Master's thesis

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

Victor Jørgensen & Hans Kristian Sande

Predictive Modeling Using Pseudo-Social Networks Derived from Transaction Data

Master's thesis in Computer Science
Supervisor: Kjetil Nørvåg
Co-supervisor: Lars Ivar Hagfors
June 2022

NTNU
Norwegian University of
Science and Technology

Victor Jørgensen & Hans Kristian Sande

Predictive Modeling Using Pseudo-Social Networks Derived from Transaction Data

Master's thesis in Computer Science
Supervisor: Kjetil Nørvåg
Co-supervisor: Lars Ivar Hagfors
June 2022

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

NTNU
Norwegian University of
Science and Technology

Abstract

Staying ahead of consumer trends is important for businesses in the retail market to maintain a competitive advantage. In a world becoming increasingly digitalized, retailers who wish to remain competitive must adapt to the technological changes by incorporating new technological capabilities. Fortunately for retail banks, they possess a valuable asset that can be leveraged to increase consumer engagement. This asset is transaction data and may be used to predict future events.

In this thesis, transaction data is transformed into a *pseudo-social network* to capture the inherent similarity among consumers. Several methods for calculating similarity are explored and further extracted as features into a machine learning model to predict future buyers. The experiments are conducted on product offerings with varying target response in cooperation with a retail bank to measure the predictive performance in a realistic setting.

The experimental results show that the features extracted from the pseudo-social network may significantly increase the predictive performance of predictive models, especially when used in combination with traditional features from customer data. Albeit, the significance is determined by the quality of the extracted features.

This thesis provides an extensive evaluation of the application of pseudo-social networks in predictive modeling. Secondly, it provides new methods for capturing similarity features from pseudo-social networks that achieve higher quality than previous studies.

Sammendrag

Å ligge et steg foran trender er viktig for bedrifter som ønsker å ivareta et konkurransefortrinn. I en verden som blir stadig mer digitalisert, så må bedrifter tilpasse seg det teknologiske skiftet ved å tilegne seg nye teknologiske kompetanser. Heldigvis for kommersielle banker har de en verdifull ressurs som kan utnyttes til å ligge et steg foran. Denne ressursen er transaksjonsdata, og det kan brukes til å predikere fremtidige hendelser.

I denne oppgaven transformeres transaksjonsdata til et *pseudo-sosialt nettverk* for å fange opp de iboende likhetene blant forbrukere. Flere metoder for å beregne likhet utforskes og anvendes videre i en maskinlæringsmodell for å predikere fremtidige kjøpere. Eksperimentene utføres på tjenester med varierende oppslutning i samarbeid med en kommersiell bank for å måle modellens prediksjonsevne i en realistisk setting.

De eksperimentelle resultatene viser at likhet-attributter utregnet fra det pseudo-sosiale nettverket kan øke den prediktive ytelsen til prediktive modeller betydelig, spesielt når de brukes i kombinasjon med tradisjonelle attributter fra kundedata. Økningen i prediktiv ytelse avhenger riktignok av kvaliteten på de utregnede likhet-attributtene.

Denne oppgaven gir en omfattende evaluering av pseudo-sosiale nettverk i prediktiv analyse. I tillegg fremgår det nye metoder for å måle likhet mellom forbrukere i det pseudo-sosiale nettverket som oppnår høyere kvalitet enn tidligere.

Preface

This thesis is submitted as part of the course *TDT4900 - Computer Science, Master's Thesis* at the *Norwegian University of Science and Technology* (NTNU), marking the end of a five-year-long journey to achieve the degree of *Master of Science in Computer Science*. The research was conducted under the supervision and co-supervision of respectively Professor Kjetil Nørvåg at NTNU's *Department of Computer Science* and Lars Ivar Hagfors, Data Scientist at *SpareBank 1 SMN*. We want to extend our gratitude to our supervisors for providing us with invaluable feedback and guidance throughout the research.

Contents

Chapter 1

Introduction

This chapter introduces the motivation and research topic. Two research questions define the scope of the thesis. The main contributions are summarized, followed by an outline of the remaining chapters.

1.1 Motivation

Marketing campaigns are more likely to succeed if they reach the right consumers. Knowing which consumers to contact with a product offering is an important step in a successful marketing campaign. Predictive models can help identify the most likely buyers in selected target groups. Introducing features from new data sources to the predictive model pipeline may enhance the predictive performance, illustrated in Figure 1.1. One such method that has shown promising results is *pseudo-social network targeting* [1]. By deriving a pseudo-social network from a transaction data set, it may be possible to capture behavioral similarities between consumers. These similarity features may be incorporated with existing predictive models to hopefully increase the predictive performance, thereby aiding the marketers in reaching the right consumers.

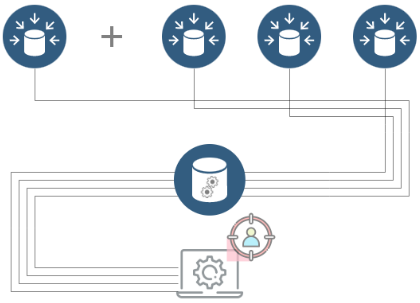


Figure 1.1: Adding new data sources to the predictive modeling pipeline.

1.2 Pseudo-Social Network Targeting

Most people perhaps think of *social media* when they hear the term *social network*. While social media platforms such as Facebook and Twitter indeed represent social networks, a social network may be described more generally. Social networks are social structures generally consisting of social actors in the form of individuals or organizations and their connections. Analyzing these social structures is known as *social network analysis* [2]. Information from social network analysis may be utilized to target consumers with the intent of selling a product or service, referred to as social network targeting. Social network targeting in Figure 1.2 is performed by targeting the social actors that exhibit proximity in the network to the individuals of interest, such as the known buyers of a product. Social network targeting has been justified based on theories of homophily, and social influence [3]. In the case of homophily, the assumption is that similar individuals are more likely to make connections [4]. This may be exploited in targeting where individuals who previously purchased a product would suggest that similar individuals may be interested in purchasing the same product. In the case of social influence, customer satisfaction may propagate among the network in a word-of-mouth fashion, suggesting that customers that were satisfied (or not satisfied) with a purchased product might affect whether similar customers will purchase the product themselves.

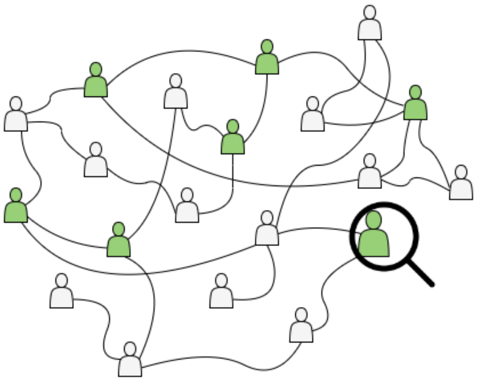


Figure 1.2: Social network targeting.

The premise for social network targeting is naturally a social network. Many companies, however, lack adequate data to construct such a network. An approach that mimics social network targeting is *pseudo*-social network targeting. A pseudo-social network (PSN) is an inferred social network, not a true social network. It is *pseudo* because the connected social actors probably do not have social relationships in real life. The lack of real social relationships weakens the argument for social influence. However, the underlying assumption is that homophily is still restored in the inferred network. That is, the network also possesses similarities between the social actors that may be used to target consumers. In this thesis, the constructed pseudo-social networks are based on behavioral data from fine-grained transaction data. The social actors are the consumers and they are connected in the network if they have made a payment to the same merchant.

1.3 Ethical and Privacy Concerns

This study is completed in collaboration with a medium-sized Norwegian commercial bank who have granted access to sensitive information about consumers' purchasing patterns. All consumers have consented to the usage of their data, and respecting their privacy is a primary concern. The data provided by the bank is fully anonymized, so it is impossible to identify the consumers. The data is securely stored on a virtual machine to lower the risk of information leakage. A VPN connection is required to access the virtual machine, which is only possible through enterprise machines provided by the bank. However, these security protocols restrain the research possibilities.

1.4 Research Questions

The research goal for this thesis is to explore new methods for pseudo-social network targeting by extracting new features from transaction data to be used in predictive modeling. The following two research questions set the scope for this thesis to help achieve this goal. The research questions are revisited and answered in the final chapter.

RQ1: *How may fine-grained transaction data be leveraged to increase the predictive performance of predictive models?*

RQ2: *What features from pseudo-social networks help increase the predictive performance of predictive models?*

1.5 Contributions

The contributions of this thesis are two-fold. Most importantly, this study provides a more extensive evaluation of pseudo-social network targeting. The results are interpreted across a broad range of metrics to gain more insights into actual predictive performance. The results confirm that pseudo-social network targeting is effective but dependent on data balance and featurization.

Secondly, the proposed methodology extends upon existing methods by extracting new features from transaction data. The results show that a selection of the new features from the proposed methodology possess significantly more predictive quality than previous features. This study is also the first in its domain to evaluate the features in more detail.

1.6 Thesis Outline

Chapter 1 - Introduction This chapter presents the context of the thesis. This includes motivation, domain, research questions and contributions.

Chapter 2 - Background This chapter gives an overview of theoretical background information. This includes graphs, machine learning and pseudo-social network targeting.

Chapter 3 - Related Work This chapter summarizes the approaches and findings by previous related studies.

Chapter 4 - Methodology This chapter gives an in-depth explanation of the proposed methodology to featurization of pseudo-social networks which extends upon the research presented in the preceding chapter.

Chapter 5 - Experiments This chapter describes how the experiments were conducted. This includes a description of the data used, defining models, experimental setup and criteria of evaluation.

Chapter 6 - Results This chapter presents, evaluates and discusses the experimental results.

Chapter 7 - Conclusions The final chapter answer the research questions and proposes unexplored directions for future work.

Chapter 2

Background

The chapter will cover graphs, machine learning, and pseudo-social network targeting.

2.1 Graphs

Graphs represent relationships between objects, e.g., a social network. Formally a graph G can be defined as a pair of disjoint sets (V, E) where V is a set of vertices, often called nodes, and E is a set of edges where E is a subset of the set V^2 of unordered pairs of V.

Definition 2.1.1. A graph $G = (V, E)$ is a set of **vertices** $V = \{v_1, v_2, ..., v_i\}$ connected pairwise by a set of **edges** $E = \{e_1, e_2, ..., e_j\}$.

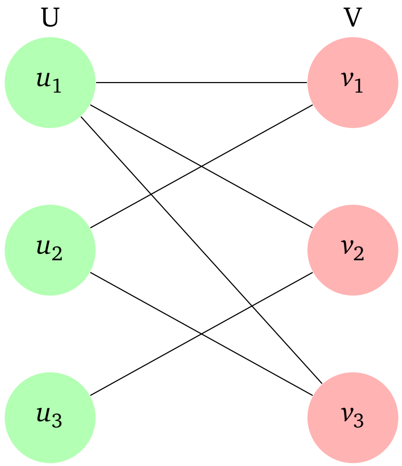


Figure 2.1: A bipartite graph consisting of the two disjoint sets U and V connected pairwise by the edges E, represented by the black lines.

2.1.1 Bipartite Graphs

A bipartite graph (or *bigraph*) is a type of graph in which a graph's vertices can be divided into two disjoint and independent sets U and V such that every edge is a connection between a vertex in U and a vertex in V, illustrated in Section 2.1.1.

Definition 2.1.2. A bipartite graph $B = (U, V, E)$ where $U = \{u_1, u_2, ..., u_r\}$ is a set of vertices disjoint to another set of vertices $V = \{v_1, v_2, ..., v_s\}$. The edges E may only contain pair with exactly one vertex from each of the sets U and V.

2.1.2 Graph Representations

Graphs may be represented as an *adjacency matrix*, *adjacency list* or an *edge list*.

Adjacency Matrix

Matrices can represent graphs. Element x_{ij} is a binary indicator of the existence of an edge between node i and node j. A *biadjacency* matrix is simply a matrix representation of a bipartite graph. In the case of a bipartite graph, the representation is a rectangular $N \times M$ matrix where N is the number of bottom nodes, and M is the number of top nodes. Because each entry in an adjacency matrix only occupies one bit, dense data can be represented very compactly. However, for sparse data, adjacency matrices are inefficient with regard to memory.

Definition 2.1.3. An adjacency matrix is a matrix representation of a graph $G = (V, E)$ consisting of ones and zeroes, where element x_{ij} is a binary indicator of whether the vertices v_i and v_j are adjacent or not. The space requirement for an adjacency matrix is $O(|V^2|)$

Adjacency List

Adjacency lists can also represent graphs and bipartite graphs. For sparse data, adjacency lists are more memory efficient than adjacency matrices because they do not waste space representing edges that are not present.

Definition 2.1.4. An adjacency list is a representation of a graph $G = (V, E)$. A collection of unordered lists represent the graph, where each unordered list describes the neighbors of a particular node in the graph. The space requirement for an adjacency list is $O(|V| + |E|)$

Edge List

An edge list can be considered a simpler variation of an adjacency list.

Definition 2.1.5. An edge list is a list representation of a graph $G = (V, E)$. The graph is represented by a list of its edges E. The space requirement for an edge list is $\Theta(|E|)$.

2.2 Machine Learning

Machine learning is a broad term with multiple definitions in the literature. Arthur Samuel, a pioneer in computer science, defined *machine learning* [5] as "a field of study that gives computers the ability to learn without being explicitly programmed." Machine learning algorithms need input data to learn. In the era of big data, the amount of available data is continuously increasing, and new use-cases for machine learning continuously emerge. With great success, machine learning has been applied to various fields such as fraud detection, image classification, and medical diagnosis.

Machine learning is a subfield of artificial intelligence (AI), but the two terms are not interchangeable. Artificial intelligence refers to intelligence demonstrated by a machine, as opposed to the natural intelligence that humans and animals demonstrate. Artificial intelligence differs from machine learning because an AI does not require the ability to learn and can be explicitly programmed to display specific behavior. In this section, we focus on machine learning and not other subfields of AI. Figure 2.2 illustrates three main categories of machine learning with a highlight on supervised learning, which is the most relevant to this thesis.

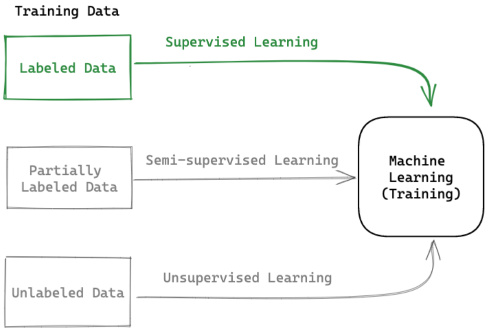


Figure 2.2: Categories of machine learning algorithms.

2.2.1 Supervised Learning

Machine learning can be applied to a broad set of problems. Classification and regression problems reside in the domain of supervised learning. Supervised machine learning algorithms are used to make future predictions based on labeled historical data. The algorithm will predict a discrete or continuous value when presented with new data.

Classification is the task of dividing data into classes. If there are only two possible classes, it is called a *binary classification* problem. If there are three or more possible classes to classify the instances to, the problem is a *multinomial classification* problem. Spam detection is a classification problem. An email is classified as either spam or not spam based on the available information in the email.

Regression algorithms aim to find a function that maps the input variable to a continuous output variable. This involves identifying correlations between dependent and independent variables. Regression algorithms can be further divided into *linear* and *non-linear* regression. These are defined mathematically by Equation (2.1) and Equation (2.2). Regression can be used to predict prices in a market, the weather, or a person's height, to name a few use cases.

Logistic regression is a classification algorithm that can be used to solve classification problems by predicting the probability of an outcome. It uses a *sigmoid function* as defined by Equation (2.3) to map the input to a discrete outcome by rounding values above or below specified thresholds.

$$Y = \beta_0 + \beta_1 X_1 \tag{2.1}$$

$$Y = f(X, \beta) + \epsilon \tag{2.2}$$

$$S(X) = \frac{1}{1 + e^{-X}} \tag{2.3}$$

where

Y = dependent variable you are trying to predict,

β_0 = the intercept, the predicted value of Y when X is 0,

β_1 = regression coefficient, how much Y is expected to change as X increases,

X_1 = independent variable,

X = a vector of p predictors,

β = a vector of k parameters,

f = a known regression function,

ϵ = the error estimate.

