Ada Skarsholt Larsen

Using Multivariate Methods To Predict Financial Default

Master's thesis in Cybernetics and Robotics
Supervisor: Frank Westad

December 2019

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Ada Skarsholt Larsen

Using Multivariate Methods To Predict Financial Default

Master's thesis in Cybernetics and Robotics
Supervisor: Frank Westad
December 2019

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

NTNU
Norwegian University of
Science and Technology

Sammendrag

Fintech er en voksende bransje for banker og andre virksomheter som tilbyr lån, forsikring, kunderådgiving og andre finansielle tjenester [PWC]. Et viktig aspekt er hvordan data om kunder og tjenester bør analyseres for å gi de beste råd samt minske risiko for tap både for kunder og tilbyder. I dette arbeidet har jeg benyttet ulike multivariate analysemetoder for å vurdere risiko for mislighold av lån med utgangspunkt i tilgjengelig informasjon om enkeltkunder.

Banker står på en stor mengde data for hver kunde, blant annet personlig information, kredittkort transaksjoner og avdrag og renter på lån. Åpen kildekode bestående av tjekkisk bankdata med tabeller med transaksjoner, lån og kredittkort informasjon i tillegg til demografiske data er i denne oppgaven brukt til å utvikle klassifikasjonsmodeller for å separere kunder med mislighold og kunder uten.

Et egenskapssett eller feature sett blir samlet sammen ved å bruke relasjonene mellom tabellene i datasettet. Dette settet blir deretter delt i to etter numeriske og kategoriske variabler. Deretter benyttes Principal Component Analysis (PCA) og Logistic Principal Component Analysis (LPCA/ Logistic PCA) til å lage to forskjellige modeller. Disse modellene i tillegg til feature settet blir deretter brukt som input til klassifikasjonsmodellene Support Vector Machines (SVM), Random Forest (RF), Logistic Regression (LR), Partial Least Squares Regression (PLSR) og Principal Component Regression (PCR). Test sett validering og kryssvalidering, i tillegg til forklart varians og korrelasjons loadings blir analysert for å bestemme modellkompleksitet.

Modellene har vist at det å klassifisere mislighold er vanskelig, mye forårsaket av et veldig ubalansert datasett. Dette fører til veldig partiske modeller mot ikke-mislighold. Det kategoriske datasettet med kjønn, region, korttype og alder viste veldig lite underliggende korrelasjon mellom variablene, noe som undergraver potensialet til Logistic PCA. Med mindre enn 25 variabler i hvert sett kan dette også føre til begrenset varians og beslutningsgrunnlag.

Følgelig fra et begrenset data sett er ikke resultatene gode nok til å bli brukt i en beslutningsprosess for å bestemme betalingsevne. Klassifikasjonsmodellen til Random Forest med feature settet som input viste de beste resultatene, men med en fremdeles begrenset evne til å klassifisere mislighold.

Abstract

Fintech, short for Financial technology, describes technology that seeks to improve and automate financial services. The idea is to help companies, business owners and consumers better manage their finances, processes and lives while minimizing the risk of doing so for both consumers and product owners. This thesis applies different multivariate methods of analysis in order to determine the risk of default, meaning failing to meet the legal obligations of loan, based on financial client data.

Banks gather a large amount of data on each client, such as personal details, all credit card transactions and loan repayments. Open source Czech banking data containing tables of transactions, loan and credit card information as well as demographic data is in this thesis used to develop classification classification models to separate default clients from non-default clients.

A feature set is aggregated using the relationships between the different tables. This feature set is then divided into numeric and categorical variable frames. Afterwards Principal Component Analysis (PCA) and Logistic Principal Component Analysis (LPCA/Logistic PCA) models are built with two different response variables, for the numeric and categorical frames respectively. These models as well as the feature set is used as input to classification models Support Vector Machines, Random Forest, Logistic Regression, Partial Least Squares Regression and Principal Component Regression. Test set validation and cross validation, as well as explained variance and correlation loadings are interpreted to determine model complexity.

The models showed that classifying default clients is extremely difficult, largely caused by a highly imbalanced data set. Which in turn causes very biased models towards non-default clients. The categorical data set containing variables gender, region, card type and age showed very little underlying correlation, undermining the potential of Logistic PCA. Having less than 25 variables in each data frame might not be enough information to classify default clients, as important variables might not be present in the data.

With limitations in the data set itself, the results are not good enough to use in a payment ability decision process. The classification models with feature set as input showed slightly better results, but with an insufficient ability to classify the default clients.

Preface

This master thesis was written during the Fall of 2019 at the Norwegian University of Science and Technology. The thesis is the final stage of the 5 years M.Sc program Cybernetics and Robotics with specialization in Autonomous Systems and Control.

I would like to thank my fantastic supervisor Frank Westad, for his thorough feedback and guidance whenever needed, even in the most inconvenient of times. Frank Westad is Adjunct Professor at Department of Engineering Cybernetics, working with multivariate modeling.

The thesis is a continuation of the project thesis written during the Spring of 2019, which was a collaboration with DNB ASA. Chapter (2) is based upon this project. In the project thesis, PCA was performed in addition to a research study of a variety of pre processing and time series data analysis methods.

Topics for future work included performing prediction of financial default from financial customer data. This has been the basis for the master thesis. The main challenge in the project thesis was that performing PCA on the given data set yielded very little explained variance for the components, meaning there was a limited amount of underlying correlation between the variables. For the master thesis, the data set was therefore replaced with a data set containing more underlying information.

The data set in question is an open source Czech financial data set provided by the PKDD'99 Discovery Challenge. The work was done on a Intel(R) Core(TM) i7-8700 processor CPU with 3.2 GHz and 32 GB RAM running Ubuntu 18.04. Jupyter Lab was used as the primary development tool, with Python and R as programming languages.

