DEGREE PROJECT IN THE FIELD OF TECHNOLOGY
INFORMATION AND COMMUNICATION TECHNOLOGY
AND THE MAIN FIELD OF STUDY
COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Machine learning and spending patterns

A study on the possibility of identifying riskily spending behaviour

MATHIAS HOLM

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

Machine learning and spending patterns

A study on the possibility of identifying riskily spending behaviour

MATHIAS HOLM

Master's Programme in Machine Learning
Date: December 21, 2017
E-mail: mathholm@kth.se
Supervisor: Jana Tumova
Examiner: Patric Jensfelt
Swedish title: Maskininlärning och utgiftsmönster
School of Computer Science and Communication

Abstract

The aim of this study is to research the possibility of using customer transactional data to identify spending patterns among individuals, that in turn can be used to assess creditworthiness. Two different approaches to unsupervised clustering are used and compared in the study, one being K-means and the other an hierarchical approach. The features used in both clustering techniques are extracted from customer transactional data collected from the customers banks. Internal cluster validity indices and credit scores, calculated by credit institutes, are used to evaluate the results of the clustering techniques. Based on the experiments in this report, we believe that the approach exhibit interesting results and that further research with evaluation on a larger dataset is desired. Proposed future work is to append additional features to the models and study the effect on the resulting clusters.

Sammanfattning

Målet med detta arbete är att studera möjligheten att använda data om individers kontotransaktioner för att identifiera utgiftsmönster hos individer, som i sin tur kan användas för att utvärdera kreditvärdighet. Två olika tillvägagångssätt som använder oövervakad klustring (eng. unsupervised clustering) används och utvärderas i rapporten, den ena är K-means och den andra är en hierarkisk teknik. De attribut (eng. features) som används i de båda klustrings teknikerna utvinns från data som innehåller kontotransaktioner och som erhålls från banker. Interna kluster värde index (eng. cluster validity indices) och individers riskprognoser, som beräknats av ett kreditinstitut, används för att utvärdera resultaten från klustrings teknikerna. Vi menar att resultaten som presenteras i denna rapport visar att målet till viss del uppnåtts, men att mer data och forskning krävs. Vidare forskning som föreslås är att lägga till fler attribut (eng. features) till modellerna och utvärdera effekten på de resulterande klusterna.

Contents

Chapter 1

Introduction

As the amount of data in the world are growing, one question that could be asked is: What are we supposed to do with this data and how can the data benefit us? To sort through the data manually would be complicated and time consuming, but due to computers and their increasing processing power we can now use advanced algorithms to evaluate the data and find complex patterns. The idea of machine learning algorithms have been around for some time, but in recent years it has become more generally known and the knowledge of such algorithms is now a desired competence in the industry. A lot of companies are putting effort into understanding how machine learning can help them become more efficient or how they can use data to build new products.

Within the banking and financial sector there is a demand for technologies that enable their services and these kind of technologies have even gotten their own name, i.e. fintech. One service in the financial sector that can benefit from the use of technology is assessing the creditworthiness of people, i.e. how likely is it that this person will default on a debt. In some countries, institutes or credit bureaus exist that will give companies a credit score on an individual looking to apply for a loan, but the way these scores are calculated differ. What most of these credit institutes have in common is that they provide a probability on how likely the individual is to default within a specific period of time depending on several attributes of the individual, such as income, amount of loans or previous default history.

But perhaps the approach of assessing creditworthiness can be done in a different way. Perhaps other parameters could be useful to the lender than those received from the credit institutes. The primary aim of this study is to investigate if the information content of bank statements is sufficient to find different spending patterns among individuals, that in turn could be used to assess the creditworthiness of the individuals. The project has been carried out in collaboration with a newly-formed startup within the financial sector that aims to evolve the way interest rates on loans are decided upon. From the point of view of the company, there is a desire to learn and understand more about the benefits of using machine learning in their line of work. A first step to achieving this outcome is being examined in this study by investigating patterns in the spending behaviour of individuals.

1.1 Research Question

The research focuses on assessing the possibility of finding certain spending patterns among individuals based on features extracted from bank statements, and what information these patterns may reveal. The hypothesis is that with complete information on the transactions and spendings of an individual, it is possible to find these patterns and create a sound model where the individuals are sorted into different clusters based on their spending behaviour. For the model to be sound, it is required that the individuals in the different clusters share some characteristic traits which relate to their creditworthiness.

In any kind of research having access to complete information is desirable, but often difficult to obtain. Therefore the information used in this project will consist of information that can be extracted from bank statements. The research question that is examined is therefore:

- Is it possible to use an unsupervised clustering algorithm to create a sound model where individuals are sorted into different groups based on data extracted from the individual's bank statement?

1.1.1 Challenges

The problem is that even though a bank statement contains detailed data on the transactions made by the individual, it does not give us complete information. One notable problem is cash transactions, since these will not be visible in the bank statement. If the individual withdraws the cash from an ATM this will be noted in the bank statement, but there is no way of knowing how the individual spent that money. Also, when it comes to transactions made to another bank account, the bank statement will give incomplete data because it may be hard to identify if the transaction is a payment or if the individual merely transferred money into another savings account.

The most prominent challenges in this project are to find a way to represent and extract features from the bank statements, but also to find an algorithm to process the features and discover spending patterns among individuals. Regarding the features, part of the project is to identify what features are more likely to result in a high information content, i.e. desirable features, and what features are less important, i.e. low information content and therefore less desirable. This has been done by studying the raw data but also by analyzing the information content in the different features using dimensionality reduction methods such as feature selection and Principal Component Analysis (PCA).

1.2 Contribution to the field of finance

Assessing the creditworthiness of an individual is of great importance to a lending company when accepting new customers, but it may also be of use for existing customers. By studying spending behaviour, a company can also pay attention to behavioural changes in existing customers. Even if a new customer receives an acceptable score from a credit institute, it is not certain that this score will stay the same over a period of several months or years. It may furthermore be difficult for a company to justify conducting a new credit assessment on a existing customer. Therefore the company will have to rely on available data to monitor existing customers. One indication of a customer dealing with financial difficulties may be that they are falling behind on their payments, but at this stage it may be too late to recover and the

customer may default. If the company can utilize bank statements to continuously monitor changes in the behaviour of customers, this could result in problems being detected before a customer ends up in deep financial trouble.

As the collaborating company is a newly-founded startup, the amount of data is limited and no clear labels are present. There exist credit scores, calculated by a credit institute, for a majority of the individuals. These credit scores could be used as labels in a supervised setting, but for the purpose of this study the scores are merely used to evaluate the resulting clusters. At a later stage, the company may have been able to collect additional data on customers that default. This would create another set of labels that could be used, but at this stage no such data exist. It could be argued that data can be shared among different lending institutes, but this may prove difficult. The main issue is that lending institutes may be hesitant to share customers' data with their competitors, and doing so may also increase the risk of exposing private data.

The main objective of this project is to use bank statements in order to study the spending behaviour of individuals. This has been done by extracting features from bank statements and using unsupervised clustering techniques in order to analyze and find spending patterns in the data. These spending patterns are studied in order to determine if they can be related to the creditworthiness of individuals. The reason for using clustering algorithms is that they are commonly used to discover patterns in data and also because relatively simple algorithms exist with suitable visualization capabilities. The choice of algorithm used to generate the clusters is somewhat dependent on the number of features in the data set and also the size of the data set. Therefore, algorithms from two different clustering methods are tested and compared against one another. Of the two algorithms examined, one is hierarchical and one is partitional.

1.3 Delimitations

The only features used as input to the clustering algorithms in this study are the features extracted from lists of transactions that are col-

lected from banks. This data have been provided by the collaborating company, more information about the raw data can be found in Section 3.2.1. Even though adding additional features may improve the clustering results, this is not covered in the study. Regarding the hierarchical clustering method, this study contains some comparisons between different linkage methods, but no deeper experiments into the different methods are conducted. The reason for this is that the primary goal of the study is not to compare or evaluate different algorithms, but instead to investigate if it is possible to discover different spending patterns in the data.

1.4 Sustainability and ethics

As previously mentioned, many companies are putting effort into researching how machine learning techniques can make their business more efficient. From an economical perspective the research conducted in this study can help companies working within the financial sector reduce their workload. If companies are able to make more accurate credit assessments with less amount of data, and removing the need for credit institutes, this can not only speed up the assessment process but also reduce costs for the company. With a quicker process individuals can also be granted loans faster, and with a more accurate credit assessment get an fairer interest rate. The benefits of not having to rely on credit institutes are even greater for multinational corporations, since the credit institutes may differ between different countries. In this way the corporation can use the same credit assessment process in every country where they are active.

Within most fields of science, including machine learning, it is important to always reflect on the ethical aspect of the research, especially when working with data that can be sensitive. No sensitive personal information, such as ethnicity, religious beliefs or medical status, has been used in this study. Despite this, the aspect that spending patterns may be viewed as sensitive data has been registered and caution has been exercised when working with this data. In practice this means that the data have been kept strictly confidential and password protected files have been used when transferring the data. The files and password have then been communicated through two

different secure channels. Furthermore, the data has been anonymized by removing personal numbers that are used in Sweden to identify a specific individual.

In theory, the work conducted in this study could potentially make the credit assessment process more ethical and fair. If it is possible to exclusively conduct a credit assessment based on the recent transaction history of an individual, we could remove the use of previous history of defaulting or failure to pay bills on time. This would suggest that if a lender is satisfied with the current spending pattern of an individual, the lender would not have to be concerned with the previous financial history of the individual. In this way, an individual who at some point in time has had financial problems would not have to be affected by these in the future, as long as the individual solves the problems and sorts out the private financial situation. Additionally, a bank or other company could use a system similar to the one studied in this report in order to detect changes in the spending patterns of individuals. For example, if it is noticed that an individual alter the spending behaviour toward one associated with a higher risk, the company could make the individual aware of this before any major financial problems arise. Similarly, if an individual start to exhibit a less risky spending behaviour, the lender could award the individual with, for example, a lower interest rate and thus encourage more stable spending behaviours. Additionally, creating a sound and accurate credit assessment system would make it less likely that a person has to view the data, making the process more anonymous.

Chapter 2

Previous work

Two areas of previous work related to this project are being studied. The first of these is the use of machine learning techniques for evaluating the creditworthiness of individuals and the second is studying unsupervised learning algorithms and their applications.

2.1 Credit assessment

The process of assessing the creditworthiness of individuals may be of great value to banks and loan givers when assessing applicants. Therefore, methods are continuously being developed to increase the accuracy of the credit assessment process. In this section, studies regarding machine learning and credit assessment will be presented.

2.1.1 Early utilization

In 1987 Carter and Catlett [1] were early to propose the idea of using machine learning techniques to perform the task of credit assessment. They argue that the traditional approach to credit assessment, where human experts use their experience combined with guidelines from lending institutions, is inflexible, inconsistent, inefficient and has poor performance. They also argue that the use of discriminant analysis, which in 1987 became increasingly popular, may solve some of the problems with the traditional approach but that this method makes some inappropriate assumptions, thus raising the question if expert systems could perform better. It should be mentioned that the authors use the term expert systems for both conventional rule-based systems

derived from expert opinions and machine learning techniques.

The authors study a decision tree approach to assess credit card applicants. They argue that this approach can be quicker and more cost-effective to develop compared to rule-based expert systems, where the expert data is gathered by conducting interviews with several human experts. Using the so called C4 algorithm, they are able to prove a statistically significant improvment compared to the existing methods.

2.1.2 Machine learning and expert systems

The results found by Carter and Catlett [1] are possibly not significant enough to conclude that machine learning techniques are to be used for credit assessment, but it is an interesting early utilization of machine learning for credit assessment. In their paper they also address both machine learning techniques and conventional rule-based system as expert systems without elaborating the difference between the two. In a paper by Ben-David and Frank [2], the authors compare the accuracy of machine learning techniques and conventional expert systems on credit scoring. They describe how expert systems became popular in the 80s and machine learning during the 90s, but how there are rather few publications comparing the two approaches. They also mention how machine learning components began to be included in expert systems and therefore the two became less distinguishable. Despite this, the definition they use of an expert system is a system that encapsulates human knowledge without the use of machine learning techniques.

According to Ben-David and Frank [2], it is difficult to compare the two approaches because commercial implementations are rarely released to the public and few organizations develop both types of systems for the same problem. Despite this, they manage to compare the accuracy of a commercial credit scoring expert system with several machine learning models. They find that while some machine learning models had a higher accuracy than the expert system in a classification experiment, no statistically significant advantage could be proven. Although in a regression experiment they were able

to show that some of the tested machine learning models have a statistically significant advantage over the expert system. The authors conclude that while the results showed that their machine learning approach was not superior to the expert system in respect to accuracy, the machine learning models were developed in a matter of days while the expert system took several months and that none of the models were shown to be inferior relative the expert system.

2.1.3 Neural network approach

The work in [2] includes little optimization of the machine learning models tested. Instead they strive to show how relatively simple models could achieve results similar or even better than that of an expert system, which took several months to develop. Other scientists have studied more advanced and optimized algorithms to be used for credit assessment. In his paper from the year 2000, West [3] studies five neural network architectures using two real world data sets to investigate the possibility of improving credit scoring accuracy. The data sets contain information such as applicant credit history, account balances, personal information and employment status. Other models such as linear discriminant analysis, logistic regression, k nearest neighbor and decision trees are also studied to compare with the neural network models. He is able to show that neural network models can improve the credit scoring accuracy, however he mentions that the use of these will require modeling skills to construct the training methods and network topologies. It is also observed that logistic regression is more accurate than both the neural network models and linear discriminant analysis for the two data sets considered.

Lee, Chiu, Lu, *et al.* [4] study a more sophisticated hybrid neural discriminant approach to the credit scoring problem. By first using a traditional discriminant analysis approach and using the credit scoring result and significant predictor variables as input to the neural network model they are able to improve the credit scoring accuracy. The data set used in this work is the same as one of the data sets used by West [3]. It is also concluded by Lee, Chiu, Lu, *et al.* [4] that the hybrid model converges faster and has the lowest Type II error, i.e. misclassifying a customer with bad credit, compared to using

logistic regression, traditional discriminant analysis or normal neural networks. Other scientists have tried using ensembles of neural networks, i.e. combining multiple classifiers, to improve the credit scoring accuracy, although without promising experimental results [5].

2.1.4 Benchmarking study

In a paper from 2003, Baesens, Van Gestel, Viaene, *et al.* [6] address the problem of deciding what classification technique to be used for credit scoring. The paper claims that most previous studies assess the performance of a small number of techniques on one specific credit scoring data set, which causes conflicting results in different studies. To overcome this problem, the authors conduct a benchmarking study evaluating the performance of several classification algorithms, including logistic regression, linear and quadratic discriminant analysis, support vector machines (SVMs) and neural networks. To further strengthen the validity of the study, eight real-life data sets from different locations around the world are used to evaluate the techniques. The studied techniques are compared by both calculating the classification accuracy and the receiver operating characteristic (ROC) curve, i.e. plotting the true positive rate against the false positive rate. The paper concludes that while many of the evaluated classification techniques were found to have similar performance, few methods were shown to be inferior to the others while some methods performed better. The techniques that had the best performance were the neural network classifiers and a least squares SVM using a RBF kernel, although both linear discriminant analysis and logistic regression yielded similar results which indicates that the non-linearity of data sets is relatively weak.