2.2.2 Unsupervised Learning

Unsupervised learning cannot be used directly to solve classification or regression problems as the input data is unlabeled. Instead, unsupervised learning models is used to solve *clustering* and *association* problems. Unsupervised learning is also used to reduce the dimensionality of data before feeding it to a supervised model.

Clustering is the process of grouping similar objects into clusters based on some similarity measure. Clustering is useful for detecting patterns in data that humans may be unable to identify.

Association is a method in which the model finds *association rules* between variables. A classic example of an association rule is market basket analysis, in which the model can discover that customers who purchase a specific product are also, in fact, likely to purchase another specified product.

2.2.3 Semi-supervised Learning

If the collected data contains both labeled and unlabeled data, the task falls in the final category of semi-supervised learning. The conjunction of labeled and unlabeled data may lead to improvements in learning accuracy. Labeling data may be costly, as this is often done by humans. Working with large data sets would increase this cost, and with a lack of resources it may be a better option to opt for the semi-supervised approach.

2.2.4 Ensemble Machine Learning Algorithms

The notion of *ensemble* learning models is to combine multiple weak learners into a single strong learner to improve the accuracy of the final model. The two algorithms presented below are both ensemble algorithms, and their main difference is *how* they combine models.

Gradient Boosting

By default, the algorithm uses decision trees as weak learners combined sequentially to create a single ensemble model. Each weak learner focuses on reducing the errors made by the previous model by updating the values of the observations. *Boosting* refers to the method of combing multiple homogeneous models sequentially, as illustrated in Figure 2.3. Boosting methods generally reduce bias in predictions.

Random Forest

Random forest divides data into N subsets, which are then used to train N homogeneous models independently in parallel. To make a prediction each of the weak learners are given the test data. In regression problems, the output of all the weak learners are averaged across for a final output. In classification tasks the final output is decided by a majority vote of the models. The method of combining the models in parallel is known as *bagging*, which is illustrated in Figure 2.4. Bagging methods are generally used to decrease prediction variance.

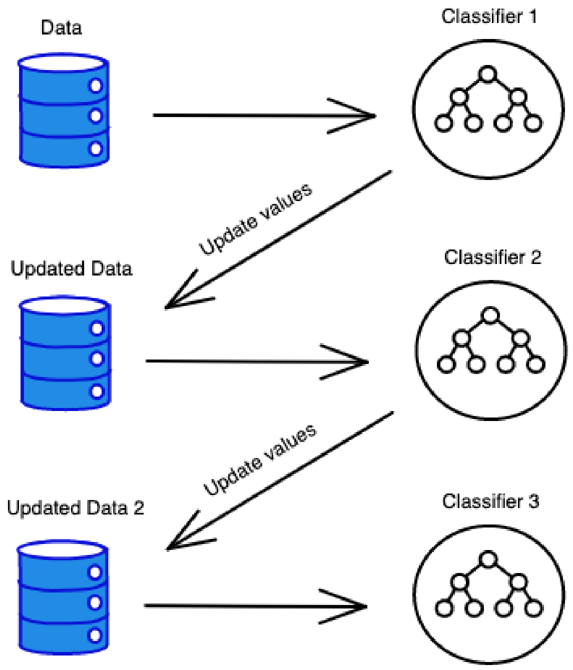


Figure 2.3: Weak learners combined sequentially in gradient boosting.

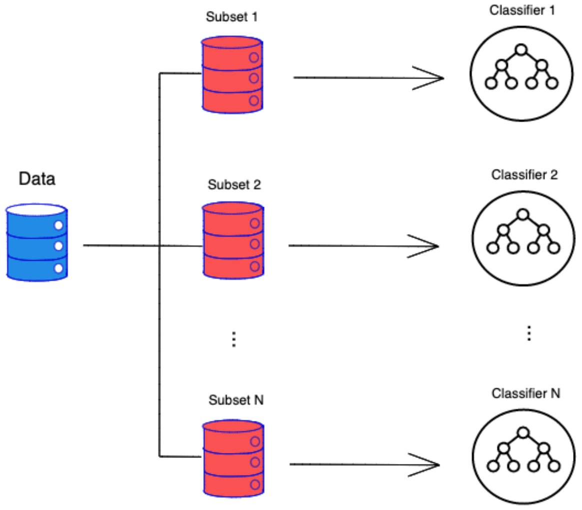


Figure 2.4: Weak learners combined in parallel in random forest.

2.3 Pseudo-Social Network Targeting

Previous sections describe essential concepts relevant to pseudo-social network targeting. This section elaborates on the general methodology for pseudo-social network targeting and the considerations involved in the different steps of the methodology.

2.3.1 Creating a Pseudo-Social Network (PSN)

Pseudo-social networks have been used in different domains for network analytics and predictive modeling [6–8]. A (pseudo)-social network can be represented as a graph, which is beneficial as it allows for the application of graph theory and techniques to analyze the network. Whether a graph representation of a social network is the best model for analysis and measuring similarity is debatable among mathematicians, and sociologists [2]. However, in a data science context, it is appropriate.

A pseudo-social network is inferred by projecting a bigraph to a unigraph consisting of only a single type of entity. A unigraph is applicable because no general network-oriented methodology has been proposed yet to perform classification or predictive modeling on vertices in bigraphs, and the current techniques do not scale well in settings of big data [9]. Bipartite network projection is a method to compress information about bipartite networks. The initial bigraph consists of entities from two disjoint sets, e.g., consumer vertices and merchant vertices [1]. The resulting (pseudo-social) network, or unigraph after projection, consists of merely consumer vertices. The edges between the consumer vertices signify that they share a common neighbor in the bigraph. The projection from a bigraph to unigraph is visualized in Figure 2.5.

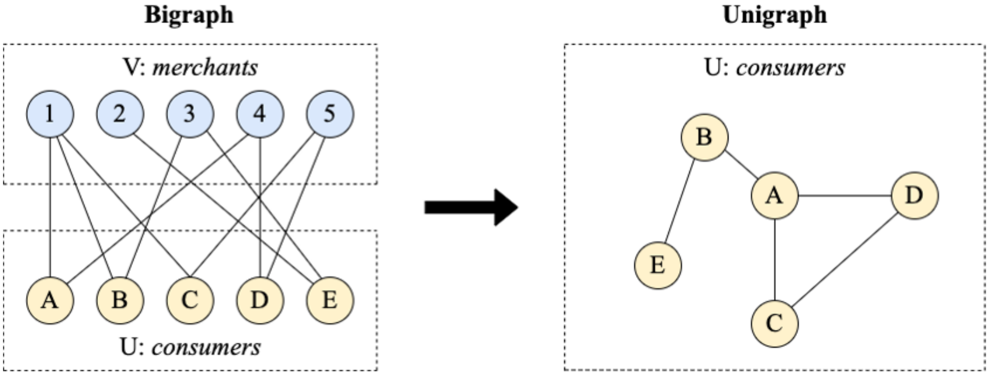


Figure 2.5: A network projection from a bigraph of consumers and merchants to a unigraph of consumers only.

2.3.2 Weighting Pseudo-Social Networks

A drawback of the projection from bigraph to unigraph is that the unigraph is always less informative than the original bigraph [10]. Many properties of the bigraph are lost in the projection, such as the topology of the network and the frequency of connections between the vertices in the disjoint sets. It is, therefore, often required to weigh the network to minimize information loss. The optimal weighting method should reflect the nature of the specific network and indicate how strong the connections are between the consumers in the social network. Stankova et al. [9] present different ways of weighting a pseudo-social network based on transaction data. Martens et al. [11] and Caigny et al. [12] demonstrate how different weighting schemes impact the final accuracy of their predictive models based on transaction data – signifying the importance of applying an appropriate weighting method.

2.3.3 Calculating Similarity Between the Consumers in a PSN

The next step after weighting the network is to measure the similarity among the consumers to identify which consumers are the most similar to the individuals of interest. This step is part of the featurization process (also known as feature engineering), which means changing some form of data into a numerical vector (a feature) that the predictive model can read. Martens and Provost [11] introduce a behavioral similarity measure they call BeSim-score, which aims to capture the similarities between consumers based on their interactions with different merchants. Caigny et al. [12] take it a step further by incorporating RFM-values in their BeSim-scores to discriminate behavior and interactions into three dimensions. In addition to the mentioned similarity measures for graphs derived from transaction data, general-purpose similarity metrics use an embedding space to map similarity between two vertices in a graph, such as node2vec [13]. Other general-purpose similarity metrics attempt to rank vertices based on their role in the network, such as the centrality measures betweenness, closeness, and degree. For example, degree centrality refers to how many connections a vertex has in a graph [14]. The creativity of the designer essentially limits the number of similarity features. However, that does not mean continually adding new similarity features will increase the model's predictive abilities. While some measures may fail to capture the essential similarities, others may correlate with one another [15].

2.3.4 General Framework for Prediction Using Social Networks

The general framework for prediction on graph data using similarity measures is demonstrated in Figure 2.6. In their first paper, Martens et al. [1] apply a similarity-based approach in their walk-through example of identifying likely buyers before turning to the more popular and state-of-the-art learning-based approach in their experiments. The learning-based approach is more adaptable to adding multiple similarity features as each new feature becomes another variable in the input data. By the same argument, it is also easier to include other features such as socio-demographic characteristics. This makes the learning-based approach a good basis for evaluating feature importance when comparing models using fine-grained data to calculate similarity features with baseline models using structured data with socio-demographic features [11, 12].

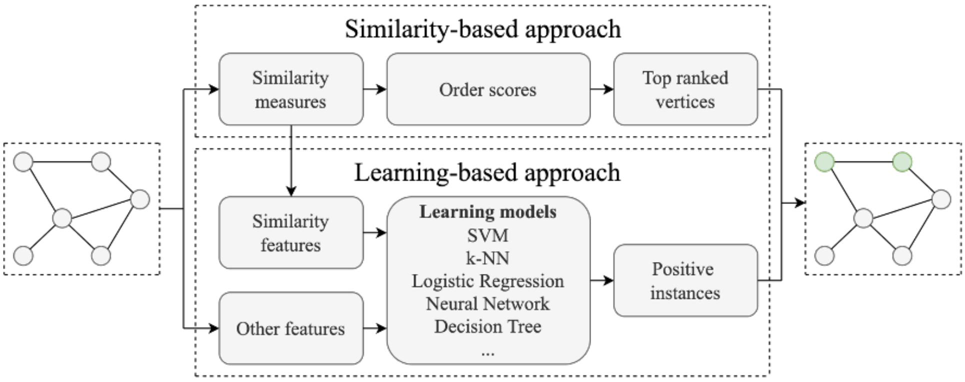


Figure 2.6: The general framework for predictive modeling using similarity measures. A green vertex in the example implies the consumer is predicted to be a likely buyer.

2.3.5 Choice of Learning Model

The next step after featurization is to choose a learning model, specifically a binary classification model. Classification is a type of supervised learning model where the categories are predefined, and the model's predictive task is to categorize new probabilistic observations into said categories. It goes to show that predictive modeling using pseudo-social networks can be applied in a broad range of other domains such as predicting anti-money laundering [16], credit scoring [7], and product attrition [6].

There are several considerations when deciding which model to implement, both from a technical and business perspective. Scalability is a desired property when working with big data sets, especially in the absence of GPUs. Another technical aspect to consider is the quality of data and how it affects the performance of different classification models. Kaur et al. [17] present several limitations of popular learning models in the case of noisy data, imbalanced data sets, and the shortcomings of different models when features correlate. There are ways to combat some of these limitations, such as random oversampling or undersampling in the case of imbalanced data sets [18], but there is no model that is perfect for every setting.

In terms of a business perspective, there is commonly an explainability requirement. The model must be easy to interpret and comprehend for managerial approval and deployment policies. Linear models predict the target as a weighted sum of the feature inputs, such that the coefficients (significance) of the features are directly interpretable. Both Martens et al. [1, 11] and Caigny et al. [12] emphasize their choices of linear models concerning the explainability and comprehensibility requirements. Other research suggests that a non-linear model can provide better results at classification using transactional and social network data [6] while being able to explain the importance of features [19].

Chapter 3

Related Work

This section presents the practices and results from related work in pseudo-social network targeting, with an emphasis on (1) weighting of networks, (2) similarity measures and network features, and (3) different learning models.

3.1 Weighting Projected Unigraphs

Given a bipartite graph $G = (U, V, E)$, the general method for weighting the projected unigraph is to capture information about the vertices in set V of the initial bigraph that is lost in the projection and further pass it along to the unigraph. Stankova et al. [9] entitle the vertices in V as the *top nodes* and present several weighting functions to capture their significance in the initial network. The weighting functions are designed in a posteriori fashion in which the functions utilize known (observed) information about which vertices in U are the individuals of interest (namely the targets), except for the simple weighting function, which assigns equal weights between the vertices in U in the unigraph based on the existence of shared connections between the vertices in U and V. In their experiments, the weighting function combining beta distribution yields the best overall performance on different datasets. The optimal parameters for beta and alpha were discovered using a grid search on the specific dataset that provides the best predictive performance (AUC-score) on a held-out validation set. Both Martens et al. [11], and Caigny et al. [12] apply the beta weighting function in their research on transaction data and compare it with other weighting functions such as the unnuanced simple weighting, the empiric probability weighting, and an inverse frequency weighting method. Caigny et al. [12] show that the beta-distribution and inverse frequency yields the best predictive results in predicting customer life events. The results from Martens et al. [11] show that the beta-distribution yields the best predictive scores when predicting customer purchases. However, they also remark how the shape of the beta-distribution conforms to that of the inverse frequency measure, indicating that they encapture the notion that merchants with many connections should be down-weighted as they provide little discriminativeness. The functions do, however, output quite different weights. It is apparent from prior research that the beta-distribution weighting method is generally the best. Nevertheless, it requires much more computation as it requires fine-tuning parameters using the specific cross-validation holdout technique. The second-best weighting method using inverse frequency might be a better option when considering scalability.

The next step after weighting the top nodes in the bigraph is determining the link weights in the projection. This step contemplates several aggregation functions that calculate the weights of the edges between the vertices in the unigraph as an aggregation of the weights from the shared top nodes. Stankova et al. [9] present several of these aggregation functions, such as the sum of shared nodes, the maximum of shared nodes, cosine-similarity, and Jaccard-similarity. They observe that the aggregation functions Jaccard-similarity, cosine-similarity, and maximum of shared nodes do not scale well to data sets with high dimensionality. Their general results show that the cosine function and the sum of shared nodes are the most suitable methods. However, they conclude that the latter is more favorable because it can be combined with specific classifier models to scale easily with very large data sets. Martens et al. [11] and Caigny et al. [12] also opted for the aggregation function that is the sum of shared nodes in their experiments.

3.2 Similarity Measures and Network Features

Martens et al. [11] introduce a versatile and scalable behavioral similarity measure *BeSim* that identifies those consumers most similar to the individuals of interest. Their results show that the BeSim-model and the traditional model capture complementary information and, in combination, are better at predicting which consumers have a higher response than the average response. Caigny et al. [12] expand the BeSim-score to include the dimensions of recency, frequency, and monetary value *RFM* of transactions when calculating the similarity between consumers in the unigraph. The scores are calculated by first aggregating the values of the consumer's transactions across the given dimensions between the respective consumer and its connected merchants in the bigraph, secondly calculating the deviation between the respective consumer's value with the average value of the known buyers and the average value of the known non-buyers. Finally, they apply a penalty function that yields a higher score to the consumers more similar to the known buyers and a lower score to those consumers more similar to the known non-buyers. Their results show that the models incorporating BeSim-scores on RFM-values deliver an increased predictive performance compared to the models using the standard BeSim-score from Martens and Provost [11]. Nonetheless, both papers support that using fine-grained data to calculate similarity features enhances predictive performance. These similarity measures are calculated in an a posteriori manner, as they both require known information about who purchased the offered product.

Lismont et al. [6] deduce several other similarity features from their network in their prediction of product attrition, which they refer to as *network features*. They also compare the significance of network features to the *local features* derived from structured data and finally to a hybrid model comprising both local and network features. Their network features incorporate another angle: measuring a consumer's influence in the network, using a modified PageRank algorithm on the bigraph, and centrality measures on the unigraph. In all cases except for their neural network model, the models that apply network features and hybrid features outperform the models using local features. Another interesting approach in their research is that they include the time period as a categorical variable in their models to encapsulate the dynamic aspect of the network as it evolves, as opposed to the BeSim-feature from Martens et al. [11] which looks at the network as a static graph. Caigny et al. [12] also compute their BeSim-scores on a static graph in which the time period for the basis of calculation is set to a year.