Table of Contents

List of Tables

List of Figures

Abbreviations

PCA	=	Principal Component Analysis
PC	=	Principal Component
LPCA	=	Logistic Principal Component Analysis
PCR	=	Principal Component Regression
LPCR	=	Logistic Principal Component Regression
PLSR	=	Partial Least Squares Regression
MLR	=	Multiple Linear Regression
RF	=	Random Forest
LR	=	Logistic Regression
SVM	=	Support Vector Machines
TP	=	True Positives
TN	=	True Negatives
FP	=	False Positives
FN	=	False Negatives
PPV	=	Positive Predictive Value
TPR	=	True Positive Rate
KDD	=	Knowledge Discovery in Databases

Chapter 1

Introduction

1.1 Background

A loan is money or other material goods given to individuals, organizations, corporations or other parties in exchange for a promise of repayment along with interest [Investopedia (2018)]. A loan can be for a specific, final amount or as a line of credit to a specified limit as given by use of a credit card. The interest and fees acquired from loans are for many banks the primary source of revenue. A client in default means that the client has failed to meet the legal obligations or conditions for a loan. This is for example a mortgage or some type of loan with obligations within a certain time limit. Consequences includes being reported to credit bureaus, collection procedures or lawsuits [Carlson (2019)].

In Norway, loans in default increased with 4 billion NOK in 2018, to a total of 46 billion NOK. While the amount of Norwegians with payment remarks, meaning a warning of an overdue invoice after a debt collection process, remains stable at 6%, the amount owed has increased. Experts have expressed concerns of the rising use of credit cards and the following rising debts. Although it presents a considerate risk for the Norwegian economy, it is momentarily very profitable for financial institutions.

A Norwegian online newspaper for economics and business, E24, reported in late 2018 that the Norwegian bank Bank Norwegian had continued growth within number of clients and income, but the loan loss provision increased faster than the income. This halted the surplus growth the bank could have had with increased revenue [Marius Lorentzen (2018)].

In order to avoid such a situation, banks want to figure out how to detect problems and make predictions based on client behavior. Therefore the goal of this thesis is to build supervised models for loan default prediction using multivariate models and to gain insight and understanding around the prediction result.

1.2 Problem Description

The candidate will work on the problem of successfully classifying default customers on a Czech banking data set through applying multivariate classification methods. The following process is followed:

1. Explore methods of multivariate analysis to gain knowledge of suitable exploratory analysis methods and classification methods

2. Creating a structure of the frames of the data set to easily understand the relationships.

3. Aggregate the data set in an informative manner to create a feature set and identify the response variables.

4. Implement dimension reduction methods to explore underlying information.

5. Develop and train classification models with test set validation and cross validation in order to achieve satisfactory results.

6. Compare models with different data sets as input

The candidate will in addition try to answer the following research question:

1. What variables contribute the most in the classification decision process?

1.3 Software

The programming languages used in this thesis is `python` (v3.6.8) and `R` (v3.4.4). `python` is an ideal language for this task as it has a large ecosystem of built in packages and is the main tool for all multivariate analysis. Using Jupyter lab, a web-based user interface for Project Jupyter, it is very effective for combining runnable code with text, images and interactive visualization for both `python` code and `R` code. `R` was used primarily as a substitute for functions and models not available in `python` to save time and effort.

For assembling the data sets, `pandas` was heavily used, as it is very intuitive, high performing and cooperates well with other packages. `pandas` is crucial for this thesis, as the experiments are built upon data frames, which are very similar to matrices. They make it easy to define rows of samples and columns with variables. The rest of the thesis will therefore refer to matrices or tables as frames.
`numpy`, much used by `pandas`, was used for all mathematical operations and matrix-operations. For model implementation, `statsmodels` and `scikit-learn` is widely used because of their easy user interfaces. Finally, `matplotlib` is responsible for figures and visualization.

1.4 Report Structure

1. Chapter 1 is an introduction to the task and process
2. Chapter 2 concerns relevant background theory including an outline of the methods for multivariate analysis
3. Chapter 3 is an introduction to the data set, the acquisition and initial processing.
4. Chapter 4 regards applying the methods on the data set followed by a discussion of results
5. Chapter 5 concludes, evaluates and considers further work

Chapter 2 is based upon the thesis written in the course ’TTK4550 - Engineering Cybernetics, Specialization Project’ the Spring of 2019. The future work mentioned in the project thesis is largely explored in this thesis, however the work is done with a different data set. Therefore some of the work in the project thesis is similar to the first part of the master thesis, but with different results and interpretation.

Chapter 2

Theory

The following chapter provides the theory on which this thesis is built upon, ranging from the very beginning with pre-processing to the very end with prediction methods producing the main results. Matrices and vectors in the text will be marked in italic, while matrices in equations are written in bold and vectors in italic.

2.1 Pre-processing

A series of methods contribute to the success of applying multivariate methods. Implementing the methods of choice (such as for example PCR) alone does not guarantee good results. The data itself needs to have certain qualities. A noisy data set that is not properly pre-processed and validated correctly can lead to misleading and unreliable results. Outliers and missing data are also problems that challenge the success of the multivariate methods. Therefore the next section will present a few pre-processing algorithms often performed to avoid among other things, unwanted noise and variation [Kotsiantis et al. (2006)].

2.1.1 Missing values

A common problem is missing data from either X or Y sets. There are a number of strategies to handle missing values. Esbensen (2001) proposes the following two methods:

1. Skip missing values
   Compute scores and loadings and prediction frames as usual, but omit missing values in calculating the square sums as in e.g. the NIPALS algorithm.

2. Replace missing values
   Replacing missing values with for example the mean value of the variable where the

missing value occurs. It is however impossible to know if this object contained an extreme value or not. Another strategy is therefore to find the two most correlated variables, and interpolate the missing value with pair-wise correlation. One may also run PCA iteratively until convergence (imputation).

