

---

# Preface

This project report is a research study of a variety of preprocessing and time series data analysis methods used with the objective of prediction of financial default from financial customer data.

The report is written as a course report for 'TTK4550 Engineering Cybernetics, Specialization Project' at the Norwegian University of Science and Technology during the spring of 2019. The report is created and completed at the request of DNB ASA.

I would like to extend my thanks to my project supervisor Frank Ove Westad for his counselling and guidance throughout the project. I would also like to thank Aria Rahmati representing DNB ASA for providing me with the datasets and answering all my questions regarding the project.

---

---

---

---

# Summary

This report aims to explore and investigate the differences within a data set of high risk and low risk banking customers. The goal is to identify the differences in features using a credit card account data set and checking account data set consisting of 12 months of data between January 2017 and December 2017. By performing Principal Component Analysis on the distribution for each customer record, various trends and patterns are discovered in scores and especially loadings.

---

# Table of Contents

<b>Preface</b>	<b>1</b>
<b>Summary</b>	<b>i</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>Abbreviations</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 Related Work . . . . .	1
1.2 Goal . . . . .	2
1.3 Structure of the report . . . . .	2
<b>2 Theory and methods</b>	<b>3</b>
2.1 Time Series . . . . .	3
2.2 Pre-processing . . . . .	3
2.2.1 Missing values . . . . .	4
2.2.2 Duplicates and inconsistent values . . . . .	4
2.2.3 Aggregation . . . . .	4
2.2.4 Standardization . . . . .	5
2.3 Principal Component Analysis (PCA) . . . . .	5
2.3.1 Singular Value Decomposition . . . . .	7
2.3.2 NIPALS algorithm . . . . .	7
2.4 Independent Component Analysis (ICA) . . . . .	8
2.5 Multivariate Regression Methods . . . . .	8
2.5.1 Principal Component Regression (PCR) . . . . .	8

---

2.5.2	Partial Least Squares Regression (PLSR)	9
2.6	Random Forests	9
2.7	Validation	9
<b>3</b>	<b>Problem definition and dataset</b>	<b>11</b>
3.1	Problem definition	11
3.2	Data set	11
3.2.1	Limitations	13
<b>4</b>	<b>Results</b>	<b>15</b>
4.1	Data Visualization	15
4.1.1	Method	15
4.1.2	Results	16
4.2	Discussion	17
<b>5</b>	<b>Future work</b>	<b>25</b>
	<b>Bibliography</b>	<b>27</b>

# List of Tables

3.1	Time Series . . . . .	12
3.2	Metadata credit card account . . . . .	12

---



# List of Figures

2.1	PCA conceptual model . . . . .	6
4.1	Incoming and Outgoing transactions for three selected customers . . . . .	15
4.2	PCA component 1 vs 2 . . . . .	17
4.3	Loading 1 for all variables . . . . .	17
4.4	Full loadings matrix for 'x4', 'x9', 'netto', 'x8', 'x13' . . . . .	18
4.5	Loading 1 vs Loading 2 cluster . . . . .	19
4.6	Loadings for variable 'x4' for PC2 vs PC1 grouped according to Bins and Months respectively . . . . .	20
4.7	Loadings for each month for all five bins . . . . .	22
4.8	Loading for each bin over a year . . . . .	23

---

# Abbreviations

TS	=	Time Series
PCA	=	Principal Component Analysis
SVD	=	Singular Value Decomposition
NIPALS	=	Nonlinear Iterative Partial Least Squares
ICA	=	Independent Component Analysis
PCR	=	Principal Component Regression
PLSR	=	Partial Least Squares Regression
MLR	=	Multiple Linear Regression
RF	=	Random Forest
LOOCV	=	Leave-One-Out Cross-Validation
RAM	=	Random Access Memory
GDPR	=	General Data Protection Regulation
PARAFAC	=	Parallel Factor Analysis

# Introduction

## 1.1 Background

To be able to differ between high risk customers and low risk customers is an important topic to financial services, especially banks. Predicting high risk customers can avoid financial default and it is therefore profitable for both the bank and the customer.

In this paper I will investigate how financial data can be used to find common patterns differentiating default customers from non-default customers. This is a paper written in collaboration with Norway's largest financial services group, DNB ASA. Using a small subset of DNB's 2.1 million retail customers [DNB (2019a)] might gain some insight in how customer behaviour affects credit and mortgage risk.

The financial data set in question is a time series for each customer over the range of a year. The set provides information on daily incoming and outgoing transactions from both credit and checking accounts as well as the amount of transactions per day. This information can then be aggregated to weekly, monthly and yearly transactions, meaning the average transaction for each week, month and year. This is useful as it makes it easier to spot trends and seasonal patterns.

### 1.1.1 Related Work

Within the banking section, the three maybe most common areas of research is mortgage default prediction, customer churn prediction and sales prediction. These areas have a lot in common and similar methods of analysis are therefore used for all three. For customer churn prediction, Kaur et al. (2013) uses Decision Trees and Support-Vector Machine, Bilal Zorić (2016) use Neural Networks and Kumar et al. (2008) uses Logistic Regression and Random Forests among other methods. To pre-process the data, Kumar et al. (2008) uses class distribution altering sampling methods like SMOTE and under-sampling and

Classification and Regression Tree (CART) for feature selection. Bilal Zorić (2016) applied data cleansing to detect corrupt, inaccurate and irrelevant records while Kaur et al. (2013) cleaned data by removing noisy data such as special symbols, missing values and duplicate information.

Within predicting mortgage default Kvamme et al. (2018) uses Neural Networks and Butaru et al. (2016) uses Decision Trees, Logistic Regression and Random Forests. For variable selection, Butaru et al. (2016) applies methods Stepwise, Resampling and Intersection as described in Glennon et al. (2008) while Kvamme et al. (2018) only standardizes the data.

Kou et al. (2014) uses Principal Component Analysis and Independent Component Analysis as a feature reduction methods before applying the results to classification algorithms such as Logistic Regression and Naive Bayes etc.

## 1.2 Goal

The main goal is to investigate different ways to visualize and gather knowledge around the difference in behaviour for default and non-default customers. Default customers is the term used for a customer failing to meet the legal obligations of a loan.

A theory is that there is a common set of features differing a group of default customers from a group of non-default customers. This is the basis of the exploration and implementation of methods in this project. However other trends and seasonal components are also interesting and will be investigated and discussed later on.