Munoz-Cancino et al. [20] perform extensive techniques in their search for better network features when attempting to predict loan defaulters. The transaction dataset consists of 7.65 million people and 245 thousand firms. They apply complex techniques from graph representation learning [21] such as graph embeddings (node2vec), graph convolutional networks, and graph autoencoders. The purpose of using methods from graph representation learning is to train the models to capture the important features themselves, replacing the featurization process, which usually requires domain knowledge. Another great benefit is that the feature vectors are dynamic and encapture the network's evolution. Their results show that the best-performing model comprises both traditional features from structured data and graph representation learning features. However, it should be noted that the runtime of feature extraction from the graph embedding and the graph convolution network alone was approximately 15 thousand minutes. Although their research is impressive, their methods require more powerful computer specifications than standard so scalability could become a concern in general settings.

3.3 Learning Models

Martens et al. [11] use both linear and non-linear SVMs in their experiments, of which the linear SVM is the best performing model, which is promising in terms of the explainability requirement for managerial approval. However, as their results show, it does not scale with big data sets as its training complexity is highly dependent on the size of the data set. Caigny et al. [12] apply logistic regression. Lismont et al. [6] evaluate different learning models and show that the ensemble method using random forests outperforms neural networks, decision trees, and namely logistic regression in all cases using both local features, network features, and the hybrid model. Stankova et al. [9] apply relational classifiers for node classification over bipartite graphs, which captures information about the entire network instead of just local information. Another benefit of relational classifiers is that they are easily scalable when working with big data sets. Oskarsdottir et al. [7] show that non-relational classifiers enriched with network features, without collective inference, using binary weights and undirected networks provide better predictive performance on credit scoring. The non-relation classifiers in their experiments included logistic regression, neural networks, and random forests. Munoz-Cancino et al. [20] opted for regularized logistic regression, random forest, and gradient boosting. The latter delivered consistently better results. Regarding the requirement of explainability, gradient boosting models can easily be combined with SHAP-values to describe feature importance [19]. In general, ensemble learning algorithms (random forest and gradient boosting) seem to provide the best performance across the domains using big data in predictive modeling. Gradient boosting may capture more complex patterns in data than random forest but is also more prone to overfit if the data is noisy. This problem may be mitigated by using cross-validation to tune the hyper-parameters but ensuring data quality remains a focal point.

Chapter 4

Methodology

This chapter describes the proposed methodology for targeting consumers using transaction data.

4.1 General Framework

The general overview of the framework is illustrated in Figure 4.1, and the following sections describe each phase in-depth. The first phase involves the preprocessing of transaction data to represent a bigraph. The second step includes calculating the similarity between the consumers in the bigraph. The third step is the construction and weighting of the pseudo-social network. Finally, the last step presents the feature collection from the featurization process, which will be used for feature selection for the different models. The structured data also require some preprocessing to extract features. However, these steps are not expanded in this chapter, as the motivation for this chapter is to extract features from transaction data. The features from structured data are however included in the total feature collection in Table 4.6 in Section 4.5, and are further elaborated in Chapter 5 along with the extraction of targets.

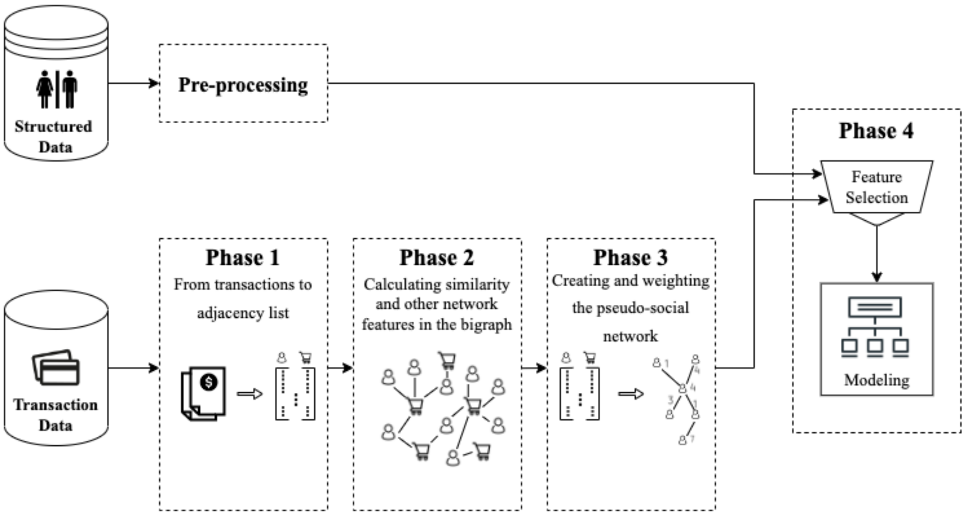
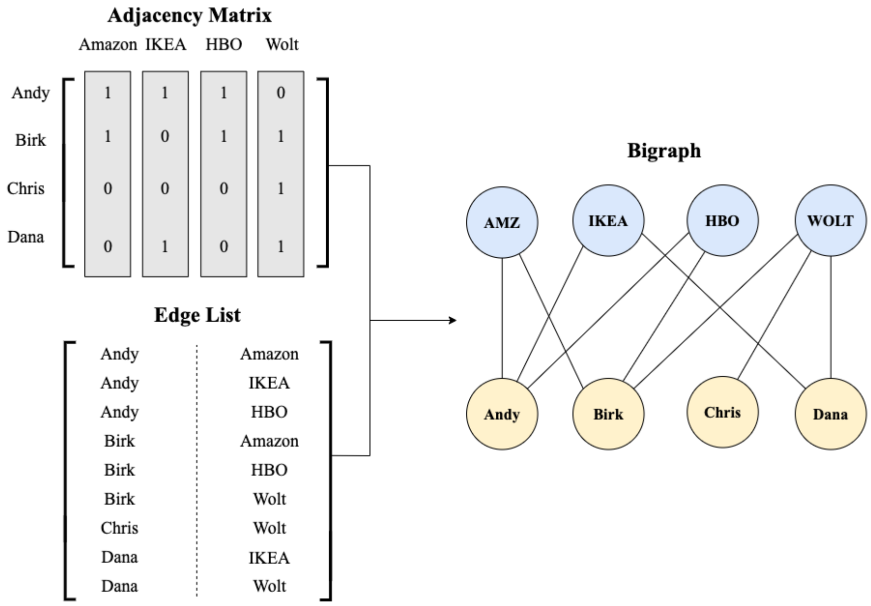

Structured Data
Pre-processing
Phase 4
Feature Selection
Modeling
Transaction Data
Phase 1
From transactions to adjacency list
Phase 2
Calculating similarity and other network features in the bigraph
Phase 3
Creating and weighting the pseudo-social network

Figure 4.1: General framework for targeting consumers using transaction data.

4.2 From Transaction Data to a Bigraph

The first phase involves manipulating the transaction data into a graph representation for the initial bigraph. In its modest form, the transactions (money transfers) in the transaction data consist of a source and a target. The source shall be referred to as the *consumer*, and the target shall be referred to as the *merchant*. All tuples in the transaction data indicate the merchants at which different consumers have made payments. This information may be utilized to construct a bigraph of consumers and merchants. Previous studies have represented this bigraph using an adjacency matrix [1, 11, 12], in which a value of 1 indicates that the given consumer has made a payment to the corresponding merchant, and 0 if the consumer has not made a payment to that specific merchant. Although this representation perhaps makes it conceptually easier to visualize the connections in the bigraph, it is not the most memory-efficient representation when working with big data sets and sparse graphs. Other alternative representations that are more efficient for sparse graphs are an adjacency list or an edge list. This proposed implementation applies edge lists to represent the connections in the bigraph, thereby avoiding the necessity to represent the connections that are not present. The edge list can easily be generated from the list of transactions simply by dropping duplicate entries. An example comparison of the representations are demonstrated in Figure 4.2. Notice that both representations yield the same resulting bigraph, despite the edge list requiring fewer representations of connections in the network.



Adjacency Matrix

	Amazon	IKEA	HBO	Wolt
Andy	1	1	1	0
Birk	1	0	1	1
Chris	0	0	0	1
Dana	0	1	0	1

Edge List

Andy	Amazon
Andy	IKEA
Andy	HBO
Birk	Amazon
Birk	HBO
Birk	Wolt
Chris	Wolt
Dana	IKEA
Dana	Wolt

Figure 4.2: Edge list compared to adjacency matrix for creating a bigraph.

4.3 Capturing Information from the Bigraph

In the general methodology for pseudo-social network targeting presented in Section 2.3 the order is to first project and weight the unigraph before calculating similarity among the consumers. These steps may be interchangeable because the bigraph also possesses information that may be used to calculate similarity. The second phase in the proposed methodology begins with capturing information from the bigraph by weighting the merchants, removing noise in the data set, and finally calculating similarity features.

4.3.1 Weighting the Merchants

It is important to look at the properties of the consumers' adjacent vertices in the bigraph because their significance in the network should be a factor when calculating consumer similarity. This assumption is derived from the idea of homophily: that the consumers with shared merchants in the bigraph indicate similarity. Merchants may be weighted with respect to their connecting consumers who are known to have purchased the target product. A selection of the general weighting methods of top nodes from Stankova et al. [9] are shown in Table 4.1, in which the top nodes are equivalent to merchants in this specific bigraph.

Table 4.1: Weighting methods for merchants.

Weighting Function	Formula	Description
Simple	$w = 1$	Simple weighting with no nuance
Inverse degree	$w = \frac{1}{C_i}$	Each merchant is weighted based on the inverse of its amount of consumers
Inverse frequency	$w = log_{10}(\frac{N}{C_i})$	Each merchant is weighted based on the log of the total amount of consumers in the graph N, divided by its amount of consumers
Empiric probability	$w = \frac{KB_i}{C_i}$	Each merchant is weighted based on its amount of known buyers of the target product, divided by its total amount of consumers

The first equation for *simple weighting* is exemplified in Figure 4.2, in which the weight of the merchant nodes and their edges between consumers are unnuanced and simply 1 if there exists a connection between the consumer and merchant. The second equation *inverse degree* attempts to capture the uniqueness of a merchant concerning how many connections a given merchant has. This notion of uniqueness is further expanded in the third equation *inverse frequency* or *inverse consumer frequency (ICF)*, which looks at the bigraph altogether to capture the discriminativeness of merchants. The idea of inverse frequency stems from a common metric used in information retrieval, namely the *inverse document frequency (IDF)* [22]. This weighting function down-weights merchants connected to many consumers because they provide little discriminativeness and rewards merchants connected to few consumers because they provide more discriminativeness. To illustrate the point in case, consider the simple bigraph consisting of the two merchants *Tax Office* and *Guitarshop*: most consumers pay their taxes and will therefore be connected to the Tax Office in the bigraph, but the number of consumers interested in guitars should be substantially less, so comparatively few will be connected the Guitarshop. By this assumption, the consumers who share the merchant Guitarshop as a common neighbor in the bigraph should generally indicate more similarity compared to the consumers who share Tax Office as a common neighbor. The number of consumers connected to the Tax Office will be far greater, suggesting a higher diversity and thus less discriminativeness between its connected consumers.

The final equation *empiric probability* differs as a weighting function compared to the other equations. We refer to it as an a posteriori method because it utilizes empiric evidence about which consumers are known to have purchased the target product. This weighting function requires special care when preparing the data for modeling because it can lead to data leakage. Data leakage is essentially the leakage of information about the target variable, which should not be legitimately available beforehand [23]. The empiric probability must be calculated using only the consumers in the training data set and thus isolated from the consumers in the test set. These considerations are further elaborated in the experimental setup in Section 5.3. Table 4.2 shows what the different weighted values for each merchant would be from the running example using the latter three weighting functions.

Table 4.2: The different weighted values of merchants.

Consumers	Merchant	Inv Deg	Inv Freq	Empiric Probability
Andy, Birk	Amazon	0.50	0.30	0.50
Andy, **Dana**	IKEA	0.50	0.30	1.0
Andy, Birk	HBO	0.50	0.30	0.50
Birk, Chris, **Dana**	Wolt	0.33	0.13	0.33

The consumers denoted in bold are the known buyers.

Stankova et al. [9] concluded that the beta-distribution density function was the best general alternative for weighting top nodes. However, related research issues its scalability concerns [11]. Therefore, the proposed methodology applies the following weighting functions: inverse frequency and empiric probability. Both functions are less computationally expensive than the beta distribution function and have shown better performance for prediction based on transaction data [12].

4.3.2 Removal of Noise

The weightings of merchants may help remove noise in the bigraph, such as the notion of discriminativeness in the inverse frequency weighting function. However, such functions are inadequate to ensure data quality, and scalability remains an important point in terms of utility. Merchants with considerably more consumers than the average are removed from the bigraph to eliminate noise. Thus a merchant like Tax Office could typically be removed. Other merchants are removed based on the two following axioms:

1. Remove merchants with one or fewer consumers
2. Remove merchants without any known buyers

The removal of merchants by the two axioms is not only beneficial for scalability reasons. Firstly, these merchants' connecting consumers can not be connected to the known buyers in the projected unigraph, as they share no merchants in the bigraph. Secondly, the purpose of calculating similarity among the consumers is ultimately to measure a consumer's similarity to a known buyer. In the running example from Figure 4.3, the merchant Guitarshop would also be removed from the bigraph, as it has no connecting known buyers.

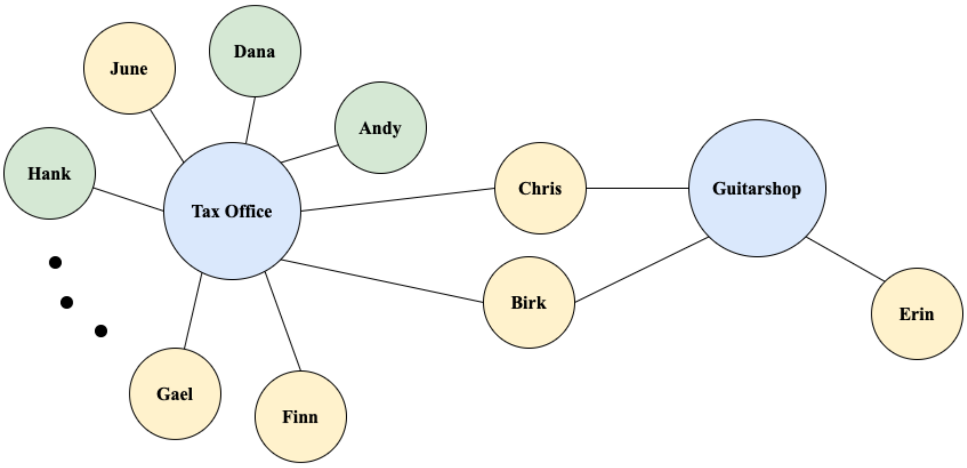


Figure 4.3: The green consumers are the known buyers of the target product.

4.3.3 Calculating Similarity in the Bigraph

Let us now expand the representation of transactions introduced in Section 4.2 to its true form, which along with source and target, includes: the date of the transaction, the monetary value of the transaction, and the category of transaction. These values can be exploited to calculate similarity along different dimensions. Although the binary links between consumers and merchants indicate similarity through shared neighbors, they do not give the whole picture. Table 4.3 shows the diversity of behaviors that can exist between consumers despite having shared neighbors in the bigraph. Birk, Chris, Dana, and Erin would be connected in the pseudo-social network derived from Table 4.3, suggesting they are similar. However, further inspection of their transactions reveals that Birk is more alike Erin, and Chris is more alike Dana. It is known that Dana is a known buyer, which suggests that Chris is more likely to purchase the target product in this simplified example because he is more similar to a consumer who has purchased the product. The goal of the behavioral similarity measures should be to capture these nuances and thus reward those consumers that are more similar to the known buyers.

Table 4.3: Edge list of transactions with monetary value.

Consumer	**Merchant**	**Monetary Value** $
Birk	Wolt	90
Chris	Wolt	850
Dana	Wolt	700
Erin	Wolt	140

The consumers denoted in bold are the known buyers.