It is also emphasized that a missing value should never be replaced by 0 as it causes false results and can lead to unreliable interpretation.

2.1.2 Aggregation

Aggregation is a method used to combine two or more variables into a single variable. The result is variable reduction which leads to less processing power demand and the behaviour of variables is often more stable. The disadvantage is a reduction of variability. For example for a time series of hourly entries, reducing from hours to weekly by combining the hours into a single week will greatly reduce the amount of variables. For quantitative entries, the values are replaced by the sum of all values or the mean. For qualitative entries, a new vector entry of all entries is created. Depending on the type of data and the objective of the analysis, objects may also be aggregated.

2.1.3 Standardization

Standardization, also known as standard score or z-score, is the process of transforming individual variables with a mean μ and a variance σ^2 into having a mean $\mu = 0$ and a standard deviation $\sigma = 1$. It is calculated by the formula:

$$Z = \frac{X - \mu}{\sigma} \tag{2.1}$$

$$\tag{2.2}$$

With X as the raw value, with μ mean and σ standard deviation. With an unknown mean and standard deviation, a set of measurements X_i's are used to approximate the score by using the sample mean $\bar{X}$ and sampled standard deviation S_x.

$$Z_x = \frac{X - \bar{X}}{S_x} \tag{2.3}$$

$$\bar{X} = \frac{1}{n}\sum_{i=0}^{n} X_i \tag{2.4}$$

$$S_x = \sqrt{\frac{\sum_{i=0}^{n}(X_i - \bar{X})^2}{n-1}} \tag{2.5}$$

Standardization as a pre-processing technique is used to compensate that variables have different units. The idea is that all variables should have equal influence.

2.1.4 Categorical variables

Categorical variables, variables that can take on a value of a fixed amount of possibilities. An example of this is a category of payment types where the options are 'household payment', 'loan payment', 'pension payment', 'insurance payment' and 'payment for statement'. If the number of possible values is limited to two, the categories can be converted to binary values 0 and 1. However in this case, the number of possible values is larger and *one-hot-encoding* is therefore used. The different types is replaced with a series of binary variables. Problems arise when variables have a very large number of possibilities, like a data set with greater than 50 number of countries. In this case a better solution might be to for example convert countries to world continents, reducing the number of variables and sparsity significantly.

2.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis, also known as data mining, is a collection of methods with the purpose of finding a hidden structure in large, complex data sets. EDA finds this structure by investigating the systematic variation when all variables are acting simultaneously, not just independently. The EDA methods featured in this section is Principal Component Analysis and Logistic Principal Component Analysis.

2.2.1 Principal Component Analysis (PCA)

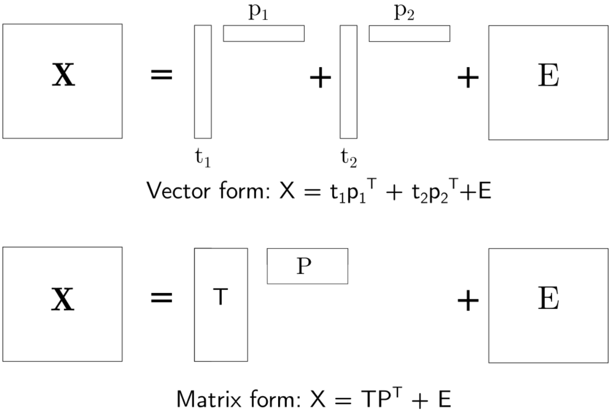


Figure 2.1: PCA conceptual model

Principal Component Analysis (PCA) is a bilinear method for reducing dimensionality. PCA is used under the assumption that the data set is structured with correlated variables such that there exist latent variables describing the data in a lower dimension variable space. 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 are 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 [James and Tibshirani (2013)].

PCA utilizes bilinear subspace models to model the structure. Bilinear modelling describes the data X $(m \times n)$ by using T scores and P loadings as shown in Figure (2.1). In mathematical terms, the idea is to create a $m \times a$ matrix T where m is the number of rows in the data set X and a is the number of principal components and a corresponding loading matrix P of size $a \times n$. The column T_i of X is the i-th score vector.

In the model p_1 refers to the loading vector of the first principal component (PC) and t_1 is the first score vector. The goal is to extract the first principal component and find the direction of the first PC that explains the most of the variance in the data, meaning the direction where the observations vary the most. The t_i 's are also orthogonal to each other. The loadings vectors are the same as the eigenvectors in linear algebra theory.

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

$$\sum_{j=1}^{n} Var(X_j) = \sum_{j=1}^{n} \frac{1}{m} \sum_{i=1}^{m} x_{ij}^2 \tag{2.6}$$

And variance explained by the k-th component:

$$\frac{1}{m} \sum_{i=1}^{m} t_{ik}^2 = \frac{1}{m} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} p_{jk} x_{ij} \right)^2 \tag{2.7}$$

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

$$\frac{\sum_{i=1}^{m} \left(\sum_{j=1}^{n} p_{jk} x_{ij} \right)^2}{\sum_{i=1}^{m} x_{ij}^2} \tag{2.8}$$

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

2.2.2 Logistic PCA

Logistic Principal Component Analysis, is an alternative to PCA for binary data. It is very similar to regular PCA, as the scores are a linear combination of the natural parameters from the saturated model. The principal components are easily interpretable, and requires only matrix multiplication to be applied to a new set of data [Landgraf (2013)].

