis and the four experiments. Table 5.1 provides information on the corresponding iteration belonging to each experiment. The depicted Rating, User-Personality, and Movie data are presented in Section 4.1.2.

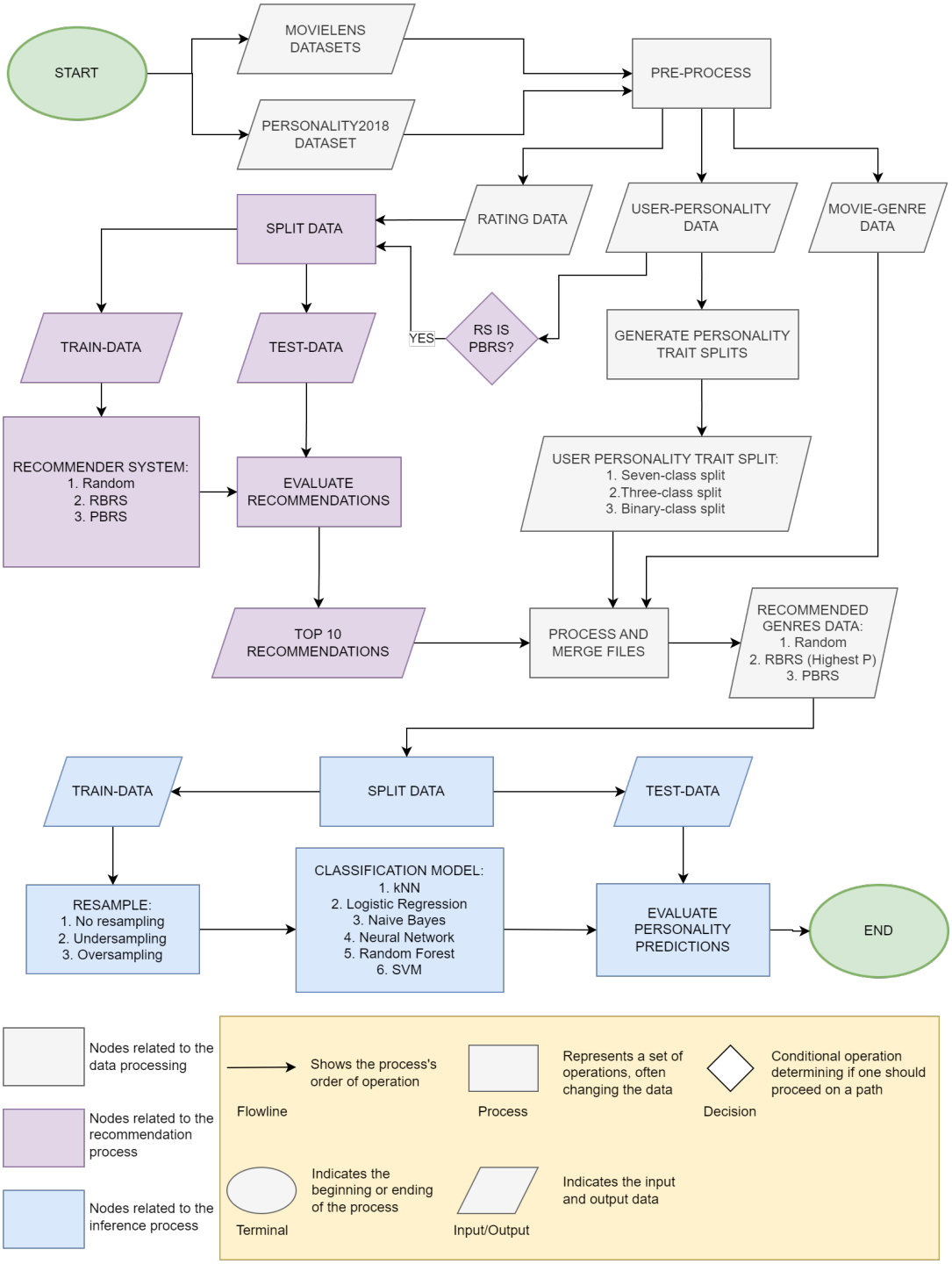


Figure 5.2: Diagram showing the entire experiment process, showing how Experiments 0 through 3 are connected. The diagram shows the data processing process, the recommendation process, and the steps included to infer user personality from the movie recommendations.

Table 5.1: Iterations of each experiment. Each iteration is done for each OCEAN value with all classifiers listed in Section 5.1.

	Recommender System	**User-Personality Split**	**Resample**
Experiment 1			
Iteration 1	Random	Seven-class split	No resampling
Iteration 2	RBRS	Seven-class split	No resampling
Iteration 3	Random	Three-class split	No resampling
Iteration 4	RBRS	Three-class split	No resampling
Iteration 5	Random	Binary-class split	No resampling
Iteration 6	RBRS	Binary-class split	No resampling
Experiment 2			
Iteration 1	RBRS	Binary-class split	No resampling
Iteration 2	PBRS	Binary-class split	No resampling
Experiment 3			
Iteration 1	PBRS	Binary-class split	No resampling
Iteration 2	PBRS	Binary-class split	Undersampling
Iteration 2	PBRS	Binary-class split	Oversampling

Chapter 6

Results

This chapter presents the results from the four experiments described in Chapter 5. The experiments involved generating and evaluating recommendations, inferring and evaluating personality traits with different numbers of classes for each personality trait, inferring based on personality-based recommendations, and experimenting with resampling. For related recommender and classification model theory, please refer to Chapter 2. Additionally, Chapter 4 provides information on all datasets and technology used in the experiments.

6.1 Experiment 0: Generating and Evaluating Recommendations

6.1.1 Rating-based and Baseline Recommender System

This section presents the evaluation results for the rating-based recommender algorithms from Lenskit and Surprise explored in the experiment. It also presents results for the baseline recommender systems. The code for evaluating algorithms and generating top n recommendations is publically available in the GitHub repository[1].

Lenskit
We experimented with various recommender algorithms from Lenskit using the Rating data defined in Section 4.1.2. The evaluation metrics of the generated recommendations are presented in Table 6.1. The algorithm that performed best regarding precision@10 was the Popular algorithm, with a precision of 0.38, meaning the Lenskit non-personalized algorithm performed better than the personalized rating-based algorithms.

The best personalized rating-based algorithm, ImplicitMF, had practically half the precision as the Popular algorithm. All other algorithms were less than five times

[1] https://github.com/hanntorj/masters-thesis.git

Table 6.1: Evaluation metrics of the recommender algorithms from Lenskit training on Rating data with ratings from Personality2018.

Algorithm	Precision@10	Recall@10	F1
Popular	0.38	0.07	0.12
ImplicitMF	0.21	0.07	*0.1*
FunkSVD	0.07	0.01	0.01
BiasedMF	0.03	0.01	*0.01*
Itemtem	0.0	0.0	0.0
Random	0.0	0.0	0.0
UserUser	0.0	0.0	0.0

as good as the Popular, indicating that the algorithms performed relatively poorly on the Rating data that included only ratings from the Personality2018 dataset. Therefore, we sought to enhance our recommender system by expanding our dataset to include ratings from the MovieLens datasets as defined in Section 4.1.2. After running the Rating-extended data through a selection of the Lenskit recommender algorithms, we evaluated the algorithms using the same metrics as previously to see if the results indicated that the algorithms performed better with a bigger dataset. The evaluation results are shown in Table 6.2 and imply that the Popular algorithm still had the highest precision; however, it decreased to 0.14 compared to 0.38 previously. The other algorithms also saw a decrease in performance, leading to the conclusion that there was no registered improvement in performance when adding ratings from the Movielens dataset compared to only Personality2018 ratings. Ultimately, we decided that the Lenskit recommender algorithms did not perform to our expectations, and we proceeded without using the recommendations generated by the Lenskit library.

Table 6.2: Evaluation metrics of the recommender algorithms from Lenskit training on Rating-extended data with ratings from both Personality2018 and Movielens.

Algorithm	Precision@10	Recall@10	F1
Popular	0.14	0.07	0.1
ImplicitMF	0.11	0.09	0.1
BiasedMF	0.02	0.01	0.01
Itemtem	0.01	0.0	0.0

Surprise

We applied various recommender algorithms to the Rating data using the Surprise toolkit. Evaluating the algorithms resulted in the metrics presented in Table 6.3. Specifically, the SVD algorithm demonstrated the highest precision@10 from Surprise with a precision of 0.87.

We also wanted to evaluate the SVD algorithm training on the Rating-extended

data as attempted on the Lenskit algorithms; results are presented in Table 6.4. This gave a precision of 0.76, displaying the same decrease in performance in terms of precision@10 when adding MovieLens ratings as the Lenskit algorithms displayed, and we decided not to continue with the Rating-extended data any further.

As we wanted to use the rating-based algorithm with the highest Precision@10 in the next experiments, we decided to proceed with the Surprise SVD recommendations trained on the Rating data.

In the end, as we utilized the Surprise package for generating rating-based recommendations, we also wanted to opt for a Surprise algorithm for our baseline recommendations to compare the personalized RBRS with a random RS within our inference model. We further decided to incorporate the NormalPredictor from Surprise as that baseline recommender system to generate random recommendations for each user.

Table 6.3: Evaluation metrics of the recommender algorithms from Surprise training on Rating data with ratings from Personality2018.

Algorithm	**Precision@10**	**Recall@10**	**F1**
SVD	0.87	0.3	0.45
KNNBasic	0.86	0.32	0.46
NMF	0.81	0.29	0.42
NormalPredictor	0.69	0.21	0.33

Table 6.4: Evaluation metrics of the SVD algorithms from Surprise training on Rating-extended data with ratings from both Personality2018 and Movielens.

Algorithm	**Precision@10**	**Recall@10**	**F1**
SVD	0.76	0.57	0.65

6.1.2 Personality-based Recommender System

For the Personality-based Recommender System (PBRS), we utilized the NCF+soft-labeled recommender system developed by Lu and Kan [71], discussed in detail in Chapter 3. This recommender system uses neural networks to recommend movies based on the ratings from the Personality2018 dataset while also incorporating the users' personality traits into the recommendation process. We obtained access to the code upon request from the author, as it is not publically available.

In our evaluation of the PBRS, we utilized the same two key metrics as Lu and Kan [71], namely Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG). Evaluating the NCF+soft-labeled personality approach, the best epoch on our dataset resulted in a HR@10 of 0.83 with a NDCG@10 of 0.59. Our results were in line with Lu and Kan's findings, who reported an HR@10 of 0.805 and

an NDCG@10 of 0.511 for the entire Personality2018 dataset with the NCF+soft-labeled approach.

6.2 Experiment 1: Inference of User Personality From Personalized Movie Recommendations

Experiment 1 was conducted using several classification models separately on rating-based recommendations generated in experiment 0 with different personality trait splits. The experiment was also conducted on the non-personalized recommendations from experiment 0 as a baseline to compare results.

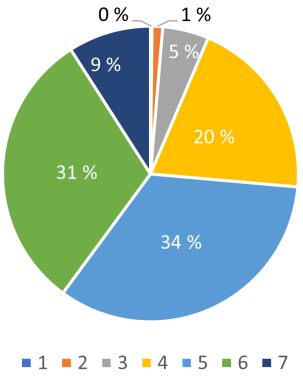
Figure 6.1, Figure 6.2 and Figure 6.3 show pie charts of the resulting distribution of the three personality trait splits: seven-class split, three-class split, and binary-class split. The coefficient of variation (also referred to as relative standard deviation) for each trait per split is listed in Table 6.5.

Table 6.5: Coefficient of Variation (CV) for each trait per personality split, with the highest CV per split shown in bold and the lowest CV underlined.

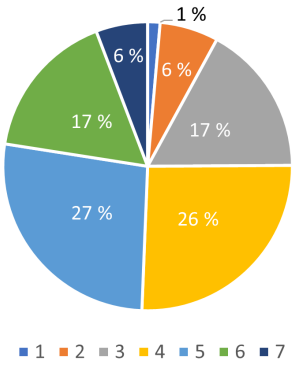
	Seven-class split	Three-class split	Binary-class split
Openness	**90.4**	70.17	**70.32**
Conscientiousness	64.68	59.22	22.07
Extraversion	<u>53.89</u>	<u>47.2</u>	40.30
Agreeableness	85.13	**84.07**	<u>12.17</u>
Neuroticism	59.83	52.86	43.08

The key results and added calculations from the inference follow below. The full evaluation of the inferences for each personality trait, using the different personality trait splits and recommendation inputs, can be found in Appendix B.

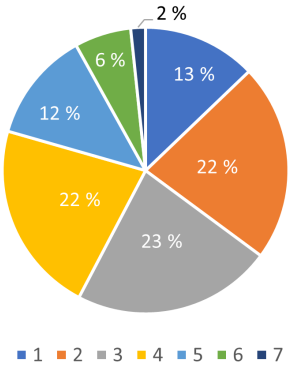
Table 6.6 represent the average Classification Accuracy (CA) for the different personality trait splits. Logistic Regression was, for every split and for both rating-based and randomly generated recommendation, the best-performing overall classifier when averaging the highest classification accuracy for the OCEAN traits. However, with LR, the inference using the random recommendations outperforms the classification accuracies of rating-based recommendations. For kNN and Neural Networks, the inference from RBRS got the highest CA compared to random recommendations. Additionally, the recommendation inputs that obtained the highest average CA among Naive Bayes, Random Forest, and SVM varied depending on the number of classes employed for personality classification.



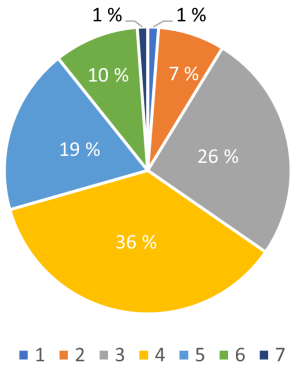
(a) Openness



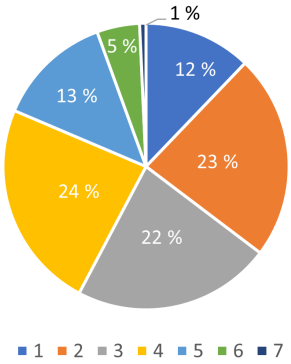
(b) Conscientiousness



(c) Extraversion

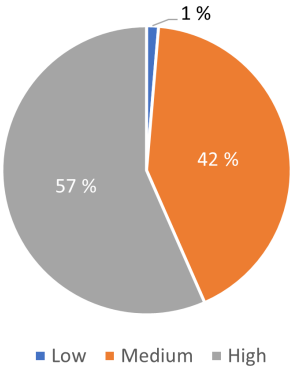


(d) Agreeableness

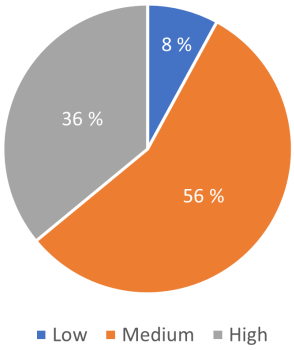


(e) Neuroticism

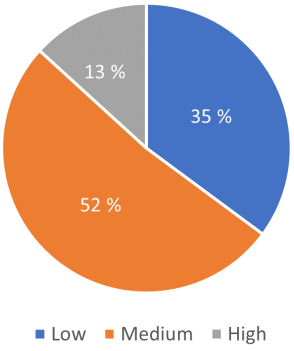
Figure 6.1: Distributions for each personality trait with the seven-class split.