Aggregating the transactions allows for calculating behavioral similarity scores for the dimensions recency, frequency, and monetary value (*RFM*). Caigny et al. [12] incorporates these RFM-dimensions in their behavioral similarity scores, and their work is the inspiration for how the proposed methodology calculates similarity scores along these dimensions. All three dimensions may be calculated in the same fashion, in which the transactions are aggregated for every consumer with its connecting merchants. In the case of recency, the date of the last transaction between $consumer_i$ and $merchant_j$ is added to the initial edge list. For frequency, the aggregated count of transactions between $consumer_i$ and $merchant_j$ is added to another separate edge list. Equally for monetary value, which is the total amount spent at $merchant_j$ by $consumer_i$.

After all person-level data has been summarized for every merchant, the next step is to evaluate each consumer with respect to the known buyers and known non-buyers by calculating their deviation from the mean. That is, how similar is a given consumer compared to the average known buyer and the average non-known buyer of a merchant's consumers. The first calculation is demonstrated in Table 4.4 using monetary value, and the second calculation comparing specific consumers to the monetary value averages for the merchant Wolt is demonstrated in Table 4.5.

Table 4.4: Aggregated sums and averages of monetary value (MV) for known buyers (KB) and known non-buyers (KNB).

Consumers	**Merchant**	$\sum \mathbf{MV}_{KB}$	$\sum \mathbf{MV}_{KNB}$	$\overline{\mathbf{MV}}_{KB}$	$\overline{\mathbf{MV}}_{KNB}$
Andy, Birk, Erin	Amazon	25000	30000	25000	15000
Andy, **Dana**, June	IKEA	60000	3000	30000	3000
Andy, Birk, Gael	HBO	7000	2500	7000	1250
Birk, Chris, **Dana**, Erin	Wolt	700	1080	700	360

The consumers denoted in bold are the known buyers.

Table 4.5: Comparing consumer similarity to known buyers (KB) and known non-buyers (KNB) using monetary value (MV) for Wolt.

Consumer	**Merchant**	$\mathbf{MV}_i$	$\lvert\mathbf{MV}_i - \overline{\mathbf{MV}}_{KB}\rvert$	$\lvert\mathbf{MV}_i - \overline{\mathbf{MV}}_{KNB}\rvert$
Birk	Wolt	90	610	270
Chris	Wolt	850	150	490
Dana	Wolt	700	0	340
Erin	Wolt	140	560	220

It can be observed from the results of the calculations in Table 4.5 that each consumer is assigned a distance from the average monetary value. These distances indicate the similarity between known buyers and known non-buyers and will be further used to calculate similarity features. The procedure is the same for the recency and frequency dimensions.

The category type is the next similarity measure that can be derived from the bigraph. As previously mentioned, each transaction is labeled with a category, e.g., travel, health, and groceries. By grouping the consumers' transactions on category type, it is easy to derive the category distribution for a consumer's spending. The assumption is that consumers similar to the known buyers will have similar spending habits and generally be interested in the same products. The category type features are included in Table 4.6 in Section 4.5.

The last similarity measure from the bigraph is inspired by degree centrality. This feature is the aggregation of merchants connected to every consumer. That is, the total amount of distinct merchants in which a specific consumer has made a payment. This feature's goal is to capture a consumer's activity level and influence in the bigraph.

4.4 Constructing and Weighting the PSN

The third phase is the construction and weighting of the pseudo-social network. The construction is performed by bipartite network projection, in which the resulting unigraph is the pseudo-social network. Although there are many variants for weighting the consumers in the unigraph, previous research indicates that the *sum of shared nodes* is the state-of-the-art method [1, 9, 11, 12]. Figure 4.4 shows an example of how the pseudo-social network can be weighted using the sum of shared nodes for inverse frequency. Notice that the weight between Andy and Birk is the sum of the inverse frequency values for their shared merchants, which are Amazon and HBO.

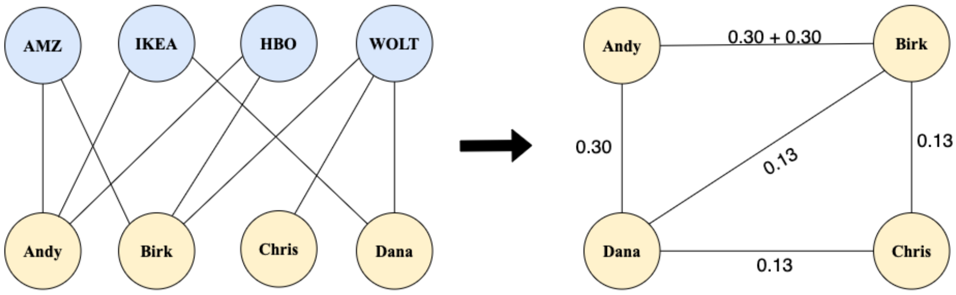


Figure 4.4: Bipartite network projection using sum of shared nodes for inverse frequency.

The sum of shared nodes is used to calculate the behavioral similarity score S_{ICF}. The calculation of this similarity measure follows the steps in prior research by Martens et al. [1, 11] and is demonstrated in Equation (4.1). This measure represents the binary connections between consumers and merchants and does not incorporate the nuances that may reside in their relationships.

$$S_{ICF}(x_i) = \sum E(j) \times ICF(j) \tag{4.1}$$

$$S_{RFM}(x_i) = \sum R_{ij} \tag{4.2}$$

$$S_{RFM}(x_i)_{ICF} = \sum R_{ij} \times ICF(j) \tag{4.3}$$

$$S_{RFM}(x_i)_{E \times ICF} = \sum R_{ij} \times (E(j) \times ICF(j)) \tag{4.4}$$

in which

x_i = the specific consumer i,
R_{ij} = the deviation value for consumer i with respect to the average for the merchant j,
$ICF(j)$ = the inverse frequency for merchant j connected to x_i,
$E(j)$ = the empiric probability for merchant j connected to x_i

In the same fashion, it is possible to calculate the similarity measures for RFM by summing their values across the common neighbors, such as in Equation (4.2), Equation (4.3) and Equation (4.4). Caigny et al. [12] show that these scores can further be scaled (weighted) with respect to the weighted merchants in the same manner. Early results in Appendix A in Table A.1 showed however that the best models with RFM features did not use scaling.

Alternative Approaches

Caigny et al. combine the BeSim-scores using a penalty function with two inputs. The consumers' similarity distances between the average known buyers and average known non-buyers for every merchant are combined as demonstrated in Equation (4.5). The output of this function is a single-value feature for each RFM dimension in which the consumers resembling known non-buyers are scored negatively, and the consumers more similar to known buyers are scored positively. This was an important step in their method when using logistic regression because the model assumes absence of multicollinearity. Intuitively, this penalty function seems reasonable, but the outputted single-value feature lowered the AUC score when compared to the models using two features for similarity to known buyers and similarity to known non-buyers. This is likely because the two-sided features capture different nuances.

$$S_{RFM}(x_i) = \sum \log(\frac{DC_{ij} + 1}{DS_{ij} + 1}) \tag{4.5}$$

in which

DC_{ij} = the deviation from the consumer x_i to the average known non-buyers

DS_{ij} = the deviation from the consumer x_i to the average known buyers

4.5 Feature Selection

After calculating similarity measures and extracting features, there are 30 features, of which 26 are derived from transactional data. The last four features are from structured data, which required little preprocessing. All features are enlisted in Table 4.6 and will form the basis for the feature selection for each model in Chapter 5.

Table 4.6: Feature collection.

Subset	Name	Description
SD	Alder_Y	The age of the consumer
	Kjonn_cd	The (categorical) gender of the consumer
	Utlaan_Sum_Amt	The consumer's total loan
	Innskudd_Sum_Amt	The consumer's total deposit
PSN	S_icf	The BeSim-score using ICF times E
RFM	S_r_kb	Recency BeSim-score (from KB mean)
	S_r_knb	Recency BeSim-score (from KNB mean)
	S_f_kb	Frequency BeSim-score (from KB mean)
	S_f_knb	Frequency BeSim-score (from KNB mean)
	S_m_kb	Monetary Value BeSim-score (KB mean)
	S_m_knb	Monetary Value BeSim-score (KNB mean)
TF	E	The aggregated sum of empiric probability
	BiDegree	The degree centrality for every consumer
	Tr_Count	The amount of transactions
	Cat_1	Entertainment (% of Tr_Count)
	Cat_2	Groceries (% of Tr_Count)
	Cat_3	Transport (% of Tr_Count)
	Cat_4	Restaurant (% of Tr_Count)
	Cat_5	ATM (% of Tr_Count)
	Cat_6	Children (% of Tr_Count)
	Cat_7	Home (% of Tr_Count)
	Cat_8	Cosmetics (% of Tr_Count)
	Cat_9	Financial Services (% of Tr_Count)
	Cat_10	Transfer (% of Tr_Count)
	Cat_11	Health (% of Tr_Count)
	Cat_12	Travel (% of Tr_Count)
	Cat_13	Professional Services (% of Tr_Count)
	Cat_14	Taxes (% of Tr_Count)
	Cat_15	Services (% of Tr_Count)
	Cat_16	Pets (% of Tr_Count)

Figure 4.5 shows the correlation between all features using the Pearson Correlation Coefficient ρ [24]. The correlation value lies in the range of [-1, 1], in which a correlation of -1.0 indicates a perfect negative correlation, and a correlation of 1.0 indicates a perfect positive correlation. The feature *Produktbredde (Product Range)* was omitted as it had a medium positive correlation with the target variable *has_purchased* of $\rho = 0.41$. An interesting observation from this correlation matrix is the high correlation between the different similarity features derived from transaction data (located in the second quadrant). It is well known that empirical estimation of a linear model, be it linear or logistic, may suffer from multicollinearity. On the other hand, tree-based ensemble models can treat correlated covariates (features) without any problems [25]. The next chapter will appoint a classification model and define different models using a selection of these features to measure their predictive quality.

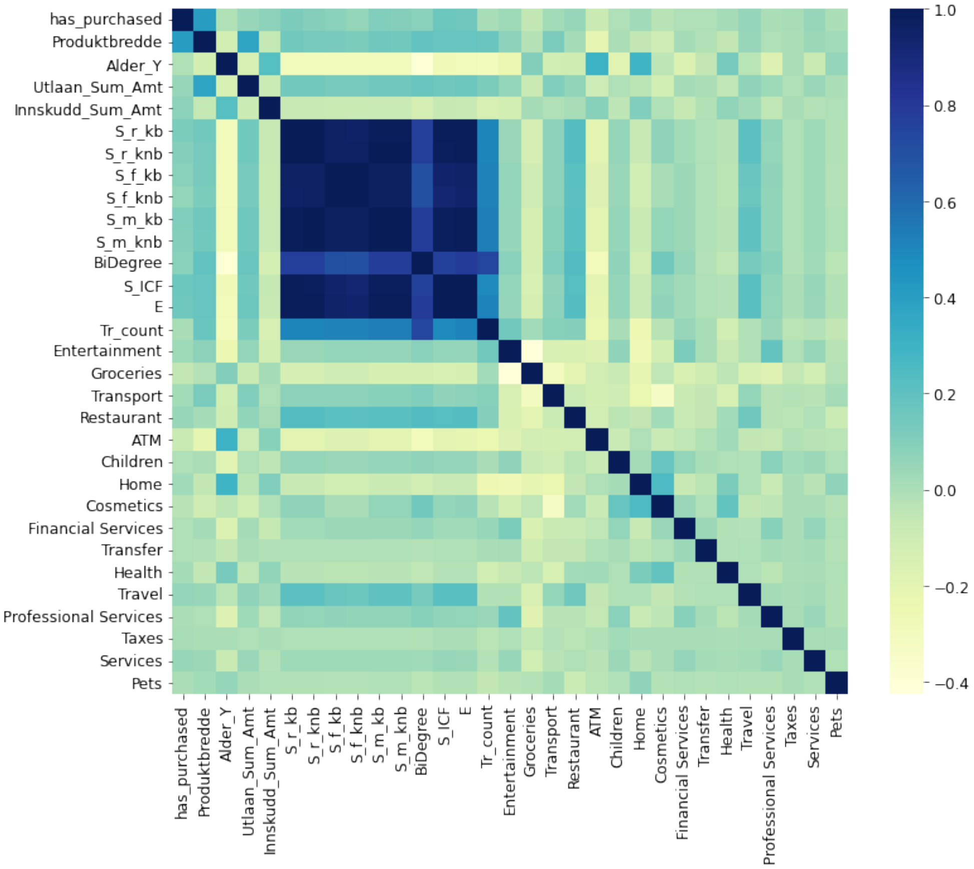


Figure 4.5: Feature correlation matrix.

Chapter 5

Experiments

This chapter describes the data used to define the models, the experimental setup, and the criteria for evaluating the models.

5.1 The Data

This section briefly describes the data. The first data set is a debit transaction log, referred to as *(fine-grained) transaction data*. The second data set contains structured and socio-demographic information about the consumers, referred to as *structured data* (SD).

5.1.1 Fine-grained Transaction Data

This data set is a log of debit transactions spanning two years. Approximately 100,000 unique customers and 1,370,000 unique merchants can be observed from the 110,000,000 transactions. However, many merchants identified from the transactions yield no value. These merchants were removed with respect to the two axioms presented in Section 4.3.2, reducing the complexity of the PSN and the memory required. The format of the transactions is shown in Table 5.1.

Table 5.1: Format of fine-grained transaction data.

CustomerID	MerchantID	Date	Amount	Category
100001	102	2019-12-14	989.0	Transport
100002	105	2019-12-15	438.6	Groceries
100003	103	2019-12-15	438.6	Entertainment
100004	101	2019-12-17	32.0	Transport

All values are fictional.

5.1.2 Structured Data

Initially, this data set had 1,700,000 entries, but there are only 100,000 consumers because each consumer has one entry per month. This format enables more advanced analysis of the consumers, as the changes in customer data over time are observable. However, such analysis is out of scope for this study. This research actively ignores the temporal dimension of the data by reducing the number of entries to one per customer. This was effectively done by keeping the most recently observed information about each customer. The original format of the structured data can be seen in Table 5.2.

Table 5.2: Format of structured data.

CustomerID	Period	Deposit Sum	Loan Sum	Age	Sex	Fund
100001	2020M1	34962.0	2653098.0	42	M	0
100001	2020M2	31029.0	2638238.0	42	M	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
200000	2021M11	14923.0	0.0	23	F	1

All values are fictional.

5.1.3 Target Variables

Two target variables were selected from the bank product offerings to perform the method described in Chapter 4. All potential target variables are listed in Table 5.3, which may be extracted from the structured data. The consumers who purchased the products are the *known buyers*, and the consumers who did not are the *known non-buyers*. These variables are binary indicators that receive the value of 1 if a consumer has purchased the product and 0 otherwise. Predicting the value of these variables will be the predictive task for the models in this experiment and thus be the basis for evaluating the five predictive models.

At first, the target variables were extracted by selecting the consumers with an observed change for the product variable during the 24 months. However, this sampling method granted very few positive target labels and was discarded in favor of a more straightforward extraction method. The current targets use the most recent entry for each consumer as their final target label. This changed the predictive task to predicting which consumers *have* purchased the offered product, compared to predicting which consumers *will* purchase the offered product. As mentioned in Section 5.1.2, this would effectively discard most of the data for each customer and ignore the time dimension.

Product 1

The first target variable is a savings fund. It was selected to measure the performance on a reasonably balanced data set. It has 44% known buyers, the highest percentage of all the product offerings. The savings fund will be referred to as *Product 1*

Product 2

The second target variable is a disability insurance, with 17% known buyers. It was selected to measure the performance of an imbalanced data set. The disability insurance will be referred to as *Product 2*.

Table 5.3: Potential target variables.

Feature	Incidences
Savings Fund	**44%**
Vehicle Insurance	33%
Property Insurance	31%
Travel Insurance	28%
Life Insurance	25%
Disability Insurance	**17%**
Child Insurance	8%

Selected target variables are boldfaced.

5.1.4 Data Preparation

Low-quality data leads to low-quality results. Characteristics of low-quality data may be that it is incomplete, noisy, and inconsistent. An important step before conducting any experiment is analyzing the data and processing it accordingly. All data preparation was executed before any further implementation.

The fine-grained transaction data included 110,000,000 transactions. Transactions with missing values were deemed incomplete and removed. The fine-grained transaction data required no further preprocessing. Irrelevant merchants and their received transactions were also removed.

Analysis of the structured data revealed that this data had a greater number of missing values. The values for the target variables were not registered in the final month and thus discarded as missing values. A small number of the consumers were missing information for most of their variables in an inconsistent pattern. These consumers were dropped, as they only accounted for a couple of hundred consumers. As the final step of preparation, all consumers that could not be observed in the transaction log were also filtered out because it would be impossible to derive their features. About 2000 consumers were removed on this basis.