2.1.5 Support vector machine

In [6], it was found that a least square SVM were among the best performing techniques evaluated. Several other researchers have also studied the use of SVMs for the problem of credit assessment [7]–[10]. Huang, Chen, and Wang [7] propose a hybrid genetic algorithm (GA) support vector machine (SVM). Apart from achieving a promising

classification accuracy, this GA-SVM hybrid is shown to simultaneous be able to optimize the model parameters and perform feature selection. Correspondingly, Bellotti and Crook [8] also determine that SVMs can be used as a basis to discover the most significant features.

2.1.6 Non-parametric approach

As can be seen from the previously presented studies, the task of credit assessment is often regarded as a classification problem where the applicants are either accepted or declined. Instead of just using a binary classification, Kruppa, Schwarz, Arminger, *et al.* [11] calculates the default probability of an applicant by using a non-parametric regression approach. They mention that the standard approach to predicting default probabilities is by using logistic regression, but this entails that some rather strict assumptions has to be made and the model is unable to deal with high correlation between independent variables. By embedding the problem as a non-parametric regression task, they are able to use machine learning methods such as random forest with probability estimation trees, k-nearest neighbors and bagged k-nearest neighbors to estimate the default probability.

What is interesting about [11], in relation to the work conducted in our study, is that the authors of the paper did not have access to information about the individuals that commonly serve as the most prominent predictors in credit assessment, such as employment status, income or information about previous unpaid bills. Instead the data set used stem from a household appliance company that offers payment by installment. The data set contain information such as if the applicant has ordered before, residential region, downpayment amount, age, amount of each installment, the number of installments and a dependent variable to indicate if the individual defaulted or not. In this experimental setting, they are able to establish that the random forest implementation outperform not only standard logistic regression but also a optimized logistic regression where variables were selected using expert knowledge.

2.1.7 Transactional data

Another paper that is highly relevant to this project is by Khandani, Kim, and Lo [12]. In this paper the authors are using account-level transactions data combined with data from credit bureaus to increase the accuracy of classifying default rates. Due to the fact that each transaction in the transactional data used in their study contain a category field to disclose the nature of the transaction, the authors can extract customer features such as phone expenses, travel expenses, bar expenses and income. The authors do however emphasize that the accuracy of these categories vary and that this is an incomplete representation of a customer's consumption. The reason for this is, among others, that the purpose of an ATM transaction is unknown and the customer may also have multiple banks. Despite this, the features extracted from the transactions data are combined with account balance databases and credit scores to be used in a decision tree learning approach. The authors are able to prove encouraging results when trying to predict customers that will be 90 days or more behind on their payments when training the model on a 3-month window.

2.1.8 Additional approaches

The idea of employing machine learning techniques for the task of credit assessment has been studied since the 1980s, and it is still being studied. In a recent paper by Luo, Wu, and Wu [13], the idea of using deep learning algorithms for credit scoring is explored. By using a Deep Belief Network with Restricted Boltzmann Machines and comparing this to logistic regression, SVMs and Multilayer Perceptron, they are able to conclude that the deep learning algorithm outperforms all of the other approaches. Other recent papers are studying the use of a single Restricted Boltzmann Machine classifier [14] or the use of hybrid and ensemble classification techniques [15]–[17] for credit scoring.

Researchers have also tried to improve the popular SVM classifier. Yi and Zhu [18] are examining a fuzzy Support Vector Machine to reduce the problems caused by noise and outliers while Shi, Zhang, and Qiu [19] study a feature-weighted SVM to increase the classification

accuracy. Harris [20] tries to improve the computational expenses of SVMs on large datasets, while preserving the accuracy, by introducing a clustered Support Vector Machine to be used for credit scoring. Studies on using semi-supervised algorithms for datasets where a low level of default history exists have also been conducted [20], [21]. By using a semi-supervised SVM, Li, Tian, Li, *et al.* [21] are able to include rejected applicants in the datasets even though no information on the rejected applicants behaviour or repayment history exist.

2.1.9 Concluding notes

What can be observed is that all the previous work presented in this section use some kind of supervised or semi-supervised classification algorithm to study the credit assessment problem. Most of them are also using the same data sets or data sets that are similar in information content and features. During the literature study conducted, no articles on the use of unsupervised clustering algorithms for credit assessment could be found. The data set used in this study differs from the data sets commonly used in the studies presented above, primarily in that no data such as credit history, employment status or other personal data is used. This may support the idea of using different techniques than those usually studied in relation to credit assessment. The fact that the data available to us at this point is treated as unlabelled further supports the idea of using unsupervised learning techniques in this project.

2.2 Unsupervised Learning

Unlike supervised learning, where each data point in the training data is pre-classified with a label or target value, unsupervised learning algorithms are dealing with datasets without labeled data. The task of discovering patterns or the structure of datasets without any labels when using unsupervised algorithms is referred to as clustering, or occasionally unsupervised classification. The purpose of clustering is to find groups in the dataset where objects belonging to the same group are similar to each other. To achieve this, a similarity definition is used to measure the similarity between two data points. Several different similarity definitions exist and the one that is used affects the

outcome of the clustering algorithm and the shape of the discovered clusters, thus making the similarity definition data dependent. One way of measuring the similarity of two data points is to use the distance between them, such as the Euclidian distance, which often results in hyperspherical clusters [22]–[26]. Another way is to use a probabilistic model that assumes the data were generated from a mixture of distributions, such as a mixture of Gaussians [22], [26].

Two different categories of clustering problems exist. In hard clustering, or sometimes called crisp clustering, each data point can only belong to one cluster while in soft clustering, or fuzzy clustering, the data points can, with some probabilities, belong to multiple clusters [23], [25]. Clustering algorithms can also be divided into hierarchical algorithms, where a hierarchy of clusters is generated, or partitional algorithms, where one-level clusters are discovered without any hierarchical relations between the clusters [23]. Density based and grid based clustering algorithms also exist [27].

According to Gan, Ma, and Wu [23], there are four design phases involved in developing a clustering algorithm. These four phases are data representation, modeling, optimization and validation. The choice of how to represent the data determines the structures of the clusters that can be found in the data. Feature selection is also an important part in the development of a clustering algorithm. Many models tend to overfit the data and degenerate when the feature space is large, which is commonly referred to as the curse of dimensionality. Because of this, the dimensionality is often reduced by selecting a smaller subset of relevant features to be used in the learning algorithm [25], [28]. Removing irrelevant features from the data set may lead to a more accurate model and lower computational costs but it is important to notice that using different subsets of features may lead to different clusters being produced [28]. Feature selection is also related to dimensionality reduction techniques, such as principal component analysis (PCA), in which linear combinations of features are generated [26].

Unlike supervised learning, where the training data points are labeled, there rarely exist any direct measurements of success when using unsupervised clustering algorithms. Yet, there has to be a way

of validating the clusters found by an algorithm in order to compare the results of different algorithms. To achieve this, it is important to use some validity criteria. It is also important to use some cluster validity method for when there is no prior knowledge about the optimal number of clusters in the data set. This situation is relatively common in unsupervised clustering [29], [30]. The two main categories of validation criteria are external and internal validation. If there exist any prior knowledge not present in the data, such as the structure specified by a class label, external validation is used to compare the result of a clustering algorithm to that of the prior knowledge. In the case where a class label exist, entropy can be used as a external validation to evaluate the clusters [30].

When there is no prior knowledge about the data, internal validation criteria are used to evaluate the clusters based on information in the data. Both external and internal validation are based on statistical testing, which can be computationally expensive [29]. Several different validation measurements exist and despite extensive research, there is no conclusive solution to cluster validation and the relationships between different validation measures are not fully established. The choice of clustering algorithm may also affect the choice of validation measures because some measures may be more appropriate for different algorithms [30]. Despite the existence of validation criteria, the task of evaluating clusters sometimes has to be based on subjective and heuristic evaluations, such as case studies, to assess the results [24], [30].

2.2.1 Common application areas

In this section some common application areas for unsupervised clustering will be presented in order to gain more insight into the advantages of clustering analysis. Clustering can be used in market research in order to identify market segments who tends to purchase less or in health psychology in order to identify groups of people at risk or people that may benefit from a specific service [23]. Other common uses for clustering includes collaborative filtering, in order to provide recommendations in a range of applications from similar users, or dynamic trend detection in social networking application. Due to its potential data summarization capabilities, clustering can

also be used as an intermediate stage for other machine learning problems such as classification or outlier analysis [26].

Unsupervised learning techniques, such as clustering analysis, are commonly used in the field of genomics to analyse gene expression data. Gene expression data is collected by using cDNA or oligonucleotide microarrays which measures the expression level of a DNA sequence or gene. The collected data can be represented as a real-valued $N \times D$ matrix where N is the number of genes and D is the number of samples. Each x_{ij} value represents the expression level of the gene *i* in sample *j*. Therefore a column in the matrix represents the expression profile for a specific sample and a row represents a expression pattern for a specific gene over all samples [31], [32]. To analyze large amounts of gene expression data it is advantageous to use computational methods, collectively referred to as bioinformatics or computational biology. These methods can be used to study the similarity between DNA sequences by aligning them, predict protein structures or to study evolutionary relationships between various species [31]. Cluster analysis is commonly used to study the function and regulation of a gene as well as the cellular process by identifying genes with similar expression patterns, referred to as coexpressed genes. When clustering the gene expression data, the data can be clustered either based on the genes, i.e. the rows of the matrix, or based on the samples, i.e. the columns of the matrix. The goal of gene-based clustering is to find genes that share similar characteristics, i.e. coexpressed genes, while the goal of sample-based clustering is to form groups of sample to identify phenotype structures among the samples [32].

The dataset used in our project can be represented by a real-valued $N \times D$ matrix similar to how the gene expression data is represented, thus indicating that algorithms used for gene expression clustering can also be used in this project. It is generally believed that a single-linkage clustering algorithm is outperformed by the K-means algorithm on gene expression data [32]. The single-linkage algorithm pertains to the hierarchical clustering algorithms while *K*-means belongs the partitional clustering algorithms.

2.2.2 Hierarchical clustering and K-means

According to Reddy and Vinzamuri [33], partitional and hierarchical algorithms are the most extensively researched and used clustering algorithms, mainly because of their simplicity. The hierarchical algorithms generates a hierarchy of clusters by developing a tree-based structure referred to as a dendrogram. The amount of clusters can then be decided by splitting the dendrogram at different levels. The dendrogram can be constructed either in a bottom-up or top-down manner and the two approaches may produce different results. Instead of constructing a hierarchy of clusters, partitional methods aims generate a number of one-level distinct groups. Hierarchical methods operates by starting of with the individual data points and gradually builds the clusters while partitional methods has to be initialized with a set of seeds or clusters that are iteratively revised [33].

The *K*-means algorithm is a simple and commonly used partitional clustering algorithm. *K*-means is a so called center-based algorithm and the clusters generated are spherical and represented by a center. Therefore, the *K*-means algorithm is a poor choice of algorithm for finding clusters of arbitrary shapes [34]. The number of clusters, *K*, has to be chosen beforehand and the algorithm starts by assigning *K* points to be the initial centroids. By using a predetermined distance measure, i.e. similarity measure, the data points are then assigned to the closest centroid and the centroids are updated with regards to the new members of the cluster. The assignment of points to the closest centroid and the recomputing of the centroid is repeated until a specific convergence criterion is met, if such criterion exists, or until the centroids stop changing [33], [34]. The performance of *K*-means is affected by the number of estimated clusters, *K*, and by the initialization of the centroids. Therefore several approaches to estimating the number of clusters and how to initialize the centroids has been developed. Several variations of the *K*-means algorithm exist, such as *K*-medians, *K*-modes and also fuzzy *K*-means [33].

Since partitional clustering methods generally require the user to specify the number of clusters, hierarchical clustering was proposed in order to solve this and some other disadvantages with partitional algorithms. The idea is that the hierarchical algorithms are to be more

deterministic and flexible [33]. As previously mentioned, hierarchical methods can be divided into bottom-up, also referred to as agglomerative, and top-down, referred to as divisive. The divisive method starts by grouping all data points into one cluster and then gradually splits the clusters into smaller ones while the agglomerative start with every data point in its own cluster and gradually merge clusters into larger ones. The root level of a generated dendrogram represents the entire dataset and when traversing the tree, every child node contains a subset of the data points contained in the parent cluster. Each leaf node in the dendrogram corresponds to a data point in the dataset. Therefore, different numbers of clusters can be obtained by splitting the dendrogram at any level without having to predefine the number of clusters or to recompute the dendrogram [33], [35].

Unlike partitional clustering, the hierarchical methods cannot iteratively optimize the clustering quality by revising previously completed merges or splits of clusters. This is often considered to be one of the disadvantages of using hierarchical clustering compared to partitional. The simplicity and computationally efficiency of partitional algorithms also makes them advantageous. However, the visual representation of the dendrograms generated by the hierarchical algorithms can benefit the end-user by helping them understand more about the clusters, especially for high dimensional data [33].

Hierarchical similarity measures

Just as with many other clustering algorithms, different similarity measures can be employed for the hierarchical clustering algorithms. Commonly used measures the in agglomerative hierarchical approach are single linkage, complete linkage, average linkage, centroid-based method and Ward's criterion. Single and complete linkage are the most popular agglomerative clustering measurements and they are also among the simplest. Single linkage employs a nearest neighbor approach, where the similarity of two clusters is equal to the similarity of their closest neighboring members. Instead of using the closest neighbors to represent the similarity of two clusters, complete linkage uses the similarity of the most dissimilar members, i.e. the two neighbors that are the furthest apart. While the single and complete linkage methods are only using the similarity between one point in

each cluster to represent the similarity of two clusters, the average linkage uses the average similarity of all pairs of points from each cluster to represent the similarity of the clusters. This measurement is however computationally expensive, especially as the number of data points grow. To reduce the computational expenses the centroid-based method uses only the distance between the centroid of two clusters to represent their similarity [33], [35].

Ward's criterion seeks to form clusters in a way that minimizes the loss of information correlated to merging clusters. It is also called the minimum variance method because it commonly uses the squared error criterion to calculate the distance between clusters, i.e. quantifying the information loss. The two clusters that will result in the least possible increase in information loss will be merged in every step of the agglomerative clustering algorithm when using Ward's criterion [35].

Several other agglomerative similarity methods exist, but the ones presented above are among the most commonly used ones [35]. Ward's criterion can also be used in divisive clustering to calculate the most efficient split of clusters when all the data is numerical. To handle nominal data, the Gini index can be utilized. The divisive clustering algorithm can be stopped at any level and does not necessarily have to generate the dendrogram all the way to the individual leaves, which suggests that it can be more efficient than an agglomerative method [33]. Finding an optimal bipartition for some clustering criteria is however difficult and the problem has been proven to be NP-hard [35].

2.2.3 Other methods

There are also other kind of clustering algorithms apart from the partitional and hierarchical ones. Model-based clustering is another commonly used approach which is based on probability models. In these clustering algorithms, the data is assumed to have been generated by a mixture of probability distributions and the algorithm try to find models to fit the data. The Gaussian mixture model (GMM) is a popular mixture model in which each cluster is represented by a Gaussian distribution, also referred to as a normal distribution.