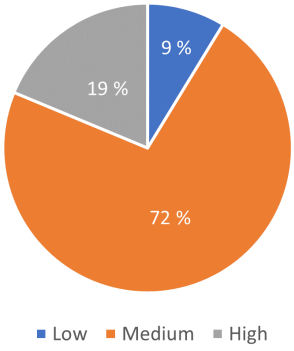
(a) Openness



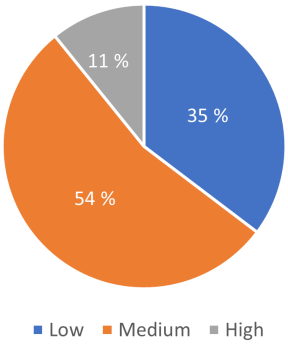
(b) Conscientiousness



(c) Extraversion

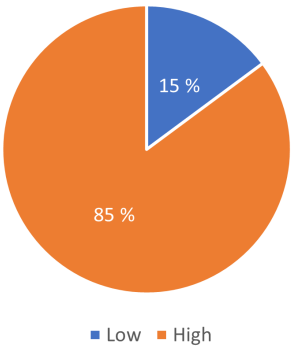


(d) Agreeableness

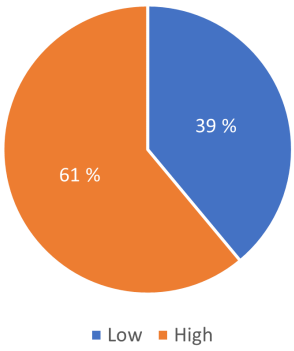


(e) Neuroticism

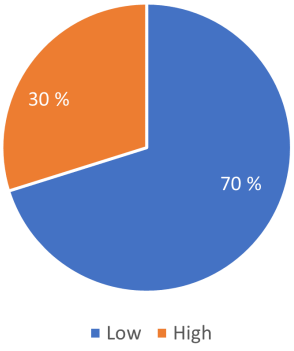
Figure 6.2: Distributions for each personality trait with the three-class split.



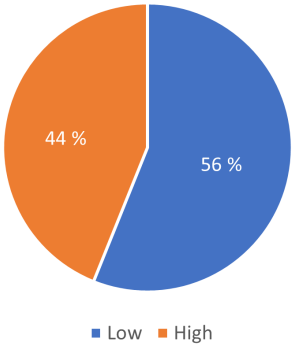
(a) Openness



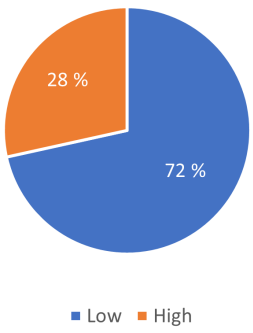
(b) Conscientiousness



(c) Extraversion



(d) Agreeableness



(e) Neuroticism

Figure 6.3: Distributions for each personality trait with the binary-class split.

Table 6.6: Average Classification Accuracy (CA) across all OCEAN traits for each classifier shown per personality trait split. The best CA for each recommendation and split are marked in bold.

	Random RS			Rating-based RS		
	Seven-class split	Three-class split	Binary-class split	Seven-class split	Three-class split	Binary-class split
kNN	0.248	0.502	0.642	0.267	0.503	0.655
LR	**0.287**	**0.607**	**0.686**	**0.271**	**0.592**	**0.683**
NB	0.184	0.582	**0.686**	0.236	0.570	0.681
NN	0.246	0.557	0.648	0.274	0.561	0.663
RF	0.232	0.572	0.646	0.269	0.527	0.668
SVM	0.248	0.471	0.534	0.256	0.503	0.502

In Table 6.7 and Table 6.8, we can see a summarized version of the best-performing classifier for each trait using random recommendations and rating-based recommendations, respectively. Regarding rating-based recommendations, Agreeableness was the most accurately classified personality trait for the seven-class split and the three-class split. In contrast, Openness achieved the highest accuracy in the binary-class split. There was a significant jump in classification accuracy for Openness from the three-class split to the binary-class split, going from 0.622 to 0.861. Agreeableness, on the other hand, went from a high accuracy of 0.789 with the three-class split to a lower accuracy of 0.583 with the binary-class split. These trends were also found for inference using random recommendations.

Table 6.7: Best classifier in terms of Classification Accuracy for each OCEAN trait, inferred from random recommendations. The classifier and CA for the trait with the highest CA for each split are marked in bold.

	Seven-class split		Three-class split		Binary-class split	
	Best classifier	CA	Best classifier	CA	Best classifier	CA
Openness	**LR**	**0.406**	LR	0.561	**NN**	**0.856**
Conscientiousness	SVM	0.278	LR	0.567	NN	0.594
Extraversion	NB	0.222	LR	0.539	LR	0.711
Agreeableness	LR	0.356	**LR/NB**	**0.789**	NB	0.600
Neuroticism	LR	0.233	LR	0.578	LR	0.706
Average	-	0.299	-	0.607	-	0.693

For Openness and Extraversion, the highest CA was higher for the rating-based recommendations for two of the splits, while for the rest of the OCEAN traits, it was only higher in one of the splits. Despite this, when averaging the highest CA score for each OCEAN trait, the value was slightly higher for the rating-based recommendations for all three splits. The traits with the highest observed increase of CA were Extraversion for the seven-class and binary-class split and Openness

Table 6.8: Best classifier in terms of Classification Accuracy for each OCEAN trait, inferred from rating-based recommendations. The classifier and CA for the trait with the highest CA for each split are marked in bold.

	Seven-class split		Three-class split		Binary-class split	
	Best classifier	CA	Best classifier	CA	Best classifier	CA
Openness	NN	0.361	NN	0.622	**LR**	**0.861**
Conscientiousness	RF	0.256	LR	0.556	NB	0.606
Extraversion	LR	0.278	NB	0.506	NB	0.728
Agreeableness	**LR**	**0.372**	**LR**	**0.789**	kNN	0.583
Neuroticism	NN	0.239	NB	0.566	LR	0.694
Average	-	0.301	-	0.608	-	0.694

for the three-class split, which can be seen in Table 6.9.

Table 6.9: The table shows observed change of classification accuracy when inferring from the rating-based recommendations compared to from the randomly generated recommendations. The numbers are calculated by subtracting the numbers in Table 6.8 from the corresponding numbers in Table 6.7.

	Seven-class split	**Three-class split**	**Binary-class split**
Openness	-0.145	0.061	0.005
Conscientiousness	-0.022	-0.011	0.012
Extroversion	0.056	-0.033	0.017
Agreeableness	0.016	0.000	-0.017
Neurotisism	0.006	-0.012	-0.012
Average	0.002	0.001	0.001

Looking at the inference from rating-based recommendations, only considering Classification Accuracy (CA), Table 6.8 suggests that Agreeableness was the most successfully inferred personality trait using a seven-class split (CA = 0.372 and AUC = 0.514) and a three-class split (CA = 0.789 and AUC = 0.593). On the other hand, Openness was the most successfully inferred personality trait using a binary-class split (CA=0.861 and AUC = 0.492). This was also the highest CA over all the splits. In all three splits, LR gave the highest CA.

Considering that inferring Openness using the binary-class split using LR gave the highest overall accuracy both from rating-based recommendations and random generations, it is worth examining the confusion matrix. This confusion matrix for the scenario of inferring from rating-based recommendations is shown in Figure 6.4. The figure shows that all instances were predicted as the majority class. The same can be seen in the confusion matrix in Figure 6.5 for inferring Agreeableness from rating-based recommendations with the three-class split. Recall Agreeableness was the only attribute where the accuracy decreased from three-class to

binary-class-split. As seen in the distributions of the binary-class split for Openness in Figure 6.3 and Agreeableness in the three-class split in Figure 6.2, both personality traits had skewed distributions and a clear majority class: High for Openness in the binary-class split, and Medium for Agreeableness in the three-class split.

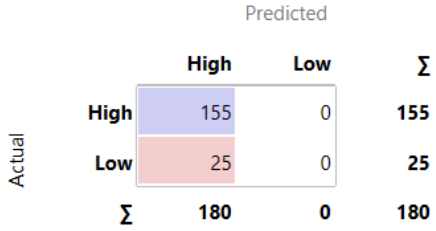


Figure 6.4: Confusion matrix for inference using Logistic Regression for Openness with the binary-class split on rating-based recommendations (CA=0.861 and AUC = 0.492).

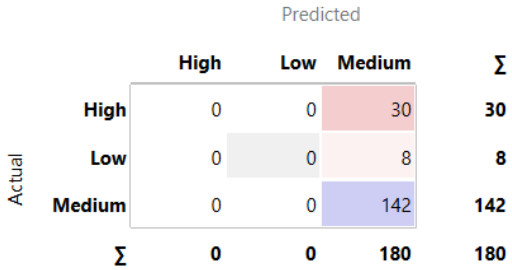


Figure 6.5: Confusion matrix for inference using Logistic Regression for Agreeableness with the three-class split on rating-based recommendations (CA = 0.789 and AUC = 0.593).

6.3 Experiment 2: Inference of User Personality Traits from Personality-based Recommendations

For Experiment 2, we implemented a Personality-based Recommender System (PBRS) and used it as the input for the inference model to compare with the inference using Rating-based Recommender System (RBRS) in Experiment 1. We hypothesized that using a PBRS would consequently lead to more information leakage of personality traits, as the recommendations are generated using personality data.

Looking at the binary-class split for the inference, Table 6.10 compares the average values for Classification Accuracy (CA) and Area Under Curve (AUC) for each classifier using a RBRS and a PBRS. Repeating how calculations were done in Table 6.6, the value was calculated by averaging the result of the classifiers for

Table 6.10: Average CA and AUC for each classifier across OCEAN traits, comparing RBRS and PBRS. Best Classification Accuracy (CA) results for each recommendations are marked in bold.

	RBRS		PBRS	
	Avg CA	Avg AUC	Avg CA	Avg AUC
kNN	0.655	0.509	0.652	0.548
LR	**0.683**	0.552	**0.689**	0.528
NB	0.681	0.528	0.654	0.528
NN	0.663	0.522	0.672	0.529
RF	0.668	0.518	0.658	0.522
SVM	0.502	0.523	0.568	0.483

each OCEAN trait. For both the RBRS and the PBRS, Logistic Regression (LR) was the best classifier when averaging the accuracy for all personality traits. The inference with PBRS performed slightly better with an average CA of 0.689, compared to 0.683 using RBRS. However, the PBRS had an average AUC score of 0.528, whereas the RBRS had an average AUC score of 0.552.

Table 6.11: Best classifier per trait, RBRS versus PBRS. The best Classification Accuracy (CA) results for each trait, comparing inference from rating-based and personality-based recommendations, are indicated by bold.

	RBRS		PBRS	
	Best classifier	CA	Best classifier	CA
Openness	**LR**	**0.861**	**LR**	**0.861**
Conscientiousness	**NB**	**0.606**	LR	0.583
Extraversion	**NB**	**0.728**	LR	0.706
Agreeableness	**kNN**	**0.583**	**LR/NN**	**0.583**
Neuroticism	LR	0.694	**LR**	**0.711**

Table 6.11 illustrates the difference between RBRS and PBRS when comparing the most accurate classifiers for each trait. Openness was the most accurately inferred personality trait for both recommendation inputs, with both systems achieving an accuracy of 0.861. The confusion matrix looked the same for both classifiers, with all instances being predicted as majority class, as previously shown in Figure 6.4. RBRS got a higher classification result than PBRS for Conscientiousness and Extraversion. For Agreeableness, both the systems achieved the same result of 0.583 CA. However, regarding Neuroticism, PBRS obtained the highest accuracy.

6.4 Experiment 3: Resampling

As introduced in Section 5.1, we resampled the training datasets to balance the dataset used in the classification.

Table 6.12: Openness - The table shows the difference in CA and AUC for inferring based on personality-based recommendations with no resampling, undersampling and oversampling. The best result for each classifier is marked in bold.

	No resampling		Undersampling		Oversampling	
	CA	AUC	CA	AUC	CA	AUC
kNN	0.833	**0.575**	-0.283	-0.055	-0.166	-0.012
LR	0.861	0.463	-0.394	**+0.010**	-0.333	+0.006
NB	0.844	0.535	-0.277	**+0.034**	-0.266	-0.005
NN	0.850	0.511	-0.283	+0.050	-0.078	**+0.075**
RF	0.850	**0.559**	-0.294	-0.003	-0.028	-0.015
SVM	0.783	0.433	-0.433	-0.031	-0.261	**+0.005**

Table 6.13: Conscientiousness - The table shows the difference in CA and AUC for inferring based on personality-based recommendations with no resampling, undersampling and oversampling. The best result for each classifier is marked in bold.

	No resampling		Undersampling		Oversampling	
	CA	AUC	CA	AUC	CA	AUC
kNN	0.561	**0.525**	-0.039	-0.017	-0.061	-0.078
LR	0.583	**0.506**	-0.094	-0.024	-0.100	-0.001
NB	0.539	0.395	+0.000	+0.142	-0.022	**+0.143**
NN	0.578	**0.528**	-0.106	-0.031	-0.084	-0.031
RF	0.572	0.476	-0.061	**+0.084**	-0.050	+0.028
SVM	0.444	0.436	+0.106	**+0.112**	+0.112	+0.059

Table 6.14: Extraversion - The table shows the difference in CA and AUC for inferring based on personality-based recommendations with no resampling, undersampling and oversampling. The best result for each classifier is marked in bold.

	No resampling		Undersampling		Oversampling	
	CA	AUC	CA	AUC	CA	AUC
kNN	0.661	**0.594**	-0.205	-0.132	-0.194	-0.131
LR	0.706	0.643	-0.134	-0.071	-0.112	**+0.008**
NB	0.694	0.644	-0.166	-0.116	-0.144	**+0.017**
NN	0.678	**0.557**	-0.206	-0.085	-0.128	-0.027
RF	0.672	**0.543**	-0.233	-0.104	-0.078	-0.048
SVM	0.583	**0.517**	-0.183	-0.117	-0.100	-0.071

Table 6.15: Agreeableness - The table shows the difference in CA and AUC for inferring based on personality-based recommendations with no resampling, undersampling and oversampling. The best result for each classifier is marked in bold.

	No resampling		Undersampling		Oversampling	
	CA	AUC	CA	AUC	CA	AUC
kNN	0.578	0.564	-0.056	-0.072	-0.039	**+0.002**
LR	0.583	0.516	-0.066	**+0.030**	-0.000	-0.058
NB	0.561	0.533	-0.011	**+0.030**	-0.039	+0.014
NN	0.583	**0.567**	-0.066	-0.046	-0.027	-0.009
RF	0.528	0.511	-0.050	+0.024	+0.033	**+0.038**
SVM	0.511	0.481	+0.078	+0.013	-0.022	**+0.019**

Table 6.16: Neuroticism - The table shows the difference in CA and AUC for inferring based on personality-based recommendations with no resampling, undersampling and oversampling. The best result for each classifier is marked in bold.