5.2 Models

Four feature subsets are defined to measure the effect of using fine-grained data in predictive models. These feature subsets are summarized in Table 5.4. A more detailed description of these subsets are presented in Table 4.6.

Table 5.4: Summary of the four features sets.

Feature Set	Description
SD	Structured Data
PSN	Behvaioral Similarity (BeSim) scores
RFM	BeSim scores for Recency, Frequency and Monetary Value
TF	Transaction Features

These subsets are the defining features of five different models. The first model *SD*, uses no features from the fine-grained transactions. This model will be the baseline for comparison with the other models to evaluate the influence of fine-grained transaction data for predictive accuracy. The models PSN and RFM both use *exclusively* features derived from transaction data but are compared to evaluate the significance of the different similarity features. The two remaining models use a combination of structured and transaction data features to assess the predictive accuracy using complementary features. All five models and the subsets that define them are listed in Table 5.5.

Table 5.5: Models defined by feature subsets.

Model	Feature Subsets	Structured Data	Transaction Data
SD	{SD}	✓	
PSN	{PSN + TF}		✓
RFM	{RFM + TF}		✓
PSN+SD	{SD + PSN + TF}	✓	✓
RFM+SD	{SD + RFM + TF}	✓	✓

Classification Model

This experiment is executed by a binary classifier using the gradient boosting algorithm. The hyperparameters are tuned manually. The predictive task is to classify consumers as buyers or non-buyers.

Results from related work indicate that random forests and gradient boosting are the best models in terms of predictive performance. Both learning algorithms were explored in this study's initial stages, but the gradient boosting models consistently outperformed the random forest models. The comparison of the models for a subset of features is included in Table A.2 in Appendix A, and the results are supported by the findings of Muñoz-Cancino et al. [20]. Another benefit is that the retail bank of which this thesis is written in cooperation with also operate gradient boosting in much of their work, making it more applicable.

5.3 Experimental Setup

The setup for this experiment is similar to that of previous research. The data is split into training and testing sets. For this study, the data is divided into 80% for training the models and the remaining 20% for testing the models. The models are trained using stratified cross-validation with ten splits, such that each set contain approximately the same percentage of samples for the target variables.

The experiment for Product 1 and Product 2 are carried out sequentially. This will be the basis for assessing how the models with features derived from the proposed methodology perform in different settings of balanced and imbalanced data sets.

The behavioral similarity scores for the consumers in the test set *must* be estimated using the information from the training set to avoid data leakage. That is, the test set must be treated as an isolated and unobserved state before prediction. The similarity features for the consumers in the test set are thereby calculated using only the merchants observed in the training set, and for the RFM similarity features using observed averages from the consumers in the training set.

5.4 Environment

All experiments in this study were conducted on a virtual machine set up by the bank. The hardware specifications for the machine is shown in Table 5.6. Processing of data and implementation of models is done in Jupyter[1] notebooks with Python 3, utilizing scikit-learn[2], LightGBM[3] and Pandas[4].

Table 5.6: Hardware specifications.

Memory	64GB
CPU	Intel(R) Xeon(R) Silver 4114 CPU
CPU Cores	4
CPU Frequency	2.19GHz
OS	Windows 10 Enterprise

[1] https://jupyter.org/

[2] https://scikit-learn.org/stable/

[3] https://lightgbm.readthedocs.io/en/latest/

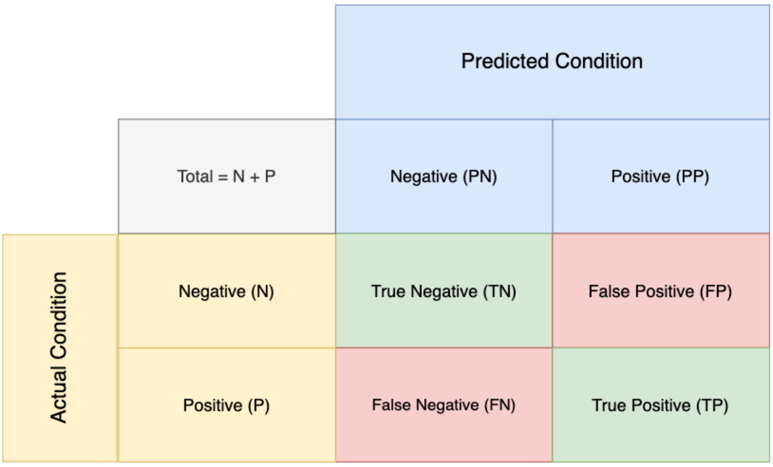
[4] https://pandas.pydata.org/

5.5 Evaluation Criteria

This section presents the evaluation metrics to be used to interpret and evaluate each model's performance. These metrics capture different aspects of performance and give varied insights into the strengths and weaknesses of the different models. The results in Chapter 6 will primarily be discussed in terms of these evaluation metrics. The metrics are elaborated beforehand to help the reader understand the metrics before the results from the experiments are presented.

Confusion Matrix

A confusion matrix is a table that makes for an easier interpretation of the performance of a predictive model. Several performance metrics may be derived from a confusion matrix. A general template for a confusion matrix is illustrated in Figure 5.1



		Predicted Condition	
	Total = N + P	Negative (PN)	Positive (PP)
Actual Condition	Negative (N)	True Negative (TN)	False Positive (FP)
	Positive (P)	False Negative (FN)	True Positive (TP)

Figure 5.1: Confusion matrix template.

Performance Metrics

The definitions presented below are performance metrics commonly used in machine learning. These metrics form the basis for more advanced metrics and are derivable from confusion matrices.

Definition 5.5.1. Sensitivity, or true positive rate (TPR), is the fraction of positives that were correctly predicted to be positive. Sensitivity can be considered the probability that a positive prediction is correct.

$Sensitivity = \frac{TP}{P} = \frac{TP}{TP+FN}$.

Definition 5.5.2. Specificity, or true negative rate (TNR), is the fraction of negatives that were correctly predicted to be negative.

$Specificity = \frac{TN}{N} = \frac{TN}{TN+FP}$

Definition 5.5.3. Precision is the fraction of predicted positives that were correctly predicted to be positive.

$Precision = \frac{TP}{TP+FP}$

Definition 5.5.4. False negative rate (FNR) is a measure of what proportion of the data was incorrectly classified as negatives.

$FNR = \frac{FN}{P} = \frac{FN}{FN+TP}$

Definition 5.5.5. False positive rate (FPR) is a measure of what proportion of the data was incorrectly classified as positives.

$FPR = \frac{FP}{N} = \frac{FP}{FP+TN}$

Definition 5.5.6. Accuracy is the proportion of correct predictions among all predictions made.

$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

ROC Curve

Receiver Operating Characteristic (ROC) is a graphical plot that illustrates the diagnostic ability of a binary classifier system [26]. ROC curves can be created by plotting the TPR against the TNR. Generally, the closer the curve is to the top left corner, the better. However, if the ROC curve is too steep, this may be a sign of overfitting. Figure 5.2 illustrates how a ROC curve may look for a classifier performing better than random guessing.

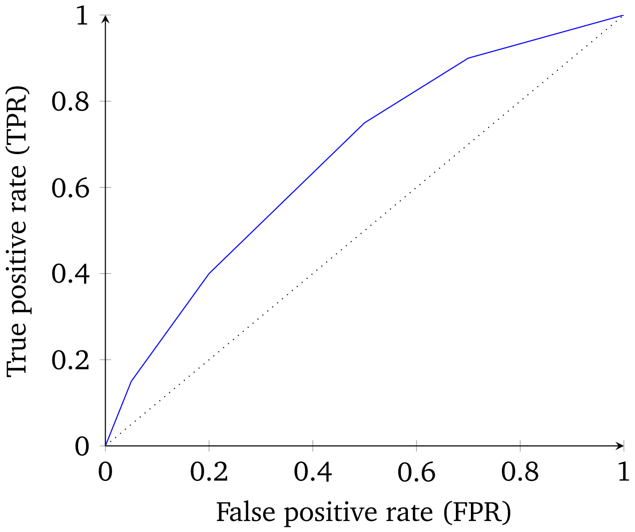


Figure 5.2: Example ROC curve. Dotted line illustrates random guessing.

AUC

The Area Under the ROC Curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. AUC is one of the most widely used single-value evaluation metrics to measure the performance of binary classifiers [27]. The AUC score always lies in the interval between 0 and 1. The higher the value, the better the model predicts 0 classes to be 0 and 1 classes to be 1. An AUC score of 0.5 indicates that the classifier is no better than random guessing, while an AUC score of 1 represents a perfect model.

Lift

Lift is most often used in data mining and association rule learning to measure the performance of a targeting model compared against a random choice model [28]. Mathematically lift is defined by Equation (5.1). For example, if a data set initially contains 5% positive instances, but the model can identify a segment of the data with 25% positive instances, then the achieved lift is 5 ($= \frac{25\%}{5\%}$).

$$Lift = \frac{P(A \cap B)}{P(A) \times P(B)} \tag{5.1}$$

Lift will be used in the evaluation by looking at the lift rates for the consumers assigned with the highest probability of being known buyers. The lift rates are considered for the top 1%, 5%, and 10%.

Feature Importance

SHapley Additive exPlanations (SHAP) [29] is commonly used to discover the relative importance of each feature when working with non-linear models. SHAP calculates the importance of each feature by comparing the prediction made by the model with and without the feature. SHAP values make it easier to explain and interpret machine learning models by visualizing the importance of each individual feature. This is an important metric for the explainability requirement for managerial approval discussed in Section 2.3.5 in Chapter 2. The SHAP algorithm that evaluates feature importance for the experimental models is called Tree SHAP and is a fast and exact method to estimate SHAP values for tree models and ensembles of trees under several different possible assumptions about feature dependence [5].

[5] https://shap.readthedocs.io/en/latest/generated/shap.explainers.Tree.html

Chapter 6

Results and Discussion

This chapter presents the experimental results from Product 1 and Product 2. The results are interpreted using the evaluation criteria defined in Section 5.5 and subsequently discussed in more detail. Finally, the overall performance is summarized and compared with the results of related work.

6.1 AUC and Lift

The AUC-scores from the experimental results are listed in Table 6.1, along with their corresponding ROC curves in Figure 6.1 and Figure 6.2. The lift rates from the experimental results are listed in Table 6.2 and Table 6.3. The results from Table 6.1 show that all five models are better than the random guessing model with AUC scores above 0.5, suggesting that both structured data and transaction data have inherent predictive quality. In both experiments from Product 1 and Product 2, it is apparent that the best performing model is the RFM+SD model compromising features from both structured and transaction data.

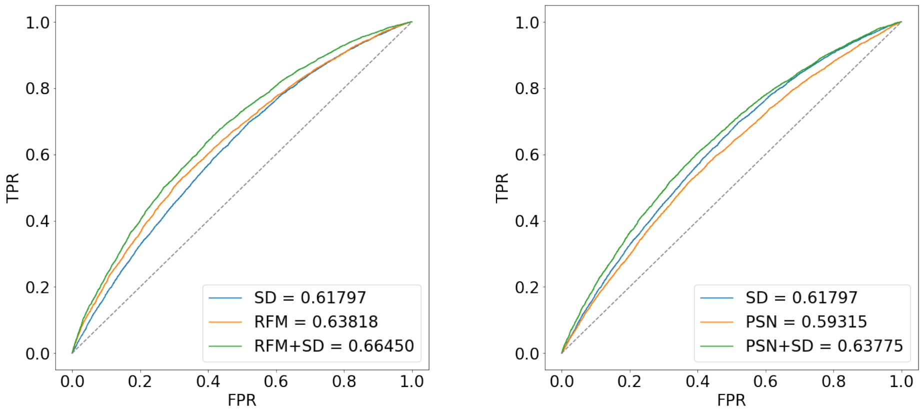
Product 1

The results from Table 6.1 are striking regarding the influence of transaction data. The best-performing model for Product 1 is the RFM+SD model. There is a noticeably increase in performance for the PSN+SD model as well, compared to their stand-alone PSN and SD models. This indicates that the models using structured data and the models using transaction data capture complementary information. Although the PSN model being outperformed by the baseline SD model would suggest that structured data possess more predictive quality, the best performing stand-alone model is the RFM model. The RFM model even marginally outperforms the PSN+SD model.

Table 6.1: AUC scores.

Target	SD	PSN	RFM	PSN+SD	RFM+SD
Product 1	0.61797	0.59315	0.63818	0.63775	**0.66450**
Product 2	0.76577	0.60629	0.67861	0.75230	**0.76748**

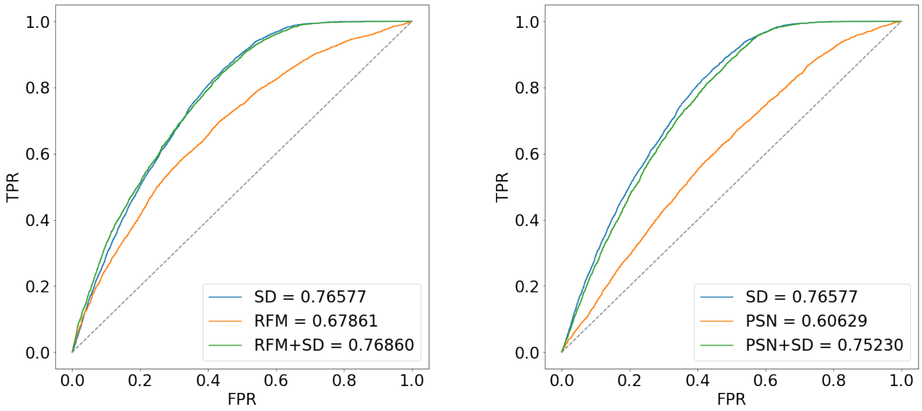
The highest scores are denoted in bold



(a) SD, RFM and RFM+SD.

(b) SD, PSN and PSN+SD.

Figure 6.1: ROC curves Product 1.



(a) SD, RFM and RFM+SD.

(b) SD, PSN and PSN+SD.

Figure 6.2: ROC curves Product 2.

Table 6.2: Lift for Product 1.

Model	Lift 1%	Lift 5%	Lift 10%
SD	1.37669	1.35124	1.34314
PSN	1.33042	1.31653	1.30149
PSN+SD	1.63121	1.49238	1.44032
RFM	1.66591	1.57568	1.46924
RFM+SD	**1.74689**	**1.61964**	**1.53981**

The highest scores are denoted in bold.

Table 6.3: Lift for Product 2.

Model	Lift 1%	Lift 5%	Lift 10%
SD	2.15224	2.32088	2.24316
PSN	1.65104	1.41383	1.35119
PSN+SD	2.12276	2.15010	2.09891
RFM	2.65345	2.35622	2.06064
RFM+SD	**2.77138**	**2.62718**	**2.40212**

The highest scores are denoted in bold.

The lift rates from the experiment in Table 6.2 strongly suggest that structured data combined with transaction data possess more predictive quality. The highest possible lift score for Product 1 is 2.27, of which 44% are positive incidences ($2.27 = \frac{1}{0.44}$). The models incorporating `RFM` similarity features are superior across all percentiles, with the RFM+SD model performing best.

Product 2

The results in Table 6.1 vary in consistency with the results from Product 1. The best performing model is still the RFM+SD model, however the best stand-alone model is the SD model, and the difference in the AUC score is insignificant. The RFM model is again better than the PSN model, but both are significantly worse than the SD model. An interesting result is that the PSN+SD model performs worse than the stand-alone SD model.

The lift rates from the experiment on Product 2 are listed in Table 6.3. The highest possible lift score for Product 2 is 5.88, of which 17% are positive incidences ($5.88 = \frac{1}{0.17}$). The lift rates for Product 2 tell a different story than the AUC scores. Even though the PSN and PSN+SD models are consistently worse than the SD model, the best-performing models incorporate RFM similarity features from transaction data. The stand-alone RFM model performs better than the SD model for all percentiles except the top 10%, while the RFM+SD model is consistently better across all percentiles.

Summary

Although the AUC scores and lift rates indicate an enhanced effect using transaction data, the results indicate that the models strongly depend on balanced data sets. The results must be further inspected and evaluated using other metrics that capture different insights.

6.2 More Performance Metrics

This section interprets model performance with performance metrics derived from the confusion matrices listed in Appendix A.2. These metrics may provide more insights into the performances of the models.