The extension from PCA to binary data, the natural parameters forming a Bernoulli saturated model is projected and the Bernoulli deviance is minimized. If X is Bernoulli distributed with probability p_{ij}, then the natural parameter θ_{ij} is the logit of the probability

$$\theta_{ij} = log\frac{p_{ij}}{1 - p_{ij}} \tag{2.9}$$

The natural parameter from the saturated model is ∞ if $X = 1$ and $-\infty$ if $X = 0$ with a large number $\pm d$ instead of ∞. Thereby if $\bar{\theta}_i$ is the m-dimensional vector of natural parameters from the saturated model, the natural parameters are estimated as such

$$\hat{\theta}_i = \mu - UU^T(\hat{\theta}_i - \mu) \tag{2.10}$$

The PC scores for the i'th observation are

$$U^T(\hat{\theta}_i - \mu) \tag{2.11}$$

2.3 Multivariate Regression Modeling

The multivariate model for X and Y is a regression relationship between the X and Y sets. The first stage in multivariate modeling is therefore the calibration stage, followed by prediction.

For the calibration stage, a known X with corresponding Y data has to exist. Following is the development of the relevant multivariate regression model. Then the model can be used on new X measurements to predict new Y-values, this is the prediction part. The difference between the calibration and prediction stage is in addition explained by Figure (2.2).

The classical regression method is Multiple linear regression (MLR). It combines a set of several X-variables in linear combination by means of least squares and is the method of choice when the variables are orthogonal. However when the X-variables are correlated, the solution might be unstable and lead to erroneous interpretation. The solution to this problem is to use methods using latent variables such as Principal Component Regression (PCR) [Esbensen (2001)].

2.3.1 Principal Component Regression (PCR)

Principal Component Regression is a technique using latent structures in the data set to build a regression model from explanatory variables X [Esbensen (2001)]. It solves the collinearity problem by using scores from PCA in place of the original variables in X for

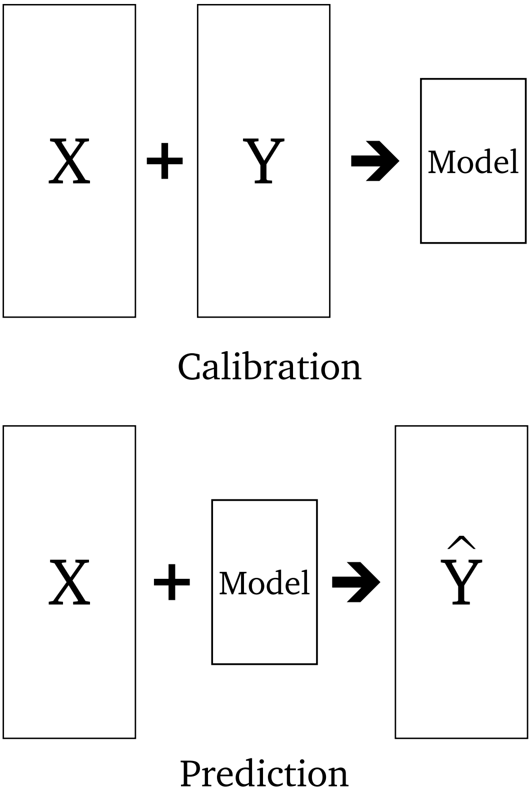


Figure 2.2: The two stages of multivariate modeling

MLR and solves the inversion problems and minimizes noise in X by finding the correct rank before regression to Y.

The first a components $T_1, ..., T_a$ from Section 2.2.1 are used as predictors in a linear regression model that is fit using least squares, described in Figure (2.3). The underlying assumption is that in the directions in which X show the must variation is the direction that is associated with Y. This assumption does not necessarily hold but is reasonable enough to give decent results. Overfitting is avoided by estimating only $a < m$ coefficients.

2.3.2 Partial Least Squares Regression (PLSR)

Partial Least Squares Regression (PLSR) focuses on the co-varying relationship between the Y-variance and the X-variance [Esbensen (2001)]. It connects Y directly to the decomposition of X such that the outcome constitutes an optimal balance between fit and precision. Figure (2.4) represent a view of PLS where the two equations symbolize two PCA models, PCA of X and PCA of Y. P and T represent loadings and scores, as for

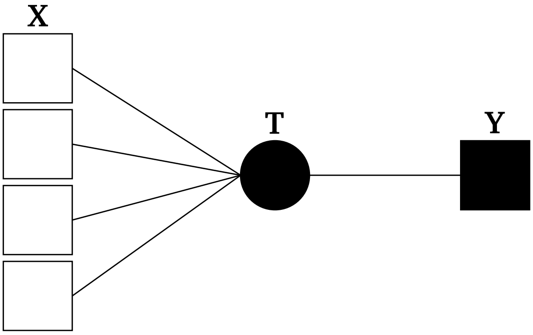


Figure 2.3: PCR [Esbensen (2001)]

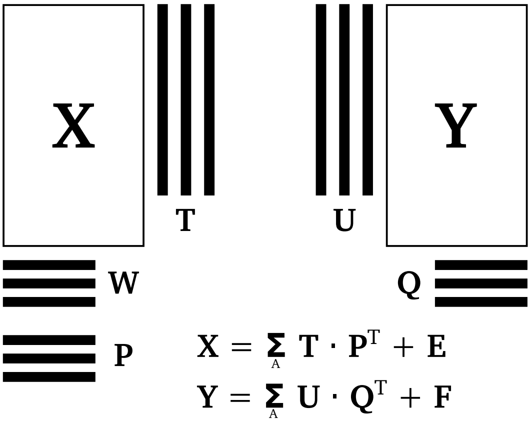


Figure 2.4: PLSR [Esbensen (2001)]

regular PCA of X, while Q and U is the equivalent for Y. W is called *loading-weights*, which is the connection between X and Y, or more precisely, the residuals in X and Y as the PLS factors are extracted and residual matrices are estimated. In the X-decomposition, it acts as a starting point for the t-score vectors. This way it does not really perform two different PCAs, but rather two connected ones.