	No resampling		Undersampling		Oversampling	
	CA	AUC	CA	AUC	CA	AUC
kNN	0.628	0.483	-0.189	-0.009	-0.106	**+0.061**
LR	0.711	0.513	-0.128	+0.115	-0.111	**+0.131**
NB	0.633	0.532	-0.100	+0.047	-0.039	**+0.076**
NN	0.672	0.483	-0.189	+0.007	-0.050	**+0.070**
RF	0.662	0.521	-0.173	+0.032	+0.044	**+0.033**
SVM	0.517	0.548	-0.011	**+0.028**	+0.027	-0.036

Based on the provided corresponding tables, we can draw the following conclusions comparing the resampling techniques to no resampling.

The results for Openness are shown in Table 6.12. Undersampling had a negative impact on Classification Accuracy for all classifiers. However, half of the classifiers, specifically LR, Naive Bayes, and Neural Network, experienced an increase in Area Under Curve (AUC) despite this. On the other hand, oversampling also negatively impacted Classification Accuracy, but slightly less than undersampling. Also here, half of the classifiers, which here was LR, Neural Network, and SVM, experienced an increase in Area Under Curve (AUC). In general, oversampling performed better than undersampling.

Table 6.13 presents the results of Conscientiousness. Undersampling had a mixed impact on CA and AUC across different classifiers. Naive Bayes, Random Forest, and SVM showed an improvement for AUC, while CA also increased for SVM, it decreased for RF. k-Nearest Neighbors, LR, and Neural Network, on the other hand, showed a decrease in both CA and AUC. Oversampling also had a mixed impact on AUC, but it was less consistent between the two metrics. Only SVM show an increase for both CA and AUC; all other classifiers negatively impact CA. Other than that, only Naive Bayes and Random Forest show increased values for AUC.

In Table 6.14, the results for Extraversion are provided. Undersampling indicated a decrease in CA and AUC for all classifiers. This applied to oversampling as well, except for Logistic Regression and Naive Bayes (NB), which showed a slight improvement in AUC.

Table 6.15 shows the results for Agreeableness. Undersampling had a negative impact on CA for all classifiers except for SVM, which showed an increase. Contrary, AUC experienced an increase for all classifiers except kNN and Neural Network. Similar results could be found for oversampling. All classifiers showed a decrease in CA with the exception of Random Forest. The only classifiers that did not show an increased AUC were LR and Neural Network.

Lastly, the results for Neuroticism are illustrated in Table 6.16. Undersampling generally led to a decrease in CA for all classifiers. In contrast, the AUC increased for all values except kNN. Oversampling performed better where the CA increased for Random Forest and SVM and decreased for the rest. Further, the AUC generally increased more than for undersampling, except for SVM, which decreased. Logistic Regression was the classifier with the highest CA before any resampling and showed the highest increase for oversampling.

Chapter 7

Discussion

This chapter will thoroughly discuss the results related to the research questions presented in Section 1.2. Through our experiment, a novel contribution has been made by exploring the inference of personality traits based on the top n movie recommendations. We will thoroughly explore and examine the extent to which it is possible to infer personality traits from these movie recommendations (RQ1), considering factors such as the number of classes for each personality trait and its impact on the accuracy of inference (RQ1.1). Additionally, we will study the influence of incorporating personality traits into the recommender system and how it affects the ability to infer personality from the recommendations (RQ2). Furthermore, we will examine the effects of resampling techniques on key evaluation metrics such as Classification Accuracy (CA) and Area Under the Curve (AUC) in inferring users' personality traits from movie recommendations (RQ3). Moreover, we will discuss potential threats to validity and identify key points for future work to further our understanding of personality inference in recommender systems.

7.1 RQ1: To What Extent Is It Possible To Infer Users' Personality Traits From Movie Recommendations?

After analyzing the average classification accuracy across the OCEAN traits of various classifiers, it was discovered that the accuracy of predicting personality traits from top n recommendations varies depending on the classifier used. As presented in Table 6.10, logistic regression emerged as the overall best classifier, however, it performed better using the baseline recommendations that do not have any information about the users as they were randomly generated. Still, the results indicate that kNN and Neural Networks perform better on the rating-based recommendations than the baseline recommendations across all splits, indicating that these classifiers effectively capture the underlying patterns in the data.

For the remaining classifiers, it is uncertain whether the rating-based recommendations perform better than the baseline in terms of average CA across the OCEAN attributes. This varies between different personality splits, and it is unclear which recommendations provide the highest average accuracy for classification. This suggests that the choice of classification approach and the number of classes the personality traits are separated in can influence inference accuracy. The latter will be discussed further in RQ1.1.

When analyzing the best results for each OCEAN trait instead of comparing classifiers, we find that although no single trait consistently outperformed the baseline recommendations, averaging the highest CA values for each trait showed slightly higher CA for rating-based recommendations for all splits. These results are shown in Appendix B. The traits that saw the highest increase in CA using rating-based recommendations were Extraversion and Openness. The results show that it is not always the case, but that for all attributes, there are some cases where the rating-based recommendations do leak some information about personality.

Top n recommendations generated by Rating-based Recommender System can potentially reveal personal information, which could put user privacy at risk. Developers of recommender systems should be aware that even with limited accuracy, inferences can still expose user personality and potentially violate user privacy. It is essential for developers to inform users about this possibility and take steps to protect their data. Robust security measures should be implemented to prevent unauthorized access or data breaches of sensitive information, not only of the user ratings but also the generated recommendations.

Overall, Experiment 1 indicates that the choice of classifiers influences the effectiveness of inferring personality traits from top n movie recommendations and that it is not universally applicable to all traits. However, it is evident that rating-based recommendations do reveal some information about personality across all attributes in some cases and for some classifiers. Therefore, while there is no definitive rule for inferring certain traits, the results suggest that the use of rating-based recommendations can still provide insights into individuals' personality character-

istics, making it possible to infer users' personality traits from movie recommendations in certain cases.

7.1.1 RQ1.1: How Does the Number of Classes for Each Personality Trait Impact the Inference Accuracy?

RQ1.1 examines the impact of different personality trait splits on inferring personality traits' accuracy. We analyze Experiment 1 findings in the inference results from rating-based recommendations, including inference results from randomly generated recommendations as a baseline. This section explores how the choice of personality trait splits influences classification accuracy when inferring personality.

For all but one personality trait, we consistently observe that accuracy increases as the number of classes decreases. This pattern holds across both rating-based and randomly generated recommendations as shown in Table 6.6. Intuitively, with fewer classes, there is a higher likelihood of randomly placing the individuals' personality traits into the correct category. This indicates that optimal accuracy is achieved with splits having fewer classes, as they simplify the categorization process. The accuracy for the three personality trait splits would statistically fall around the inverse of the number of classes when classifying at random. In our experiment, we get higher values than those at random. If this only occurred with the rating-based recommendations, it could prove information leakage from the input. However, since the observed phenomena hold for the non-personalized random recommendations as well, it suggests the presence of additional factors influencing the classification accuracies, such as skewed datasets or other effects.

The exception for the results showing higher accuracy with fewer classes arises for the trait of Agreeableness; the accuracy decreases for the binary-class split compared to the three-class split. This unexpected finding may be attributed to Agreeableness being the most heavily skewed trait in the three-class split and the least skewed in the binary-class split regarding the coefficient of variation as presented in Table 6.5. The skewed distribution of Agreeableness in the three-class split indicates how it is important to acknowledge that the presence of trait imbalance can disrupt established accuracy trends, emphasizing the need for careful handling of imbalanced traits. We also observed that heavily imbalanced personality trait splits tend to exhibit higher classification accuracy, indicating a potential bias towards the majority class, a common occurrence in imbalanced datasets. This suspicion is further supported by the confusion matrices. In Experiment 3, strategies are explored to mitigate the impact of trait imbalance on classification accuracy.

Despite the overall best accuracy being achieved with the binary-class split, it is essential to consider the potential oversimplification of user personalities when using fewer classes. If the inferred personality becomes too simple, it may not be a useful attribute for practical applications. For other attributes like gender, which is

commonly perceived as having two distinct categories, an inference attack aiming to identify the binary classes may be reasonable in certain contexts. However, when attempting to infer attributes that normally come in multiple classes, like age, inferring only two classes would likely provide limited information making it practically useless if able to infer. In potential real-life scenarios where someone wishes to infer personality, such as inferring personality from a movie-watching site, overly simplified personality classes may not yield precise and helpful results for the attacker to use in their own domain.

We examined how the accuracy of personality trait inference is affected by using different splits for personality traits. The results indicate that personality trait splits with fewer classes tend to exhibit higher classification accuracy across both rating-based and randomly generated recommendations. This suggests that the choice of personality trait splits does impact the accuracy of inferring personality traits. However, an exception is observed for the trait of Agreeableness, where results point to differences of imbalances also affecting the accuracy. Further, it is crucial to balance simplicity and accuracy in practical applications. It is worth noting that the extent to which the binary-class split simplifies the complexity of personality has not been explored in this thesis. Further research should investigate alternative personality trait splits that effectively balance simplicity and accuracy, ensuring practical usefulness and maintaining the necessary level of complexity.

7.2 RQ2: How Does Incorporating Personality Traits Into a Recommender System Influence the Ability To Infer Personality From the Recommendations?

The goal with RQ2 was to explore how the accuracy of inference of personality traits would be influenced by incorporating personality into the recommender system we used as input for our inference model. We hypothesized that if personalized recommendations incorporate a user's personality as an additional factor, inferring personality traits would be more accurate and feasible. Contrary to the initial hypothesis, the results from Experiment 2 demonstrated no significant improvement in inference accuracy and, in some cases, even performed slightly worse using the personality-based recommendations. A number of different factors could explain these unexpected results.

In Lu and Kan [71]'s recommender system, they found an improvement in hit rate for users with high Conscientiousness, Extraversion, or Agreeableness when incorporating personality into the recommender system using the NCF+soft-label. Interestingly, our inference results reveal that inferring the aforementioned personality traits gave better results when utilizing rating-based recommendations without explicit personality information than with personality-based recommendations. This raises questions about why the accuracy decreases when personality information is included in the recommender system. The results can indicate that the incorporation of personality into the recommender system might not accurately represent the personality information or may interpret it in a manner that leads to overfitting. This potential overfitting occurs when the system relies excessively on the personality information, potentially overshadowing other relevant factors captured by the ratings.

Looking at Table 6.11, Neuroticism was the only trait that showed a slight improvement for Classification Accuracy (CA) with personality information, while Openness and Agreeableness achieved the same result for both recommendation inputs. On the other hand, Extraversion and Conscientiousness obtained better accuracy with the rating-based recommendations. Nevertheless, it is worth noting that most of the average AUC scores from the different classifiers increase from inferring using the RBRS to inference using the PBRS, as shown in Table 6.10. However, for Logistic Regression (LR), which was the most accurate classifier for both recommender systems, the AUC is higher for the RBRS. Therefore it is hard to find a cohesive pattern that says whether or not the inference has improved after using personality information in the recommender system. The inference values obtained from our experiment indicate that the results are inconclusive.

In our experiment, incorporating personality in a recommender system did not improve the classification accuracy for inferring the OCEAN personality traits. This indicates that the Personality-based Recommender System did not provide more information leakage regarding personality information than the Rating-based Rec-

ommender System. While the overall results imply no additional personality data leakage in a PBRS, it is important to acknowledge that certain issues identified in Experiment 1 may still persist in Experiment 2. Thus, it is essential to avoid prematurely asserting that it is impossible to infer personality with acceptable accuracy, given the potential inaccuracies in our initial setup for the inference process.

7.3 RQ3: How Does the Application of Resampling Affect the Classification Accuracy and AUC for Inferring Users' Personality Traits From Movie Recommendations?

The aim of the resampling in Experiment 3 was to balance out errors that came from the majority class bias in the classification accuracies identified in Experiment 1. In Figure 6.4 and Figure 6.5, we saw that the classifiers with the highest CA often classified all users as the majority class, meaning the high values for CA were sometimes only a representation of the percentage of majority samples. Experiment 3 resampled the train set to reach a balance between the majority and minority classes for each personality trait.

In most cases, as shown in Section 6.4 the classification accuracy decreases when undersampling or oversampling techniques are applied. However, there were some exceptions. Specifically, when undersampling was employed, Conscientiousness and Agreeableness exhibited improved CA when using SVM as the classifier. These two traits were the least skewed in the binary-class split.

Similarly, in the case of oversampling, CA increased for Conscientiousness and Agreeableness but also for Neuroticism. The classifiers demonstrating these results were SVM again for Conscientiousness and Neuroticism, but Random Forest for Agreeableness and Neuroticism. RF had the second lowest CA before resampling for Agreeableness. Before resampling, SVM had the lowest CA of the different classifiers among all OCEAN traits, and Random Forest the second lowest.

These results indicate that undersampling and oversampling both generally decrease CA unless the classifier already performs poorly on the dataset. Additionally, it suggests that traits with lower skewness respond better to a 1:1 ratio resampling than heavily skewed traits. One possible explanation for this could be that the resampled training data has a comparably more similar ratio to the test set and fewer alterations, minimizing duplications of the minority class during oversampling and removing important instances during undersampling. However, this hypothesis requires further investigation, potentially by attempting to resample based on the number of samples in each class rather than aiming for a specific ratio.

Even though the AUC often shows improvement, the impact of resampling techniques on AUC is inconsistent across different personality traits and classifiers, indicating that factors that vary between the traits may influence the effects of resampling, and no straightforward answer can be given. Before any resampling, the best performance in terms of AUC belonged to Extraversion, while Conscientiousness and then Agreeableness performed the worst. The traits that showed the biggest improvement in AUC were Neuroticism and then Agreeableness. Extraversion generally performs worse after resampling, while Openness and Con-

scientiousness are more inconclusive and dependent on the classifier; they are also the only traits with High as the majority class. Interestingly both Extraversion and Neuroticism were in the middle when ranking skewness in the binary-class split but are oppositely affected by resampling. This might indicate that resampling performs worse with a higher original AUC, and a lower original AUC may be positively affected by resampling but that there might not be a direct correlation between the level of imbalance in the original data.

Generally, oversampling shows a more significant improvement than undersampling across traits and classifiers for AUC. In almost all cases, the AUC score for oversampling is better than undersampling. This implies that there might be a loss of data during undersampling that negatively impacts the classification. For CA, oversampling also outperforms undersampling, but the accuracy for both resampling techniques is often lower than the accuracy of the no resampling baseline.

In summary, the Classification Accuracy (CA) generally decreases when employing undersampling or oversampling to reach a 1:1 ratio. However, the Area Under Curve (AUC) tends to increase for more than half of the tested classifiers across all attributes when using these resampling methods, except for in the case of Extraversion. The results suggest that oversampling may yield better outcomes than undersampling, as undersampling potentially loses crucial data. Nevertheless, as observed earlier, oversampling yielded improved results for traits that were originally less skewed, indicating that excessive oversampling could potentially disrupt the benefit of resampling. Future research could explore resampling methods that strike a balance by applying oversampling up to a certain threshold and undersampling to ensure the alterations of the trainset are not too significant.