Product 1

Table 6.4 lists multiple performance metrics for all models from Product 1. The RFM+SD model has the best overall performance. It performs best in terms of precision and accuracy and second-best in terms of sensitivity. The best stand-alone model is yet again the RFM model. It achieves the second-best precision and specificity, and in terms of accuracy, it performs practically equal to the PSN+SD model. The RFM model is significantly better than the PSN model for all metrics besides specificity, in which the PSN model scores best out of all the models. However, high specificity by itself yields little value. A model that predicts all instances to be negatives will achieve a specificity of 100%. The RFM and PSN models both have higher precision and specificity than the SD model. This is a strong indicator that the features derived from fine-grained transaction data capture more predictive quality compared to the traditional features from structured data, at least for balanced data sets.

Table 6.4: Performance metrics of all models for Product 1.

Model	Precision	Sensitivity	Specificity	Accuracy
SD	54.7%	**40.6%**	73.4%	59.0%
RFM	60.9%	27.3%	86.2%	60.2%
PSN	57.6%	12.1%	**93.0%**	57.3%
PSN+SD	60.0%	29.8%	84.3%	60.3%
RFM+SD	**62.8%**	33.8%	84.2%	**62.0%**

The best scores are denoted in bold.

Product 2

Table 6.5 reports the same performance metrics from the experiment on Product 2. The model performances are not consistent with Product 1. Interestingly, both the SD and PSN models have a precision of 0%. The confusion matrices in Appendix A.2 reveal that this is because the SD model does not predict any consumer to be a buyer. The PSN model only makes one positive prediction, which was an incorrect prediction. Despite the lack of precision, the SD model and the PSN model still achieve the highest accuracy, which is approximately the same across all the models. This shows that accuracy can be a misleading performance metric in the case of imbalanced data sets and that high AUC scores and lift rates do not necessarily mean the model is useful.

For Product 2, the RFM+SD model is still the model with the highest precision, followed by the RFM model. However, the low sensitivity for both models reveals that they make very few positive predictions. The sensitivity is still significantly higher compared to the SD and PSN models, which again shows that the `RFM` similarity features from transaction data are much more capable of identifying buyers for an imbalanced data set.

Table 6.5: Performance metrics of all models for Product 2.

Model	Precision	Sensitivity	Specificity	Accuracy
SD	0%	0%	**100%**	**82.7%**
RFM	44.3%	3.1%	99.2%	82.6%
PSN	0%	0%	99.9%	**82.7%**
PSN+SD	36.1%	0.4%	82.7%	82.6%
RFM+SD	**49.1%**	**4.8%**	99.0%	**82.7%**

The highest best are denoted in bold.

Summary

From a business perspective, one might say that precision is more important than specificity because it indicates how capable a model is at identifying the actual buyers. In general, it is believed to be more costly to miss the products of interests than the other way around. Therefore, precision may be a more relevant metric. By this argument, the RFM+SD model is significantly better than the SD model for the experiment on Product 2, despite little difference in AUC score.

The results using performance metrics from confusion matrices also indicate that the similarity features for the RFM model are superior to the similarity features for the PSN model. The `RFM` feature subset possess more predictive quality for both fairly balanced and imbalanced data sets. Analyzing the feature importance for the PSN+SD model and the RFM+SD model may grant further insights into the influence of each feature subset from the fine-grained transaction data.

6.3 SHAP: Feature Importance

The analysis and evaluation of feature importance are primarily centered around the experiment on Product 1 in which both models using the feature subsets PSN and RFM performed the best. However, the best performing models on Product 2 from Section 6.2 are also evaluated and compared.

Product 1

The most important features for the RFM+SD model and the PSN+SD model from Product 1 are listed respectively in Figure 6.3 and Figure 6.6 in terms of Shapley values. The summary plots explain the effect each feature has on the target prediction. That is, to what degree a feature influences the prediction of being a buyer. The most important feature(s) for both models are derived from the pseudo-social network. The fact that features from transaction data outperform the structured data features in terms of feature importance is auspicious and emphasizes the potential in the featurization of fine-grained transaction data.

RFM+SD Model

The empiric probability feature E in Figure 6.3 is the feature with the most significant impact on the predictions by the RFM+SD model. As the summary plot indicates, a higher empiric probability value reflects positively on the likelihood of being a buyer. However, the impact of the feature E does not continuously increase in Figure 6.4 although its feature value may increase. It reaches a plateau at which its impact is constant.

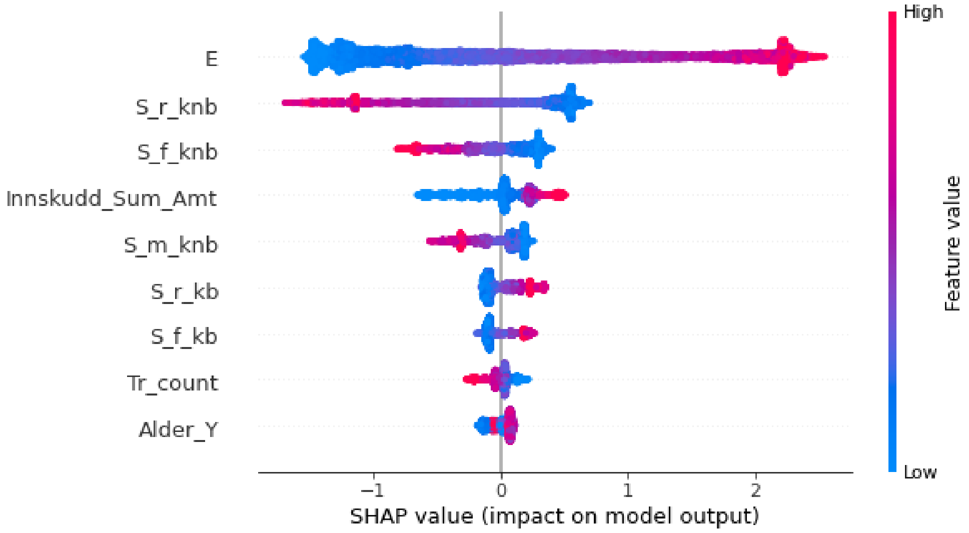


Figure 6.3: Most important features for the RFM+SD model from Product 1.

The RFM similarity features are also important for the RFM+SD model. The SHAP values of the similarity features *S_r_kb* and *S_r_knb* in Figure 6.3 illustrate that the decision of using two features for each RFM dimension instead of combining them to a singular feature is indeed appropriate. The feature correlation matrix in Chapter 4 showed that the RFM similarity features were strongly correlated. This is substantiated by Figure 6.5, in which the inverse shape of the SHAP values for *S_r_kb* resembles the shape of the SHAP values for *S_r_knb*. This indicates that they capture related information, which is respectively the similarity and dissimilarity to known buyers. The *S_r_knb* feature negatively impacts the model output when the feature value increases, suggesting that being similar to a known non-buyer decreases the likelihood of being a buyer. By the same argument, the *S_r_kb* feature indicates that being more similar to the known buyers increases the likelihood of being a buyer.

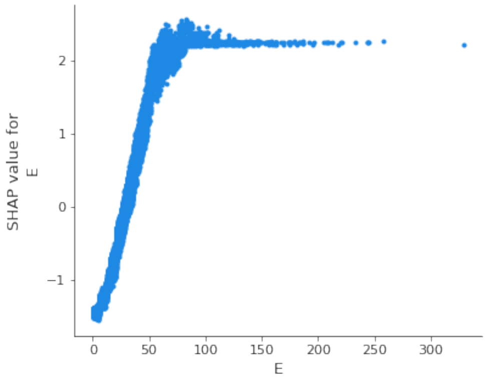
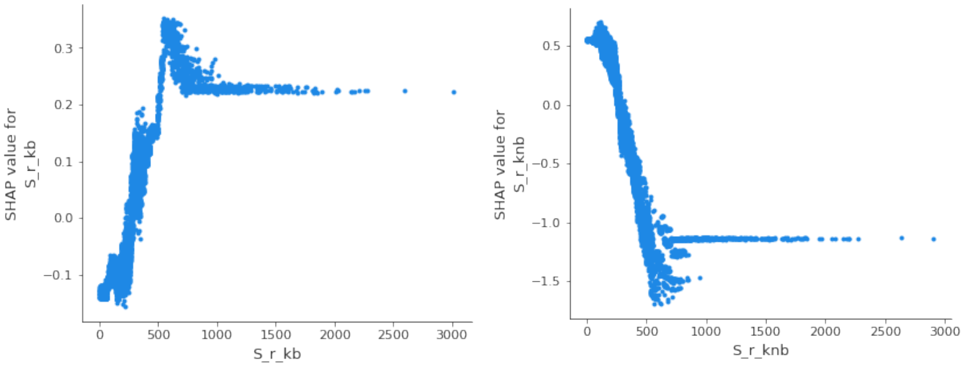


Figure 6.4: The most important feature for RFM+SD from Product 1.



(a) S_r_kb.

(b) S_r_knb.

Figure 6.5: Recency similarity features for RFM+SD from Product 1.

PSN+SD Model

Figure 6.6 shows that the transaction data features are among the most important features for the PSN+SD model. The *S_ICF* is the most important feature, which combines the empiric probability and inverse frequency of merchants. It is not a big surprise that its shape in the dependence plot in Figure 6.7a resembles that of *E* for RFM+SD in Figure 6.4. The importance of *E* is also much less important for the PSN+SD model, which is likely because its predictive quality is already included in *S_ICF*. It seems that the RFM similarity features combined with *E* possess more predictive quality than the *S_ICF* feature alone when comparing the AUC scores of the RFM+SD and PSN+SD models.

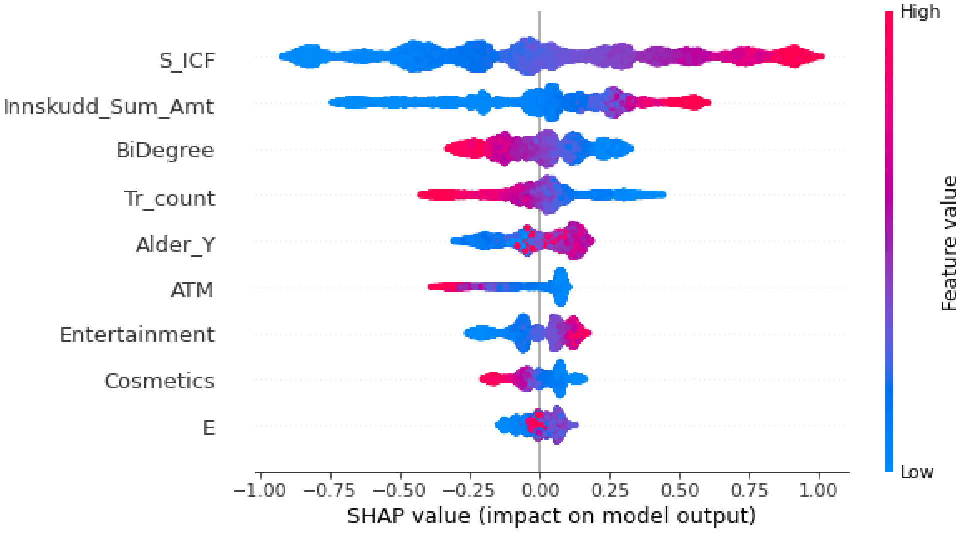
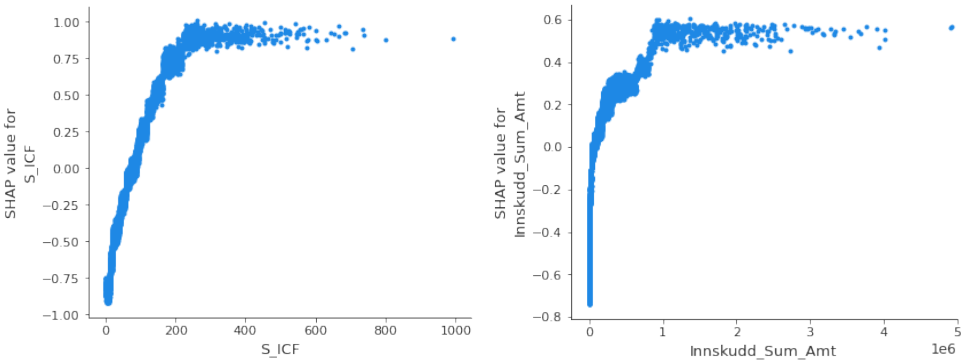


Figure 6.6: Most important features for the PSN+SD model from Product 1.

The most important feature from `SD` for both models is *Innskudd_Sum_Amt*. Figure 6.7b shows that in the case of the PSN+SD model, consumers who have deposited about 1 million NOK are more likely to be buyers.

The similarity feature based on degree centrality *BiDegree* is also more influential in the PSN+SD model than in the RFM+SD model. Both models also have *Tr_count* among their most important features.



(a) Most important feature for PSN+SD. **(b)** 2nd most important feature for PSN+SD.

Figure 6.7: The two most important features for PSN+SD model from Product 1.

Product 2

The results from the confusion matrices reveal that the best performing models on Product 2 are the models using RFM similarity features. This section, therefore, evaluates the feature importance of the RFM and RFM+SD model.

RFM+SD Model

Figure 6.8 shows the feature importance for the RFM+SD model from Product 2. The RFM similarity features dominate importance as they do for Product 1. However, the most important feature stems from the SD feature subset, namely *Alder_Y*. The dependence plot in Figure 6.9 shows the feature's impact as a consumer's age increases. The model evaluates consumers over the age of 50 to be less likely buyers, as the impact on model output changes at this point. This may indicate that Product 2 heavily depends on consumer age or that the model overfits and has a strong bias toward age. The latter is more likely given the fact that the RFM+SD model's predictions of buyers are correct approximately 50% of the time.

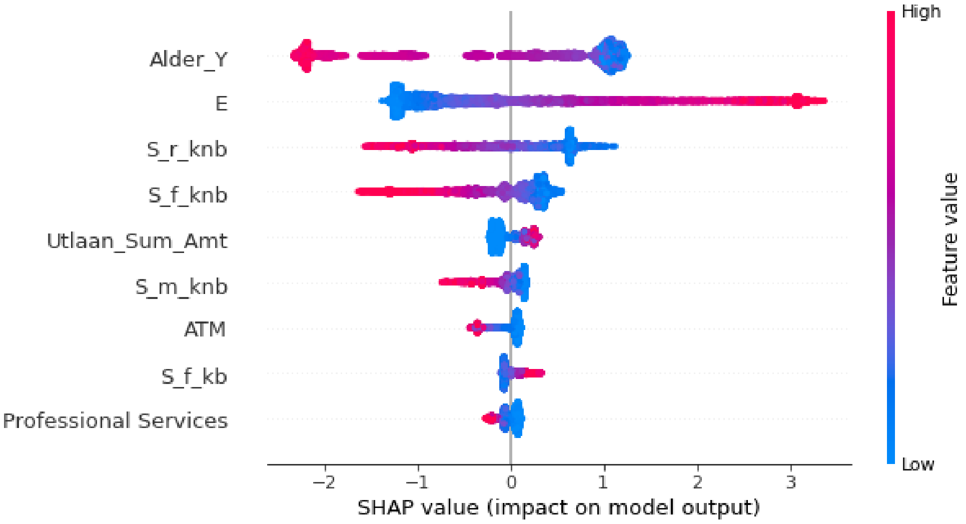


Figure 6.8: Most important features for RFM+SD model from Product 2.

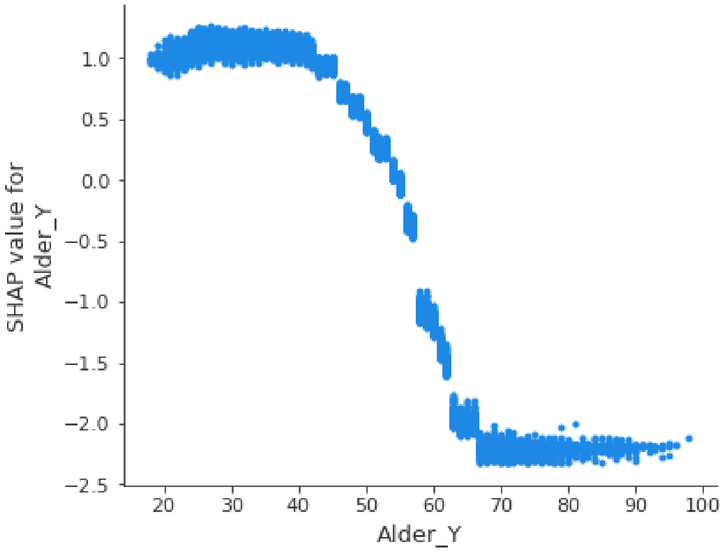


Figure 6.9: Most important feature for RFM+SD model from Product 2.