The effect of this combination is that the X- and Y-space is now modeled interdependently. In other words, PLS uses the information in Y to find the Y-relevant structure in X where W represents the maximum (T,U)-covariance/correlation. Since PLS focuses on Y, the Y-relevant information is often found in the earlier components. However the effect can be subtle, so including a number of components can be beneficial to accumulate the impact.

2.3.3 Logistic Regression

The building blocks of a linear classifier is passing the output of a linear function through the threshold function [Russell (2016)]. A disadvantage of this approach is that this linear function is non-differentiable and discontinuous. This makes learning with the perceptron learning rule (Eq. 2.12) very unpredictable.

$$w_i \leftarrow w_i + \alpha(y - h_w(\boldsymbol{x})) \times x_i \tag{2.12}$$

This issue can be solved by approximating the hard threshold with a continuous, differentiable function like the logistic function (Eq. 2.13). Logistic regression converges more slowly, but behaves more reliably.

$$Logistic(Z) = \frac{1}{1 + e^{-z}} \tag{2.13}$$

Logistic Regression is a discriminative classifier, meaning the method models the probability of a data point belonging to a class directly without paying attention to the joint distribution.

2.3.4 Random Forest

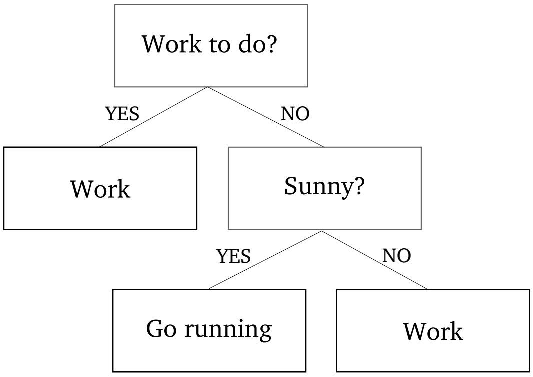


Figure 2.5: Decision Tree example

The building blocks of any forest is trees. In random forest the building blocks are decision trees (Figure 2.5). These consist of observations represented by branches and conclusions or classes represented by leaves. Decision trees are structures looking for data to split based on the largest difference [Russell (2016)].

A decision tree algorithm is a divide-and-conquer algorithm, which tests the most important attributes first, meaning the features that create the best splitting of the data. The

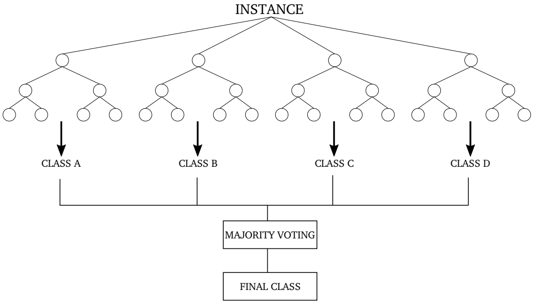


Figure 2.6: Random Forest classifier

Gini impurity or entropy measures are then used to measure the importance of a feature.

A forest model creates hundreds of decision trees, called an ensemble. Each tree is created by different randomly generated subsets of the original variables. The ensemble as a whole then creates a prediction. Random forest trees fixes a common issue with decision trees called overfitting, where the model fits the sample data a little too well and does not meet the prediction requirements. Each individual tree still has the issue of overfitting, however on average over hundreds of trees the overfitting is averaged out. Figure (2.6) displays an example of a simple random forest model.

2.3.5 Support Vector Machines

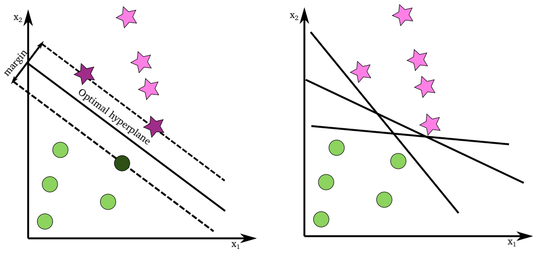


Figure 2.7: Support Vector Machines example

Support Vector Machines (SVMs) is a very popular classification and regression algorithm as it often produces significant accuracy with limited computational power [Bennett and Campbell (2000)]. The algorithm finds a hyperplane in a N-dimensional space that distinctly classifies the data points. The first graph of Figure (2.7) displays such a hyperplane. The second graph displays how there are many possible hyperplanes that can be chosen, however the goal is to find a plane that maximizes the margin, also known as the distance between data points of both classes. This is done such that future data points can be classified with more confidence. Support vectors are points that are closer to the hyperplane and therefore decides the position and orientation of the plane. In Figure (2.7) these are the darker colored points.

Mathematically, given a set of n data points such that there exist a set of (x_i, y_i) where x_i is a m-dimensional feature vector and y_i is a response class that take the values $y_i = \pm 1$. The output is then a set of weights w, one for each feature and the surface separating the classes form a hyperplane of the form:

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0 \tag{2.14}$$

Where w is the weight vector, x is the input vector and b is the bias. $\frac{b}{||w||}$ represents the offset from the origin. SVM seeks to maximize the margin of separation (d). In order to do this we have to minimize $||w||$ with the condition that are no data points between the neighboring planes such that

$$\begin{align} \boldsymbol{x}_i \boldsymbol{w} + b \geq +1 \, \textit{when} \; y_i &= +1 \tag{2.15} \\ \boldsymbol{x}_i \boldsymbol{w} + b \leq -1 \, \textit{for} \; y_i &= -1 \tag{2.16} \\ & \tag{2.17} \end{align}$$

This is a constrained, quadratic optimization problem solvable by the Lagrangian multiplier method and guaranteed to have a unique optimum. In practice, this is not always feasible.

2.3.6 Scoring Metrics

In order to compare the different classification methods, a measure of performance is needed. The three most common metrics to evaluate predictions are precision, recall and F_1-score. These metric consists of the number of *true positives* (tp), *true negatives* (tn), *false positives* (fp) and *false negatives* (fn). *True positives* and *true negatives* are the number of correctly labeled items belonging to the positive and negative class, respectively. *False positives* and *false negatives* are the number of items incorrectly labeled as belonging to the opposite class. Figure (2.8) displays a confusion matrix, used to elaborate on the meaning of these values.