The iterative Expectation-Maximization (EM) algorithm is commonly used to find the parameters of the GMM model [36], [37].

Unsupervised artificial neural network techniques for clustering also exist, such as the self-organizing map (SOM). A SOM perform clustering and dimensionality reduction by mapping the input space onto a feature space of lower dimensions, often one- or two-dimensional. This is done by defining a grid of points in a D-dimensional space, where D is the selected dimension of the feature space. Each point in this grid represents a neuron and each neuron holds a vector that defines a location in the input space. This location in the input space ultimately serves as a cluster center. The SOM is calculated by initializing neuron centroids and iteratively updating these in regard to the closest data points, similar to K-means. The difference is that when a centroid is updated, the neighbouring neurons in the grid are also updated [22], [33], [38], [39]. The dimension of the neuron grid and the amount of neurons, i.e. the amount of clusters, has to be given as user input, along with other parameters such as the shape and width of the neighbourhood function and the learning rate. Choosing a two-dimensional grid may be beneficial for visualization purposes but this may also result in a bad fit to the data [22].

2.2.4 Contribution of this project

As mentioned in section 2.1.9, no articles has been found to use unsupervised clustering techniques to study the problem of credit assessment. But as discussed in this section, clustering can often be helpful to find complex patterns and gain insight into the data. In this project, unsupervised clustering techniques has been applied to data extracted from bank statements in order to study the information content of this kind of data. By doing this, the primary goal is to determine if this information is enough to assess the financial situation of an individual.

In this project we have used two different clustering algorithms, K-means and one hierarchical algorithm. The reason for using K-means is that this is a common and well-known algorithm which can serve as a benchmark for other algorithms. Unlike K-means, the hierarchical algorithm does not need to be initialized with the amount of resulting

clusters and the generated dendrogram provide a practical visualization for high-dimensional data. It may be argued that a unsupervised neural network approach, such as the self-organizing map, or a deep learning approach could be used for the work conducted in this report. The reason for not using the SOM approach is because there are a lot of parameters that has to be initialized by the user, similar to K-means. Since the is no prior information on the amount of clusters in the data and there is no decisive way to determine this, it would mean that the SOM would have to be calculated several times for different initializations and evaluated, just like K-means. Furthermore, even though the relations between neighbouring neurons in a SOM provides some hierarchical information and a two-dimensional SOM constitute a suitable visualization, we believe that the hierarchical clustering algorithm and associated dendrogram can provide more information. Regarding deep learning, it is generally known that these kind of algorithms tend to require a large amount of data. If the amount of data is limited, as in this project, it may be favourable to use a more simplistic algorithm such as K-means [40]. Methods to deal with limited datasets for artificial neural networks and deep learning exist [41], [42], but we have chosen to not investigate these further.

Chapter 3

Methods

In this chapter the experiments and methods used in this project are described in more detail. As mentioned in Chapter 2, little to no previous work was found to study the use of unsupervised clustering methods on customer transactions data in order to conduct credit assessment. Therefore the research presented in this report can be seen as a type of exploratory research in order to study the informativeness of this kind of data.

The chapter includes a description of the experimental setup, the feature extraction process, the clustering algorithms that are used and the indices used to evaluate the results of the clustering algorithms.

3.1 Experimental setup

This section contains a description of the hardware and software utilized to conduct the experiments. Regarding software, the only programming language that has been used is Python[1] version 2.7.12 and Python libraries compatible with this version. No further information on the Python programming language will be given but libraries that are crucial for the experiments and not part of the Python standard library[2] will be mentioned.

[1] https://www.python.org/

[2] https://docs.python.org/2/library/index.html

3.1.1 NumPy

NumPy[3] is a Python library commonly used in scientific studies for simplifying work conducted on multi-dimensional arrays and matrices. The library also contains implementations for various functions that can be applied to vectors and matrices, as well as functions such as finding the maximum value or the unique values of a vector or matrix. Other libraries used in these experiments are dependent on NumPy to work and it is also used throughout the study when working with vectors or matrices.

3.1.2 SciPy

The SciPy[4] library contains a large amount of functions for scientific computations, such as numerical algorithms and linear algebra but also domain-specific modules. In this study the function for calculating pairwise distances between data points, pdist[5], and the module for hierarchical clustering[6] has been utilized.

3.1.3 Scikit-learn

Scikit-learn[7], also referred to as sklearn, is a machine learning library for Python that is built on NumPy and SciPy. The library contains modules to satisfy the needs of most machine learning scientists, such as preprocessing, classification and clustering modules. In this study the preprocessing, dimensionality reduction and clustering modules have been utilized. The preprocessing module is used to normalize the data and the dimensionality reduction module is used to conduct feature selection and Principal component analysis. While the SciPy library, described in Section 3.1.2, also contains implementations of the K-means algorithm, the implementation in Scikit-learn was used due to its extended capabilities and built-in parallel computation option.

---

[3] http://www.numpy.org/

[4] https://www.scipy.org/

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html

[6] https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html

[7] http://scikit-learn.org/stable/

3.1.4 Matplotlib

Matplotlib[8] is a Python library that simplifies the creation of graphs and figures. It provides the capability to create a large amount of different types of graphs and gives full control over options such as font setting and the properties of axes. Matplotlib is used to create all the plots of data used in this report.

3.1.5 Hardware

The experiments in this study has been executed on a desktop computer with 16 GiB of RAM using a 8-core Intel Core i7-2700K processor with a clock rate of 3.50 GHz. The operating system used is Ubuntu 16.04.2 LTS desktop version.

3.2 Feature extraction

In this section the work conducted to extract features from the raw data is presented. The section starts off by describing the content of the raw data and continues with a description of the features extracted and how these are calculated.

3.2.1 The raw data

The raw data used in this study contains information on individuals collected from their respective bank and stored in a JSON format. Each data point contains some personal information about the individual, a list of all bank accounts for this individual at the bank from where the data is collected and in some cases information on previous loans held by the individual. Each account in the list of bank accounts also contains a list of the transactions made to and from this account dating back a certain amount of months, generally between 6 and 12 months. The information within this list of bank accounts is the data from which the features are extracted, or more specific the primary account within this list. To better understand the structure of the raw data, see Appendix A.

[8] `https://matplotlib.org/`

Since the list contain all of the individual's bank accounts, it also contains savings accounts and other accounts. Although in this research, the goal is to study the spending behaviour of an individual and therefore we want to use an account that is connected to a debit or credit card. The list of bank accounts did not contain any information on the purpose of each account apart from a name selected by the individual. Therefore it is assumed that the account with the most amount of transactions is the individual's primary account, and that this one is connected to a debit or credit card.

Each element in the list of transactions for an account contains a date for when the transaction was processed, the amount, the account balance after the transaction has been made and a short description. These elements are the ones that are processed in order to calculate statistics and extract the features to be used in the clustering algorithms. To study current spending behaviour of individuals we decided to use transactions made within the last three months before the data was collected from the bank when calculating the features. In this way it is also possible to study if an individual changes their spending behaviour, so long as new data is continuously collected from the banks.

3.2.2 Calculating features

The features extracted from the raw data can be divided into two groups depending on how they are calculated. We have decided to call the groups the category statistics and the general statistics. The category statistics, not to be confused with categorical features, got the name from the fact that the different transactions are being classified into different categories based on the nature of the transaction, such as if the transaction is an ATM withdrawal or a purchase in a grocery store. The general statistics group on the other hand contains features such as the average balance kept or the spendings in each week of the month. A more detailed description of the two groups of features and how they are calculated follow below.

Category statistics

When the transactions are classified into categories based on the nature of the transaction, different relationships exist between the

categories. Some categories are inherently mutually exclusive, for example an ATM withdrawal cannot belong to any other category. Other transactions can belong to several categories, such as the bank payment and telecommunication categories if an invoice is paid through a bank payment. Therefore the number of categories that a transaction belong to is dependent on the nature of the transaction. The categories selected are based on what we believed to give interesting results in the clustering algorithms based on studying the raw data. For a more detailed description of the different categories and the number of features extracted for each category see Table 3.1.

Category	**#Features**	**Description**
Grocery	2	Statistics on purchases made at grocery stores. The two features are the number of purchases and the total amount spent.
Swish	4	Swish is a Swedish mobile payment system. The features are the number of transfers in and out from the account and the amounts.
Bank transfers	4	Statistics on the number of bank transfers in and out from the account and the amounts.
Travel	2	The number of purchases and total amount made to travel agencies or other companies working within the travel industry.
ATM	2	The number of ATM withdrawals and the total amount.
Systembolaget	2	Systembolaget is the Swedish government owned chain of liquor stores. This category contains statistics on the number of purchases and total amount.
Telecommunication	2	The number of transactions made to companies within the telecommunication industry.

Gambling	4	Statistics on the amount of transactions made to and from companies within the gambling industry. The features are the number of transactions to and from the account in question and the amounts.
Bank payments	2	Statistics on the number of bank payments and the total amount made from the account.
Hospitality	2	The number of purchases and the total amount made to companies within the hospitality business, such as restaurants or bars.
Convenience store	2	The number of purchases and the total amount made at convenience stores.
Autogiro	2	Autogiro is the automatic payment system in Sweden that can be set up to pay a amount of money at a specific date, generally used to pay recurring payments. Contains statistics on the number of payments and the total amount.
Gas	2	The number of payments and the total amount made to gas stations.
Other	2	The number of purchases and the total amount of transactions that were not classified into any previous category.

Table 3.1: The different categories that the features are split into.

The transactions are classified into the different categories using the short description string present for each transaction. The content of this description string is generally selected by the company at which the purchase was made, or for bank transfers selected by the individual. To classify a transaction into a different category the content of the description string tries to be matched against words or company names common for each category. Swedish bank payments, for example, usually contains the phrases *bg-bet* or *pg-bet* which stands for *bankgiro* and *postgiro*, respectively, and are two

payment systems used in Sweden. Furthermore, transactions that are classified into the gambling category commonly contains the terms *casino*, *bingo* or *slots* while transactions that adhere to the hospitality category commonly contains words such as *bar*, *restaurant* or *café*.

To further improve the categorization of transaction the names of the most common companies within some categories are also tried to be matched against the description string of each transaction. A web data extraction tool was also developed using Python in order to extract data from the web site *allabolag.se*[9], which contains lists of Swedish companies within different industries. The industries that were extracted are hospitality, travel and tourism, convenience stores, gambling companies and telecommunication companies. The names of companies working within the different fields were then tried to be matched against the description string of each transaction.

General statistics

The general statistics group contain features such as the total number of transactions, average spending per purchase or the total spending for each day of the week. Since each transaction also contains the account balance after the transaction has been made we can calculate statistics on the account balance. The features in this group has further been divided into groups that share similar characteristic for readability, see Table 3.2 for more information.

Since most months of the year do not contain an even number of weeks, the features in the spending per week group are not actual weeks. Instead, a month is divided into four parts. Different dates will be placed into different groups based on the number of the date and the total number of days in that month using some arithmetic operators. This entails that some of the calculated weeks will be 8 days long when there are more than 28 days in the month. When calculating spending on different days of the week the *datetime* library, which is part of the Python standard library, has been used.

[9] `http://www.allabolag.se/`

Group	#Features	Description
Transaction statistics	6	Statistics on the total number of transactions made and the average number of transfers per month. The transactions are further divided into incoming transactions and outgoing transaction and the total amount for each of these two.
Balance statistics	2	Statistics on the average balance kept in the account and the minumum account balance.
Spending per week	8	The number of transactions made for each week of the month and the total amount spent during those weeks
Purchase statistics	2	The total amount of purchases made and the average spending per purchase. For these features transactions that are bank transfers and payments are not included
Spending on weekday	14	This group contains statistics on the number of transactions and the total spending on each day of the week.

Table 3.2: The different groups of general statistic features.

The full feature data set

After calculating all the statistics for each individual the features from both groups are merged into a full feature data set containing a total of 66 features. To reduce the computational time of the feature extraction, predominantly caused by the large amount of string parsing and matching to calculate the category features, multiprocessing is used. This is achieved using the Python library *multiprocessing*[10], which is part of the Python standard library.

[10] `https://docs.python.org/2/library/multiprocessing.html`

3.3 Preprocessing

The preprocessing module in Scikit-learn, mentioned in Section 3.1.3, has been used to process the data before inputting it to the clustering algorithms. The features of the data were scaled by using the normalize[11] function on the column axis of the matrix. The L2 norm was used to divide each element by the length of the corresponding column vector.

3.4 Dimensionality reduction

In this study two different types of dimensionality reduction has been employed. The Scikit-learn library, described in Section 3.1.3, contain dimensionality reduction tools and both of the methods used are contained in this library.

3.4.1 Feature selection

The first method that was implemented was the VarianceThreshold function[12]. This feature selection method removes all features with a variance lower than a given threshold.

3.4.2 Principal component analysis

Apart from the feature selection method, principal component analysis (PCA)[13] was also used. PCA reduces the amount of dimensions of the data by projecting it onto a lower dimensional space.

3.5 Clustering algorithms

In this section, the clustering algorithms used in the experiments are presented. Two different types of algorithms are used, one hierarchical

[11] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html

[12] http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

[13] http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

and one partitional. Both of the algorithms are crisp clustering algorithms, i.e. a data point can only belong to one cluster once a clustering solution has been found. Crisp clustering algorithms are used because they tend to be less complicated compared to fuzzy algorithms. There may exist algorithms that would yield better results than the ones used in this study. Despite this, we have chosen to employ two commonly used and fairly straightforward algorithms because the primary objective of this study is to prove the presence of patterns in the data, and not to compare the performance of different algorithms.

3.5.1 Hierarchical algorithm

In Section 2.2.2, two different approaches to constructing the dendrogram using a hierarchical algorithm are described, either in a top-down or bottom-up manner. In this experiment, functions from the clustering package[14] contained in the SciPy library, see Section 3.1.2, are used. These algorithms perform bottom-up, also called agglomerative, clustering. The agglomerative clustering algorithms start off with every single data point in its own cluster and then gradually merge them together using a given method for calculating the distance between clusters. This is done until all the data points are merged into one cluster, i.e. the root node of the dendrogram.

The SciPy implementation supports 7 different methods for calculating the distance between clusters. Test have been conducted using several of these different methods but the primary distance method used is the ward method, which uses the Ward variance minimization algorithm. Apart from functions to execute the hierarchical algorithm and generate a dendrogram, the library also contains functions to cut the dendrogram and return flat cluster assignments based on a given criterion.

3.5.2 K-means algorithm

The second clustering algorithm used in this study is the partitional K-means algorithm. As mentioned in Section 3.1.3, the implementation

[14] `https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html`

utilized is part of the Scikit-learn library[15]. For the K-means algorithm the number of cluster to generate, k, has to be given as an input to the function. Apart from number of clusters the implementation also supports settings for different methods of centroid initialization and different algorithms to solve the problem.

3.6 Evaluation of clustering

To evaluate the clusters generated by the clustering algorithms three different internal cluster validation scores were used. Internal validation metrics are used when there is no prior knowledge about the ground truth labels or structure of the data. For some of the individuals in the data set, knowledge about the credit score calculated by a credit institute exist. Therefore these credit scores were also used to evaluate the formed clusters.