RFM Model

The feature importance for the RFM model is shown in Figure 6.10. The summary plot resembles the feature importance in Figure 6.3, in which the most important feature is E. For Product 2, however, the feature E is more influential in the RFM model than in the RFM+SD model. Nevertheless, the precision of the RFM model is lower than the RFM+SD model, which indicates that the stand-alone RFM model is indeed better when combined with features from SD.

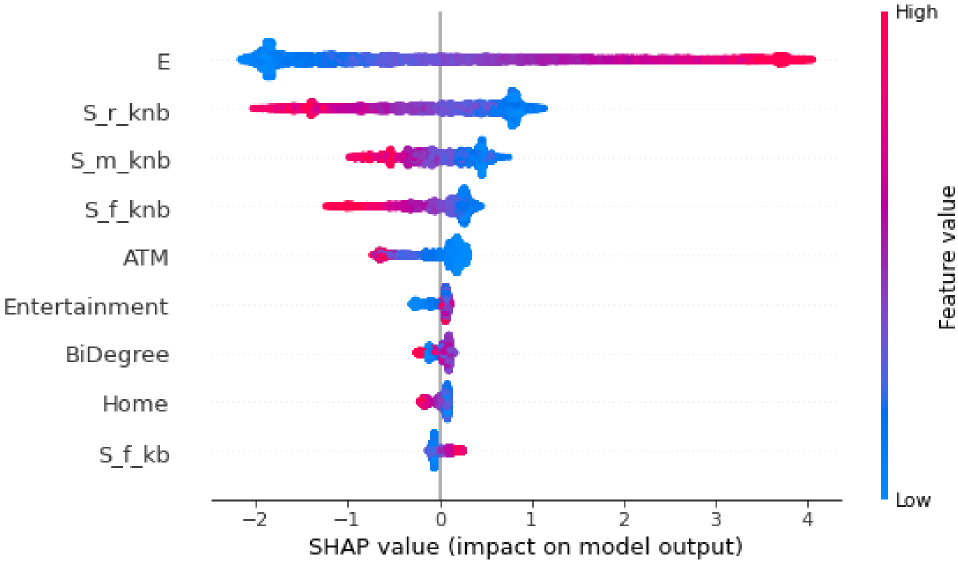


Figure 6.10: Most important features for RFM model from Product 2.

6.4 Overall Performance

The results presented in this chapter show that the RFM+SD model is superior to all the other models when it comes to predictive performance.

The results from the experiment on Product 1 imply that predictive models using structured data experience an enhanced effect when combined with transaction data. Martens et al. [11] discovered similar results from their experiments, in which their PSN+SD model compromising features from both data sets yielded a higher AUC score than the stand-alone models PSN and SD. In addition, their stand-alone PSN model using S_{ICF} as a similarity feature performed worse than the SD model, which is also consistent with the findings in this research. Another important notice from their study is that they use 10% of the test data to estimate the similarity features for the remaining consumers in the test set. This is problematic because the measure incorporates information about the targets it is trying to predict, thus potentially causing data leakage. The test set must remain unseen before the prediction takes place to give a more realistic performance result. The different data and testing procedures makes a direct comparison of AUC scores and lift rates indifferent.

The stand-alone model from this study using RFM similarity features outperforms the SD model and the PSN+SD model for Product 1. This result is profound in terms of applicability. It means that companies that do not possess adequate structured data for all consumers may also apply predictive modeling if they possess transaction data for those missing consumers. Another interesting result regarding the RFM similarity features is that they provide better results when they are not combined to single-value features. This is exemplified by the summary plots of feature importance in Section 6.3. This method deviates from the proposed methodology by Caigny et al. [12], but the results in terms of predictive performance are clear. Figure A.4 in Appendix A also shows that combining the features in the function they proposed yield lower AUC scores and lift rates. A major benefit of using ensemble methods like gradient boosting compared to logistic regression is that ensemble methods are robust to multicollinearity problems [30]. Logistic regression on the other hand assumes absence of multicollinearity, which is an important case for single-valued RFM features.

Caigny et al. [12] and Martens et al. [11] also presented several ways of scaling the pseudo-social networks using both ICF and S_ICF as scaling factors. Despite the difference in methods, the results from Caigny et al. [12] also suggest that the RFM+SD model is generally superior to all the other models. These results coincide with the indication that the featurization phase is of great importance in harnessing predictive quality from transaction data.

The AUC scores from the experiments on Product 2 indicate that the difference between models using only structured data and structured data in combination with transaction data is much less significant. This opposes the results of Martens et al. [11] in which the combined models are significantly better across both of their target products. However, in line with their results, the experiments on Product 1 and Product 2 show an increase in AUC scores when the imbalance of the data sets increases. The same adheres to the lift rates, which experience an increase on Product 2. Closer inspection using more informative performance metrics reveals that these results are likely because the models are biased toward predicting non-buyers for the imbalanced data sets. These results is an example of how AUC can be a misleading measure of the performance of predictive models [31].

The problem with imbalanced data sets for the models incorporating features from transaction data is the size of the projected pseudo-social networks. The second axiom in Chapter 4 says to remove merchants that do not have connecting known buyers. For Product 2, of which 17% of the consumers are known buyers, it means that the pseudo-social network is notably smaller than for Product 1. Thus, the data basis for deriving the similarity features from the pseudo-social network is also substantially less. In essence, although the summary plots for both RFM and RFM+SD from Product 2 suggest that the RFM similarity features are important, their predictive quality may be ineffectual compared to the RFM similarity features derived from the pseudo-social network from Product 1.

Chapter 7

Conclusions and Future Work

Reports by McKinsey show that commercial banks could see reductions in revenue between 10% to 40% by 2025 if they fail to react accordingly to increasing competition [32]. Predictive modeling is increasingly becoming a part of businesses' targeting methods because staying a step ahead of consumer trends is essential to maintaining a competitive advantage. Introducing new data sources could potentially increase predictive performance.

This research incorporates fine-grained transaction data as a new data source. By inferring a pseudo-social network from the transaction data, it is possible to extract features that improve predictive performance. Higher predictive performance facilitates more accurate targeting and may further assist marketing decision-making.

7.1 Conclusions

This study set out to explore the domain and existing literature before proposing a new methodology for pseudo-social network targeting that extends upon the approaches in previous research. Multiple features were derived from the pseudo-social network and divided into feature subsets. The four subsets laid the foundation of the five models used in the experiments to measure the predictive quality of the features derived from the pseudo-social network compared to the features from structured data. The models were tested on two data sets to measure how the models adapt to different settings, one being a fairly balanced data set and the other an imbalanced data set. Finally, the results were presented, evaluated, and discussed.

The two research questions raised in the introduction defined the scope of this thesis. The following answers are meant to complement the findings from the experimental results in a concise matter.

RQ1 *How may fine-grained transaction data be leveraged to increase the predictive performance of predictive models?*

The proposed methodology in Chapter 4 shows how a pseudo-social network may be constructed from fine-grained transaction data to further derive new features to predictive models. The experimental results show that fine-grained transaction data can significantly increase the overall predictive performance. The best-performing models incorporate features from structured data combined with features from the pseudo-social network. However, the effect of including different features from the pseudo-social network also shows that the increase in predictive performance is dependent on the predictive quality of the extracted features.

RQ2 *What features from pseudo-social networks help increase the predictive performance for predictive models?*

This study primarily explores two methods of deriving similarity features from a pseudo-social network. The overall performance concludes that both the model using the `PSN` feature subset and the model using the `RFM` feature subset help increase the predictive performance when combined with structured data. The PSN models are, however, outperformed by the RFM models. A merit result from the experiments is that the stand-alone RFM model is better than both the PSN and PSN+SD models. This shows that the similarity features in the RFM models capture more predictive quality from the pseudo-social network than the similarity feature in the PSN models.

7.2 Future Work

This section presents the limitations of this research and points to new directions in which future work could yield better results and new insights into pseudo-social network targeting as a method in predictive modeling.

The experimental results show the significance of adding a new data source to the predictive model pipeline. While the results indicate a boost in predictive performance, they are limited to the predictive quality that exists in the two data sources of transaction data and structured data. Extracting features from new data sources such as more descriptive social-demographic data and social network data may further excel the predictive performance. Exploring alternative calculations of RFM similarity features in the proposed methodology may further enhance the predictive performance when using transaction data.

A major contribution of this thesis is the extensive evaluation of pseudo-social network targeting as a method in predictive modeling. Although evaluating the results using more classification performance metrics provides a better picture of predictive performance, it does not reflect the true performance in a real-world setting. This research is limited by using historical data collected by the bank. However, the testing phase should incorporate new data [33]. Deployment of the model in production systems would provide new data and give more insights into how well the model actually fares in real-world applications.

The security protocols enforced by the bank predetermined the software and hardware used in this research. State-of-the-art machine learning methods such as graph convolutional networks [34] could potentially be used for more extensive analysis of the pseudo-social networks and further increase predictive performance. Such methods are computationally expensive and rarely feasible without access to a GPU for more efficient computations.

True social networks are dynamic structures that are likely to change over time. New connections are likely to appear while others disappear. The methodology proposed in this thesis constructs a static pseudo-social network that does not account for the temporal dimension. Some consumers have longer/shorter relationships with the bank at different intervals. Future work may extend the proposed implementation with temporal information to create dynamic networks. Such networks may enable more advanced network analysis and potentially broaden the spectrum of applications for pseudo-social networks.

Appendix A

Additional Material

A.1 Alternative Methods

RFM Similarity Features with Scaling Factors

Table A.1: AUC using scaled RFM features for the RFM+SD model.

	No scaling	ICF Scaling	S_ICF Scaling
AUC	0.66450	0.66211	0.62392

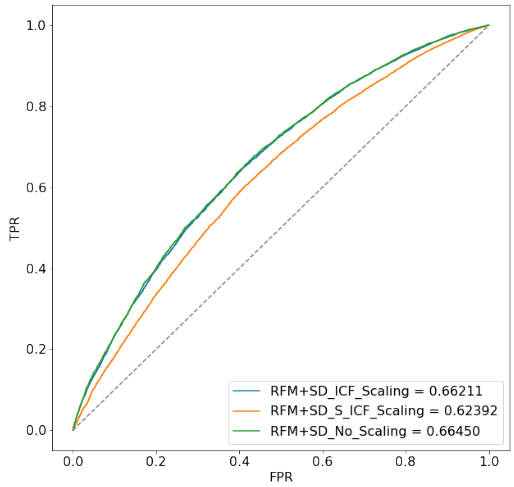
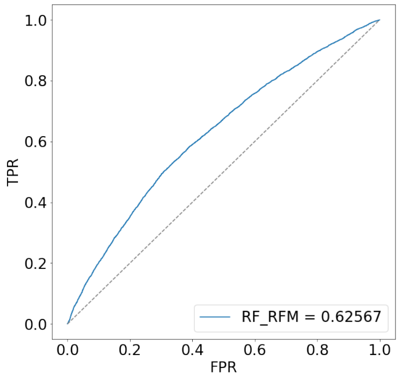


Figure A.1: Comparison of scaling factors for RFM features using the best model.

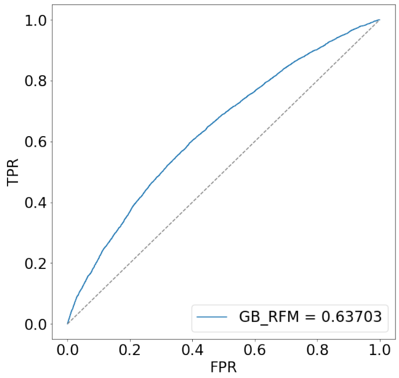
Random Forest vs. Gradient Boosting

Table A.2: Early results for Random Forest and Gradient Boosting.

Classification Algorithm	**AUC**	**Lift 1%**	**Lift 5%**	**Lift 10%**
Random Forest	0.62567	1.37344	1.47022	1.43995
Gradient Boosting	0.63703	1.61732	1.59316	1.48347



(a) Random Forest with RFM-features



(b) Gradient Boosting with RFM-features

Alternative RFM Computation

Table A.3: AUC and Lift using alternative RFM features.

AUC	**Lift 1%**	**Lift 5%**	**Lift 10%**
0.55749	1.34198	1.27026	1.22399

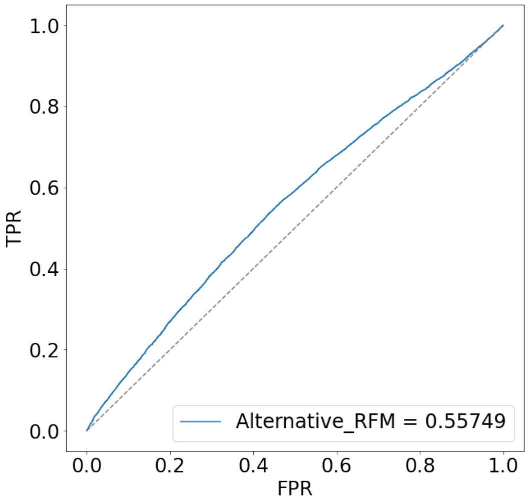


Figure A.3: ROC curve using alternative RFM features.

Using Only Single-Valued RFM Similarity Features

Table A.4: AUC and Lift using single-valued RFM features.

AUC	Lift 1%	Lift 5%	Lift 10%
0.51593	1.21473	1.15457	1.14763

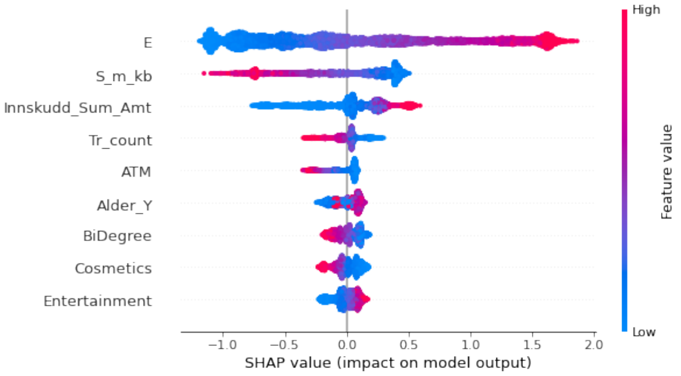


Figure A.4: Feature importance RFM+SD with single-valued RFM features.

A.2 Confusion Matrices

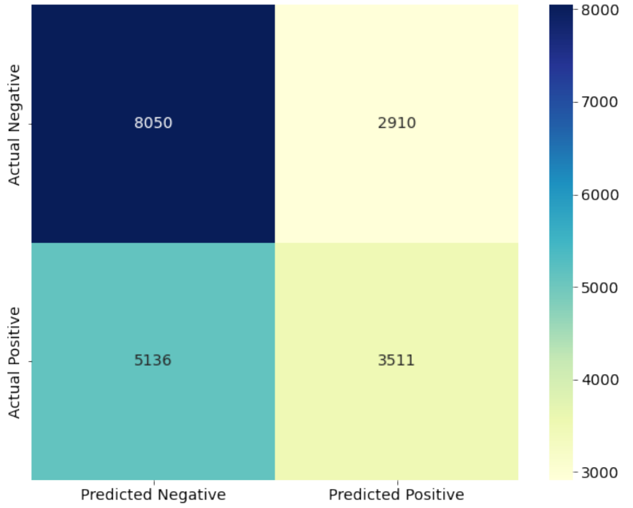


Figure A.5: Confusion matrix for the SD model from Product 1.