7.4 Threats to Validity

Several threats may influence the validity of our findings. Firstly, the utilized dataset might have been too sparse, particularly due to imbalanced personality traits and the limited number of samples in the minority class for the traits. To investigate this possibility, we could have employed multiple datasets, such as the Amazon-beauty and the Amazon-music datasets used by Lu and Kan [71]. However, privacy restrictions prevented us from accessing these alternative datasets, and our search for other datasets containing both ratings and personality information proved unsuccessful. Exploring diverse domains beyond movies could have helped validate the generalizability of our findings.

Although we obtained our dataset from a reliable source, there is a potential for sample bias. It is plausible that individuals with specific personality traits were more inclined to provide movie ratings or even engage in higher movie consumption, leading to the observed skewness in the attributes. However, it is worth noting that certain personality traits are naturally more prevalent in the real world. Therefore, the question remains whether the skewness observed in the Personality2018 dataset accurately reflects real-world distributions. While we have not delved into this aspect, we have deferred the responsibility to the original creators of the dataset.

Due to the dataset's imbalance, many classifiers were biased towards the majority class, sometimes even misclassifying all users as belonging to the majority class. Despite employing stratification during the sampling of training and test data, the ratio between the majority and minority classes fluctuated slightly. As a result, the classification accuracy also exhibited slight variations each time the inference was performed. However, after conducting multiple runs to cross-validate, we found that the changes in CA between each run were low. By ensuring consistent train-test data partitions across the same input, we could compare the results across OCEAN attributes for the same recommendations and personality trait split. This minimized the impact of these fluctuations on the overall analysis.

Given the inconsistent outcome of Experiment 2 involving the Personality-based Recommender System (PBRS), it is necessary to examine the performance of the recommender system. In our study, we implemented the NCF personality-based recommender system created by Lu and Kan [71]. Although their research reported improved performance for the recommender system by incorporating personality, the impact of this incorporated personality might not be as significant as necessary for our purpose. It is possible that utilizing an alternative personality-based recommender system that incorporates personality in a different manner could have yielded more accurate inferences. Further, the recommender system employed by Lu et al. utilized personality data from the Personality2018 dataset, which relies on the TIPI questionnaires for personality assessment. However, this assessment comprises only ten questions, which may result in a generalized and potentially inaccurate representation of a user's true personality.

Lastly, it is possible that our resampling methods were oversimplified, which could have limited their impact on classification accuracy. By employing random over/undersampling and simplifying it to achieve a 1:1 ratio, we might have encountered issues with the most skewed traits. It is plausible that these traits experienced a large number of duplicated instances or significant loss of data during the training phase. This is further discussed in Section 7.6.

7.5 Practical Implications

Our findings emphasize the practical implications of the importance of developing privacy-aware recommender systems in general, as some of our results indicated information leakage. Lu and Kan [71] showed that incorporating personality improved their recommender system. However, when comparing the inference of personality traits from rating-based recommendations and personality-based recommendations in our experiments, there were no significant differences in accuracy between the two. This can imply that personality-based recommender systems are not inherently riskier or more vulnerable to personality leakage than traditional recommender systems while still providing improved performance. In practical terms, this can mean that incorporating personality into recommender systems does not inherently present heightened ethical concerns regarding privacy. Nonetheless, it is important to be privacy aware when developing recommender systems, as the experiments revealed potential information leakage. By focusing on privacy awareness, developers can effectively find a balance between providing personalized recommendations and respecting user privacy, as well as protecting against data inference.

7.6 Future Work

To enhance the accuracy and reliability of personality inference, other feature representations beyond counting the occurrences of each genre in the top 10 recommendations may be considered. Future experiments can explore alternative approaches and assess the representativeness of genres in inferring personality. This could involve various genre-counting methods or even exploring alternative ways to represent the recommendations beyond genre-counting. It is important to consider whether the chosen representation captures sufficient information to accurately infer personality traits. Additionally, it may be valuable to examine the feature selection and whether certain genres should be excluded from the analysis, as certain genres could be more strongly correlated with specific personality traits. Sometimes, irrelevant features can negatively impact the performance of a model. Table 3.1 can serve as a starting point to identify potential correlations between genres and personality traits. By refining the representation of recommendations, we can gain a deeper understanding of the relationship between recommendations and personality, ultimately improving the accuracy of personality

inference.

In our study, the classifiers were used in their original form without hyperparameter tuning. However, it is worth noting that the choice of hyperparameters heavily influences the performance of classifiers. Therefore, future research can explore the impact of different hyperparameters, such as regularization strength or max iterations, to identify the optimal set of hyperparameters that maximize the Classification Accuracy of personality inference. By fine-tuning the hyperparameters, it is possible to enhance the performance of the classifiers and potentially achieve better results in inferring personality traits compared to our experiments.

As previously mentioned, another Personality-based Recommender System (PBRS) that utilizes personality traits differently than the PBRS created by Lu and Kan [71] might reflect personality more accurately. For example, using a PBRS trained on more comprehensive personality information, like the more extensive Big Five inventory instead of the Ten Item Personality inventory, may yield more accurate personality data into the recommender system. Alternatively, experimenting with a PBRS based on an entirely different algorithm than the NCF could also be beneficial. By implementing a recommender system that places greater emphasis on personality, it is possible that the potential information leakage associated with personality traits could have been more evident.

In future research, there is a potential for further exploration of resampling techniques beyond the simple strategies explored in this thesis. Our findings indicated that oversampling yielded better results compared to undersampling, but undersampling also showed promise. The extent of oversampling impacted the observed improvements, suggesting the need for methods that strike a balance by applying oversampling up to a certain threshold and undersampling to prevent significant alterations to the trainset. Additionally, exploring alternative ratios beyond the 1:1 ratio could provide valuable insights. If further research on resampling techniques does not yield increased classification accuracy, ensemble methods can be considered. Ensemble methods can potentially enhance the performance of a single classifier by combining multiple base classifiers that outperform any independent classifier.

Chapter 8

Conclusion

This thesis aimed to study how accurately one can infer users' personality traits from their top n movie recommendations and examine the impact of different personality trait splits on accuracy. The study also investigated how integrating personality traits into a recommender system influences the inference from the movie recommendations. Furthermore, the research explored how resampling techniques affect the classification accuracy and AUC of personality inference from the same recommendations.

Further, RQ1.1 looked at how different personality trait splits affected the accuracy of predicting those traits. The results showed that accuracy improved when the number of classes was lower, indicating that simpler splits made more accurate predictions. This was true whether the predictions were based on ratings or randomly generated suggestions, suggesting that more than just the recommendations affect accuracy. However, one trait, Agreeableness, had lower accuracy when split into only two categories than when divided into three, showing that the balance of traits can affect accuracy. The study emphasized the importance of handling imbalanced traits carefully and weighing the trade-off between simplicity and accuracy in practical applications. Future research should explore alternative personality trait splits that balance simplicity and accuracy while still being complex enough to make the inferred personality useful.

RQ2 looked at whether adding personality traits to a recommender system would make it better at accurately inferring personality. However, the findings showed that including personality information did not make a significant improvement in accuracy. At times, the accuracy even decreased when personality information was added. Although these results suggest that information leakage does not increase when incorporating personality, one might be wary that other factors contributed to the results. For example, the problems identified in the earlier RQs may still persist, or the incorporated personality information in the recommender system may not accurately reflect the user's personality or overshadow other important

factors captured by the ratings.

Lastly, RQ3 looked at how resampling techniques affect the accuracy of personality inference from movie recommendations. Our observations showed that using techniques like undersampling and oversampling generally led to decreased classification accuracy, though some exceptions existed. The AUC, on the other hand, increased in half of the cases. Traits with less skewness were more responsive to resampling, while heavily skewed traits showed less consistent effects. Oversampling tended to produce better outcomes than undersampling, but excessive oversampling seemed to interfere with the benefits of resampling.

In conclusion, this thesis examines the possibility of inferring users' personality traits based on the top 10 movie recommendations. The findings suggest that there may be some information leakage from personalized recommendations about one's personality traits. Still, there is no definite or conclusive approach to making inferences based on them. The accuracy of personality inference depends on factors such as the classifier used, choice of personality trait splits, and trait imbalance. Including personality information to generate recommendations did not significantly improve the accuracy of inferring personality traits. The practical implications of these findings indicate that although incorporating personality traits in a recommender system leads to improved recommendations, it does not significantly compromise user privacy compared to traditional recommender systems. Lastly, the effectiveness of tested resampling techniques varied.

It must be noted that this research has certain limitations. The dataset used was small and skewed, which could affect how applicable the results are in other situations. It is also possible that the results were influenced by biases in the sample and imbalanced personality traits in the data. More research is necessary to confirm and expand upon these findings using more extensive and diverse datasets, preferably in other domains as well. Furthermore, investigating different personality-based recommender systems and classification methods that can handle imbalanced classes could yield more information about how accurately personality traits can be inferred from recommendations.

This study contributes to the understanding of privacy concerns and limitations associated with movie recommendations. It is important for developers to be aware of the potential privacy risks and to inform users about the possibility of personality inference. This thesis indicates that movie recommendations should be protected and safeguarded in the same way as other sensitive information, such as user ratings and user personality traits themselves.

Bibliography

[1] C. Ike, *Personalized Product Recommendations: What It Is, Benefits, and Best Practices*, en, Oct. 2020. [Online]. Available: https://adoric.com/blog/personalized-product-recommendations-what-it-is-benefits-and-best-practices/ (visited on 12/10/2022).

[2] C. Wang, Y. Zheng, J. Jiang, and K. Ren, "Toward Privacy-Preserving Personalized Recommendation Services," en, *Engineering*, vol. 4, no. 1, pp. 21–28, Feb. 2018, ISSN: 20958099. DOI: 10.1016/j.eng.2018.02.005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2095809917303855 (visited on 12/10/2022).

[3] A. J. P. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, and Q. Tang, "Privacy in Recommender Systems," in *Social Media Retrieval*, N. Ramzan, R. van Zwol, J.-S. Lee, K. Clüver, and X.-S. Hua, Eds., London: Springer London, 2013, pp. 263–281, ISBN: 978-1-4471-4555-4. DOI: 10.1007/978-1-4471-4555-4_12. [Online]. Available: https://doi.org/10.1007/978-1-4471-4555-4_12.

[4] E. Van de Garde-Perik, B. Ruyter, P. Markopoulos, and B. Eggen, *The Sensitivities of User Profile Information in Music Recommender Systems*. Jan. 2004, Journal Abbreviation: IEEE Journal of Solid-state Circuits - IEEE J SOLID-STATE CIRCUITS Pages: 141 Publication Title: IEEE Journal of Solid-state Circuits - IEEE J SOLID-STATE CIRCUITS. [Online]. Available: https://www.researchgate.net/publication/220919925_The_Sensitivities_of_User_Profile_Information_in_Music_Recommender_Systems.

[5] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "BlurMe: Inferring and Obfuscating User Gender Based on Ratings," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, ser. RecSys '12, event-place: Dublin, Ireland, New York, NY, USA: Association for Computing Machinery, 2012, pp. 195–202, ISBN: 978-1-4503-1270-7. DOI: 10.1145/2365952.2365989. [Online]. Available: https://doi.org/10.1145/2365952.2365989.

[6] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5802–5805, Apr. 2013, Publisher: Proceedings of the National Academy of Sciences. DOI: 10.1073/

pnas.1218772110. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.1218772110 (visited on 11/25/2022).

[7] S. Milano, M. Taddeo, and L. Floridi, "Recommender systems and their ethical challenges," *AI & SOCIETY*, vol. 35, no. 4, pp. 957–967, Dec. 2020, ISSN: 1435-5655. DOI: 10.1007/s00146-020-00950-y. [Online]. Available: https://doi.org/10.1007/s00146-020-00950-y.

[8] T. T. Nguyen, F. Maxwell Harper, L. Terveen, and J. A. Konstan, "User Personality and User Satisfaction with Recommender Systems," *Information Systems Frontiers*, vol. 20, no. 6, pp. 1173–1189, Dec. 2018, ISSN: 1572-9419. DOI: 10.1007/s10796-017-9782-y. [Online]. Available: https://doi.org/10.1007/s10796-017-9782-y.

[9] R. Hu and P. Pu, "A Study on User Perception of Personality-Based Recommender Systems," in *User Modeling, Adaptation, and Personalization*, P. De Bra, A. Kobsa, and D. Chin, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 291–302, ISBN: 978-3-642-13470-8. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-13470-8_27.

[10] I. Cantador, I. Fernandez-Tobias, A. Bellogín, M. Kosinski, and D. Stillwell, "Relating Personality Types with User Preferences Multiple Entertainment Domains," vol. 997, Jan. 2013.

[11] M. A. S. Nunes and R. Hu, "Personality-Based Recommender Systems: An Overview," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, ser. RecSys '12, event-place: Dublin, Ireland, New York, NY, USA: Association for Computing Machinery, 2012, pp. 5–6, ISBN: 978-1-4503-1270-7. DOI: 10.1145/2365952.2365957. [Online]. Available: https://doi.org/10.1145/2365952.2365957.

[12] G. Dunn, J. Wiersema, J. Ham, and L. Aroyo, "Evaluating Interface Variants on Personality Acquisition for Recommender Systems," en, in *User Modeling, Adaptation, and Personalization*, G.-J. Houben, G. McCalla, F. Pianesi, and M. Zancanaro, Eds., vol. 5535, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 259–270, ISBN: 978-3-642-02246-3 978-3-642-02247-0. DOI: 10.1007/978-3-642-02247-0_25. [Online]. Available: http://link.springer.com/10.1007/978-3-642-02247-0_25 (visited on 12/07/2022).

[13] V. Tiwari, A. Ashpilaya, P. Vedita, U. Daripa, and P. P. Paltani, "Exploring Demographics and Personality Traits in Recommendation System to Address Cold Start Problem," en, in *ICT Systems and Sustainability*, M. Tuba, S. Akashe, and A. Joshi, Eds., vol. 1077, Series Title: Advances in Intelligent Systems and Computing, Singapore: Springer Singapore, 2020, pp. 361–369, ISBN: 9789811509353 9789811509360. DOI: 10.1007/978-981-15-0936-0_37. [Online]. Available: http://link.springer.com/10.1007/978-981-15-0936-0_37 (visited on 12/07/2022).

[14] M. Martijn, C. Conati, and K. Verbert, ""Knowing me, knowing you": Personalized explanations for a music recommender system," *User Modeling and User-Adapted Interaction*, vol. 32, no. 1, pp. 215–252, Apr. 2022, ISSN: 1573-1391. DOI: 10.1007/s11257-021-09304-9. [Online]. Available: https://doi.org/10.1007/s11257-021-09304-9.

[15] W. Wu and L. Chen, "Implicit Acquisition of User Personality for Augmenting Movie Recommendations," in *User Modeling, Adaptation and Personalization*, F. Ricci, K. Bontcheva, O. Conlan, and S. Lawless, Eds., Cham: Springer International Publishing, 2015, pp. 302–314, ISBN: 978-3-319-20267-9. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-20267-9_25.

[16] J. Golbeck, C. Robles, and K. Turner, "Predicting personality with social media," en, in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*, Vancouver, BC, Canada: ACM Press, 2011, p. 253, ISBN: 978-1-4503-0268-5. DOI: 10.1145/1979742.1979614. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1979742.1979614 (visited on 12/07/2022).