		Predicted	
		Positive	Negative
True	Positive	*tp*	*fn*
	Negative	*fp*	*tn*

Figure 2.8: Confusion Matrix

Precision (or Positive Predictive Value (PPV)) measures how many of the observations predicted as positive are actually positive. Consider an example of trying to predict winter sports from a series of sports. Precision is the ratio of winter sports correctly classified as a winter sport. Recall (or True Positive Rate (TPR)) measures how many observations out of all positive observation are classified as positive. In the winter sports example, it displays how many winter sports was found out of all possible winter sports.

$$PPV = \frac{tp}{tp + fp} \tag{2.18}$$

$$TPR = \frac{tp}{tp + fn} \tag{2.19}$$

F_1-score combines precision and recall, as the harmonic mean of the two.

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \tag{2.20}$$

2.4 Validation

Choosing model complexity is a crucial part of a predictive model. Model complexity is often referred to as how many latent variables should be included in the model. With too high model complexity, the model is at risk of overfitting, while too little gives poor predictive performance. Model validation is therefore very important.

Model validation is often divided into two categories: 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. There are several types of internal validation methods, such as cross-validation and independent test set validation.

2.4.1 Test Set Validation

It is common to divide data into three parts: training set to fit the 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.

2.4.2 Cross-validation

Cross validation uses training data to determine the optimal complexity of the model. It uses the most important sources of variance. 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 training set, build a model with the remaining samples and predict a response for the left out sample, and compute the residual. Put the samples back into the training 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.

2.4.3 Explained Variance

The explained variance is the variance explained by the model. Each component of the model has an explained variance and depending on the desirable total explained variance, this decides how many components to retain. A normal total is usually 90%-95%. The rule of thumb of how much explained that is above a critical threshold is highly dependent on the type of data. The variance not included is called residual or noise. Cross validation is an additional method or tool to decide on the optimal dimensionality.

2.5 Plot Interpretations

2.5.1 Score Plot

Scores from PCA are most often visualized as a 2D scatter plot (example in Figure (2.9)) where each score in a score matrix corresponds to a projected data point down on the principal component axis. The components on each axis can be any combination of the vectors in the scores.

The plot can be interpreted by detecting groups, outliers or trends in the scores. In Figure (2.9) the section of points named 'Prague' is a group clearly distanced from the rest. Each axis contains the explained variance for the component, which is important

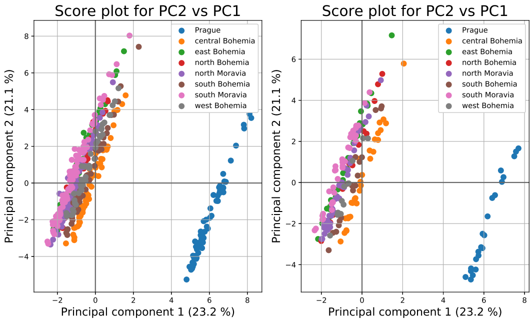


Figure 2.9: a) Training set b) Projecting the test set

because it is an argument for how important a trend is to the entire data set. For example if the explained variance for the first principal component and the second were 5% and 3%, then ’Prague’ as a separate group would not be as interesting because it would explain a very small part of the entire data set. ’Prague’ might only stand out for one or two variables in the original data set. Nevertheless, one cannot in advance for a specific data set assume that PCs with a small % explained variance do not contain relevant information or structure.

2.5.2 Loading Plot

The loading plot illustrates how much each variable contributes to each principal component. In particular, for a correlation loading plot each variables contribution is limited on a scale from 0 to 1 where 0 indicates no contribution. Figure (2.10) displays correlation loadings for PC1 vs PC2. The numbers for each point describe a variable explained in the legend. The orange circle is the 75% explained variance radii, while the blue circle is 100% explained variance. For Figure (2.10), variables 10 and 5 contributes quite a lot to principal component 1, but very little to principal component 2 as their values are ±0.30 on the ’Loading 2’ axis. Variables clustered together indicate high correlation between each other, meaning variable 10 and 7 are positively correlated while 10 and 5 have a negative correlation.

Correlation loadings is also an important factor to decide model complexity. Even if a components yields a cumulative explained variance of 90%, some of these a components might still not include usable information. This is where the explained variance radii is very helpful. Only the principal components where the correlation loadings have variables outside the 75% explained variance radii, or the very least 50% is within what is expected to be able to make a case for any interpretation.

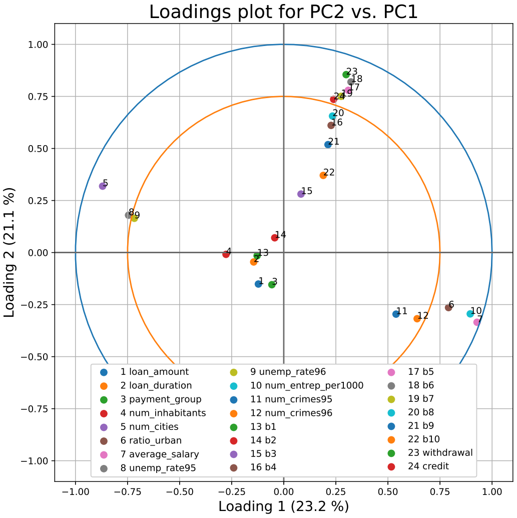


Figure 2.10: Loading Plot PC2 vs PC1

2.6 Hotelling's T^2 Statistic

Hotelling's T^2 statistics measure show close a new observation is to the model mean [Brereton (2016)]. It is used in multivariate datasets as a measure of distance from the centre of a distribution and follows the F distribution. Can be used as a means of converting the Mahalanobis distance from a centroid to a probability of belonging to a predefined multivariate distribution. Hotelling's T^2 statistic for sample i

$$T_a^2 = \sum_{i=1}^{a} \frac{t_i^2}{\lambda_i} = \sum_{i=1}^{a} \frac{t_i^2}{s_{t_i}^2} \tag{2.21}$$