## 1.3 Structure of the report

The report is structured as follows. The first chapter presents the background for the project, including the motivation and goal. The second chapter contains the underlying theory and methods used for both pre-processing and multivariate data analysis methods. The third chapter is the problem definition and description of the data sets. Chapter four summarizes and discusses the results so far and the last chapter is an introduction to the content of the master thesis during fall 2019.

# Theory and methods

The following section provides the theory behind the different stages of the project, starting with a definition of time series data sets followed by pre-processing the data, a selection of methods for exploratory data analysis, classification and regression.

## 2.1 Time Series

A time series is a series of data sampled at specific points in time. It is therefore discrete-time data [Brockwell and Davis (2002)]. There are two main categories of sampled data, being univariate and multivariate. The objective of a time series is to determine if there exists a model that can describe the features of the data in order to explain the past or predict the future.

Data collected on the same metric or same object at regular or irregular time intervals. Is there an inherent relationship or structure between data at various time points and whether we can leverage the time-ordered information.

Since time series data is sampled in equally spaced increments of time, there is often a trend built into the data. This prevents the data from being stationary. Therefore differencing and power transformation are often applied to the data to remove the trend and stabilize the variance.

## 2.2 Pre-processing

The given data sets are unaltered and raw, and not necessarily collected for the specific purpose of this project. Noisy, incomplete and inconsistent data are factors that can severely affect the success of data analysis methods [Kotsiantis et al. (2006)]. It is therefore important to do some pre-processing to minimize these effects. Examples of pre-processing is

strategies to handle missing values, duplicates and inconsistent values, as well as aggregation and scaling.

### 2.2.1 Missing values

Missing values is commonly known as values lost or missing. These should be taken into account in the analysis. Tan et al. (2006) suggests a number of ways to deal with missing values such as

- **Remove Data Objects or Attributes**  
If there is a significant amount of data in comparison to how much is needed, simply removing the data entry completely will solve the missing value problem. Bear in mind entries with missing values may contain important information.
- **Estimate Missing Values**  
Apply a method of prediction or estimation to estimate a missing value, this is known as imputation. An example is to estimate one row by interpolating surrounding rows.

### 2.2.2 Duplicates and inconsistent values

Duplicate rows are very common in data sets manually extracted due to human error, limitations in the source or in the collection process. There are two categories of duplicate rows:

- Two entries of the same object with the same content
- Two entries of the same object with different content

For the first scenario Tan et al. (2006) suggest removing the duplicate such that there is only one unique entry. For the second it is suggested to leave both as is because they represent individual unique information. This might cause a problem for some methods if two entries of the same object is not accounted for.

Inconsistent values concerns values sticking out from the rest by not following a given rule, for example that an id cannot be negative or a date needs to follow DD-MM-YYYY. This can be fixed by checking values against a given rule.

### 2.2.3 Aggregation

Aggregation means combining two or more objects into a single object. A time series example is to take daily or hourly entries and combine them into a single weekly or monthly entry. This reduces the number of variables significantly, demanding less processing power. Another advantage is that the behaviour of the series is often more stable, however reduces the variability. For quantitative entries, the values are either replaced by the sum of all values or the mean. For qualitative entries, the new entry is a vector of all entries.

### 2.2.4 Standardization

Standardization, also known as standard score or z-score, is the process of transforming the individual variables into having a mean of 0 and a standard deviation of 1. It is calculated by the formula:

$$Z = \frac{X - \mu}{\sigma} \quad (2.1)$$

With  $X$  as the raw value,  $\mu$  is the mean value and  $\sigma$  is the standard deviation. It is often used to contain outliers to a more limited scale.

However when the mean value and standard deviation is unknown, the formula becomes:

$$Z_x = \frac{X - \bar{X}}{S_x} \quad (2.2)$$

$$\bar{X} = \frac{1}{n} \sum_{i=0}^n X_i \quad (2.3)$$

$$S_x = \sqrt{\frac{\sum_{i=0}^n (X_i - \bar{X})^2}{n - 1}} \quad (2.4)$$

Where  $\bar{X}$  is the sample mean and  $S_x$  is the sample standard deviation.

## 2.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a method to reduce dimensionality. The basic idea is to replace original variables with latent variables as a linear combination of the original ones with the objective function of maximizing the explained variance for the subsequent components. These latent variables do not necessarily have a physical meaning but can nevertheless often reveal underlying patterns that is useful for the actual application when interpreted with the domain-specific knowledge at hand. By assessing the optimal number of components, PCA will place the noise or unsystematic part of the data matrix  $X$  in the residual matrix  $E$ . PCA is used in applications such as exploratory data analysis, classification and identification, process monitoring and variable reduction, which is the focus in this report [James and Tibshirani (2013)].

In mathematical terms, the idea is to create an  $n \times M$  matrix  $T$  where  $n$  is the number of chosen rows in the data set  $X$  and  $M$  is the number of principal components. The column  $T_m$  of  $X$  is the  $m$ -th principal component.

$$\mathbf{X} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \mathbf{E}$$

Vector form:  $\mathbf{X} = \mathbf{t}_1 \mathbf{p}_1^T + \mathbf{t}_2 \mathbf{p}_2^T + \mathbf{E}$

$$\mathbf{X} = \mathbf{T} \mathbf{P}^T + \mathbf{E}$$

1

Matrix form:  $\mathbf{X} = \mathbf{T} \mathbf{P}^T + \mathbf{E}$

**Figure 2.1:** PCA conceptual model

In the model given by Figure (2.1),  $\mathbf{p}_1$  refers to the loadings of the first principal component. The goal is to have  $\mathbf{t}_1$  as the highest possible variance, meaning the direction where the observations vary the most. The  $\mathbf{t}_i$  's are also orthogonal to each other.

Given a matrix  $\mathbf{X}$  with dimension  $n \times M$  where each column represent a vector of predictors. Assuming the covariance matrix associated with  $\mathbf{X}$  to be  $\Sigma$ , it has an eigen-decomposition

$$\Sigma = \mathbf{C} \mathbf{\Lambda} \mathbf{C}^{-1} \quad (2.5)$$

$\mathbf{\Lambda}$  is a diagonal matrix of non-negative eigenvalues in decreasing order and  $\mathbf{C}$  is a matrix made up by the eigenvectors of  $\Sigma$ .