[17] Y. Bachrach, M. Kosinski, T. Graepel, P. Kohli, and D. Stillwell, "Personality and patterns of Facebook usage," en, in *Proceedings of the 3rd Annual ACM Web Science Conference on - WebSci '12*, Evanston, Illinois: ACM Press, 2012, pp. 24–32, ISBN: 978-1-4503-1228-8. DOI: 10.1145/2380718.2380722. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2380718.2380722 (visited on 12/07/2022).

[18] D. Quercia, M. Kosinski, D. Stillwell, and J. Crowcroft, "Our Twitter Profiles, Our Selves: Predicting Personality with Twitter," in *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*, Boston, MA, USA: IEEE, Oct. 2011, pp. 180–185, ISBN: 978-1-4577-1931-8 978-0-7695-4578-3. DOI: 10.1109/PASSAT/SocialCom.2011.26. [Online]. Available: http://ieeexplore.ieee.org/document/6113111/ (visited on 12/07/2022).

[19] R. Gao, B. Hao, S. Bai, L. Li, A. Li, and T. Zhu, "Improving user profile with personality traits predicted from social media content," en, in *Proceedings of the 7th ACM conference on Recommender systems*, Hong Kong China: ACM, Oct. 2013, pp. 355–358, ISBN: 978-1-4503-2409-0. DOI: 10.1145/2507157.2507219. [Online]. Available: https://dl.acm.org/doi/10.1145/2507157.2507219 (visited on 12/07/2022).

[20] W. Youyou, M. Kosinski, and D. Stillwell, "Computer-based personality judgments are more accurate than those made by humans," en, *Proceedings of the National Academy of Sciences*, vol. 112, no. 4, pp. 1036–1040, Jan. 2015, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1418680112. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.1418680112 (visited on 12/07/2022).

[21] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," en, *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, Jan. 2017, Publisher: IOS Press, ISSN: 1088-467X. DOI: 10.3233/IDA-163209. [Online]. Available: https://content.iospress.com/articles/intelligent-data-analysis/ida163209 (visited on 12/04/2022).

[22] C. C. Aggarwal, *Recommender Systems*, en. Cham: Springer International Publishing, 2016, ISBN: 978-3-319-29657-9 978-3-319-29659-3. DOI: 10.1007/978-3-319-29659-3. [Online]. Available: http://link.springer.com/10.1007/978-3-319-29659-3 (visited on 12/04/2022).

[23] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," en, *Physics Reports*, vol. 519, no. 1, pp. 1–49, Oct. 2012, ISSN: 03701573. DOI: 10.1016/j.physrep.2012.02.006. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0370157312000828 (visited on 05/03/2023).

[24] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, en. New York, NY: Springer US, 2022, ISBN: 978-1-07-162196-7 978-1-07-162197-4. DOI: 10.1007/978-1-0716-2197-4. [Online]. Available: https://link.springer.com/10.1007/978-1-0716-2197-4 (visited on 10/17/2022).

[25] J. Lu, Q. Zhang, and G. Zhang, *Recommender Systems: Advanced Developments* (Intelligent Information Systems), en. WORLD SCIENTIFIC, Sep. 2020, vol. 06, ISBN: 9789811224621 9789811224638. DOI: 10.1142/11947. [Online]. Available: https://www.worldscientific.com/worldscibooks/10.1142/11947 (visited on 12/04/2022).

[26] I. Cantador, I. Fernández-Tobías, S. Berkovsky, and P. Cremonesi, "Cross-Domain Recommender Systems," en, in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., Boston, MA: Springer US, 2015, pp. 919–959, ISBN: 978-1-4899-7636-9 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_27. [Online]. Available: http://link.springer.com/10.1007/978-1-4899-7637-6_27 (visited on 12/11/2022).

[27] M. F. Dacrema, I. Cantador, I. Fernández-Tobías, S. Berkovsky, and P. Cremonesi, "Design and Evaluation of Cross-Domain Recommender Systems," en, in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., New York, NY: Springer US, 2022, pp. 485–516, ISBN: 978-1-07-162196-7 978-1-07-162197-4. DOI: 10.1007/978-1-0716-2197-4_13. [Online]. Available: https://link.springer.com/10.1007/978-1-0716-2197-4_13 (visited on 12/11/2022).

[28] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," en, in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., vol. 4321, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 325–341, ISBN: 978-3-540-72078-2. DOI: 10.

1007/978-3-540-72079-9_10. [Online]. Available: http://link.springer.com/10.1007/978-3-540-72079-9_10 (visited on 12/04/2022).

[29] Valentina, *Introduction to recommender systems*, en-US, Jan. 2021. [Online]. Available: https://thingsolver.com/introduction-to-recommender-systems/ (visited on 12/04/2022).

[30] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," en, in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., vol. 4321, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 291–324, ISBN: 978-3-540-72078-2. DOI: 10.1007/978-3-540-72079-9_9. [Online]. Available: http://link.springer.com/10.1007/978-3-540-72079-9_9 (visited on 12/04/2022).

[31] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, ser. UAI'98, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jul. 1998, pp. 43–52, ISBN: 978-1-55860-555-8. (visited on 05/04/2023).

[32] M. Tkalčič and L. Chen, "Personality and Recommender Systems," en, in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., New York, NY: Springer US, 2022, pp. 757–787, ISBN: 978-1-07-162196-7 978-1-07-162197-4. DOI: 10.1007/978-1-0716-2197-4_20. [Online]. Available: https://link.springer.com/10.1007/978-1-0716-2197-4_20 (visited on 12/11/2022).

[33] X. Amatriain, A. Jaimes*, N. Oliver, and J. M. Pujol, "Data Mining Methods for Recommender Systems," en, in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Boston, MA: Springer US, 2011, pp. 39–71, ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_2. [Online]. Available: https://doi.org/10.1007/978-0-387-85820-3_2 (visited on 06/08/2023).

[34] A. Sharma, *Neural Collaborative Filtering*, en, Dec. 2019. [Online]. Available: https://towardsdatascience.com/neural-collaborative-filtering-96cef1009401 (visited on 06/08/2023).

[35] United Nations, *Universal Declaration of Human Rights*. Dec. 1948.

[36] *What is GDPR, the EU's new data protection law?* en-US, Section: GDPR Overview, Nov. 2018. [Online]. Available: https://gdpr.eu/what-is-gdpr/ (visited on 12/03/2022).

[37] *What personal data is considered sensitive?* en, Text. [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/legal-grounds-processing-data/sensitive-data/what-personal-data-considered-sensitive_en (visited on 12/07/2022).

[38] Datatilsynet, *The anonymisation of personal data*, en. [Online]. Available: https://www.datatilsynet.no/en/regulations-and-tools/reports-on-specific-subjects/anonymisation/ (visited on 12/07/2022).

[39] United Nations Development Group, *UNSDG | Data Privacy, Ethics and Protection: Guidance Note on Big Data for Achievement of the 2030 Agenda*, en. [Online]. Available: https://unsdg.un.org/resources/data-privacy-ethics-and-protection-guidance-note-big-data-achievement-2030-agenda,%20https://unsdg.un.org/resources/data-privacy-ethics-and-protection-guidance-note-big-data-achievement-2030-agenda (visited on 12/03/2022).

[40] I. Wigmore, *What is implicit data? | Definition from TechTarget*, Dec. 2012. [Online]. Available: https://www.techtarget.com/whatis/definition/implicit-data (visited on 12/07/2022).

[41] Australian Cyber Security Centre, *Cyber adversary | Cyber.gov.au*. [Online]. Available: https://www.cyber.gov.au/acsc/view-all-content/glossary/cyber-adversary (visited on 12/07/2022).

[42] *Inference noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com*. [Online]. Available: https://www.oxfordlearnersdictionaries.com/definition/english/inference?q=inference (visited on 12/12/2022).

[43] *Understanding Machine Learning Inference*. [Online]. Available: https://www.run.ai/guides/machine-learning-inference/understanding-machine-learning-inference (visited on 12/12/2022).

[44] B. Dickson, *Inference attacks: How much information can machine learning models leak?* en, Apr. 2021. [Online]. Available: https://portswigger.net/daily-swig/inference-attacks-how-much-information-can-machine-learning-models-leak (visited on 12/07/2022).

[45] A. Friedman, B. Knijnenburg, K. Vanhecke, L. Martens, and S. Berkovsky, "Privacy Aspects of Recommender Systems," in Jan. 2015, pp. 649–688, ISBN: 978-1-4899-7636-9. DOI: 10.1007/978-1-4899-7637-6_19. [Online]. Available: https://www.researchgate.net/publication/294287414_Privacy_Aspects_of_Recommender_Systems.

[46] T. G. Mesevage, *Machine Learning Classifiers - The Algorithms & How They Work*, en, Section: Machine Learning, Dec. 2020. [Online]. Available: https://monkeylearn.com/blog/what-is-a-classifier/ (visited on 12/04/2022).

[47] S. Singh, *Understanding the Bias-Variance Tradeoff*, en, Oct. 2018. [Online]. Available: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229 (visited on 05/03/2023).

[48] *What is the k-nearest neighbors algorithm? | IBM*. [Online]. Available: https://www.ibm.com/topics/knn (visited on 05/30/2023).

[49] N. Tyagi, *L2 vs L1 Regularization in Machine Learning | Ridge and Lasso Regularization*, en. [Online]. Available: https://www.analyticssteps.com/blogs/l2-and-l1-regularization-machine-learning (visited on 05/03/2023).

[50] B. Stecanella, *Support Vector Machines (SVM) Algorithm Explained*, en, Section: Machine Learning, Jun. 2017. [Online]. Available: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/ (visited on 12/05/2022).

[51] N. Kumar, *Naive Bayes Classifiers*, en-us, Section: Python, Mar. 2017. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/ (visited on 12/04/2022).

[52] P. Gupta, *Decision Trees in Machine Learning*, en, Nov. 2017. [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 (visited on 12/05/2022).

[53] *What are Neural Networks? | IBM*, en-us. [Online]. Available: https://www.ibm.com/topics/neural-networks (visited on 05/30/2023).

[54] O. Kisi, J. Shiri, and V. Demir, "Hydrological Time Series Forecasting Using Three Different Heuristic Regression Techniques," en, in *Handbook of Neural Computation*, Elsevier, 2017, pp. 45–65, ISBN: 978-0-12-811318-9. DOI: 10.1016/B978-0-12-811318-9.00003-X. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B978012811318900003X (visited on 12/12/2022).

[55] S. E. R, *Random Forest | Introduction to Random Forest Algorithm*, en, Jun. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (visited on 12/05/2022).

[56] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," en, *Expert Systems with Applications*, vol. 73, pp. 220–239, May 2017, ISSN: 09574174. DOI: 10.1016/j.eswa.2016.12.035. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417416307175 (visited on 05/08/2023).

[57] *Data Splitting*, en-us, May 2022. [Online]. Available: https://datexdatastealth.com/blog/data-splitting (visited on 05/07/2023).

[58] *3.1. Cross-validation: Evaluating estimator performance*, en. [Online]. Available: https://scikit-learn/stable/modules/cross_validation.html (visited on 05/07/2023).

[59] Walber, *English: Precision and recall*, Nov. 2014. [Online]. Available: https://commons.wikimedia.org/wiki/File:Precisionrecall.svg (visited on 05/05/2023).

[60] B. Wang, *Ranking Evaluation Metrics for Recommender Systems*, en, Jan. 2021. [Online]. Available: https://towardsdatascience.com/ranking-evaluation-metrics-for-recommender-systems-263d0a66ef54 (visited on 05/30/2023).

[61] P. Chokhra, *Evaluating Recommender Systems*, en, Apr. 2021. [Online]. Available: https://medium.com/nerd-for-tech/evaluating-recommender-systems-590a7b87afa5 (visited on 05/30/2023).

[62] S. Sen, *Evaluating Recommender Systems*, en, Aug. 2022. [Online]. Available: https://medium.com/the-owl/evaluating-recommender-systems-749570354976 (visited on 05/30/2023).

[63] *Classification: ROC Curve and AUC | Machine Learning*, en. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (visited on 05/30/2023).

[64] L. Zhou, "Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods," en, *Knowledge-Based Systems*, vol. 41, pp. 16–25, Mar. 2013, ISSN: 09507051. DOI: 10.1016/j.knosys.2012.12.007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095070511200353X (visited on 05/08/2023).

[65] X. Y. Chin, H. Y. Lau, and Z. X. Chong, "Personality prediction using machine learning classifiers," en, vol. 5, no. 1, 2020.

[66] R. R. McCrae and P. T. Costa, "Validation of the five-factor model of personality across instruments and observers.," en, *Journal of Personality and Social Psychology*, vol. 52, no. 1, pp. 81–90, 1987, ISSN: 1939-1315, 0022-3514. DOI: 10.1037/0022-3514.52.1.81. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/0022-3514.52.1.81 (visited on 12/06/2022).

[67] S. Srivastava and J. Oliver P, "The Big-Five trait taxonomy: History, measurement, and theoretical perspectives," in University of California Berkeley, 1999, pp. 102–138.

[68] R. R. McCrae and O. P. John, "An Introduction to the Five-Factor Model and Its Applications," *Journal of Personality*, vol. 60, no. 2, pp. 175–215, Jun. 1992, Publisher: Wiley. DOI: 10.1111/j.1467-6494.1992.tb00970.x. [Online]. Available: https://doi.org/10.1111%2Fj.1467-6494.1992.tb00970.x.

[69] O. P. John, R. W. Robins, and L. A. Pervin, Eds., *Handbook of personality: theory and research*, 3rd ed. New York: Guilford Press, 2008, OCLC: ocn192109750, ISBN: 978-1-59385-836-0 978-1-60918-059-1.

[70] S. D. Gosling, P. J. Rentfrow, and W. B. Swann, "A very brief measure of the Big-Five personality domains," en, *Journal of Research in Personality*, vol. 37, no. 6, pp. 504–528, Dec. 2003, ISSN: 0092-6566. DOI: 10.1016/S0092-6566(03)00046-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0092656603000461 (visited on 06/10/2023).

[71] X. Lu and M.-Y. Kan, "Improving Recommendation Systems with User Personality Inferred from Product Reviews," 2023, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2303.05039. [Online]. Available: https://arxiv.org/abs/2303.05039 (visited on 05/16/2023).

[72] A. Kobsa and J. Schreck, "Privacy through pseudonymity in user-adaptive systems," *ACM Transactions on Internet Technology*, vol. 3, no. 2, pp. 149–183, 2003, ISSN: 1533-5399. DOI: 10.1145/767193.767196. [Online]. Available: https://doi.org/10.1145/767193.767196 (visited on 11/25/2022).

[73] Y. S. Resheff, Y. Elazar, M. Shahar, and O. S. Shalom, "Privacy and Fairness in Recommender Systems via Adversarial Training of User Representations," 2018, Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.1807.03521. [Online]. Available: https://arxiv.org/abs/1807.03521 (visited on 12/03/2022).