3.6.1 Silhouette Coefficient

The function used to calculate the silhouette score of a specific cluster assignment comes from the Scikit-learn API[16].

$$S(i) = \frac{(b(i) - a(i))}{max(a(i), b(i))} \tag{3.1}$$

By using Equation 3.1 the silhouette coefficient for a sample i is calculated. In this equation $a(i)$ is the mean intra-cluster dimension, i.e. the average distance between point i and all other data points in the same cluster, while $b(i)$ is the minimum mean distance to any other cluster. This means that $b(i)$ will be the mean distance between the sample i and all points in the closest neighboring cluster. Using this equation the silhouette coefficient for a sample will be in the range $[-1, 1]$, where negative values indicates a sample that is wrongly clustered and values close to 0 indicates overlapping clusters. The silhouette score for an entire clustering assignment is the mean silhouette coefficient over all samples [30].

[15] `http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

[16] `http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html`

3.6.2 Calinski-Harabasz Index

The Calinski-Harabasz index is calculated as the average ratio between the inter- and intra cluster distances [29], [30]. Equation 3.2 shows a definition of the index where n is the number of data points and k is the number of clusters. The matrices B and W represents the inter- and intra cluster distances, respectively, and $Tr()$ calculates the sum of squares for each cluster [29]. The implementation used in this study is part of the Scikit-learn library[17].

$$V_{CH} = \frac{(n-k)Tr(B)}{(k-1)Tr(W)} \tag{3.2}$$

3.6.3 Dunn Index

To calculate the Dunn Index an implementation has been developed. The implementation divides the minimum distance between two points in different clusters by the maximum distance between two points in the same cluster. This means that Dunn Index has a value between zero and infinity and should be maximized.

3.6.4 Credit scores

The credit scores used represent the probability of an individual defaulting on their loan within one year, i.e. a higher score represents a higher risk for the company issuing the loan. After a set of clusters have been found, the credit scores are used to calculate the average credit score and standard deviation for all individuals in the same cluster. This is then plotted in order to examine if there are any differences in credit score between the different clusters.

[17] `http://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabaz_score.html`

Chapter 4

Results

In this chapter, the results of the experiments will be presented and discussed. The outcome of both clustering algorithms and the corresponding cluster validity scores are presented separately.

4.1 PCA

As mentioned in Section 3.4, PCA has been used to reduce the dimensions of the data. When reducing the data down to two or three dimensions, we are able to plot it in order to visualize the data. Figure 4.1 visualizes the data using 2 principal components, while Figure 4.2 shows the results when using 3 principal components.

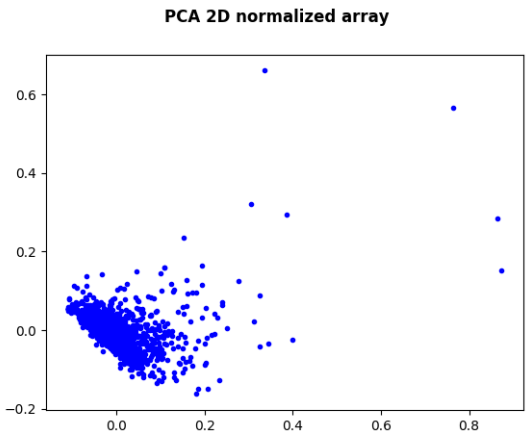

Figure 4.1: The data using 2 principal components.

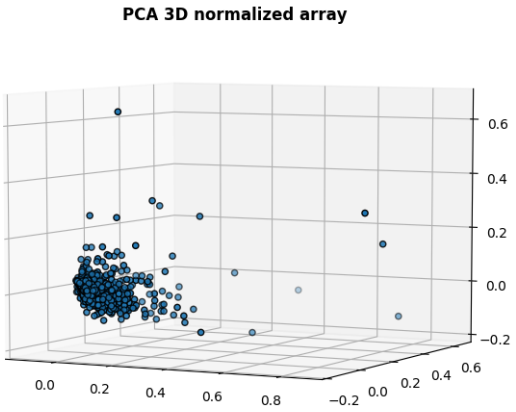

Figure 4.2: The data in 3 principal components.

4.2 Hierarchical clustering

The Ward linkage method was the primarily used metric to calculate the distance between two clusters, even though the SciPy implementation supports a variety of different methods. Therefore the majority of results and figures presented in this section will have been calculated using the Ward linkage method, and when a result utilizing any other method is presented this will be clearly declared.

4.2.1 Dendrograms

In this section a selection of the dendrograms generated in this study is presented. Some of the dendrograms contains the entire tree structure, where each leaf is an individual data point, while some of the dendrograms have been truncated.

Figure 4.3 visualize the dendrogram generated when using the Ward linkage method on the entire feature matrix, i.e. no dimensionality reduction has been applied to the data set. This figure displays the entire dendrogram, i.e. each leaf corresponds to a data point in the data set. When the data set grows, the dendrogram can be hard to analyze and therefore Figure 4.4 shows a truncated version of the same linkage matrix. For a larger figure of the entire dendrogram, see Appendix B Figure B.1. In Figure 4.4, the last 10 merging of

clusters are the only ones shown, and the rest of the merges are contracted into leaf nodes. Since a bottom-up hierarchical algorithm is used, this corresponds to starting at the maximum distance in the full dendrogram and only plotting the first 10 separations of clusters we come across when traversing down the distance axis. The amount of leaf nodes to show in the truncated dendrogram can be varied and in the same way the dendrogram can be cut at any level to retrieve a desired amount of clusters. The numbers within parentheses in Figure 4.4 indicates the amount of data points in that leaf node while a number without parentheses indicates the index for a single data point, in case a leaf node only contains one data point.

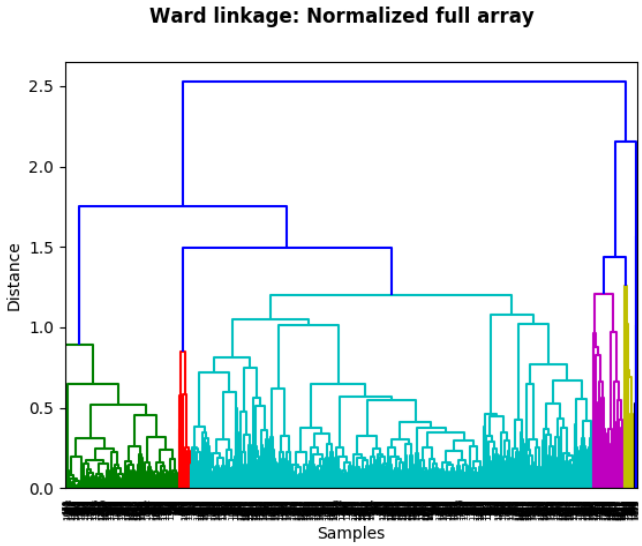

Figure 4.3: The dendrogram generated using Ward linkage without any dimensionality reduction.

Dendrograms have also been generated for data sets where PCA has been used to reduce the amount of dimensions of the data to 2 and 3 principal components. Figure 4.5 and 4.6 shows the truncated dendrograms for the data sets with 2 and 3 principal components, respectively.

Observing the two most distant clusters of the dendrogram in Figure 4.3, we can see that there is a fairly large part of the data points in the left cluster and a smaller amount in the right cluster. From the

same figure, and also from the truncated version of the dendrogram in Figure 4.4, we can see that the distances between the data points in the right cluster is shorter than the distances in the left cluster. These observations correspond well with what can be seen in the 2 and 3

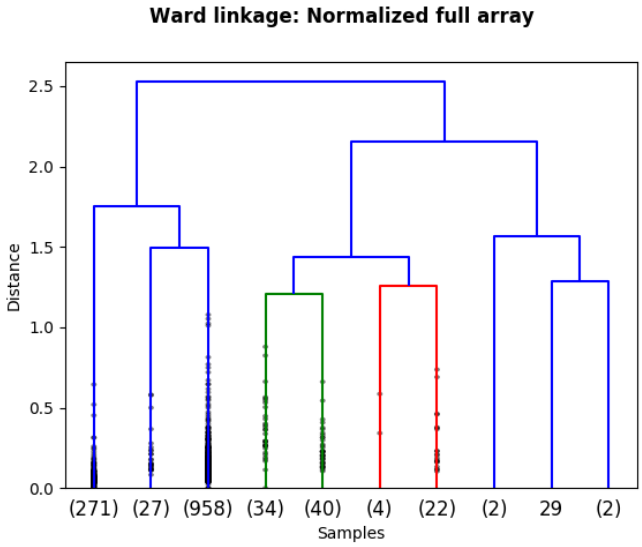

Figure 4.4: Truncated dendrogram from using Ward linkage on full feature matrix.

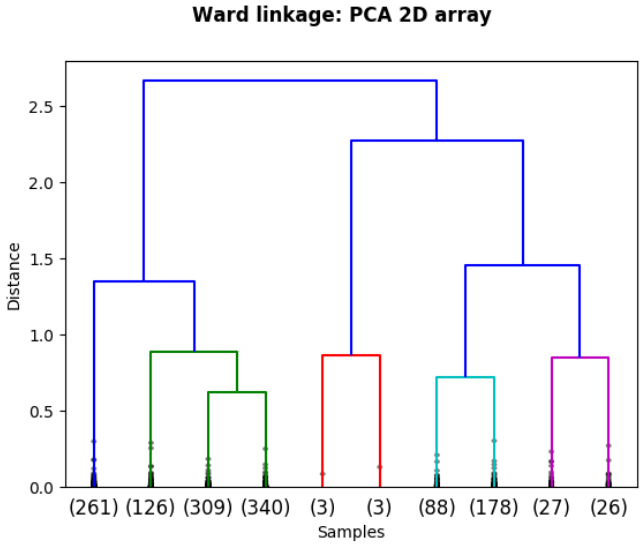

Figure 4.5: Truncated dendrogram for a data set with 2 principal components.

dimensional plots of Figure 4.1 and 4.2, where there is a large group of data points closely clustered and a smaller amount of data points with increasing distances to the large group. Similar arrangements can be observed in the dendrograms for the 2 and 3 dimensional data sets, displayed in Figure 4.5 and 4.6, although in these dendrograms it appears as if a few more data points are being classified to the right cluster.

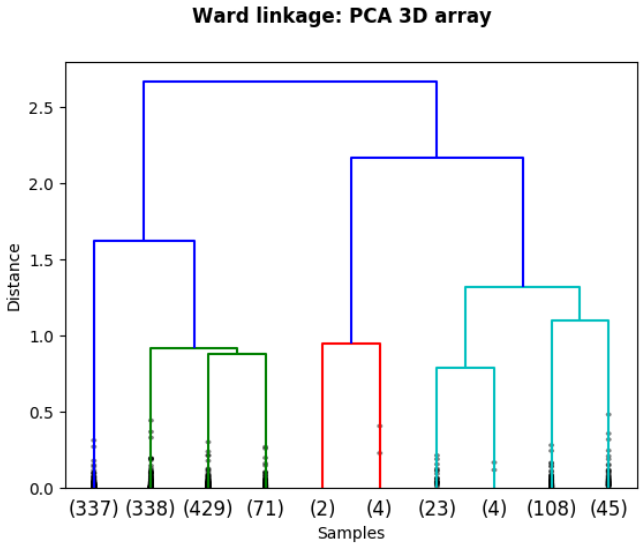

Figure 4.6: Truncated dendrogram for a data set with 3 principal components.

As previously mentioned, other linkage methods for the hierarchical clustering algorithm were also tested. Figure 4.7 shows a dendrogram generated using the Complete linkage method on a data set with 2 principal components. Figure 4.8 shows the resulting clustering assignments for two clusters when using the Complete linkage method.

If we compare the dendrogram generated using the Complete linkage method in Figure 4.7 to the previously analyzed dendrograms generated using the Ward linkage method, we can see that they differ quite substantially. The reason for this is that the Complete linkage method has a characteristic where a large cluster tend to grow even larger, compared to the Ward method where the sizes of the clusters

remain more equal. As previously mentioned, a number within parentheses in the truncated dendrograms represents the number of data points in that leaf while a number without parentheses represents the label of an individual data points. This means that the right cluster of the dendrogram generated using the Complete method only contains 6 data points.

4.2.2 Visualizing clusters

When reducing the amount of dimensions of the data that is inputted to the hierarchical algorithm down to 2 or 3 principal components, we are also able to visualize the clusters found by the algorithm. In Figure 4.9 the results of the hierarchical clustering algorithms using Ward linkage and three clusters are presented. In Figure 4.10 we see the same amount of clusters but for three dimensions, while Figure 4.11 shows the results for 6 clusters in two dimensions.

Looking at Figure 4.8, that visualizes the clustering assignments of the Complete linkage method with 2 clusters in a two-dimensional space, we can see that the 6 most distant outliers has been clustered into one cluster while the rest of the data points belongs to the other

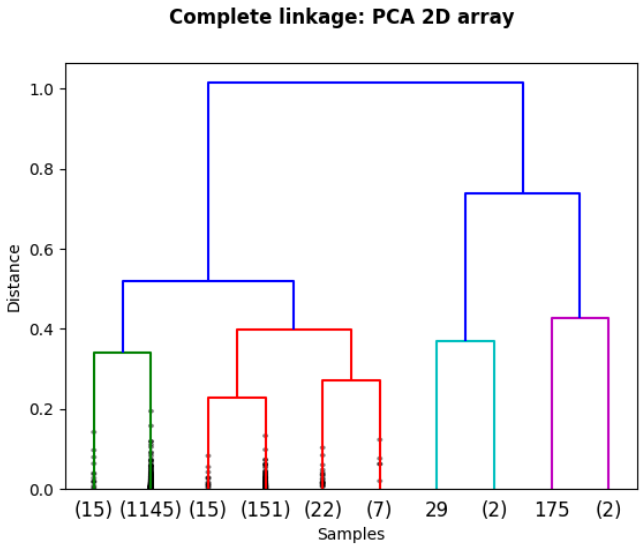

Figure 4.7: Dendrogram using Complete linkage on 2 dimensional data.

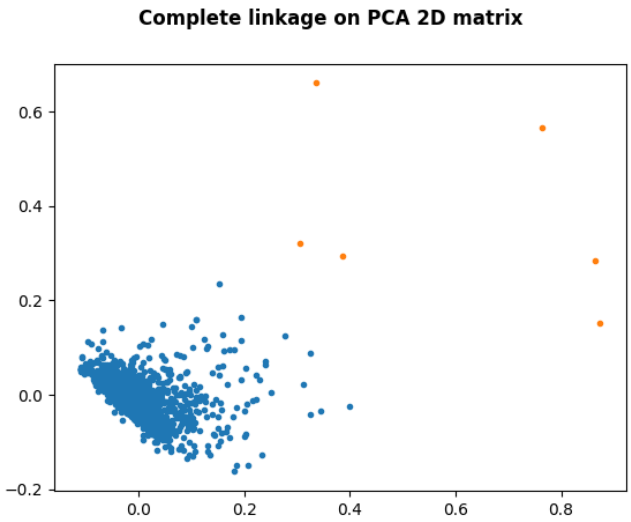

Figure 4.8: Complete linkage on 2 dimensional data with 2 clusters.