Furthermore total variance present in a data set is explained by:

$$\sum_{j=1}^p \text{Var}(\mathbf{X}_j) = \sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n x_{ij}^2 \quad (2.6)$$

And variance explained by the  $m$ -th component:

$$\frac{1}{n} \sum_{i=1}^n t_{im}^2 = \frac{1}{n} \sum_{i=1}^n \left( \sum_{j=1}^p p_{jm} x_{ij} \right)^2 \quad (2.7)$$



Combining the two yields the proportion of variance explained (PVE):

$$\frac{\sum_{i=1}^n \left( \sum_{j=1}^p p_{jm} x_{ij} \right)^2}{\sum_{i=1}^n x_{ij}^2} \quad (2.8)$$

There are  $\min(n - 1, p)$  principal components and their PVEs sum to one.

There are several methods for finding the eigenvectors and the next sections will discuss a couple of these.

### 2.3.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is the default eigen decomposition method on the covariance matrix as it is often more efficient than other methods and can handle sparse matrices.

The procedure is as follows:

- Let  $X$  be a matrix of size  $(nRows \times nCols)$
- SVD on  $X$ :  $[u, s, v] = \text{svd}(X)$   
Where  $s$  is a vector of singular values, scores  $T = u \times s$  and loadings  $P = v$
- In case of matrices which are "long thin" or "short fat":
  - Perform SVD on the covariance matrix of the smallest dimension  $\frac{X^T X}{(nRows-1)}$   
or  $\frac{X X^T}{(nCols-1)}$ 
    - \* If  $nCols < nRows$ :  $[u, s, v] = \text{svd}(\frac{X^T X}{(nRows-1)})$   
Both  $u$  and  $v$  hold the loadings  $P$
    - \* If  $nRows < nCols$ :  $[u, s, v] = \text{svd}(\frac{X X^T}{(nCols-1)})$ 
      - i.  $v = X^T u$
      - ii.  $P = \text{norm}(v)$
  - $T = X P (X^T P)^{-1}$ ;  $P^T P = I$
  - Eigenvalues =  $\text{diag}(s)$
  - Explained variance =  $\frac{\text{cumulative sum}(\text{eigenvalues})}{\text{sum}(\text{eigenvalues})}$

### 2.3.2 NIPALS algorithm

The NIPALS algorithm, Nonlinear Iterative Partial Least Squares, is the second most commonly used method for calculating the principal components of a data set. The algorithm is given as follows:

---

**Algorithm 1: NIPALS**

---

```
 $E = \frac{x - \bar{x}}{\sigma};$ 
for  $a=1:A$  do
  while  $\epsilon > tol$  do
     $t = \text{column in } E \text{ with highest variance}$ 
     $p^T = \frac{t^T E}{t^T t}$ 
     $p = \frac{\|p\|}{\|p\|}$ 
     $t = \frac{E p}{p^T p}$ 
     $\epsilon = \|t_{old} - t\|$ 
     $t_{old} = t$ 
  end
   $E = E - t p^T$ 
end
```

---

## 2.4 Independent Component Analysis (ICA)

PCA is about finding latent variables by maximizing variance. ICA is trying to maximize independence, i.e. finds linear transformation in the feature space into a new feature space in a way that each new feature is mutually independent. Mutual information of all new features and original features is as high as possible.

## 2.5 Multivariate Regression Methods

Multiple Linear Regression (MLR) estimates regression coefficients by means of least squares and is the method of choice when the variables are orthogonal as for factorial designs. However when the X-variables are correlated, the solution might be unstable and lead to erroneous interpretation. MLR requires more objects than variables and only one Y-variable can be modelled at the time. The solution is to use methods using latent variables like Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR)

### 2.5.1 Principal Component Regression (PCR)

Principal Component Regression solves the collinearity problem by using scores from PCA in place of the raw data X for MLR (Multiple Linear Regression). PCA also solves the inversion problems and minimizes noise in X by finding the correct rank before regression to Y.

Based on the first  $m$  principal components  $M_1, \dots, M_m$  from section 2.3, the idea is to use these components as the predictors in a linear regression model that is fit using least squares. The underlying assumption is that in the directions in which **X** show the most variation are the directions that are associated with Y. This assumption does not necessarily hold but is reasonable enough to give decent results. By estimating only  $m < n$  coefficients it is possible to mitigate overfitting. The more principal components are used in the regression model, the bias decreases but the variance increases. This is known as

the bias-variance trade-off.

### 2.5.2 Partial Least Squares Regression (PLSR)

The components found in PCA are found using unsupervised learning, meaning there are no label  $Y$  used to find the directions. There is consequently no guarantee that the directions that optimally explains the predictors also optimally explains the response. Partial Least Squares Regression (PLSR) is a supervised method to explain both the predictors and the response.

## 2.6 Random Forests

A generic forest model is based on a decision tree. A decision tree is a structure looking for data to split based on the largest difference. The tree goes from observations represented by branches to conclusions represented by leaves.

A forest model creates hundreds of trees called an ensemble of decision trees. Each tree is created by different randomly generated subsets of the original variables. The ensemble is then looked at as a whole to create a prediction. Random forest trees fix a common issue with decision trees called overfitting, where the model fits the sample data a little too well and is not able to predict future result as well as it should. Each individual tree still has the issue of overfitting, however on average over hundreds of trees the overfitting is averaged out.

## 2.7 Validation

For model validation it is normal to distinguish between external (hypothesis-driven) and internal (data-driven) validation. External validation examines whether the model's predictions are reliable by answering questions like "Do I use the correct information based on the theory in my model?" or "Do different methods give the same results/interpretation?". The results are confirmed by theory or existing knowledge. The internal validation uses numerical methods like cross validation, test set validation, verification set validation and cross model validation where the result is discussed within the scope of the project.

It is common to divide data into three parts: training set to fit the model, validation set to select the best model and test set to assess how well the model fits on new independent data. The addition of the test set is important because it will be too optimistic to report the error on the test set when the test set has already been used to choose the best model.

The validation error is the prediction error over a set of validation samples. The validation sample consists of data not used when training the model. The idea is for the model to capture the most important relationships between the response variable and the covariates to avoid underfitting. The trade-off in selecting a model with enough complexity and

flexibility is called the variance-bias trade-off.