[74] G. Kraaykamp and K. v. Eijck, "Personality, media preferences, and cultural participation," en, *Personality and Individual Differences*, vol. 38, no. 7, pp. 1675–1688, May 2005, ISSN: 01918869. DOI: 10.1016/j.paid.2004.11.002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0191886904003496 (visited on 12/06/2022).

[75] R. P. Karumur, T. T. Nguyen, and J. A. Konstan, "Personality, User Preferences and Behavior in Recommender systems," *Information Systems Frontiers*, vol. 20, no. 6, pp. 1241–1265, Dec. 2018, ISSN: 1572-9419. DOI: 10.1007/s10796-017-9800-0. [Online]. Available: https://doi.org/10.1007/s10796-017-9800-0.

[76] E. M. Khan, M. S. H. Mukta, M. E. Ali, and J. Mahmud, "Predicting Users' Movie Preference and Rating Behavior from Personality and Values," *ACM Transactions on Interactive Intelligent Systems*, vol. 10, no. 3, 22:1–22:25, 2020, ISSN: 2160-6455. DOI: 10.1145/3338244. [Online]. Available: https://doi.org/10.1145/3338244 (visited on 11/03/2022).

[77] R. P. Karumur, T. T. Nguyen, and J. A. Konstan, "Exploring the Value of Personality in Predicting Rating Behaviors: A Study of Category Preferences on MovieLens," en, in *Proceedings of the 10th ACM Conference on Recommender Systems*, Boston Massachusetts USA: ACM, Sep. 2016, pp. 139–142, ISBN: 978-1-4503-4035-9. DOI: 10.1145/2959100.2959140. [Online]. Available: https://dl.acm.org/doi/10.1145/2959100.2959140 (visited on 12/03/2022).

[78] R. Hu and P. Pu, "Enhancing Collaborative Filtering Systems with Personality Information," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys '11, event-place: Chicago, Illinois, USA, New York, NY, USA: Association for Computing Machinery, 2011, pp. 197–204, ISBN: 978-1-4503-0683-6. DOI: 10.1145/2043932.2043969. [Online]. Available: https://doi.org/10.1145/2043932.2043969.

[79] O. Nalmpantis and C. Tjortjis, "The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender Systems," in *Engineering Applications of Neural Networks*, G. Boracchi, L. Iliadis, C. Jayne, and A. Likas, Eds., vol. 744, Series Title: Communications in Computer and Information Science, Cham: Springer International Publishing, 2017, pp. 498–507, ISBN: 978-3-319-65171-2 978-3-319-65172-9. DOI: 10.1007/978-3-319-65172-9_42. [Online]. Available: http://link.springer.com/10.1007/978-3-319-65172-9_42 (visited on 05/16/2023).

[80] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," en, in *Proceedings of the 26th International Conference on World Wide Web*, Perth Australia: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 173–182, ISBN: 978-1-4503-4913-0. DOI: 10.1145/3038912.3052569. [Online]. Available: https://dl.acm.org/doi/10.1145/3038912.3052569 (visited on 05/22/2023).

[81] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel, "Inferring the demographics of search users: Social data meets search queries," en, in *Proceedings of the 22nd international conference on World Wide Web - WWW '13*, Rio de Janeiro, Brazil: ACM Press, 2013, pp. 131–140, ISBN: 978-1-4503-2035-1. DOI: 10.1145/2488388.2488401. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2488388.2488401 (visited on 12/05/2022).

[82] T. Feng, Y. Guo, Y. Chen, X. Tan, T. Xu, B. Shen, and W. Zhu, "Tags and titles of videos you watched tell your gender," in *2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW: IEEE, Jun. 2014, pp. 1837–1842, ISBN: 978-1-4799-2003-7. DOI: 10.1109/ICC.2014.6883590. [Online]. Available: http://ieeexplore.ieee.org/document/6883590/ (visited on 12/05/2022).

[83] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "AttriInfer: Inferring User Attributes in Online Social Networks Using Markov Random Fields," en, in *Proceedings of the 26th International Conference on World Wide Web*, Perth Australia: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 1561–1569, ISBN: 978-1-4503-4913-0. DOI: 10.1145/3038912.3052695. [Online]. Available: https://dl.acm.org/doi/10.1145/3038912.3052695 (visited on 12/03/2022).

[84] S. Salamatian, A. Zhang, F. d. P. Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "How to hide the elephant- or the donkey- in the room: Practical privacy against statistical inference for large data," in *2013*

IEEE Global Conference on Signal and Information Processing, Austin, TX, USA: IEEE, Dec. 2013, pp. 269–272, ISBN: 978-1-4799-0248-4. DOI: 10.1109/GlobalSIP.2013.6736867. [Online]. Available: http://ieeexplore.ieee.org/document/6736867/ (visited on 12/05/2022).

[85] S. Bhagat, U. Weinsberg, S. Ioannidis, and N. Taft, "Recommending with an agenda: Active learning of private attributes using matrix factorization," en, in *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*, Foster City, Silicon Valley, California, USA: ACM Press, 2014, pp. 65–72, ISBN: 978-1-4503-2668-1. DOI: 10.1145/2645710.2645747. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2645710.2645747 (visited on 12/05/2022).

[86] T. Chen, R. Boreli, M.-A. Kaafar, and A. Friedman, "On the Effectiveness of Obfuscation Techniques in Online Social Networks," in *Privacy Enhancing Technologies*, E. De Cristofaro and S. J. Murdoch, Eds., vol. 8555, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 42–62, ISBN: 978-3-319-08505-0 978-3-319-08506-7. DOI: 10.1007/978-3-319-08506-7_3. [Online]. Available: http://link.springer.com/10.1007/978-3-319-08506-7_3 (visited on 12/05/2022).

[87] N. Z. Gong and B. Liu, "Attribute Inference Attacks in Online Social Networks," en, *ACM Transactions on Privacy and Security*, vol. 21, no. 1, pp. 1–30, Jan. 2018, ISSN: 2471-2566, 2471-2574. DOI: 10.1145/3154793. [Online]. Available: https://dl.acm.org/doi/10.1145/3154793 (visited on 12/05/2022).

[88] M. Slokom, A. Hanjalic, and M. Larson, "Towards user-oriented privacy for recommender system data: A personalization-based approach to gender obfuscation for user profiles," en, *Information Processing & Management*, vol. 58, no. 6, p. 102 722, Nov. 2021, ISSN: 03064573. DOI: 10.1016/j.ipm.2021.102722. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306457321002065 (visited on 12/03/2022).

[89] O. Loyola-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and M. García-Borroto, "Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases," en, *Neurocomputing*, vol. 175, pp. 935–947, Jan. 2016, ISSN: 09252312. DOI: 10.1016/j.neucom.2015.04.120. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231215015908 (visited on 05/08/2023).

[90] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," en, *Journal of Intelligent Information Systems*, vol. 46, no. 3, pp. 563–597, Jun. 2016, ISSN: 0925-9902, 1573-7675. DOI: 10.1007/s10844-015-0368-1. [Online]. Available: http://link.springer.com/10.1007/s10844-015-0368-1 (visited on 05/08/2023).

[91] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," en, *Neurocomputing*, vol. 135, pp. 32–41, Jul. 2014, ISSN: 09252312. DOI: 10.1016/j.neucom.2013.05.059. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231213011429 (visited on 05/25/2023).

[92] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Dec. 2015, Place: New York, NY, USA Publisher: Association for Computing Machinery, ISSN: 2160-6455. DOI: 10.1145/2827872. [Online]. Available: https://doi.org/10.1145/2827872.

[93] *Projects*, en, Sep. 2013. [Online]. Available: https://grouplens.org/about/projects/ (visited on 12/05/2022).

[94] M. D. Ekstrand, "LensKit for Python: Next-Generation Software for Recommender System Experiments," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, arXiv:1809.03125 [cs], Oct. 2020, pp. 2999–3006. DOI: 10.1145/3340531.3412778. [Online]. Available: http://arxiv.org/abs/1809.03125 (visited on 12/05/2022).

Appendix A

Recommender System Evaluation Results

This appendix provides a complete overview of the evaluation results for the Lenskit and Surprise algorithms. To access the code for generating and evaluating recommendations, please refer to the GitHub repository[1].

Table A.1: Evaluation metrics of the recommender algorithms from Lenskit training on Rating data (Personality2018 ratings). All algorithms with five folds.

Algorithm	**Precision@10**	**Recall@10**	**F1**
als.BiasedMF(feature=50, regularization=0.1)	0.029	0.006	0.009
als.ImplicitMF(features=10, reg=0.1, w=40)	0.205	0.066	0.100
FunkSVD(features=50, reg=0.015)	0.070	0.006	0.011
item_knn.ItemItem(nnbrs=10, msize=None)	0.003	0.00	0.000
basic.PopScore(quantile)	0.388	0.073	0.123
basic.Popular	0.384	0.071	0.120
basic.Random	0.003	0.000	0.001
user_knn.UserUser(nnbrs=10, min_sim=0)	0.000	0.000	0.000

[1] https://github.com/hanntorj/masters-thesis.git

Table A.2: Evaluation metrics of the recommender algorithms from Lenskit training on Rating-extended data (both Personality2018 and Movielens ratings). All algorithms with five folds.

Algorithm	**Precision@10**	**Recall@10**	**F1**
als.BiasedMF(features=50, regularization=0.1)	0.016	0.011	0.013
als.ImplicitMF(features=10, reg=0.1, w=40)	0.112	0.090	0.100
item_knn.ItemItem(nnbrs=10, msize=None)	0.006	0.001	0.002
basic.PopScore(quantile)	0.135	0.074	0.096
basic.Popular	0.135	0.074	0.096

Table A.3: Evaluation metrics of the recommender algorithms from Surprise training on Rating data (Personality2018 ratings).

Algorithm	**Precision@10**	**Recall@10**	**F1**
BaselineOnly	0.825	0.293	0.433
CoClustering	0.809	0.287	0.424
KNNBaseline	0.843	0.300	0.443
KNNBasic	0.855	0.317	0.463
KNNWithMeans	0.822	0.298	0.437
NMF	0.815	0.287	0.424
NormalPredictor	0.687	0.215	0.327
SlopeOne	0.824	0.290	0.429
SVD	0.870	0.303	0.449

Table A.4: Evaluation metrics of the SVD algorithms from Surprise training on Rating-extended data (both Personality2018 and Movielens ratings).

Algorithm	**Precision@10**	**Recall@10**	**F1**
SVD	0.756	0.573	0.652

Appendix B

Inference Results

This appendix presents inference results from Orange Data Mining in full. Relevant information is presented in Chapter 6. F1, precision, and recall are calculated as a weighted average for all classes.

B.1 Experiment 1

Model	AUC	CA	F1	Precision	Recall
kNN	0.518	0.322	0.321	0.334	0.322
Logistic Regression	0.534	0.406	0.332	0.285	0.406
Naive Bayes	0.512	0.061	0.089	0.167	0.061
Neural Network	0.492	0.306	0.293	0.286	0.306
Random Forest	0.542	0.311	0.292	0.278	0.311
SVM	0.516	0.272	0.258	0.253	0.272

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.500	0.267	0.259	0.258	0.267
Logistic Regression	0.443	0.239	0.158	0.119	0.239
Naive Bayes	0.485	0.211	0.191	0.193	0.211
Neural Network	0.461	0.217	0.192	0.184	0.217
Random Forest	0.497	0.222	0.203	0.216	0.222
SVM	0.493	0.278	0.267	0.261	0.278

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.491	0.217	0.207	0.206	0.217
Logistic Regression	0.483	0.200	0.165	0.142	0.200
Naive Bayes	0.538	0.222	0.224	0.229	0.222
Neural Network	0.515	0.211	0.191	0.182	0.211
Random Forest	0.541	0.183	0.170	0.162	0.183
SVM	0.491	0.200	0.187	0.187	0.200

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.481	0.250	0.238	0.234	0.250
Logistic Regression	0.487	0.356	0.222	0.189	0.356
Naive Bayes	0.446	0.222	0.185	0.178	0.222
Neural Network	0.470	0.278	0.250	0.234	0.278
Random Forest	0.524	0.289	0.249	0.225	0.289
SVM	0.526	0.256	0.255	0.257	0.256

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.495	0.183	0.182	0.193	0.183
Logistic Regression	0.505	0.233	0.196	0.171	0.233
Naive Bayes	0.499	0.206	0.206	0.229	0.206
Neural Network	0.484	0.217	0.210	0.208	0.217
Random Forest	0.476	0.156	0.149	0.153	0.156
SVM	0.516	0.233	0.230	0.228	0.233

(e) Neuroticism

Figure B.1: Classifying users with seven-class personality trait split based on random (Normalpredictor from the Surprise python package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.508	0.322	0.317	0.328	0.322
Logistic Regression	0.516	0.239	0.191	0.162	0.239
Naive Bayes	0.522	0.233	0.263	0.314	0.233
Neural Network	0.538	0.361	0.349	0.341	0.361
Random Forest	0.530	0.306	0.296	0.287	0.306
SVM	0.506	0.328	0.322	0.340	0.328

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.526	0.239	0.222	0.215	0.239
Logistic Regression	0.525	0.233	0.163	0.218	0.233
Naive Bayes	0.510	0.194	0.158	0.151	0.194
Neural Network	0.508	0.233	0.209	0.230	0.233
Random Forest	0.503	0.256	0.224	0.227	0.256
SVM	0.553	0.239	0.193	0.275	0.239

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.515	0.200	0.184	0.170	0.200
Logistic Regression	0.518	0.278	0.238	0.259	0.278
Naive Bayes	0.558	0.222	0.207	0.215	0.222
Neural Network	0.495	0.244	0.218	0.206	0.244
Random Forest	0.557	0.256	0.238	0.228	0.256
SVM	0.489	0.267	0.247	0.256	0.267

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.526	0.339	0.311	0.293	0.339
Logistic Regression	0.514	0.372	0.269	0.236	0.372
Naive Bayes	0.523	0.311	0.289	0.281	0.311
Neural Network	0.526	0.294	0.251	0.230	0.294
Random Forest	0.500	0.333	0.299	0.290	0.333
SVM	0.492	0.244	0.242	0.244	0.244

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.533	0.233	0.223	0.225	0.233
Logistic Regression	0.554	0.233	0.203	0.292	0.233
Naive Bayes	0.525	0.222	0.217	0.291	0.222
Neural Network	0.498	0.239	0.215	0.205	0.239
Random Forest	0.504	0.194	0.179	0.174	0.194
SVM	0.529	0.200	0.188	0.188	0.200

(e) Neuroticism

Figure B.2: Classifying users with seven-class personality trait split based on rating-based recommendations (SVD with random-state=0 from the Surprise python package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.507	0.500	0.497	0.494	0.500
Logistic Regression	0.485	0.561	0.442	0.470	0.561
Naive Bayes	0.496	0.533	0.483	0.521	0.533
Neural Network	0.494	0.522	0.513	0.508	0.522
Random Forest	0.537	0.550	0.532	0.529	0.550
SVM	0.464	0.550	0.544	0.540	0.550

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.515	0.467	0.449	0.433	0.467
Logistic Regression	0.504	0.567	0.423	0.439	0.567
Naive Bayes	0.445	0.528	0.438	0.417	0.528
Neural Network	0.475	0.511	0.481	0.458	0.511
Random Forest	0.510	0.539	0.500	0.475	0.539
SVM	0.479	0.450	0.440	0.455	0.450