Where $s_{t_i}^2$ is the estimated variance of the corresponding latent variable t_i. An upper control limit based on the a first PCs is obtained using the F-distribution and given by

$$T_{a,UCL}^2 = \frac{(m^2-1)a}{m(m-a)} F_{\alpha(a,m-a)} \tag{2.22}$$

Where $F_{\alpha(A,n-A)}$ is the upper $100\alpha\%$ critical point of the F distribution with $(a, m - a)$ degrees of freedom.

Hotelling's T^2 is a metric often used to detect outliers, by looking at the variations within the model. Intuitively, Hotelling's T^2 is the distance of each sample from the ideal zero-score situation meaning perfect fit.

The residual matrix E_A is calculated directly using, where A is the optimal dimension found from interpretation of the model results (see Section (2.4)).

$$\boldsymbol{E_A} = \boldsymbol{X} - \boldsymbol{T_A} \cdot \boldsymbol{P_A^T} \tag{2.23}$$

For multivariate SPC it is important to detect outliers based on Hotelling's and sample residuals, however this is not pursued further in this thesis.

Chapter 3

Data set and Pre-processing

This section is a walk-through of everything leading up to the results of this thesis. From acquiring the data from a Czech banking competition, to combining frames and forming a feature set. Each step of the way is documented along with important decisions and assumptions.

3.1 Data Acquisition and Pre-Processing

The data used in this paper was acquired from the PKDD'99 Discovery Challenge [PKDD], a challenge where the task was to ”define a problem which can help the bank to improve their services” and ”show how Knowledge Discovery in Databases (KDD) can be used to solve the problem”. The database was prepared by Petr Berka and Marta Sochorova and contain the following relations shown in Table (3.1).

Figure (3.1) describes how the relations are related to each other. Tables 'credit card', 'disposition', 'client', 'account', 'district' and 'loan' contain static characteristics exclusively, while 'order' and 'transactions' contain dynamic characteristics. A *client id* repre-

Relation	# Objects	Explanation
account	4500	Each record describes static characteristics of an account
client	5369	Each record describes characteristics of a client
disposition	5369	Each record relates together a client with an account
permanent order	6471	Each record describes characteristics of a payment order
transaction	1056320	Each record describes one transaction on an account
loan	682	Each record describes a loan granted for a given account
credit card	892	Each record describes a credit card issued to an account
demographic data	77	Each record describes demographic characteristics of a district

Table 3.1: Data set relations

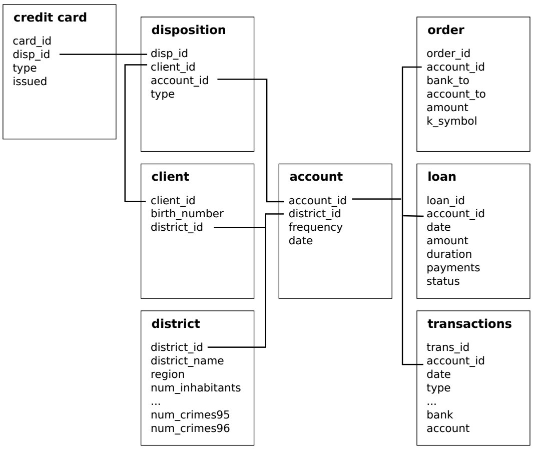


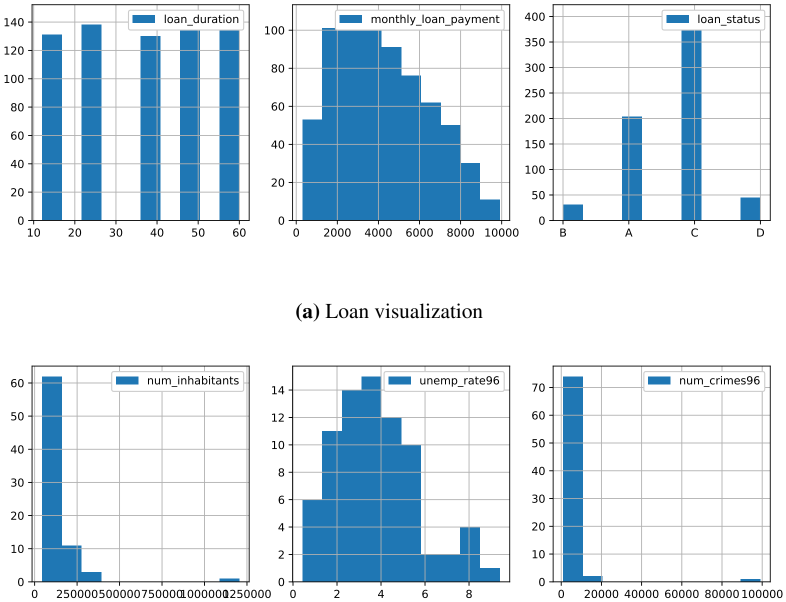
Figure 3.1: Relational table

sents any person with authority to manipulate an account, while an *account id* represent a single bank account. One client can have more than one account and more than one client can share one account. One account can have several cards but only one loan.