Test set validation tests the model on new independent samples by splitting the data into two sample sets, building the model with the training samples and then using the model to predict  $Y$  for the test samples. Finally the predicted  $Y$  is compared to the reference  $Y$  in order to compute the prediction error residual. The drawbacks with this approach is the high variability of test set error since this error is dependent on which observations are included in each set. Another drawback is that the subset of the whole data set may not represent all relevant dimensions, since the data set is split in half.

Segmented cross validation is used to find the most important sources of variance and to decide of the optimal rank of the model. It is often used in situations in which it is not possible to produce a separate representative test set. There is in general no given number of segments that ensures an optimal cross validation, but if there are known subgroups of the samples, the segments should reflect these groups as it is a way to estimate the model robustness taking into account known stratification of the samples such as age groups, year etc. The idea is to pick out one or more samples from the calibration set, build a model with the remaining samples and predict  $Y$  for the left out sample, and compute the residual. Put the samples back into the calibration set, take out other samples and do the same. After all samples have been left out once, combine the prediction residuals. One may also apply Leave-one-out cross-validation (LOOCV) as a form of lower bound for the validated error.

# Problem definition and dataset

## 3.1 Problem definition

The goal for this report is to explore and attempt different multivariate analysis methods to separate default customers from non-default customers. The idea is to find common features or trends in default customers that separate them from non-default customers. This can then be used in prediction methods to predict for example customer churn or customer default. There are several different methods that can be applied to find features. This report will focus mainly on dimensionality reduction by the use of principal component analysis. Performing dimensionality reduction on a time series data set can be done in a number of different ways and preprocessing of the data is therefore crucial.

## 3.2 Data set

The data used in paper is data provided by DNB's banking services and is collected from real customer transaction data for the purpose of data analysis. Apart from anonymisation masking the identify all customers, the data is not altered in any way.

It contains data from 3775 customers with a checking account and or credit account in the period from January 2017 to December 2017. The credit card and checking accounts given in Table 3.1 represent the two main time series used in this report. These two time series with corresponding metadata can be found in Table ?? and Table 3.2.

Both data sets are time series for each customer over the range of a year, however there not 365 data points per customer as there are only data points for days with an ingoing or outgoing transaction. In total the checking account data set is of size  $711911 \times 14$  with 3775 customer ids while the credit card account data is of size  $144085 \times 10$  with 2726 customer ids.

Series	Abbrev	Explanation
Credit card	cc	Sum of transactions on the credit card
Checking account	ch	Sum of transactions in the checking account

**Table 3.1:** Time Series

Variable	Explanation	Type
'x1'	Customer identifier masked	Int
'x2'	Date of transaction	String
'x3'	Year and month of transaction date in the form YYYYMM	Int
'x4'	Sum of all ingoing transactions into the customer's checking account	Float
'x5'	Sum of all ingoing, regular transactions into the checking accounts	Float
'x6'	Sum of all ingoing, variable transactions into the checking accounts	Float
'x7'	Sum of all ingoing, account transfer transactions into the checking accounts	Float
'x8'	The total number of ingoing transactions into the checking accounts	Int
'x9'	Sum of all outgoing transactions from the checking accounts	Float
'x10'	Sum of all outgoing, regular transactions from the checking accounts	Float
'x11'	Sum of all outgoing, variable transactions from the checking accounts	Float
'x12'	Sum of all ingoing, account transfer transactions into the checking accounts	Float
'x13'	The total number of outgoing transactions into the checking accounts	Int
'Y'	1 if the customer defaulted in the period 201801, 201812	Boolean

Variable	Explanation	Type
'x1'	Customer identifier masked	Int
'x2'	Date of transaction	String
'x3'	Year and month of transaction date in the form YYYYMM	Int
'x4'	Sum of all ingoing transactions into the customer's credit card accounts	Float
'x5'	Maximum, ingoing transaction amount into the customer's credit card accounts	Float
'x6'	The total number of ingoing transactions into the credit card accounts	Int
'x7'	Sum of all outgoing transactions from the credit card accounts	Float
'x8'	Maximum, outgoing transaction amount from the credit card accounts	Float
'x9'	The total number of outgoing transactions from the credit card accounts	Int
'Y'	1 if the customer defaulted in the period 201801, 201812	Boolean

**Table 3.2:** Metadata credit card account

### 3.2.1 Limitations

- **Privacy**  
Because the data set contains sensitive financial information, there are regulations set by GDPR to make sure personal information stays confidential. This makes the process of acquiring the data slow [DNB (2019b)].
- **Data Anonymisation**  
As a consequence of the privacy regulations mentioned above, there is no available information regarding the identity of the customer base. It is therefore difficult to make any assumptions of how likely younger customers are of going default in comparison to older customers etc.
- **Data set size**  
The size of the data set is quite large, computational power was a limitation. This was due to a limited amount of RAM on the rented Amazon Web Services server instance. The solution was to upgrade to an instance with more RAM and section-wise load the data set into memory, thereby increasing the required time by quite a lot.

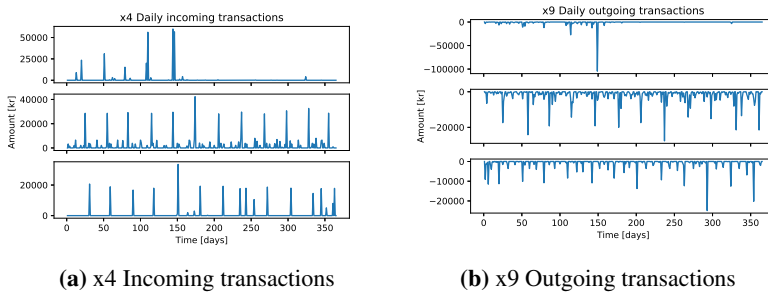




# Results

## 4.1 Data Visualization

In order to perform time series analysis and forecasting, visualization is an important first step. There is a variety of different types of visualizations used for time series data, a fundamental one being the line plot.



**Figure 4.1:** Incoming and Outgoing transactions for three selected customers

Figure (4.1) displays the outgoing and incoming transactions for three randomly selected customers. They display different patterns of transactions and can be discussed as is. However for 3557 customers, this would take a while. Therefore the idea is to explore other methods of visualization more suitable for large quantities of data.