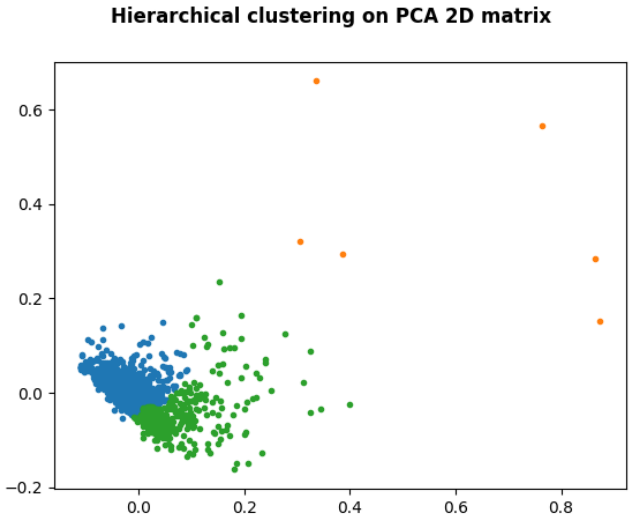

Figure 4.9: Clusters calculated using the Ward linkage method on a data set with 2 principal components.

cluster. This result differs from the Ward linkage method where at least 3 clusters are needed in order for the 6 most distant outliers to be classified in their own cluster, which is visualized in Figure 4.9 and 4.10.

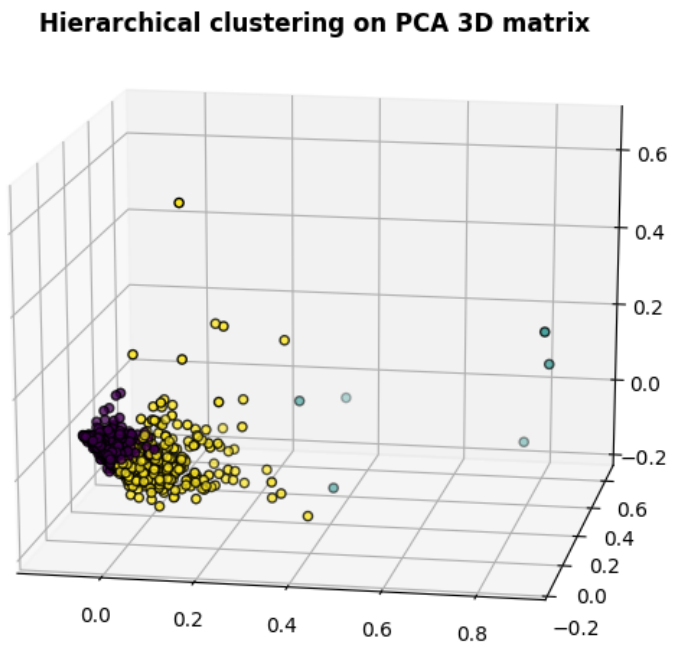

Figure 4.10: Clusters calculated using the Ward linkage method on a data set with 3 principal components.

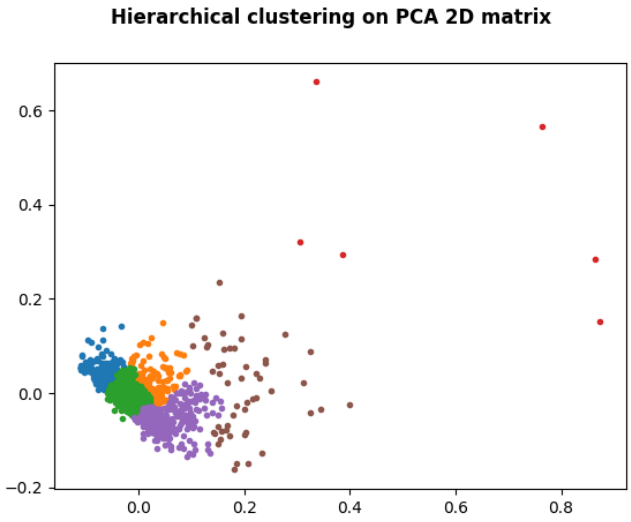

Figure 4.11: Six clusters using the Ward linkage method on a data set with 2 principal components.

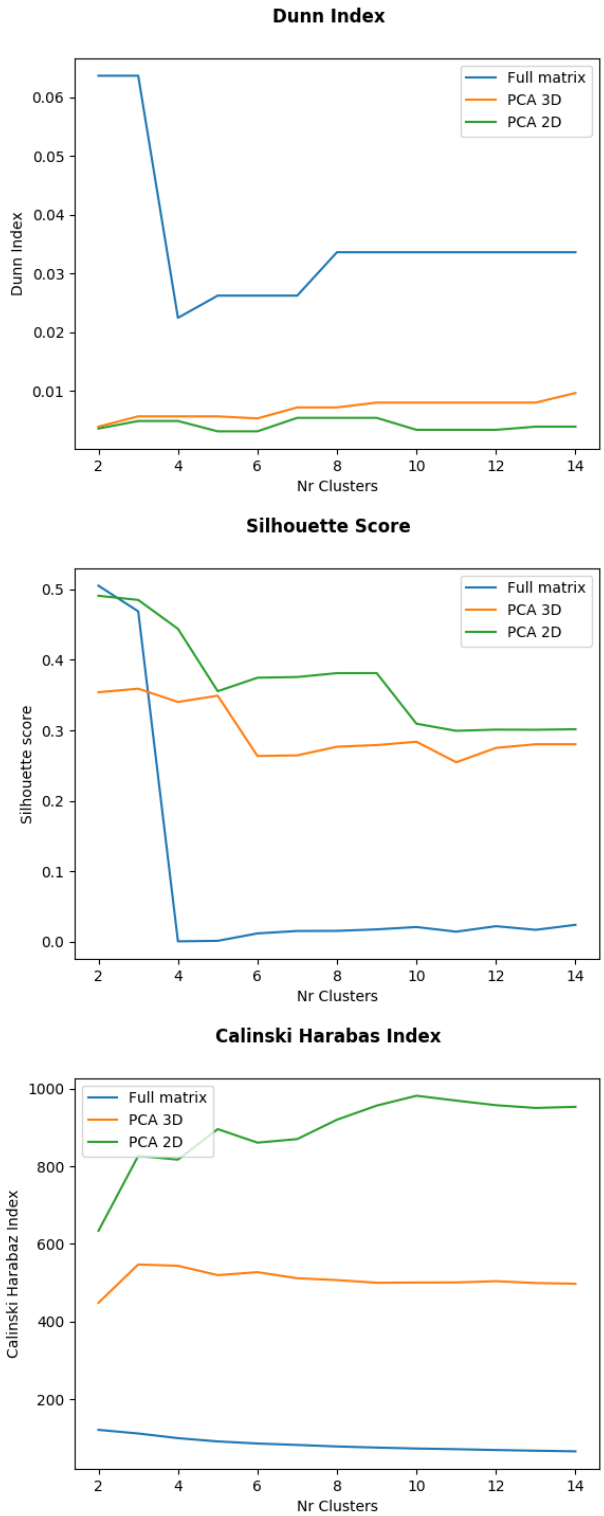

Figure 4.12: Cluster validity indices for different number of clusters and dimensions using the Ward method.

4.2.3 Cluster validity indices

Internal cluster validity indices have been calculated for different number of clusters on the full feature matrix and the data sets with 2 and 3 principal components. The results of these validity indices that have been calculated are visualized in Figure 4.12, where each line represents a different data set and the x-axis represent the number of clusters.

The internal cluster validity indices has, as previously mentioned, been used to find the optimal amount of clusters. In Section 2.2 it is mentioned that there is no mutual understanding regarding what validation measure to use in different situations. The way that the different measurements are calculated, as seen in Section 3.6, also means that the measurements are difficult to compare against each other. It is also important to note that internal validity measurements can be quite vague since they are not based on any ground truth. Therefore the results of the measurements should not be considered as any definite proof to solely rely on, it is also important to subjectively evaluate the results depending on the situation and usage of the clustering results.

Despite this, three different cluster validity indices along with statistics on credit score has been calculated to evaluate the clusterings. The silhouette index can only take on values in the range $[-1, 1]$, while the Dunn and Calinski-Harabasz does not have an upper limit. The Dunn index simply divides the minimum distance between two clusters by the maximum distance within a cluster. This means that when using PCA to reduce dimensions, some variance may dissipate and lead to a different Dunn index. Therefore the Dunn index values between different data sets, which we in some way get when using PCA, cannot be directly compared. Instead we can study the shape of the plots for the different data sets and see if we find some similar patterns. The Calinski-Harabasz index has a similar characteristic.

For the Dunn index in Figure 4.12 we can see that the value appears to increase slightly for the 2 and 3 dimensional data sets as the number of clusters increase, while for the full matrix the index has a maximum value at 2 clusters and is then decreasing. For the silhouette index in

the same figure we can see that the maximum value for all data sets appear to be for 2 clusters, which matches the observations for the Dunn index of the full matrix. The Calinski-Harabasz index differs from the Dunn index in that it calculates an average over all clusters instead of just looking at the two most extreme values. We can see that the Calinski-Harabasz index for the 2 and 3 dimensional data sets is increasing slightly as the number of clusters increase while the full feature data set decreases.

4.2.4 Credit scores

By using the credit scores available for some of the individuals in the data set, we have calculated the average credit score for all individuals in the same cluster. Figure 4.13 shows the average credit scores for individuals when splitting the dendrogram generated using the Complete linkage method into two and three clusters using the data set comprised of two principal components. In Figure 4.15 statistics for the three data sets of different dimensions are visualized. In this figure the ward linkage method has been used to calculate the clustering assignments. The numbers within parentheses on the x-axis represents the number of data points in each cluster. Figure 4.14 shows the credit score statistics of the full feature data set when a larger number of clusters are used.

Looking at the credit scores for different clusters in Figure 4.15 we can observe that clusters containing a smaller amount of data points tend to have a higher average credit score. Especially with three clusters we can see that the small cluster containing the 6 most distant outliers has a quite significantly higher average credit score than the other clusters. As previously discussed, the results of using the Complete linkage method means that only two clusters are needed to find the outliers. This can also be seen in Figure 4.13 where it can be observed that with 3 clusters there is a group of three outliers with an ever greater credit score than the average for the 6 most distant outliers.

In Figure 4.14 we can observe that when a larger amount of clusters are used, such as 6 clusters, we end up with small clusters that has a lower average credit score than the largest cluster. Even if the differ-

ence is relatively small, this could be interpreted as finding a cluster of people that has a lower credit score than the general average and therefore a more beneficial spending pattern. In the same figure we can also see that the outliers are split up into even smaller clusters, similar to the behaviour of the Complete linkage method in Figure 4.13.

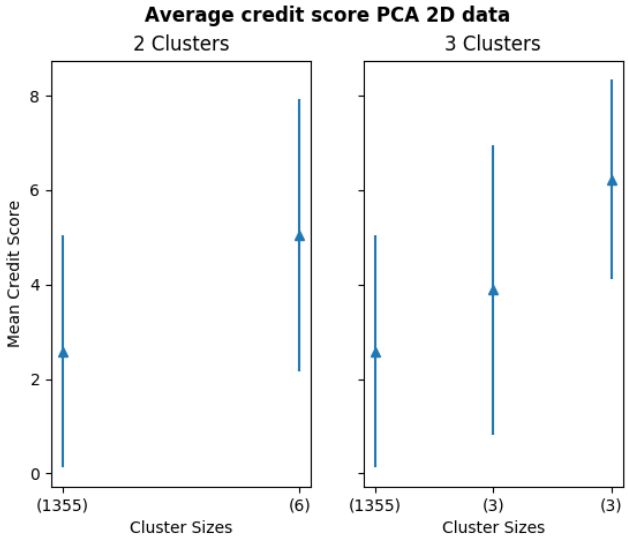

Figure 4.13: Credit score statistics using Complete linkage on 2 dimensional data.

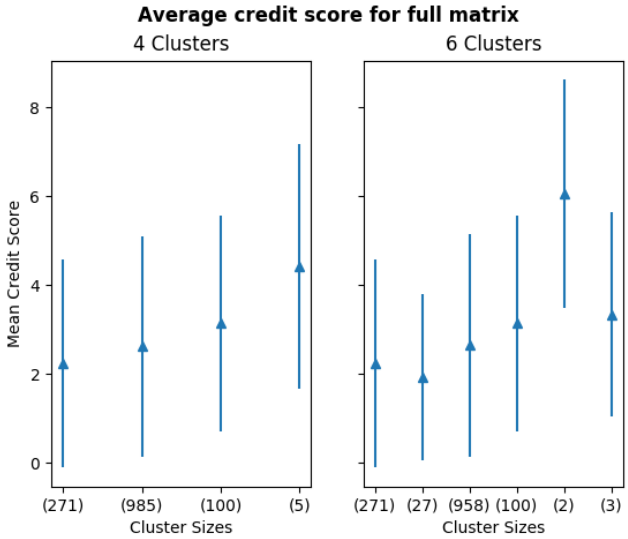

Figure 4.14: Credit score statistics on the full feature matrix.

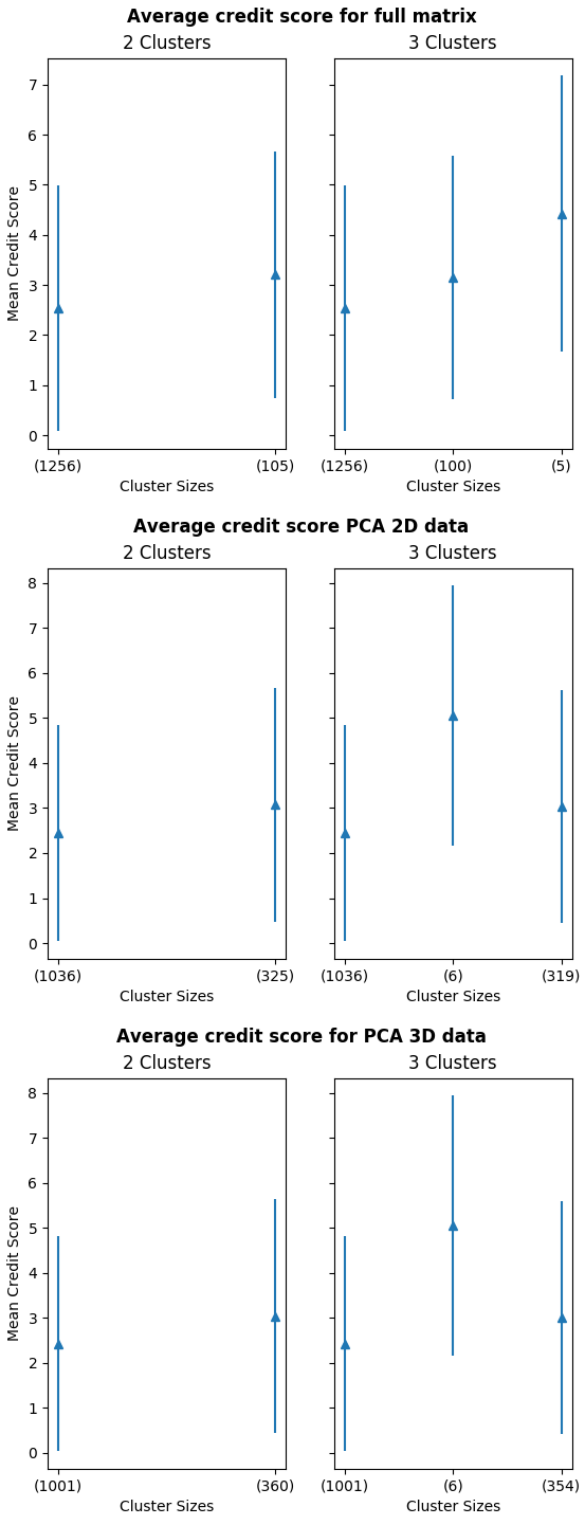

Figure 4.15: Credit score statistics when using Ward linkage.