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.464	0.389	0.389	0.389	0.389
Logistic Regression	0.493	0.539	0.377	0.290	0.539
Naive Bayes	0.535	0.511	0.444	0.419	0.511
Neural Network	0.550	0.511	0.495	0.486	0.511
Random Forest	0.551	0.500	0.472	0.462	0.500
SVM	0.447	0.394	0.400	0.421	0.394

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.489	0.722	0.700	0.684	0.722
Logistic Regression	0.492	0.789	0.696	0.622	0.789
Naive Bayes	0.469	0.789	0.696	0.622	0.789
Neural Network	0.464	0.750	0.687	0.650	0.750
Random Forest	0.564	0.783	0.703	0.680	0.783
SVM	0.451	0.606	0.628	0.656	0.606

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.516	0.433	0.435	0.436	0.433
Logistic Regression	0.510	0.578	0.468	0.511	0.578
Naive Bayes	0.511	0.550	0.478	0.479	0.550
Neural Network	0.505	0.489	0.459	0.440	0.489
Random Forest	0.495	0.489	0.454	0.432	0.489
SVM	0.559	0.356	0.360	0.440	0.356

(e) Neuroticism

Figure B.3: Classifying users with three-class personality trait split based on random (NormalPredictor from the Surprise package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.492	0.506	0.502	0.499	0.506
Logistic Regression	0.572	0.572	0.495	0.536	0.572
Naive Bayes	0.571	0.533	0.532	0.538	0.533
Neural Network	0.598	0.622	0.614	0.612	0.622
Random Forest	0.496	0.528	0.517	0.513	0.528
SVM	0.451	0.500	0.496	0.532	0.500

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.539	0.483	0.470	0.472	0.483
Logistic Regression	0.527	0.556	0.450	0.442	0.556
Naive Bayes	0.482	0.500	0.440	0.418	0.500
Neural Network	0.495	0.506	0.474	0.484	0.506
Random Forest	0.479	0.461	0.420	0.471	0.461
SVM	0.558	0.439	0.439	0.456	0.439

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.455	0.372	0.359	0.346	0.372
Logistic Regression	0.490	0.500	0.375	0.342	0.500
Naive Bayes	0.528	0.506	0.441	0.433	0.506
Neural Network	0.531	0.472	0.429	0.399	0.472
Random Forest	0.483	0.439	0.412	0.389	0.439
SVM	0.502	0.417	0.432	0.451	0.417

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.504	0.717	0.672	0.639	0.717
Logistic Regression	0.593	0.789	0.696	0.622	0.789
Naive Bayes	0.579	0.756	0.679	0.617	0.756
Neural Network	0.585	0.744	0.706	0.680	0.744
Random Forest	0.491	0.733	0.682	0.646	0.733
SVM	0.594	0.661	0.676	0.698	0.661

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.479	0.439	0.438	0.438	0.439
Logistic Regression	0.530	0.544	0.409	0.388	0.544
Naive Bayes	0.526	0.556	0.486	0.504	0.556
Neural Network	0.456	0.461	0.428	0.405	0.461
Random Forest	0.470	0.472	0.452	0.448	0.472
SVM	0.506	0.500	0.471	0.469	0.500

(e) Neuroticism

Figure B.4: Classifying users with three-class personality trait split based on rating-based (SVD with random-state=0 from the Surprise python package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.443	0.844	0.788	0.775	0.844
Logistic Regression	0.492	0.850	0.781	0.722	0.850
Naive Bayes	0.433	0.850	0.781	0.722	0.850
Neural Network	0.482	0.856	0.811	0.823	0.856
Random Forest	0.516	0.850	0.781	0.722	0.850
SVM	0.507	0.544	0.610	0.754	0.544

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.558	0.583	0.572	0.572	0.583
Logistic Regression	0.541	0.583	0.440	0.549	0.583
Naive Bayes	0.538	0.589	0.539	0.567	0.589
Neural Network	0.539	0.594	0.578	0.581	0.594
Random Forest	0.470	0.483	0.462	0.455	0.483
SVM	0.469	0.544	0.498	0.505	0.544

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.549	0.622	0.603	0.589	0.622
Logistic Regression	0.425	0.711	0.591	0.506	0.711
Naive Bayes	0.487	0.706	0.623	0.640	0.706
Neural Network	0.469	0.639	0.593	0.569	0.639
Random Forest	0.456	0.656	0.598	0.574	0.656
SVM	0.478	0.528	0.549	0.591	0.528

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.512	0.517	0.516	0.515	0.517
Logistic Regression	0.471	0.578	0.514	0.516	0.578
Naive Bayes	0.549	0.600	0.589	0.586	0.600
Neural Network	0.507	0.511	0.506	0.503	0.511
Random Forest	0.556	0.550	0.554	0.559	0.550
SVM	0.503	0.517	0.523	0.541	0.517

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.453	0.644	0.616	0.602	0.644
Logistic Regression	0.474	0.706	0.584	0.498	0.706
Naive Bayes	0.415	0.683	0.573	0.493	0.683
Neural Network	0.536	0.639	0.599	0.581	0.639
Random Forest	0.463	0.689	0.609	0.608	0.689
SVM	0.505	0.539	0.559	0.605	0.539

(e) Neuroticism

Figure B.5: Classifying users with binary-class personality trait split based on random (NormalPredictor from the Suprise python package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.466	0.844	0.788	0.739	0.844
Logistic Regression	0.492	0.861	0.797	0.742	0.861
Naive Bayes	0.479	0.856	0.794	0.741	0.856
Neural Network	0.520	0.856	0.804	0.791	0.856
Random Forest	0.502	0.856	0.804	0.791	0.856
SVM	0.507	0.594	0.656	0.766	0.594

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.490	0.550	0.531	0.531	0.550
Logistic Regression	0.580	0.600	0.491	0.626	0.600
Naive Bayes	0.585	0.606	0.552	0.595	0.606
Neural Network	0.480	0.550	0.502	0.511	0.550
Random Forest	0.553	0.589	0.566	0.572	0.589
SVM	0.641	0.411	0.409	0.406	0.411

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.559	0.667	0.611	0.593	0.667
Logistic Regression	0.628	0.711	0.591	0.506	0.711
Naive Bayes	0.638	0.728	0.644	0.724	0.728
Neural Network	0.554	0.683	0.609	0.596	0.683
Random Forest	0.595	0.678	0.624	0.612	0.678
SVM	0.481	0.556	0.575	0.615	0.556

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.545	0.583	0.583	0.582	0.583
Logistic Regression	0.539	0.550	0.508	0.500	0.550
Naive Bayes	0.485	0.544	0.525	0.517	0.544
Neural Network	0.550	0.556	0.553	0.551	0.556
Random Forest	0.562	0.567	0.559	0.555	0.567
SVM	0.423	0.461	0.468	0.484	0.461

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.483	0.633	0.580	0.554	0.633
Logistic Regression	0.519	0.694	0.578	0.495	0.694
Naive Bayes	0.452	0.672	0.605	0.591	0.672
Neural Network	0.507	0.672	0.605	0.591	0.672
Random Forest	0.398	0.650	0.597	0.576	0.650
SVM	0.565	0.489	0.509	0.545	0.489

(e) Neuroticism

Figure B.6: Classifying users with binary-class personality trait split based on rating-based (SVD with random-state=0 from the Surprise python package) recommendations.

B.2 Experiment 2

Model	AUC	CA	F1	Precision	Recall
kNN	0.509	0.306	0.301	0.297	0.306
Logistic Regression	0.478	0.372	0.316	0.299	0.372
Naive Bayes	0.492	0.028	0.050	0.243	0.028
Neural Network	0.470	0.317	0.288	0.278	0.317
Random Forest	0.500	0.356	0.325	0.307	0.356
SVM	0.480	0.300	0.281	0.277	0.300

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.464	0.183	0.174	0.169	0.183
Logistic Regression	0.524	0.278	0.193	0.207	0.278
Naive Bayes	0.509	0.206	0.201	0.210	0.206
Neural Network	0.490	0.217	0.184	0.180	0.217
Random Forest	0.558	0.256	0.233	0.229	0.256
SVM	0.509	0.256	0.230	0.234	0.256

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.509	0.228	0.220	0.228	0.228
Logistic Regression	0.523	0.217	0.185	0.209	0.217
Naive Bayes	0.535	0.194	0.200	0.210	0.194
Neural Network	0.497	0.172	0.159	0.153	0.172
Random Forest	0.515	0.194	0.186	0.187	0.194
SVM	0.521	0.178	0.154	0.158	0.178

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.480	0.267	0.245	0.239	0.267
Logistic Regression	0.544	0.367	0.270	0.334	0.367
Naive Bayes	0.542	0.272	0.272	0.287	0.272
Neural Network	0.514	0.328	0.296	0.290	0.328
Random Forest	0.529	0.289	0.255	0.237	0.289
SVM	0.493	0.244	0.234	0.243	0.244

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.546	0.217	0.208	0.213	0.217
Logistic Regression	0.492	0.217	0.179	0.162	0.217
Naive Bayes	0.563	0.206	0.216	0.252	0.206
Neural Network	0.504	0.233	0.220	0.227	0.233
Random Forest	0.553	0.306	0.298	0.308	0.306
SVM	0.534	0.222	0.213	0.210	0.222

(e) Neuroticism

Figure B.7: Classifying users with seven-class personality trait split based on personality-based (NCF+soft-labeled [71]) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.481	0.522	0.504	0.500	0.522
Logistic Regression	0.441	0.556	0.467	0.494	0.556
Naive Bayes	0.454	0.456	0.440	0.457	0.456
Neural Network	0.458	0.511	0.490	0.485	0.511
Random Forest	0.463	0.528	0.500	0.499	0.528
SVM	0.489	0.544	0.513	0.516	0.544

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.534	0.506	0.487	0.470	0.506
Logistic Regression	0.500	0.572	0.435	0.498	0.572
Naive Bayes	0.517	0.500	0.442	0.414	0.500
Neural Network	0.533	0.539	0.494	0.471	0.539
Random Forest	0.500	0.506	0.467	0.441	0.506
SVM	0.484	0.472	0.458	0.452	0.472

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.527	0.450	0.455	0.460	0.450
Logistic Regression	0.551	0.517	0.397	0.469	0.517
Naive Bayes	0.529	0.472	0.424	0.400	0.472
Neural Network	0.493	0.483	0.437	0.441	0.483
Random Forest	0.554	0.517	0.481	0.457	0.517
SVM	0.492	0.356	0.371	0.392	0.356

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.527	0.717	0.680	0.653	0.717
Logistic Regression	0.614	0.783	0.693	0.621	0.783
Naive Bayes	0.618	0.744	0.727	0.710	0.744
Neural Network	0.632	0.772	0.733	0.722	0.772
Random Forest	0.576	0.756	0.704	0.678	0.756
SVM	0.690	0.694	0.707	0.727	0.694

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.571	0.500	0.486	0.480	0.500
Logistic Regression	0.531	0.561	0.449	0.486	0.561
Naive Bayes	0.572	0.517	0.480	0.472	0.517
Neural Network	0.545	0.550	0.526	0.535	0.550
Random Forest	0.509	0.517	0.470	0.449	0.517
SVM	0.498	0.461	0.482	0.522	0.461

(e) Neuroticism

Figure B.8: Classifying users with three-class personality trait split based on personality-based (NCF+soft-labeled [71]) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.575	0.833	0.783	0.738	0.833
Logistic Regression	0.463	0.861	0.797	0.742	0.861
Naive Bayes	0.535	0.844	0.788	0.739	0.844
Neural Network	0.511	0.850	0.791	0.740	0.850
Random Forest	0.559	0.850	0.791	0.740	0.850
SVM	0.433	0.783	0.763	0.744	0.783

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.525	0.561	0.532	0.537	0.561
Logistic Regression	0.506	0.583	0.440	0.549	0.583
Naive Bayes	0.395	0.539	0.485	0.491	0.539
Neural Network	0.528	0.578	0.542	0.555	0.578
Random Forest	0.476	0.572	0.547	0.552	0.572
SVM	0.436	0.444	0.442	0.473	0.444

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.594	0.661	0.627	0.613	0.661
Logistic Regression	0.643	0.706	0.588	0.505	0.706
Naive Bayes	0.644	0.694	0.609	0.604	0.694
Neural Network	0.557	0.678	0.612	0.598	0.678
Random Forest	0.543	0.672	0.596	0.569	0.672
SVM	0.517	0.583	0.592	0.604	0.583

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.564	0.578	0.580	0.582	0.578
Logistic Regression	0.516	0.583	0.564	0.560	0.583
Naive Bayes	0.533	0.561	0.560	0.560	0.561
Neural Network	0.567	0.583	0.584	0.584	0.583
Random Forest	0.511	0.528	0.528	0.529	0.528
SVM	0.481	0.511	0.516	0.554	0.511

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.483	0.628	0.582	0.558	0.628
Logistic Regression	0.513	0.711	0.597	0.795	0.711
Naive Bayes	0.532	0.633	0.575	0.544	0.633
Neural Network	0.483	0.672	0.584	0.553	0.672
Random Forest	0.521	0.667	0.618	0.606	0.667
SVM	0.548	0.517	0.528	0.543	0.517

(e) Neuroticism

Figure B.9: Classifying users with binary-class personality trait split based on personality-based (NCF+soft-labeled [71]) recommendations.