### 4.1.1 Method

Plotting the data without a temporal ordering is often helpful to explore how the data is actually distributed. The idea was to see if the average default customer transaction distribution was different from the average non-default customer transaction distribution.

The set of chosen interesting transaction variables includes:

- 'x4' - Sum of all ingoing transactions into the customer's checking accounts.
- 'x9' - Sum of all outgoing transactions from the checking accounts.
- 'netto' - Sum of all ingoing transactions subtracted the sum of all outgoing transactions.
- 'x8' - The total number of ingoing transactions into the checking accounts.
- 'x13' - The total number of outgoing transactions into the checking accounts.

For every variable, we define  $nmonths \times nbins$  variables where the data for each month is described using a histogram with  $n$  number of bins. This gives an array of size:

$$nparams \cdot nmonths \cdot nbins = 5 \times 12 \times 5 \quad (4.1)$$

$$= 300 \quad (4.2)$$

For each customer we then compute this array yielding an  $x_{train}$  of shape  $3775 \times 300$  and  $y_{train}$  of shape  $3775 \times 1$ . The x set is then standardized and fed through a PCA algorithm with  $ncomponents = 7$ . This outputs a scores matrix and a loadings matrix as well as eigenvalues which gives an explained variance vector given in equation (4.3).

$$\begin{bmatrix} 0.11362362 & 0.05451965 & 0.02025486 & 0.01837598 \\ 0.01303826 & 0.01214935 & 0.01031114 & \end{bmatrix} \quad (4.3)$$

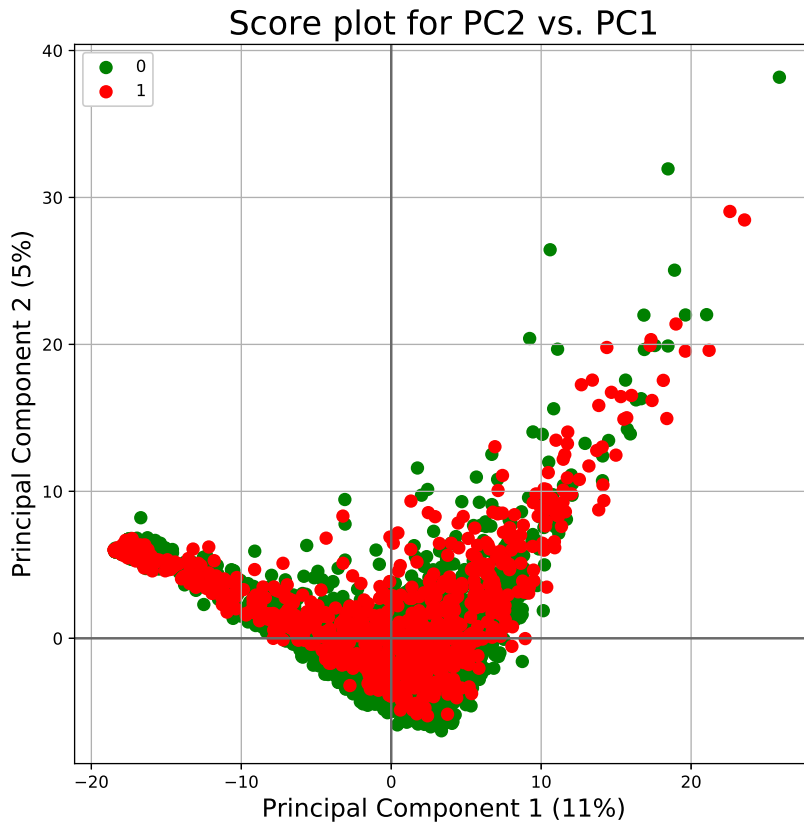
### 4.1.2 Results

Plotting the first two components from the PCA with corresponding labels (where 1 equals default and 0 equals non-default) gives Figure 4.2. The x and y label also indicate how large the explained variance is for each component.

Figure (4.3) display loading 1 for all variables over the course of 12 months and 5 bins per month. The same plot with all loadings 1-7 is given in Figure (4.4). This is referred to as the full loadings matrix.

Scatter plot of Loading 1 vs Loading 2 for all variables can be found in Figure (4.5). The variables are then grouped according to bins and months respectively. A corresponding plot with 'x4' only is displayed in Figure (4.6).

Figures (4.7) and (4.8) are heatplots displaying the same result in two different ways. Firstly, Figure (4.7) displays the year chronologically with 5 bins per month. Figure (4.8) displays the bins chronologically with 12 months per bin.



1

Figure 4.2: PCA component 1 vs 2

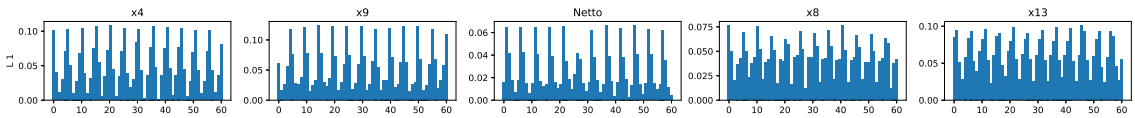
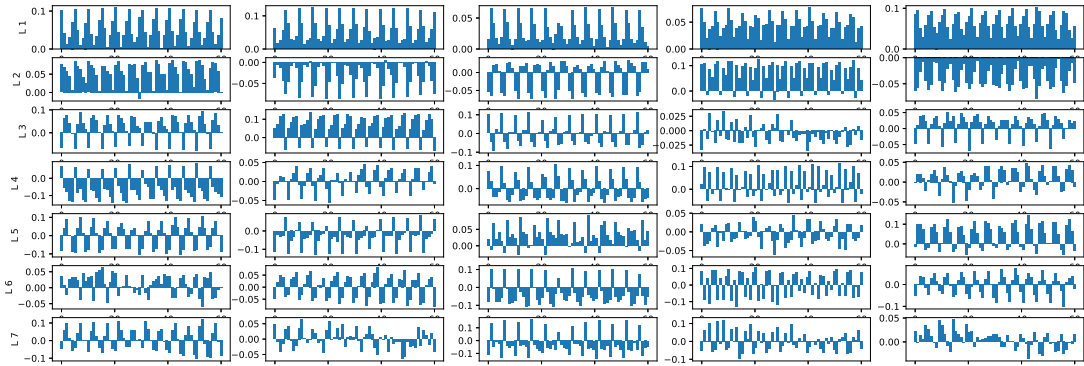