4.3 K-means clustering

The K-means clustering method is the second type of clustering algorithm that has been used in this study. This method differs from the hierarchical method in the sense that K-means is a partitional method, i.e. no hierarchical relationship exist between the clusters. This means that no visualization similar to the dendrogram can be generated when using the K-means method. Instead we will have to rely on other techniques to visualize the results of the clustering algorithm.

For the 2 and 3 dimensional data sets, it is possible to plot the individual data points and the clusters they have been assigned to, but for the full feature matrix this is not possible. Therefore the results when using the entire data set without dimensionality reduction has been evaluated through the cluster validity indices and the credit score statistics.

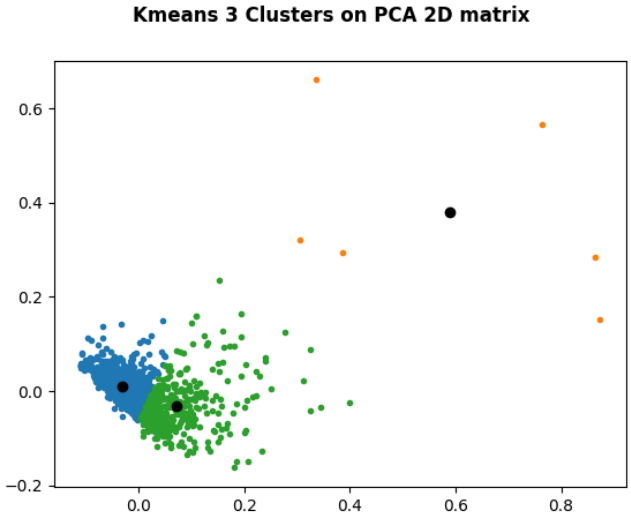

Figure 4.16: K-means results using 3 clusters.

4.3.1 Visualizing clusters

In this section figures showing the clustering assignments for the 2 and 3 dimensional data sets are presented. Figure 4.16 shows the results when executing the K-means algorithm on the data set with

2 principal components and 3 clusters. The black dots in the plot represents the centers of the different clusters. In a similar manner, Figure 4.17 show the results of the algorithm on the 3 dimensional data set with 2 clusters. Figure 4.18 shows the results for a larger amount of cluster, to be exact 6 clusters, on the 2 dimensional data set.

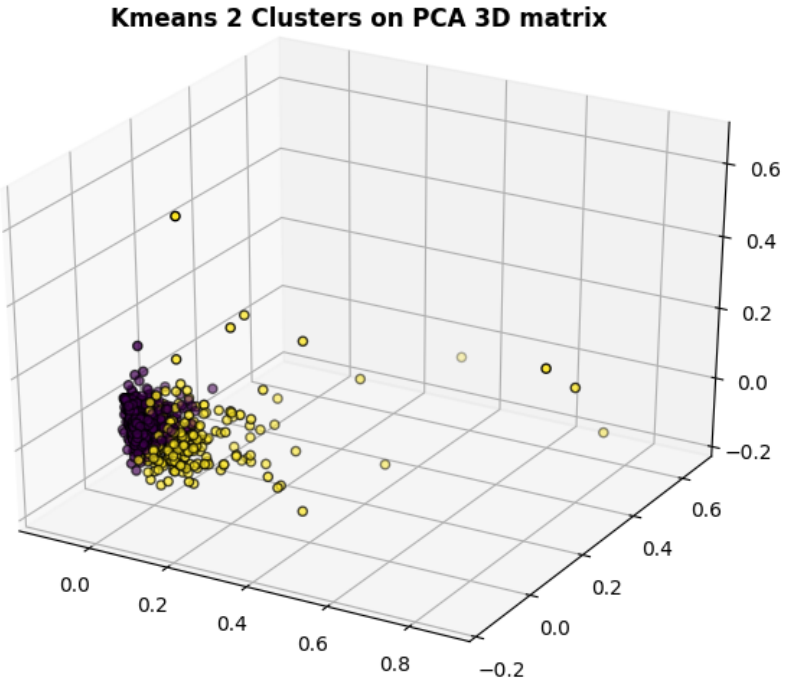

Figure 4.17: K-means results using 2 cluster on the 3 dimensional data set.

The clustering result for 3 clusters in a two-dimensional space using K-means, visualized in Figure 4.16, resemble the results when using Ward linkage method on the same data set, featured in Figure 4.9. We can also observe, in Figure 4.17, that for two clusters in 3 dimensions the K-means algorithm fails to put the 6 most outliers in a cluster of their own, just like the results using Ward linkage.

4.3.2 Cluster validity indices

Just as for the hierarchical clustering method, internal cluster validity indices has been calculated for a number of different cluster configu-

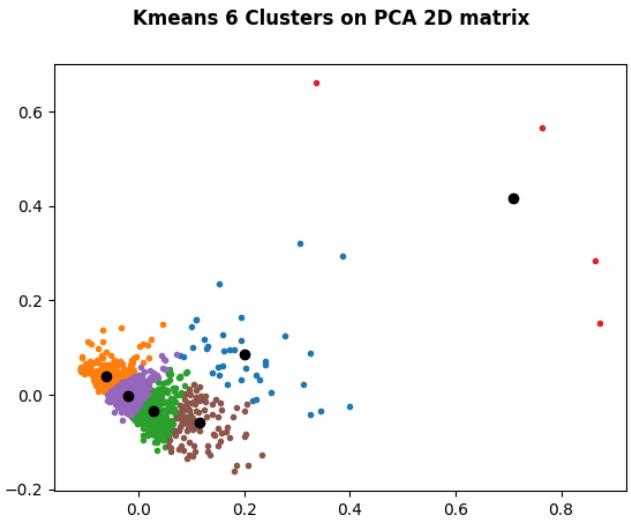

Figure 4.18: K-means results using 6 cluster on the 2 dimensional data set.

rations on the three data sets of different dimensions. The same three validity indices used for hierarchical clustering has also been used to evaluate the K-means method. In Figure 4.19 the validity indices over different number of clusters have been plotted. Each plot contains values for each of the three data sets of different dimensions.

Comparing the cluster validity indices for the hierarchical clustering in Figure 4.12 to the ones for K-means in Figure 4.19, we can see that the plots share some similar patterns. Just as for the hierarchical clustering, K-means achieves a maximum silhouette score for all data sets when the number of clusters is equal to two. The same is also true for the Dunn and Calinski-Harabasz index when clustering the full data set, while the value for the 2 and 3 dimensional data sets increases as the number of clusters grow.

4.3.3 Credit scores

By using the existing credit scores, the average score and the standard deviation for each cluster is calculated, just as for the hierarchical clustering method. Figure 4.20 shows the results of these calculations when using 2 and 3 clusters for the full feature data set and the two

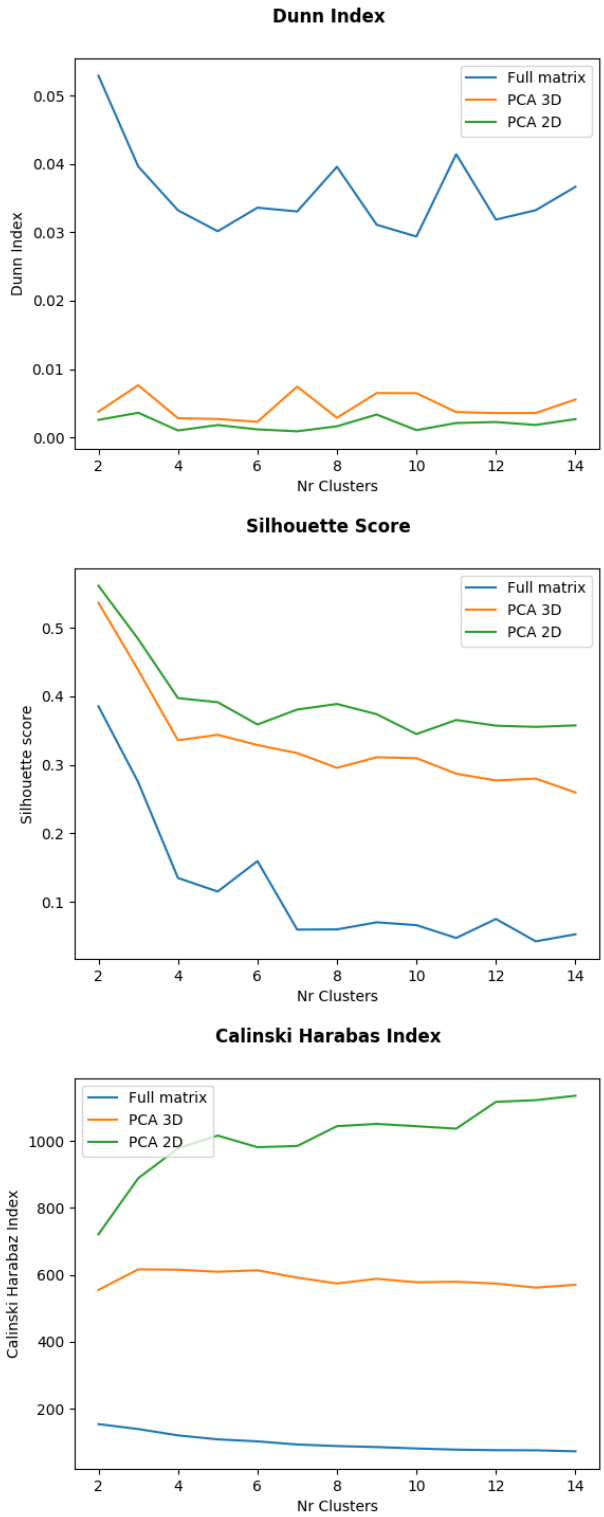

Figure 4.19: Cluster validity indices for different number of clusters using K-means.

PCA reduced data sets. Just as for the corresponding plots for the hierarchical method, the numbers on the x-axis represents the number of data points in each cluster.

Looking at the credit score statistics for K-means in Figure 4.20, we observe a similar behaviour as when using the Ward linkage in Figure 4.15. Just as with the Ward linkage, smaller clusters appear to have a slightly higher average credit score and three clusters are needed in order to capture the most distant outliers in a cluster of their own.

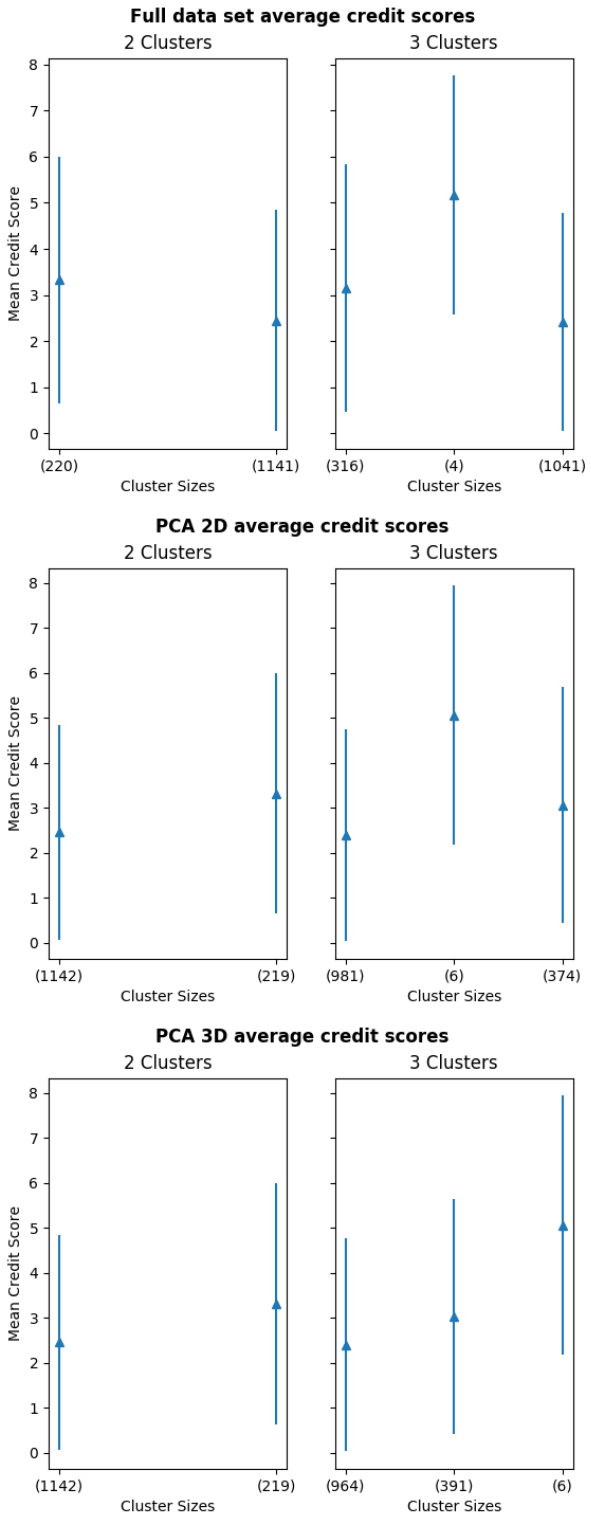

Figure 4.20: Credit score statistics for K-means algorithm.

Chapter 5

Discussion and conclusion

This chapter will start off by discussing some obstacles and problems that were encountered during the study. After this, a comparison between K-means and hierarchical clusering follow before the general conclusions are discussed.

5.1 Obstacles

During this study, some obstacles were encountered. Some of these changed the course of this study, and in effect also the outcome. The first obstacle, and perhaps also the major one, is regarding the raw data. Before the study commenced, there was no data available and little information on the informative content of the data. The reason for this is that the collaborating company is newly started and launched their business in the early stages of this project. Therefore data has gradually been collected throughout the study, and the final data set merely contain about 1300 individual data points.

Since the informative content of the data initially was unknown, we had hoped that each transaction in the data would contain some specific information that would make it easy to relate a transaction to a company, such as a organizational number. Since this is not the case, and a description string is the only source of information in each transaction, this made the feature extraction process harder. It was found that companies do not necessarily put the full company name in the description string, and occasionally they would put a different name. Therefore it proved difficult to classify a transaction

to a specific category and the results are that around 40% of all transactions remain unclassified. This could be solved by manually trying to classify transactions, but due to time restrictions this was unfeasible.

Another problem with the data that was discovered during the study was that some individuals had made little to no transactions. The reason for this may be that the individual do not use their debit card, or they may not even own one. As mentioned in Section 3.2.1, there is no information available in the data on potential debit cards being connected to a specific account, which is why the assumption is made that the primary account is the one with the most amount of transactions. This assumption may be incorrect if the amount of transactions are low. During the collection process of data from the individual, the individual is responsible for choosing what bank the data is collected from. Therefore, if the individual has accounts in multiple banks, the individual may choose to not select their primary bank. No matter what the cause is, it is difficult to study the spending behaviour of an individual if there is not enough data.