B.3 Experiment 3

Model	AUC	CA	F1	Precision	Recall
kNN	0.494	0.650	0.696	0.766	0.650
Logistic Regression	0.486	0.544	0.615	0.754	0.544
Naive Bayes	0.473	0.533	0.605	0.737	0.533
Neural Network	0.531	0.722	0.741	0.762	0.722
Random Forest	0.555	0.844	0.812	0.797	0.844
SVM	0.439	0.622	0.676	0.766	0.622

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.555	0.556	0.539	0.546	0.556
Logistic Regression	0.516	0.517	0.515	0.514	0.517
Naive Bayes	0.521	0.544	0.545	0.545	0.544
Neural Network	0.502	0.506	0.492	0.493	0.506
Random Forest	0.506	0.550	0.516	0.536	0.550
SVM	0.517	0.533	0.533	0.532	0.533

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.539	0.472	0.496	0.577	0.472
Logistic Regression	0.596	0.544	0.566	0.634	0.544
Naive Bayes	0.652	0.572	0.592	0.659	0.572
Neural Network	0.526	0.589	0.600	0.616	0.589
Random Forest	0.568	0.633	0.618	0.607	0.633
SVM	0.486	0.456	0.478	0.572	0.456

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.560	0.533	0.528	0.549	0.533
Logistic Regression	0.532	0.533	0.534	0.537	0.533
Naive Bayes	0.557	0.550	0.549	0.549	0.550
Neural Network	0.545	0.533	0.531	0.530	0.533
Random Forest	0.497	0.522	0.493	0.508	0.522
SVM	0.495	0.472	0.468	0.467	0.472

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.458	0.506	0.550	0.682	0.506
Logistic Regression	0.532	0.578	0.613	0.673	0.578
Naive Bayes	0.476	0.533	0.576	0.674	0.533
Neural Network	0.461	0.628	0.644	0.663	0.628
Random Forest	0.470	0.628	0.636	0.645	0.628
SVM	0.560	0.589	0.623	0.684	0.589

(e) Neuroticism

Figure B.10: Classifying users with binary-class personality trait split oversampled to a 1:1 ratio, based on rating-based (SVD with random-state=0 from the Surprise python package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.594	0.639	0.693	0.798	0.639
Logistic Regression	0.442	0.472	0.551	0.741	0.472
Naive Bayes	0.480	0.517	0.591	0.746	0.517
Neural Network	0.494	0.528	0.600	0.757	0.528
Random Forest	0.597	0.522	0.593	0.796	0.522
SVM	0.500	0.533	0.605	0.774	0.533

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.595	0.578	0.567	0.572	0.578
Logistic Regression	0.519	0.500	0.501	0.502	0.500
Naive Bayes	0.534	0.494	0.496	0.501	0.494
Neural Network	0.526	0.494	0.495	0.502	0.494
Random Forest	0.528	0.517	0.515	0.530	0.517
SVM	0.444	0.422	0.416	0.433	0.422

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.533	0.450	0.458	0.619	0.450
Logistic Regression	0.613	0.544	0.566	0.629	0.544
Naive Bayes	0.595	0.494	0.516	0.611	0.494
Neural Network	0.511	0.539	0.560	0.606	0.539
Random Forest	0.564	0.556	0.576	0.635	0.556
SVM	0.434	0.433	0.452	0.566	0.433

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.573	0.483	0.478	0.494	0.483
Logistic Regression	0.538	0.550	0.549	0.548	0.550
Naive Bayes	0.543	0.594	0.595	0.597	0.594
Neural Network	0.520	0.517	0.507	0.534	0.517
Random Forest	0.532	0.561	0.559	0.573	0.561
SVM	0.512	0.472	0.473	0.474	0.472

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.371	0.344	0.379	0.594	0.344
Logistic Regression	0.532	0.572	0.611	0.702	0.572
Naive Bayes	0.458	0.544	0.585	0.666	0.544
Neural Network	0.385	0.439	0.489	0.614	0.439
Random Forest	0.381	0.400	0.453	0.583	0.400
SVM	0.511	0.556	0.596	0.695	0.556

(e) Neuroticism

Figure B.11: Classifying users with binary user distribution undersampled to a 1:1 ratio, based on rating-based (SVD with random-state=0 from the Surprise python package) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.563	0.667	0.710	0.776	0.667
Logistic Regression	0.469	0.528	0.601	0.749	0.528
Naive Bayes	0.530	0.578	0.643	0.777	0.578
Neural Network	0.586	0.772	0.774	0.776	0.772
Random Forest	0.544	0.822	0.803	0.789	0.822
SVM	0.438	0.522	0.596	0.733	0.522

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.447	0.500	0.481	0.483	0.500
Logistic Regression	0.505	0.483	0.482	0.482	0.483
Naive Bayes	0.538	0.517	0.517	0.517	0.517
Neural Network	0.497	0.494	0.479	0.479	0.494
Random Forest	0.504	0.522	0.480	0.497	0.522
SVM	0.495	0.556	0.556	0.557	0.556

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.463	0.467	0.491	0.560	0.467
Logistic Regression	0.651	0.594	0.613	0.686	0.594
Naive Bayes	0.561	0.550	0.571	0.637	0.550
Neural Network	0.530	0.550	0.560	0.572	0.550
Random Forest	0.495	0.594	0.582	0.572	0.594
SVM	0.446	0.483	0.506	0.594	0.483

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.566	0.539	0.539	0.545	0.539
Logistic Regression	0.574	0.583	0.583	0.583	0.583
Naive Bayes	0.547	0.522	0.523	0.526	0.522
Neural Network	0.558	0.556	0.556	0.556	0.556
Random Forest	0.549	0.561	0.539	0.556	0.561
SVM	0.500	0.489	0.489	0.495	0.489

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.544	0.522	0.566	0.683	0.522
Logistic Regression	0.644	0.600	0.636	0.731	0.600
Naive Bayes	0.608	0.594	0.629	0.698	0.594
Neural Network	0.553	0.622	0.644	0.673	0.622
Random Forest	0.554	0.706	0.704	0.703	0.706
SVM	0.512	0.544	0.586	0.679	0.544

(e) Neuroticism

Figure B.12: Classifying users with binary user distribution oversampled to a 1:1 ratio, based on personality-based (NCF+soft-labeled [71]) recommendations.

Model	AUC	CA	F1	Precision	Recall
kNN	0.520	0.550	0.619	0.755	0.550
Logistic Regression	0.473	0.467	0.546	0.731	0.467
Naive Bayes	0.569	0.567	0.633	0.782	0.567
Neural Network	0.561	0.567	0.633	0.797	0.567
Random Forest	0.556	0.556	0.623	0.803	0.556
SVM	0.402	0.350	0.425	0.701	0.350

(a) Openness

Model	AUC	CA	F1	Precision	Recall
kNN	0.508	0.522	0.515	0.515	0.522
Logistic Regression	0.482	0.489	0.488	0.500	0.489
Naive Bayes	0.537	0.539	0.539	0.549	0.539
Neural Network	0.497	0.472	0.471	0.483	0.472
Random Forest	0.560	0.511	0.509	0.525	0.511
SVM	0.548	0.550	0.548	0.565	0.550

(b) Conscientiousness

Model	AUC	CA	F1	Precision	Recall
kNN	0.462	0.456	0.478	0.572	0.456
Logistic Regression	0.650	0.572	0.592	0.675	0.572
Naive Bayes	0.595	0.528	0.548	0.641	0.528
Neural Network	0.526	0.472	0.491	0.603	0.472
Random Forest	0.454	0.439	0.458	0.570	0.439
SVM	0.466	0.400	0.397	0.580	0.400

(c) Extraversion

Model	AUC	CA	F1	Precision	Recall
kNN	0.492	0.522	0.516	0.537	0.522
Logistic Regression	0.546	0.517	0.516	0.516	0.517
Naive Bayes	0.563	0.550	0.551	0.552	0.550
Neural Network	0.521	0.517	0.516	0.524	0.517
Random Forest	0.535	0.478	0.470	0.490	0.478
SVM	0.494	0.589	0.589	0.590	0.589

(d) Agreeableness

Model	AUC	CA	F1	Precision	Recall
kNN	0.474	0.439	0.481	0.667	0.439
Logistic Regression	0.628	0.583	0.621	0.712	0.583
Naive Bayes	0.579	0.533	0.576	0.681	0.533
Neural Network	0.490	0.483	0.530	0.660	0.483
Random Forest	0.553	0.489	0.531	0.697	0.489
SVM	0.576	0.506	0.550	0.639	0.506

(e) Neuroticism

Figure B.13: Classifying users with binary user distribution undersampled to a 1:1 ratio, based on personality-based (NCF+soft-labeled [71]) recommendations.

Appendix C

Orange Data Mining

This appendix contains snapshots of the Orange Data Mining file used to conduct the experiments. The file can be downloaded from the GitHub repository[1].

[1] https://github.com/hanntorj/masters-thesis.git

C.1 Experiment 1 and 2

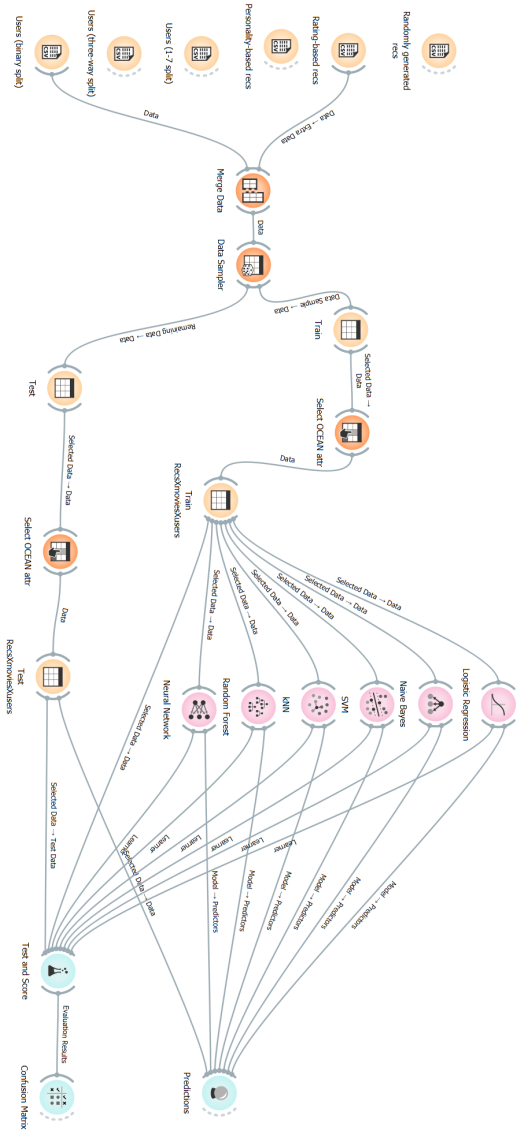


Figure C.1: Snapshot of the Orange Data Mining file used in Experiments 1 and 2.

C.2 Experiment 3

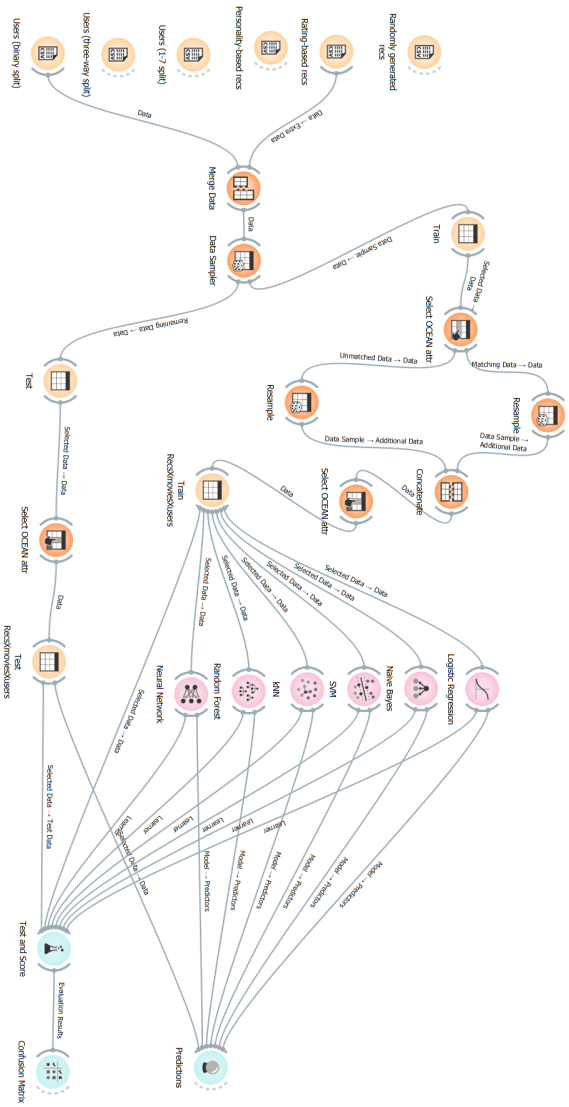
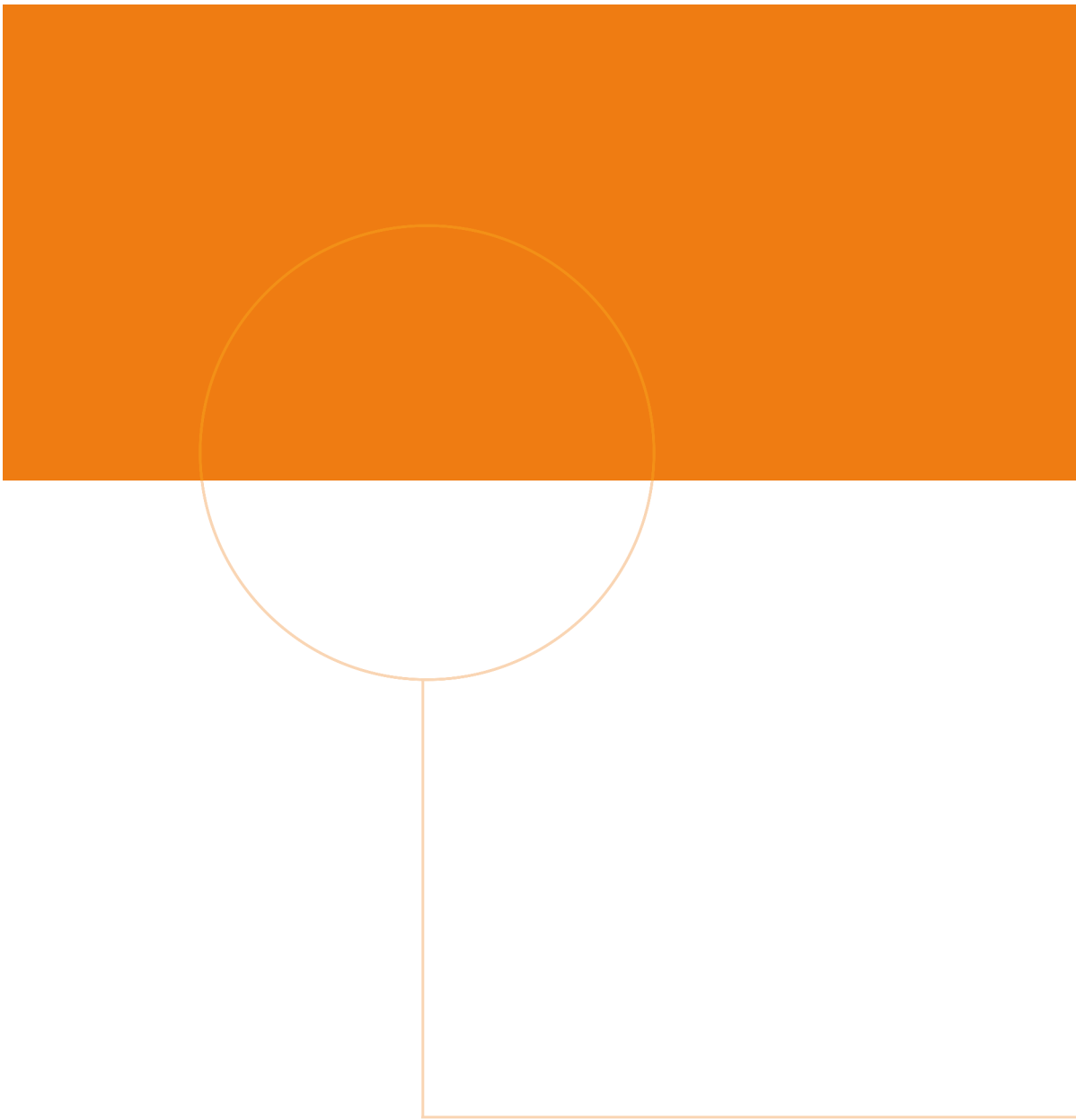


Figure C.2: Snapshot of the Orange Data Mining file used in Experiment 3.



NTNU

Norwegian University of
Science and Technology