Figure 4.3: Loading 1 for all variables

## 4.2 Discussion

Equation (4.3) shows the explained variance vector. The first principal component is always the component with the most explained variance and is in this case responsible for 11% of the explained variance. This is a pretty low number, and there might be several



**Figure 4.4:** Full loadings matrix for 'x4', 'x9', 'netto', 'x8', 'x13'

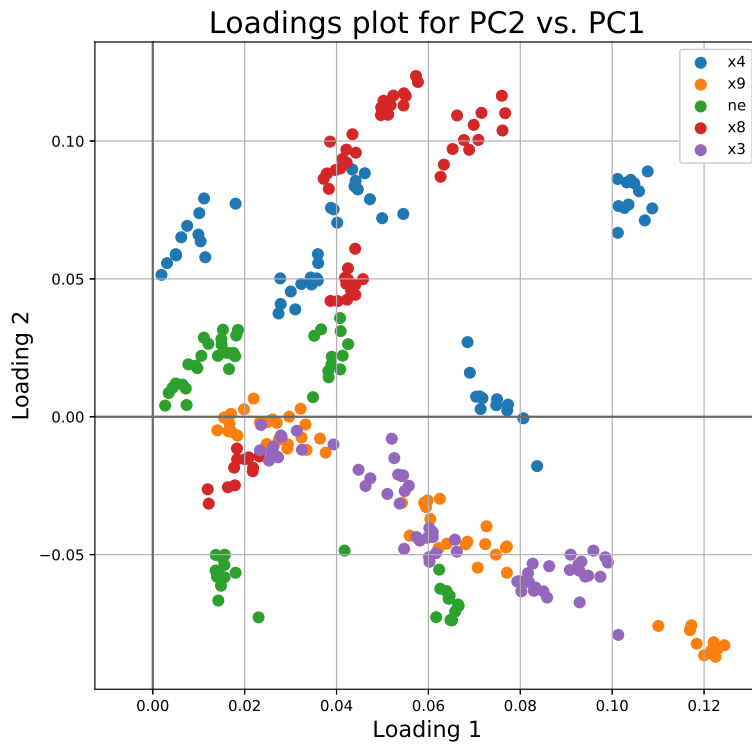
reasons for this. With a limited knowledge of the customer base, one can only speculate in how well distributed it is. A low total explained variance might signify a low amount of sparsity in the data set.

Figure (4.2) shows how default customers (given by red points) form mostly the same pattern as non-default customers (green points). This pattern continues when combining other pairs on principal components such as PC1 vs PC3, PC2 vs PC3 etc. This neither confirms nor denies the theory of a common set of features differing a group of default customers from a group of non-default customers as this unsupervised PCA-model has no "goal" of separating the two. Nevertheless, a separation of the groups in two clusters in the score plot of PC1 vs PC2 would have been beneficial given the objective of discriminating between non-default and default customers. This will be a subject of further investigation, applying regression and classification methods described above.

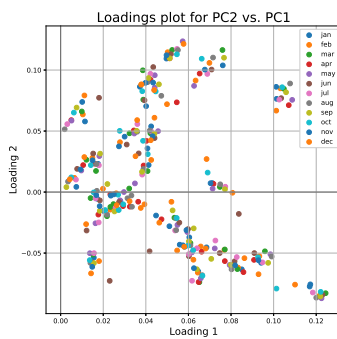
It is however interesting to look at how important each variable is for each component. A one-dimensional loading plot shows the importance of the variables for individual principal components scores as shown in Figure (4.3) for loading 1. For 'x4' the first principal component separates the smallest and largest values from the rest, and similarly the same pattern for 'x9'. Because 'netto' is the difference between 'x9' and 'x4', it makes sense to explain the loading for 'netto' by the values that fall into the middle 3 bins. Principal component 1 scores is also mostly correlated with the smaller values of 'x8' and the two first bins for 'x13'.

Figure (4.4) shows the entire loadings matrix for all seven principal components. Interpretation of the figure shows that the loadings get seemingly more random the less variance is explained by each component, which is as expected. In addition there is a clear trend, meaning very often the same bins are correlated for every single month.

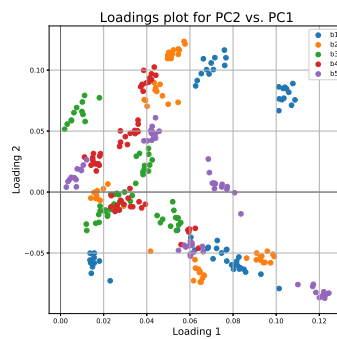
The loadings vs loadings plot (Figure (4.5)) displays how all variables are positively