In section 3.4 it is mentioned that a feature selection algorithm was tested using a variance threshold to remove low-variance features. This method proved to be ineffective because after the feature scaling process, most of the features had a fairly low and similar variance. This can also be seen in the data sets which has been reduced using PCA in Figure 4.1 and 4.2, where most data points appear to be fairly tightly clustered together with a few rare outliers. This may be due to the fact that the individuals in the data set do not represent the general population. Instead they represent people that apply for loans without insurance, and these people may share common characteristics in their spending behaviour. Another reason may be that the data is only collected from individuals that has already gone through a lighter form of credit assessment, effectively removing potential outliers.

5.2 Feature extraction

The structure and informative content of the raw data, described in Section 5.1, made the categorization of transactions difficult. Despite

this, it was possible to extract partial data and combined with features in the general statistics group the features extracted proved some variability between individuals. Some features were also found to have a bigger impact when it comes to the outliers. Most prominent of these were the gambling category that was shown to be responsible for causing the 6 most distant outliers.

5.3 K-means compared to hierarchical clustering

In general, the K-means algorithm and the hierarchical clustering algorithm using Ward linkage appear to generate similar clustering results for low amount of clusters. Although for a larger amount of cluster, such as the 6 clusters in Figure 4.11 and 4.18, we can observe that the cluster shapes starts looking slightly different. The reason for this may be that the K-means algorithm favour spherical clusters.

The difference between the two clustering methods is that while the K-means algorithm has to be executed again in order to increase or decrease the amount of clusters, the hierarchical clustering algorithm only has to be executed once. That is the benefit of having the hierarchical relationships between clusters. If the number of clusters has to be changed, the dendrogram is simply cut at a different level in order to retrieve the new amount of clusters. The dendrogram also provides a simple way of visualizing the clustering results, even for data sets in high dimensions. The K-means algorithm does not provide any similar technique that makes it easy to visualize the results for high dimensional data sets, which makes it harder to study and evaluate the clusters created.

5.4 General conclusions

The aim of this study is to try and identify spending patterns among individuals, that could be related to the creditworthiness of the individual. The features used to identify these spending patterns have come from lists of transactions extracted from the bank of an individual. Observing the results and the discussions made on the

experiments, we argue that this aim has been achieved to some extent. The individual data points has been assigned to different clusters and some differences in average credit score among these clusters has been identified. In Section 1.1 the proposed hypothesis is that with complete information, i.e. full knowledge of all the transactions made by individuals, it would be possible to identify spending patterns and create a sound model to assess the creditworthiness of individuals. Furthermore, the research question examined is if a sound clustering model can be created using features extracted from bank statements, i.e. incomplete information. Based on this study, we argue that even with this limited information it is possible to create a model that can identify an individual's spending patterns to some extent and cluster individuals with similar patterns to the same cluster.

Regarding the features extracted from the raw data and their informative content, we argue that with more information in the raw data it would be possible to improve the extraction of the category features and that this may improve the results of the clustering. Getting more information may be difficult since the raw data is extracted from the banks, but this could also be solved by doing manual labor and try to create a database of transactional description strings. Observing the spread of the data in Figure 4.1, we can see that it is rather limited. Most of the data points appear to belong to one big cluster with a few outliers. The reason for this may either be that most people have a fairly similar spending pattern or that the individuals in the data set do not represent the general population, but instead individuals that apply for loans without collateral. A larger data set than the one currently used may also result in a larger spread of the data points.

Observing the experimental results to decide on the best amount of cluster merely based on internal validity indices and the credit score statistics has proven difficult. As previously mentioned, most internal validity measures simply evaluates the spread of the different clusters and the compactness of a single cluster. This does not necessarily mean that a clustering assignment is beneficial for what the model is supposed to be used for. The credit score statistics may give a bit more information but we believe that to find the most beneficial amount of clusters, the subjective opinion of a person with experience and knowledge of the financial field is necessary.

Chapter 6

Future work

Based on the conclusions drawn from this study some future research problems have been identified. As previously mentioned, a more robust feature extraction process with more accurate category classifications of transactions could be developed to study if this will influence the clustering results. Another future study could be based on extending the current model with additional features not extracted from transactional data, such as amount of loans, occupation or even age, and study this impact on the resulting clusters.

We have also identified that the most distant outliers appear to have a higher average credit score. Therefore, instead of clustering all the data points, a study could be conducted to find the data points that deviate the most from the rest of the data points using techniques such as outlier analysis. This would somewhat change the purpose that is assessing the creditworthiness of all data points, but instead trying to find the ones that are less likely to be granted a loan. This could be used to identify individuals that may have to be further evaluated before a decision on the loan is made. For this objective, we have also seen that using the Complete linkage method appears to easily capture distant outliers in their own cluster. This means that the Complete linkage could be used to identify outliers.

Bibliography

[1] C. Carter and J. Catlett, "Assessing credit card applications using machine learning", *IEEE expert*, vol. 2, no. 3, pp. 71–79, 1987.

[2] A. Ben-David and E. Frank, "Accuracy of machine learning models versus "hand crafted" expert systems–a credit scoring case study", *Expert Systems with Applications*, vol. 36, no. 3, pp. 5264–5271, 2009.

[3] D. West, "Neural network credit scoring models", *Computers & Operations Research*, vol. 27, no. 11, pp. 1131–1152, 2000.

[4] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen, "Credit scoring using the hybrid neural discriminant technique", *Expert Systems with applications*, vol. 23, no. 3, pp. 245–254, 2002.

[5] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring", *Expert systems with applications*, vol. 34, no. 4, pp. 2639–2649, 2008.

[6] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring", *Journal of the operational research society*, vol. 54, no. 6, pp. 627–635, 2003.

[7] C.-L. Huang, M.-C. Chen, and C.-J. Wang, "Credit scoring with a data mining approach based on support vector machines", *Expert systems with applications*, vol. 33, no. 4, pp. 847–856, 2007.

[8] T. Bellotti and J. Crook, "Support vector machines for credit scoring and discovery of significant features", *Expert Systems with Applications*, vol. 36, no. 2, pp. 3302–3308, 2009.

[9] W. Chen, C. Ma, and L. Ma, "Mining the customer credit using hybrid support vector machine technique", *Expert systems with applications*, vol. 36, no. 4, pp. 7611–7616, 2009.

[10] K. B. Schebesch and R. Stecking, "Support vector machines for classifying and describing credit applicants: Detecting typical and critical regions", *Journal of the Operational Research Society*, vol. 56, no. 9, pp. 1082–1088, 2005.

[11] J. Kruppa, A. Schwarz, G. Arminger, and A. Ziegler, "Consumer credit risk: Individual probability estimates using machine learning", *Expert Systems with Applications*, vol. 40, no. 13, pp. 5125–5131, 2013.

[12] A. E. Khandani, A. J. Kim, and A. W. Lo, "Consumer credit-risk models via machine-learning algorithms", *Journal of Banking & Finance*, vol. 34, no. 11, pp. 2767–2787, 2010.

[13] C. Luo, D. Wu, and D. Wu, "A deep learning approach for credit scoring using credit default swaps", *Engineering Applications of Artificial Intelligence*, pp. –, 2016, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2016.12.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197616302299.

[14] J. M. Tomczak and M. Zięba, "Classification restricted boltzmann machine for comprehensible credit scoring model", *Expert Systems with Applications*, vol. 42, no. 4, pp. 1789–1796, 2015.

[15] M. Abedini, M. Abedini, F. Ahmadzadeh, F. Ahmadzadeh, R. Noorossana, and R. Noorossana, "Customer credit scoring using a hybrid data mining approach", *Kybernetes*, vol. 45, no. 10, pp. 1576–1588, 2016.

[16] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring", *Journal of Retailing and Consumer Services*, vol. 27, pp. 11–23, 2015.

[17] C.-F. Tsai and C. Hung, "Modeling credit scoring using neural network ensembles", *Kybernetes*, vol. 43, no. 7, pp. 1114–1123, 2014.

[18] B. Yi and J. Zhu, "Credit scoring with an improved fuzzy support vector machine based on grey incidence analysis", in *Grey Systems and Intelligent Services (GSIS), 2015 IEEE International Conference on*, IEEE, 2015, pp. 173–178.

[19] J. Shi, S.-y. Zhang, and L.-m. Qiu, "Credit scoring by feature-weighted support vector machines", *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 3, pp. 197–204, 2013.

[20] T. Harris, "Credit scoring using the clustered support vector machine", *Expert Systems with Applications*, vol. 42, no. 2, pp. 741–750, 2015.

[21] Z. Li, Y. Tian, K. Li, F. Zhou, and W. Yang, "Reject inference in credit scoring using semi-supervised support vector machines", *Expert Systems with Applications*, 2017.

[22] F. v. d. Heijden, R. Duin, D. de Ridder, and D. Tax, "Unsupervised learning", in *Classification, Parameter Estimation and State Estimation*. John Wiley & Sons, Ltd, 2005, pp. 215–251, ISBN: 9780470090152. DOI: 10.1002/0470090154.ch7. [Online]. Available: http://dx.doi.org/10.1002/0470090154.ch7.

[23] G. Gan, C. Ma, and J. Wu, "1. data clustering", in *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007, ch. 1, pp. 3–17. DOI: 10.1137/1.9780898718348.ch1. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718348.ch1. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch1.

[24] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning", in *The elements of statistical learning*, Springer, 2009, pp. 485–585.

[25] S. Bandyopadhyay and S. Saha, "Introduction", in *Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–16, ISBN: 978-3-642-32451-2. DOI: 10.1007/978-3-642-32451-2_1. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32451-2_1.

[26] C. C. Aggarwal, "An intoduction to cluster analysis", in *Data clustering: Algorithms and applications*. Chapman and Hall/CRC, 2013, ch. 1, pp. 1–27.

[27] S. Bandyopadhyay and S. Saha, "Clustering algorithms", in *Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–92, ISBN: 978-3-642-32451-2. DOI: 10.1007/978-3-642-32451-2_4. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32451-2_4.

[28] S. Alelyani, J. Tang, and H. Liu, "Feature selection for clustering: A review", in *Data clustering: Algorithms and applications*. Chapman and Hall/CRC, 2013, ch. 2, pp. 30–60.

[29] G. Gan, C. Ma, and J. Wu, "17. evaluation of clustering algorithms", in *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007, ch. 17, pp. 299–320. DOI: 10.1137/1.9780898718348.ch17. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718348.ch17. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch17.

[30] H. Xiong and Z. Li, "Clustering validation measures", in *Data clustering: Algorithms and applications*. Chapman and Hall/CRC, 2013, ch. 23, pp. 572–605.

[31] K.-C. Wong, Y. Li, and Z. Zhang, "Unsupervised learning in genome informatics", in *Unsupervised Learning Algorithms*, M. E. Celebi and K. Aydin, Eds. Cham: Springer International Publishing, 2016, pp. 405–448, ISBN: 978-3-319-24211-8. DOI: 10.1007/978-3-319-24211-8_15. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24211-8_15.

[32] G. Gan, C. Ma, and J. Wu, "18. clustering gene expression data", in *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007, ch. 18, pp. 323–340. DOI: 10.1137/1.9780898718348.ch18. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718348.ch18. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch18.

[33] C. C. Reddy and B. Vinzamuri, "A survey of partitional and hierarchical clustering algorithms", in *Data clustering: Algorithms and applications*. Chapman and Hall/CRC, 2013, ch. 4, pp. 87–110.

[34] G. Gan, C. Ma, and J. Wu, "9. center-based clustering algorithms", in *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007, ch. 9, pp. 161–182. DOI: 10.1137/1.9780898718348.ch9. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718348.ch9. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch9.

[35] ——, "7. hierarchical clustering techniques", in *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007, ch. 7, pp. 109–149. DOI: 10.1137/1.9780898718348.ch7. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718348.ch7. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch7.

[36] ——, "14. model-based clustering algorithms", in *Data Clustering: Theory, Algorithms, and Applications*. SIAM, 2007, ch. 14, pp. 227–242. DOI: 10.1137/1.9780898718348.ch14. eprint: http://epubs.siam.org/doi/pdf/10.1137/1.9780898718348.ch14. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898718348.ch14.

[37] H. Deng and J. Han, "Probabilistic models for clustering", in *Data clustering: Algorithms and applications*. Chapman and Hall/CRC, 2013, ch. 3, pp. 61–86.

[38] C. C. Reddy, M. A. Hasan, and M. J. Zaki, "Clustering biological data", in *Data clustering: Algorithms and applications*. Chapman and Hall/CRC, 2013, ch. 16, pp. 381–414.

[39] T. Kohonen, "The basic som", in *Self-Organizing Maps*.

[40] M. Dundar, Q. Kou, B. Zhang, Y. He, and B. Rajwa, "Simplicity of kmeans versus deepness of deep learning: A case of unsupervised feature learning with limited data", in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 883–888. DOI: 10.1109/ICMLA.2015.78.

[41] R. Mao, H. Zhu, L. Zhang, and A. Chen, "A new method to assist small data set neural network learning", in *Sixth International Conference on Intelligent Systems Design and Applications*, vol. 1, 2006, pp. 17–22. DOI: 10.1109/ISDA.2006.67.

[42] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data", *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 293–303, 2018, ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2756450.

Appendix A

Structure of the raw data

In this section, a JSON-style format of the raw data is presented in order to get a better understanding of the structure. Listing A.1 represent the structure of one data point, with some annotations on what type different variables are filled with. Some of the variables, such as *successfulLogin* and *status*, are used to indicate a complete record with information. Those variables have been used in this project in order to remove incomplete records. While the raw data contain several variables and objects, the majority have not been used in this study. The primary source of information has been the *accountList* objects, and more specifically the *transactionList* objects. In this representation, both the *accountList* and *transactionList* only contain one object. In reality these lists contain several objects, especially the transaction list which contain transactions for several months back.

```
{
  "reportTime": "",
  "userID": "",
  "scrape": {
    "scrapeReport": {
      "status": "",
      "comment": "",
      "reportNumber": <integer>,
      "scrapeDate": "",
      "isFinal": <boolean>
    },
    "userDetails": {
```

```
      "phone": "",
      "address": "",
      "name": "",
      "email": ""
    },
    "accountList": [
      {
      "kind": "",
      "holderName": "",
      "number": "",
      "currency": "",
      "iban": "",
      "balance": <integer> ,
      "transactionList": [
        {
        "onDate": "",
        "amount": <integer> ,
        "balance": <integer>,
        "params": {
          "bookingDate": "",
          "valueDate": ""
        },
      "description": ""
      }
      ]
    },
    ],
    "bank": {
      "abbreviation": "",
      "country": "",
      "name": ""
    }
  },
  "scrapeTime": "",
  "extended": {},
  "successfulLogin": <boolean>,
  "basicInfo": {
    "username": "",
    "source": "",
```

```
    "phone": "",
    "miscEntryList": [
      {
      "value": "",
      "key": "",
      "provider": ""
      }
    ],
    "address": "",
    "scrapeReport": {
      "status": "",
      "comment": "",
      "reportNumber": <integer>,
      "scrapeDate": "",
      "isFinal": <boolean>
    },
    "email": "",
    "bank": {
      "abbreviation": "",
      "country": "",
      "name": ""
    }
  },
  "bankReport": {
    "scrapeStatus": "",
    "averageMinimumBalanceMonth": <integer>,
    "cashFlow": [
      {
      "outgoing": <integer>,
      "incoming": <integer>,
      "minBalance": <integer>,
      "amountOfLoans": <integer>,
      "month": <integer>,
      "numberOfLoans": <integer>
      }
    ],
    "numberOfAccounts": <integer>,
    "averageNumberOfTransactionsMonth": <integer>,
    "averageAmountOfOutgoingTransactionsMonth":
```

```
        <integer>,
91      "amountOfLoans": <integer>,
92      "numberOfLoans": <integer>,
93      "averageAmountOfIncomingTransactionsMonth":
          <integer>,
94      "owner": "",
95      "totalNumberOfTransactions": <integer>,
96      "monthsAvailable": <integer>,
97      "bank": {
98        "abbreviation": "",
99        "country": "",
100       "name": ""
101     }
102   }
103 }
```

Listing A.1: Representation of the raw data.