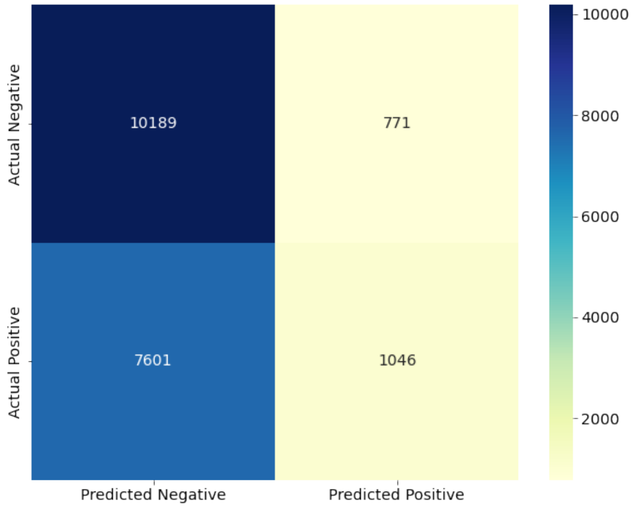


Figure A.6: Confusion matrix for the PSN model from Product 1.

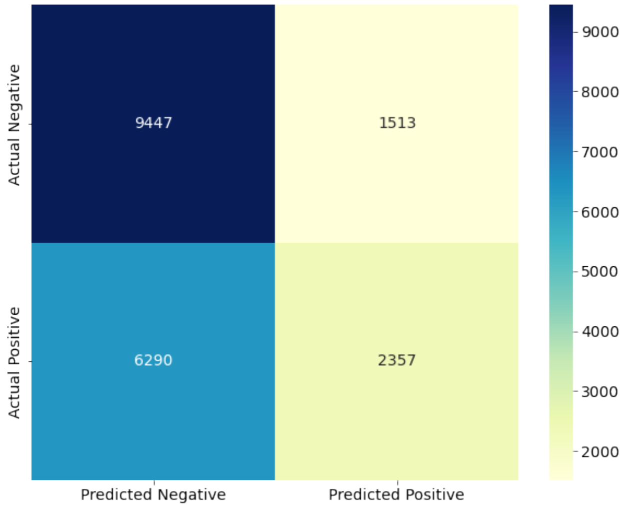


Figure A.7: Confusion matrix for the RFM model from Product 1.

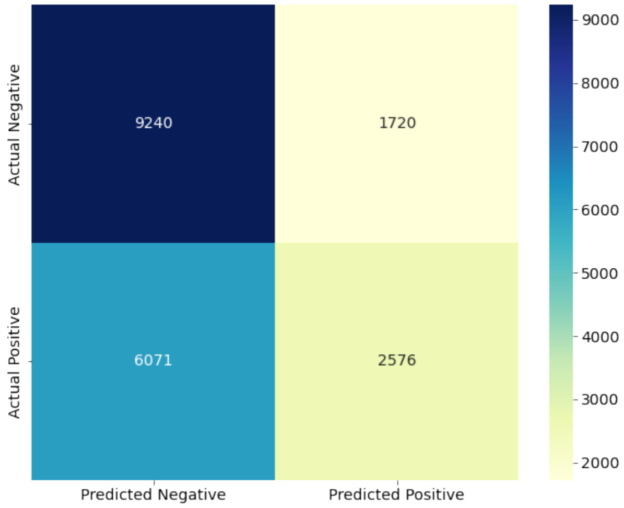


Figure A.8: Confusion matrix for the PSN+SD model from Product 1.

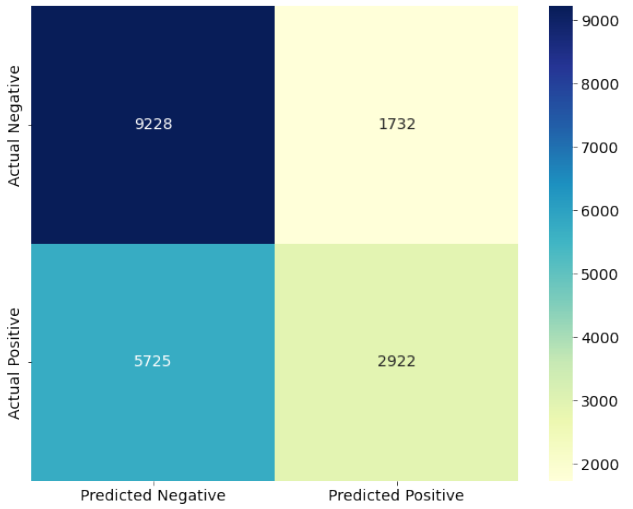


Figure A.9: Confusion matrix for the RFM+SD model from Product 1.

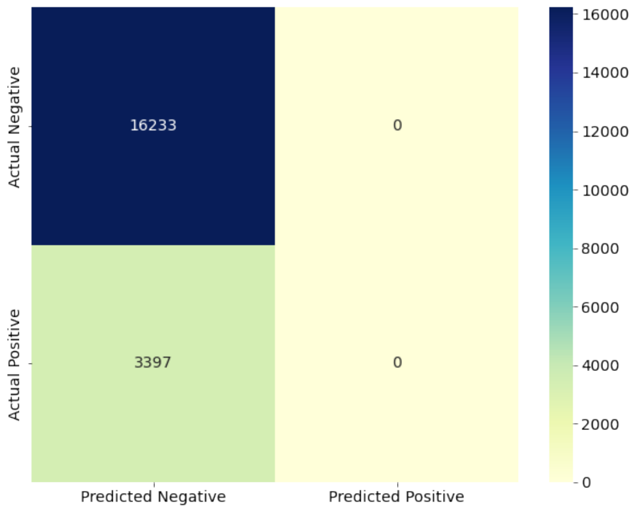


Figure A.10: Confusion matrix for the SD model from Product 2.

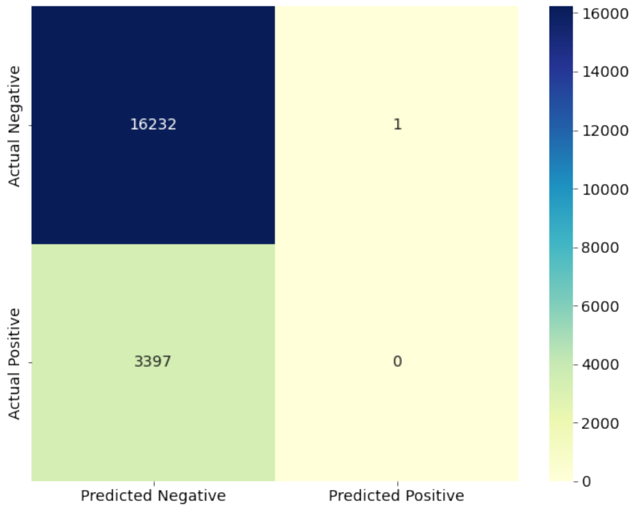


Figure A.11: Confusion matrix for the PSN model from Product 2.

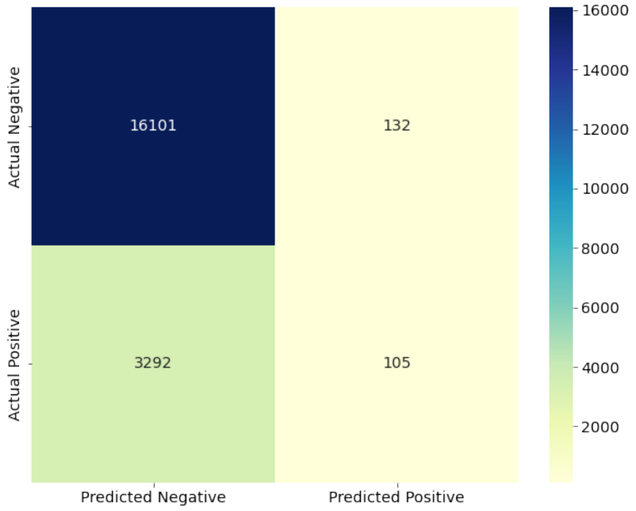


Figure A.12: Confusion matrix for the RFM model from Product 2.

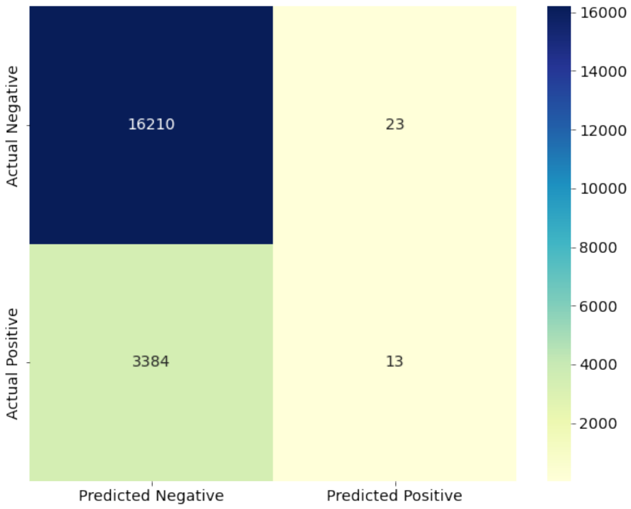


Figure A.13: Confusion matrix for the PSN+SD model from Product 2.

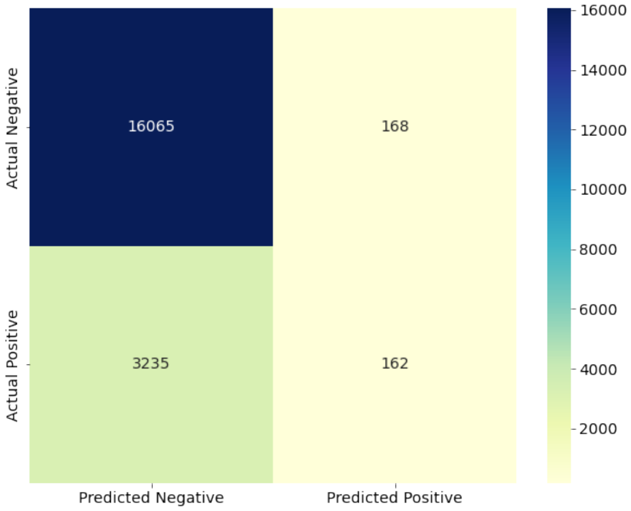


Figure A.14: Confusion matrix for the RFM+SD model from Product 2.

Bibliography

[1] D. Martens and F. Provost, ‘Pseudo-social network targeting from consumer transaction data,’ Sep. 2011.

[2] J. Scott, ‘Social network analysis,’ *Sociology*, vol. 22, no. 1, pp. 109–127, Feb. 1988.

[3] L. Ma, R. Krishnan and A. L. Montgomery, ‘Latent homophily or social influence? an empirical analysis of purchase within a social network,’ *Management Science*, vol. 61, no. 2, pp. 454–473, Feb. 2015.

[4] M. McPherson, L. Smith-Lovin and J. M. Cook, ‘Birds of a feather: Homophily in social networks,’ *Annual Review of Sociology*, vol. 27, pp. 415–444, Nov. 2001.

[5] A. L. Samuel, ‘Some studies in machine learning using the game of checkers,’ *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, Jul. 1959.

[6] J. Lismont, S. Ram, J. Vanthienen, W. Lemahieu and B. Baesens, ‘Predicting interpurchase time in a retail environment using customer-product networks: An empirical study and evaluation,’ *Expert Systems with Applications*, vol. 104, pp. 22–32, Aug. 2018.

[7] M. Óskarsdóttir, C. Sarraute, C. Bravo, B. Baesens and J. Vanthienen, ‘Credit scoring for good: Enhancing financial inclusion with smartphone-based microlending,’ *International Conference on Information Systems 2018, ICIS 2018*, Jan. 2020.

[8] A. Said, E. W. De Luca and S. Albayrak, ‘Using social and pseudo-social networks for improved recommendation quality,’ *CEUR Workshop Proceedings*, vol. 756, pp. 45–48, 2010.

[9] M. Stankova, S. Praet, D. Martens and F. Provost, ‘Node classification over bipartite graphs through projection,’ *Machine Learning*, vol. 110, no. 1, pp. 37–87, Jan. 2021.

[10] M. Latapy, C. Magnien and N. D. Vecchio, ‘Basic notions for the analysis of large two-mode networks,’ *Social Networks*, vol. 30, no. 1, pp. 31–48, Jan. 2008.

[11] D. Martens, F. Provost, J. Clark and E. J. de Fortuny, ‘Mining massive fine-grained behavior data to improve predictive analytics,’ *MIS Quarterly*, vol. 40, no. 4, pp. 869–888, 2016.

[12] A. De Caigny, K. Coussement and K. W. De Bock, ‘Leveraging fine-grained transaction data for customer life event predictions,’ *Decision Support Systems*, vol. 130, Mar. 2020.

[13] A. Grover and J. Leskovec, ‘Node2vec: Scalable feature learning for networks,’ *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 855–864, Aug. 2016.

[14] D. A. Landherr, D.-M. B. Friedl and J. Heidemann, ‘A critical review of centrality measures in social networks,’ *Business & Information Systems Engineering 2010 2:6*, vol. 2, no. 6, pp. 371–385, Oct. 2010.

[15] T. W. Valente, K. Coronges, C. Lakon and E. Costenbader, ‘How correlated are network centrality measures?’ *Connections (Toronto, Ont.)*, vol. 28, no. 1, p. 16, Jan. 2008.

[16] A. K. Shaikh, M. Al-Shamli and A. Nazir, ‘Designing a relational model to identify relationships between suspicious customers in anti-money laundering (AML) using social network analysis (SNA),’ *Journal of Big Data*, vol. 8, no. 1, pp. 1–22, Dec. 2021.

[17] H. Kaur, H. S. Pannu and A. K. Malhi, ‘A systematic review on imbalanced data challenges in machine learning,’ *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–36, Jul. 2020.

[18] R. Mohammed, J. Rawashdeh and M. Abdullah, ‘Machine learning with oversampling and undersampling techniques: Overview study and experimental results,’ in *2020 11th International Conference on Information and Communication Systems (ICICS)*, IEEE, Apr. 2020, pp. 243–248.

[19] R. Muñoz-Cancino, C. Bravo, S. A. Ríos and M. Graña, ‘On the dynamics of credit history and social interaction features, and their impact on creditworthiness assessment performance,’ Apr. 2022.

[20] R. Muñoz-Cancino, C. Bravo, S. A. Ríos and M. Graña, ‘On the combination of graph data for assessing thin-file borrowers’ creditworthiness,’ Nov. 2021.

[21] W. L. Hamilton, ‘Graph representation learning,’ *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, Sep. 2020.

[22] K. Church and W. Gale, ‘Inverse document frequency (IDF): A measure of deviations from Poisson,’ in, 1999, pp. 283–295.

[23] S. Kaufman, S. Rosset, C. Perlich and O. Stitelman, 'Leakage in data mining: Formulation, detection, and avoidance,' *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, pp. 1–21, Dec. 2012.

[24] A. E. Castro Sotos, S. Vanhoof, W. van den Noortgate and P. Onghena, 'The transitivity misconception of Pearson's correlation coefficient,' *Statistics Education Research Journal*, vol. 8, no. 2, pp. 33–55, Nov. 2009.

[25] M. Sandri and P. Zuccolotto, 'A bias correction algorithm for the Gini variable importance measure in classification trees,' *Journal of Computational and Graphical Statistics*, vol. 17, no. 3, pp. 611–628, Sep. 2008.

[26] T. Fawcett, 'An introduction to ROC analysis,' *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[27] A. P. Bradley, 'The use of the area under the ROC curve in the evaluation of machine learning algorithms,' *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[28] S. Tufféry, *Data mining and statistics for decision making*. Wiley, Mar. 2011, pp. 288–290.

[29] S. M. Lundberg and S. I. Lee, 'A unified approach to interpreting model predictions,' *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 4766–4775, May 2017.

[30] J. Yoon, 'Forecasting of real GDP growth using machine learning models: Gradient Boosting and Random Forest approach,' *Computational Economics*, vol. 57, no. 1, pp. 247–265, Jan. 2021.

[31] J. M. Lobo, A. Jiménez-Valverde and R. Real, 'AUC: a misleading measure of the performance of predictive distribution models,' *Global Ecology and Biogeography*, vol. 17, no. 2, pp. 145–151, Mar. 2008.

[32] Miklos Dietz, Somesh Khanna, Tunde Olanrewaju and Kausik Rajgopal, 'Cutting through the fintech noise: Markers of success, imperatives for banks,' *Global Banking Practice*, Dec. 2015.

[33] I. H. Sarker, 'Machine learning: Algorithms, real-world applications and research directions,' *SN Computer Science 2021 2:3*, vol. 2, no. 3, pp. 1–21, Mar. 2021.

[34] T. N. Kipf and M. Welling, 'Semi-supervised classification with graph convolutional networks,' Sep. 2016.

NTNU

Norwegian University of
Science and Technology