(a) Variables

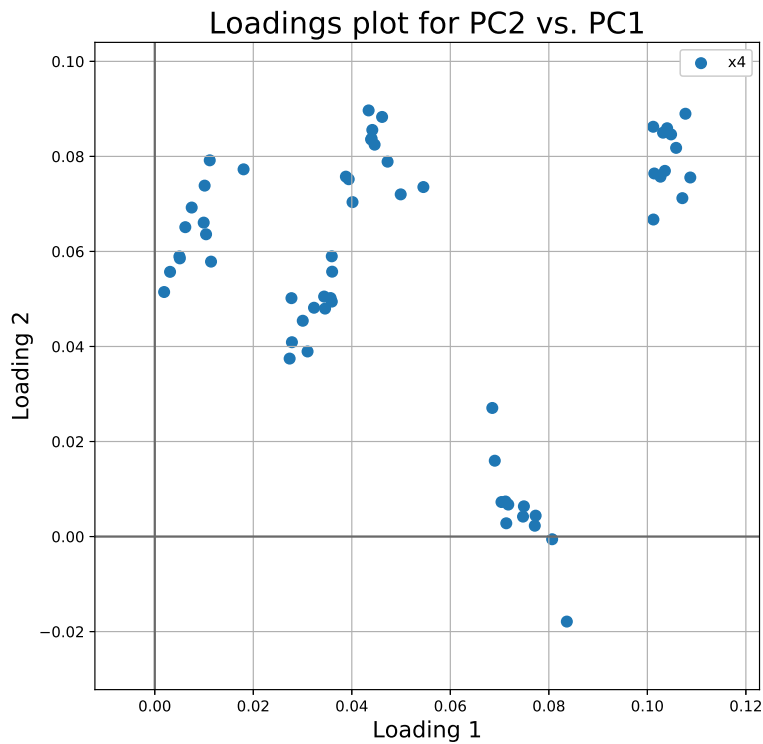


(b) Months

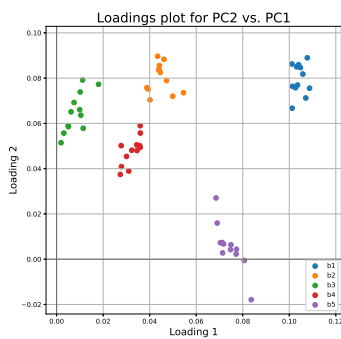


(c) Bins

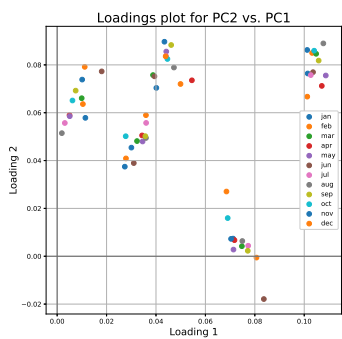
**Figure 4.5:** Loading 1 vs Loading 2 cluster



(a) x4 loadings



(b) Bins



(c) Months

**Figure 4.6:** Loadings for variable 'x4' for PC2 vs PC1 grouped according to Bins and Months respectively

correlated for loading 1 while about half is positively correlated and half is negatively correlated for loading 2. There are also clear clusters for each variable which can be explained by the different bins (Figure (4.5c)). This plot is displayed three times to differ between variables, bins and month. The idea is to find out if there is a clear pattern in either. There seems to be clear clusters for each variable however spread out, forming clusters for each bin. There looks to be no interpretable pattern for each month. Looking more closely at 'x4' in Figure (4.6) almost all values are positively correlated for both loading 1 and loading 2 and there is a clear distinction between the different bins, however no obvious explanation for each month.

Finally, Figures (4.7) and (4.8) exhibit the same three loadings with variables displayed in two different ways. Firstly, Figure (4.7) display very similar information to the barplots in Figures (4.3) and (4.4), however it is easier to see a clear pattern in what bin has the most effect on the scores. Figure (4.8) on the other hand presents the differences in the different months for each bin. For 'x4', all bins have a slight change in color around June, which can be interpreted by for example summer wages or return on tax payments which both happen in June.

Altogether, there is a lot more to be explored and tested for the data sets. The results shown in this project report reduces the dimensionality of the data set too much for it to properly represent the full set. Implementing new methods can therefore help extract the most important features. This in particular is discussed further in Section (5).