Appendix B

Additional dendrograms

Some of the dendrograms may require a larger figure in order to be interpreted correctly. Therefore this section will visualize some of the dendrograms, along with complete dendrograms that did not fit in Chapter 4, in a larger format. Figure B.1 show a larger representation of Figure 4.3 presented in Chapter 4.

In Chapter 4, a truncated version of the dendrogram generated using the 2 dimensional data set is presented in Figure 4.5. Figure B.2 presents the full dendrogram for this data set. The same also applies for the Complete linkage method, where a truncated dendrogram is presented in Figure 4.7. Therefore the entire dendrogram using this linkage method is presented in Figure B.3.

Other linkage methods, apart from Ward and Complete, have also been tested in this study, but no results are presented in the report. For those who are interested in other linkage methods, a dendrogram constructed using the Average linkage method is presented in Figure B.4.

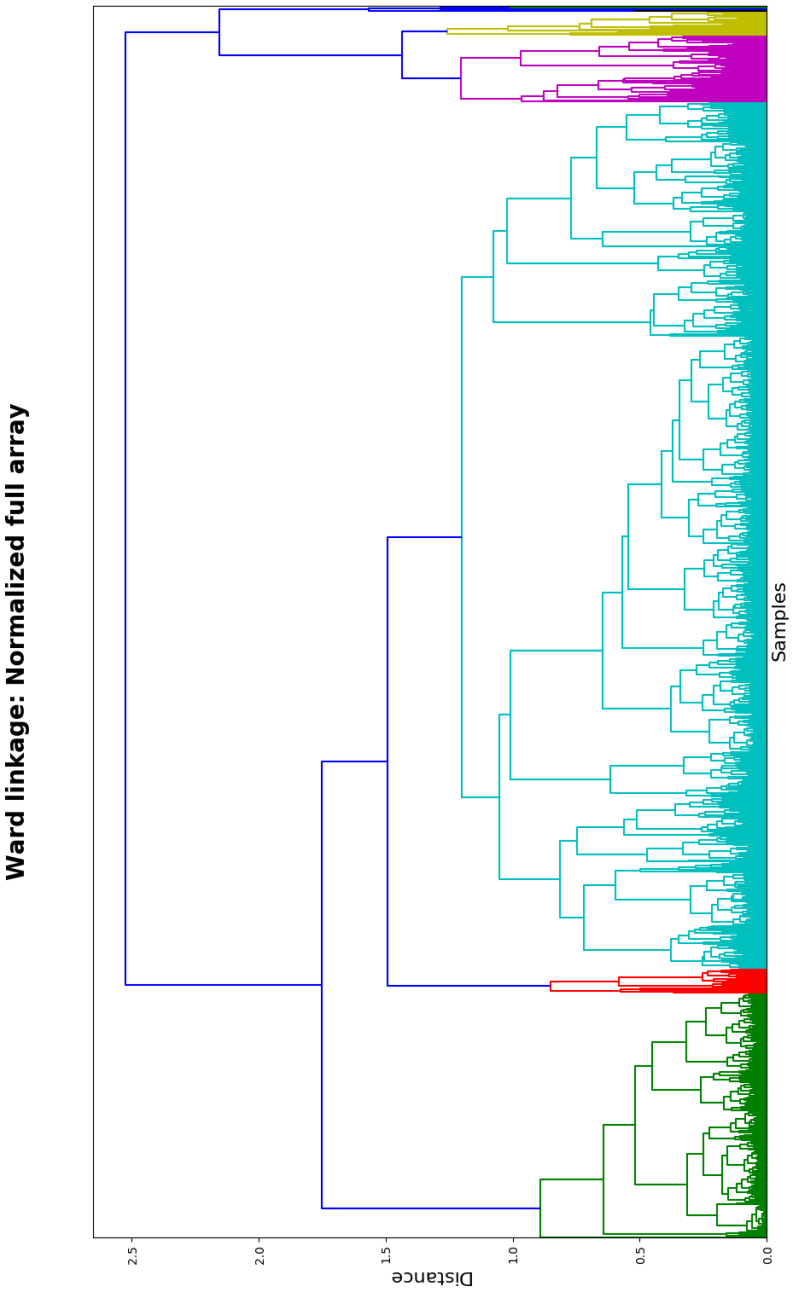

Figure B.1: Dendrogram for Ward linkage on the full data set.

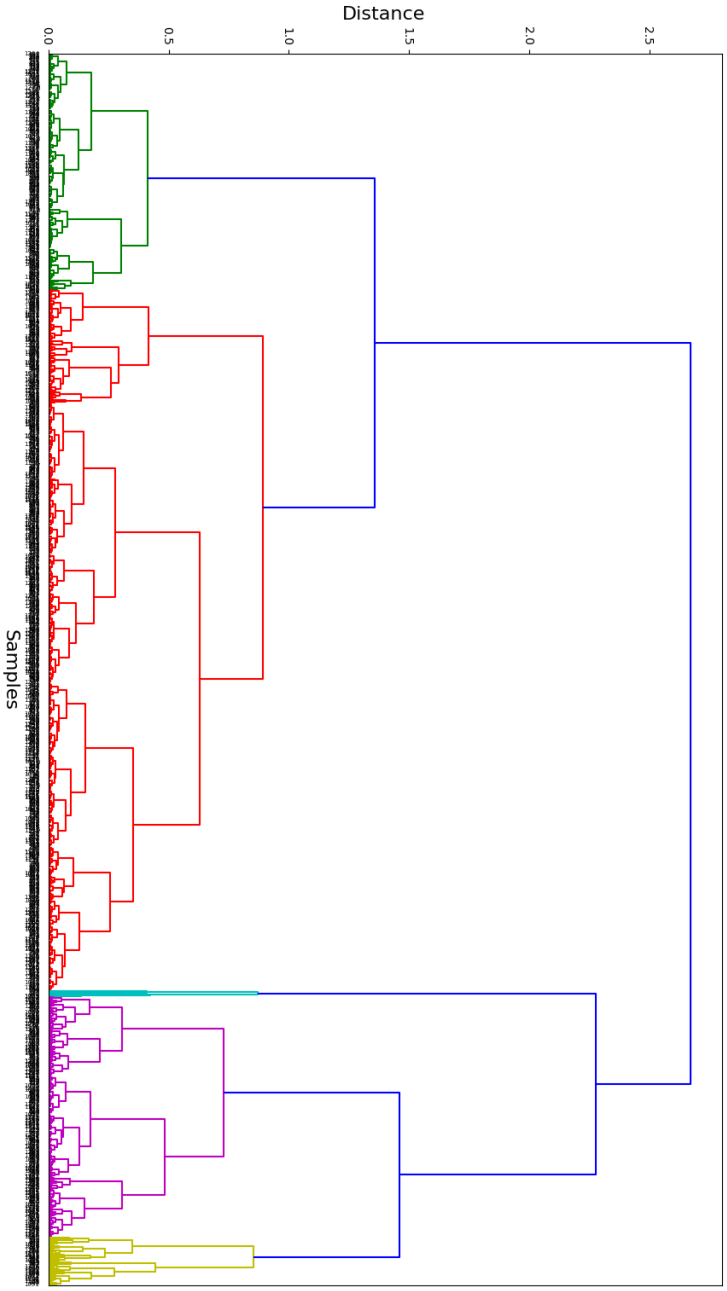

Figure B.2: Dendrogram for Ward linkage on the 2 dimensional data set.

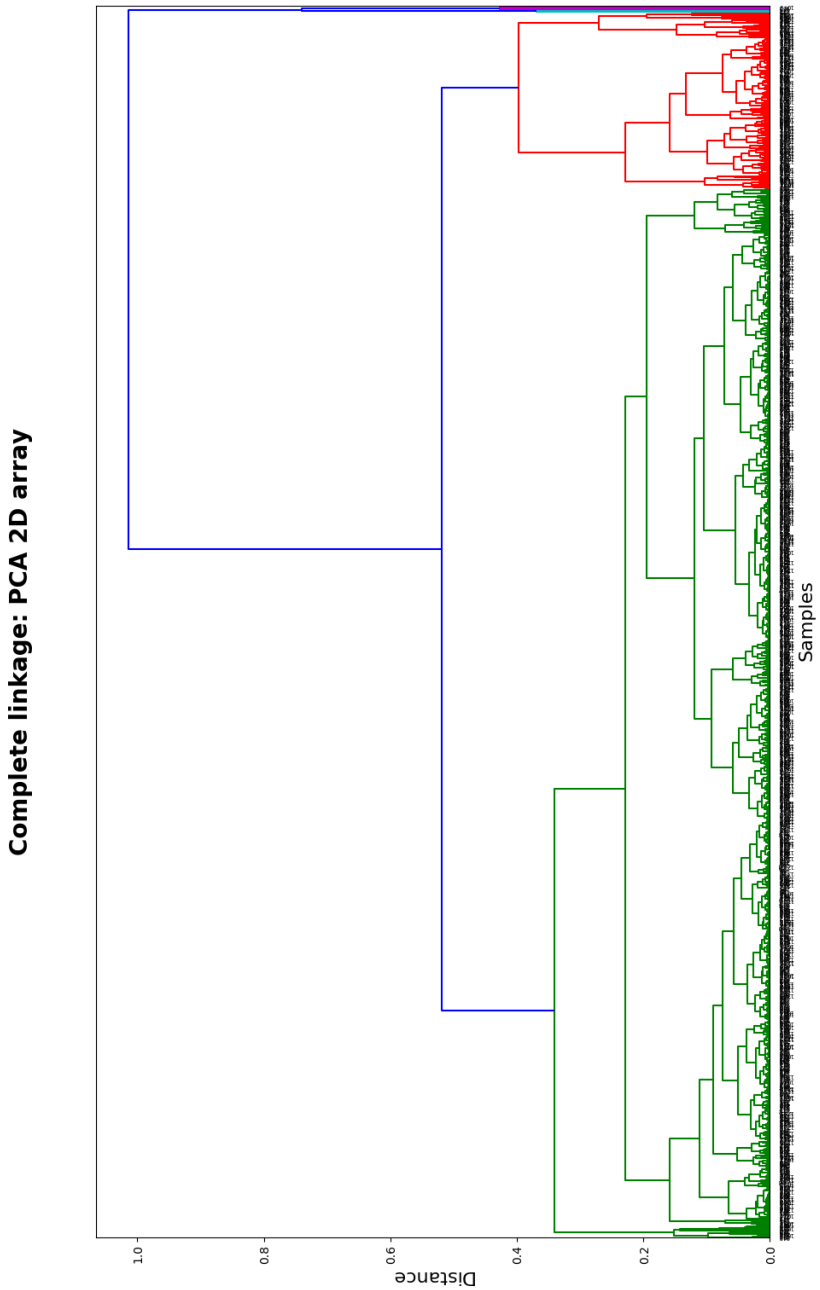

Figure B.3: Dendrogram for Complete linkage on the 2 dimensional data set.

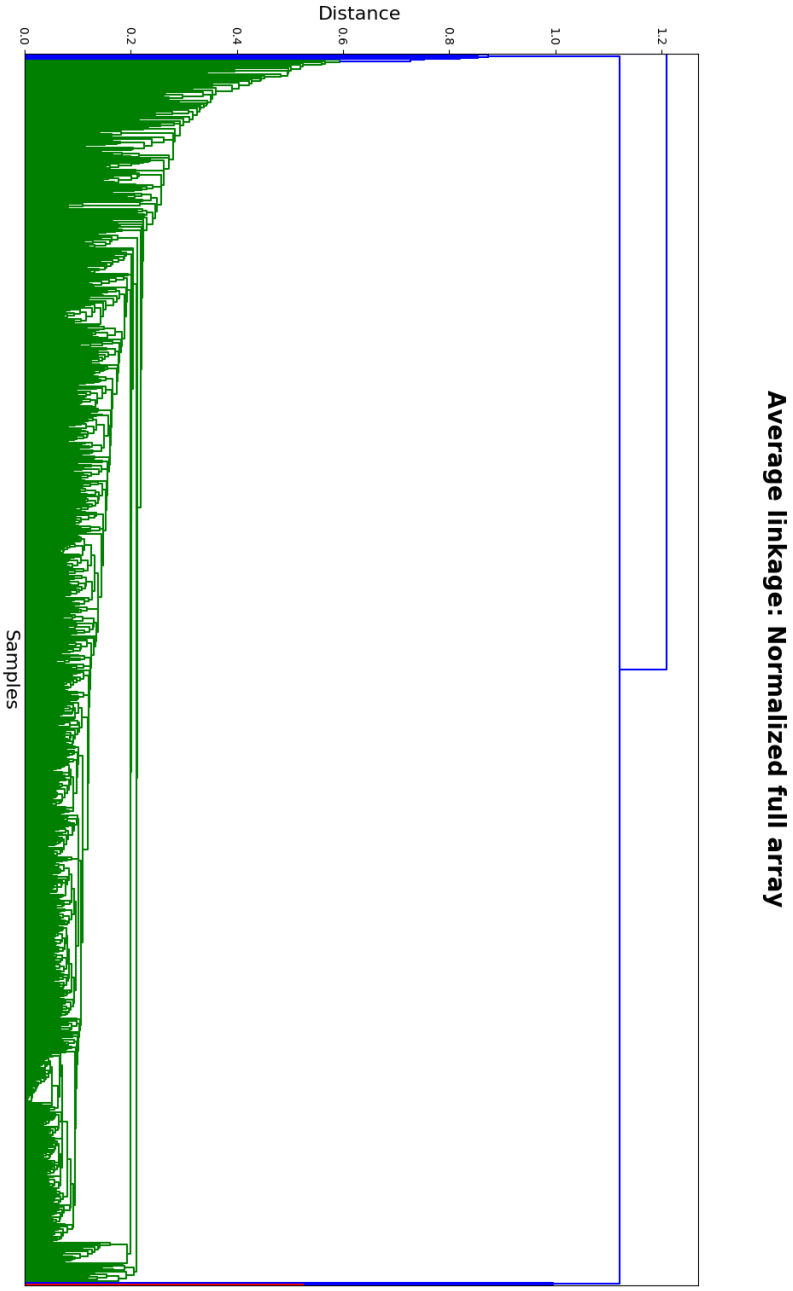

Figure B.4: Dendrogram for Average linkage on the full data set.