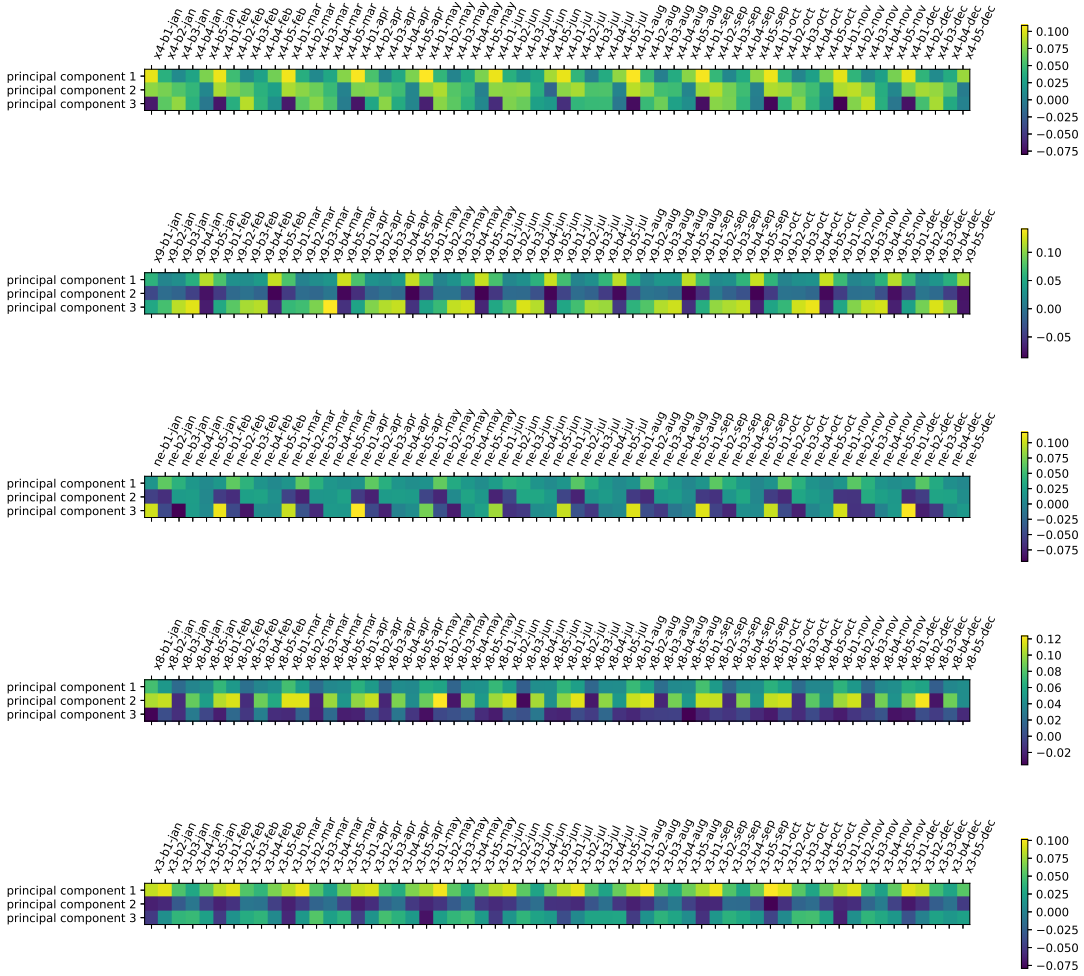


Figure 4.7: Loadings for each month for all five bins



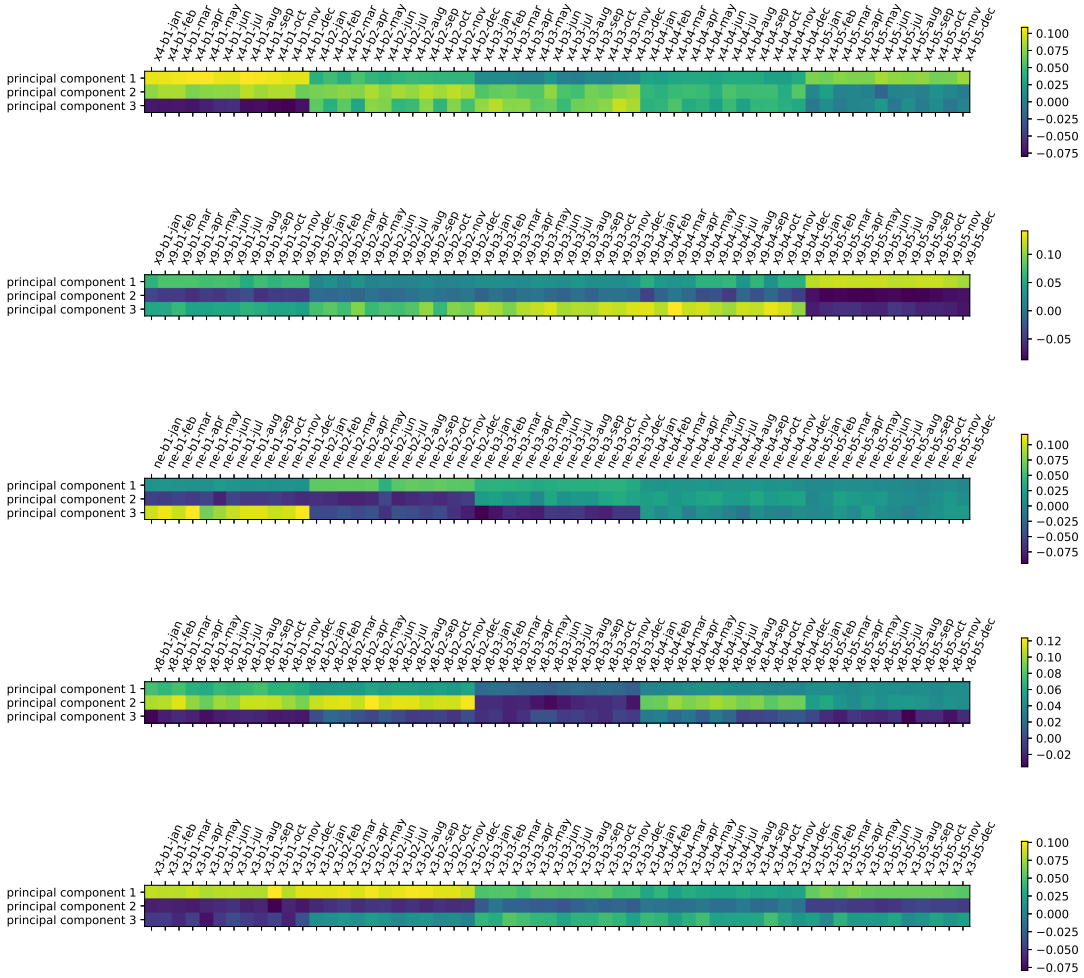


Figure 4.8: Loading for each bin over a year



## Future work

This project thesis focuses mainly on researching methods for time series data analysis. The preliminary results are limited thus providing room for further exploration in the master thesis.

The focus for the master thesis will be to continue working with time series data, implement methods mentioned in this project report as well as delve into new methods and data sets. In Section (3.2), a second data set with credit card transactions is mentioned, this set will be further analysed in the master thesis. Methods already mentioned in the project report are ICA, PCR, PLSR and Random Forests. New methods include outlier detection by use of Hotelling's  $T^2$  statistics and residual statistics, and generalization of PCA to higher order arrays by using Parallel Factor Analysis (PARAFAC). These approaches will contribute to reducing the dimensionality of the data set and extracting the most important features.

After selecting prominent features, different classifiers can be applied to a training set and validated on a test set using an appropriate validation method. The best classifier will then be chosen based on measures of classification error and accuracy.



# Bibliography

- Bilal Zorić, A., 2016. Predicting customer churn in banking industry using neural networks. *Interdisciplinary Description of Complex Systems: INDECS 14* (2), 116–124.
- Brockwell, P. J., Davis, R. A., 2002. *Introduction to Time Series and Forecasting*. Springer.
- Butaru, F., Chen, Q., Clark, B., Das, S., Lo, A. W., Siddique, A., 2016. Risk and risk management in the credit card industry. *Journal of Banking & Finance* 72, 218 – 239.  
URL <http://www.sciencedirect.com/science/article/pii/S0378426616301340>
- DNB, Jun 2019a. Dnb-konsernet.  
URL <https://www.dnb.no/om-oss/om-dnb.html>
- DNB, May 2019b. Dnbs personvernerklæring.  
URL <https://cran.r-project.org/web/packages/logisticPCA/vignettes/logisticPCA.html>
- Glennon, D., Kiefer, N. M., Larson, C. E., Choi, H.-s., 2008. Development and validation of credit scoring models. *Journal of Credit Risk*, Forthcoming.
- James, G., W. D. H. T., Tibshirani, R., 2013. *An Introduction to Statistical Learning*. Springer.  
URL <http://www-bcf.usc.edu/~gareth/ISL/>
- Kaur, M., Singh, K., Sharma, N., 2013. Data mining as a tool to predict the churn behaviour among indian bank customers. *International Journal on Recent and Innovation Trends in Computing and Communication* 1 (9), 720–725.
- Kotsiantis, S., Kanellopoulos, D., Pintelas, P., 01 2006. Data preprocessing for supervised learning. *International Journal of Computer Science* 1, 111–117.
- Kou, G., Peng, Y., Lu, C., 2014. Mcdm approach to evaluating bank loan default models. *Technological and Economic Development of Economy* 20 (2), 292–311.

---

Kumar, D. A., Ravi, V., et al., 2008. Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies* 1 (1), 4–28.

Kvamme, H., Sellereite, N., Aas, K., Sjursen, S., 2018. Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications* 102, 207 – 217.

URL <http://www.sciencedirect.com/science/article/pii/S0957417418301179>

Tan, P.-N., Steinbach, M., Kumar, V., 2006. *Introduction to data mining*.