The data set is originally from real anonymized Czech bank transactions. Therefore characteristics such as type, symbol, and frequency are named in czech. For example ”polatek tydne”, meaning ”weekly” and ”sipo”, meaning ”household”, needs to be translated to get a better understanding of the data set. Converting dates to datetime, birth number to gender and age and monthly payment to loan payment group are all measures to condense information to ease the computational burden but most importantly condense the information in order to feed the most crucial information for the multivariate analysis. Therefore age is also converted to age groups. Districts are converted to region as there are 77 different districts, with approximately 60 inhabitants per district. Compared to 560 inhabitants per region, it is logical to switch from one to the other. Some variables contained missing values, marked by a single ’?’. This problem was solved omitting the samples with missing values. Alternatively one could have used imputation, however with many binary variables one has to be careful trying to estimate the missing values.

Figures (3.2) visualizes the distribution of the different variables for the region frame and the loan frame. For regions, it is clear that one region stands out from the rest when it comes to both crime and inhabitants, which is Prague. For the loan frame, loan status contains a lot more ’A’ and ’C’ than ’B’ and ’D’, and these are the non-default customers.

The transaction data frame is a time series ranging up to several years. To concatenate this data directly with the static frames will lead to the transaction related variables being of size up to 1000 while the card frame only forms 2 or 3 variables. If these two sets of variables are combined in a common frame, care must be taken of how to give weights to the two sets in the analysis.



(a) Loan visualization

(b) Region visualization

Figure 3.2: Histogram of Region and Loan

The transactions are ordered without temporal ordering. This is helpful to explore how the data is distributed. A theory explored in the project thesis was to see if the average default customer transaction distribution is different from the average non-default customer transaction distribution. If so, is the underlying reason related to small or large purchases. To view this distribution, a histogram is calculated.

The transaction data frame includes the transaction amount in absolute value and the balance after the transaction is made. Because all accounts begin at zero, it is therefore logical to view the transactions as the balance at all times. This meaning the histogram of the balance at all times yields a distribution of how often a customer carries a large amount of money and the other way around. This histogram is computed with 10 bins and an example is displayed in Figure (3.3).

3.2 Data Aggregation

To create a feature set, a decision has to be made on what information is relevant for the problem description. A set of decisions on relevant variables are therefore made:

1. As two people can share an account and there is no way to differentiate between

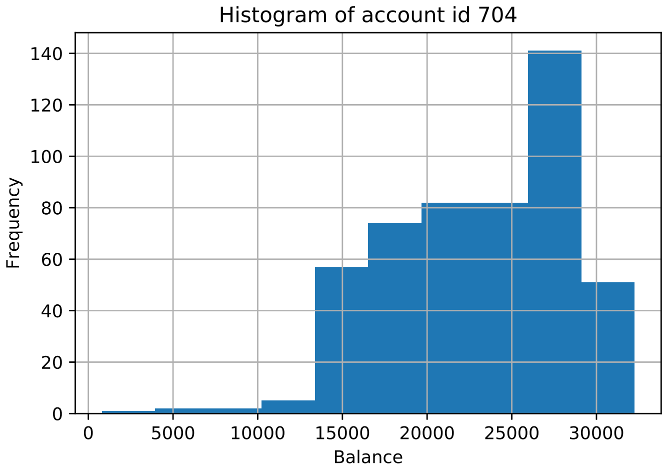


Figure 3.3: Histogram example

who made a transaction or acquired a loan, only the account owner ids are included.

2. Because one of the main response variables is loan status, all account ids are required to have acquired a loan. Therefore all ids without a loan is removed.

3. All variables containing 'number of municipalities' are excluded as they have little relevance to regions.

Data frames account, disposition, loan, client and card are then merged together. A response data frame is also aggregated, including loan status and region. Loan status is furthermore divided into one column with A, B, C and D, and one column with 1 and 0 (bad and good).

Afterwards the feature set is divided into separate numeric and categorical data frames, dividing the variables as shown in Table (3.2). Multi-valued categorical values are encoded into numerical variables by use of one-hot-encoding.

The categorical (binary) data frame is distributed as given by Table (3.3). It is clearly very sparse, as less than 20% of the data set contains 1's. A more evenly distributed set would be closer to 50/50. However, as some of the columns originate from an ordinal variable, there is closure present, this the sum of these will always be 1. One example is Region.

To get some initial intuition about the data, a correlation matrix is made (Figure 3.4). It clearly shows that the variables related to region are very correlated to each other in one direction or the other and likewise for loan variables, however the two groups do not correlate much with the other group. Likewise, Figure (3.6) shows the relationship between loan status and variables 'loan amount' and 'num crimes95' in a boxplot. For the number of crimes it doesn't look like any amount of crimes distinguishes itself from the rest, and

	Variables
Numeric	'loan amount', 'loan duration', 'payment group', 'num inhabitants', 'num cities', 'ratio urban', 'average salary', 'unemp rate95', 'unemp rate96', 'num entrep per1000', 'num crimes95', 'num crimes96', 'b1', 'b2', 'b3', 'b4', 'b5', 'b6', 'b7', 'b8', 'b9', 'b10', 'withdrawal', 'credit'
Categorical	'gender', 'has card', 'frequency after tr', 'frequency monthly', 'frequency weekly', 'card type classic', 'card type gold', 'card type junior', 'region Prague', 'region central Bohemia', 'region east Bohemia', 'region north Bohemia', 'region north Moravia', 'region south Bohemia', 'region south Moravia', 'region west Bohemia', 'loan status A', 'loan status B', 'loan status C', 'loan status D', 'age group 1', 'age group 2', 'age group 3', 'age group 4', 'age group 5', 'age group 6'

Table 3.2: Feature sets

contrary to what one would think, the districts with less crime can be interpreted as being more likely to default. Looking at the loan amount boxplot, it indicates that the less the loan amount is, the more likely the customer is to be able to pay it back and the other way around for larger loan amounts.

Variable	1	0
gender	0.51	0.49
frequency after tr	0.05	0.95
frequency monthly	0.82	0.18
frequency weekly	0.13	0.87
card type classic	0.13	0.87
card type gold	0.20	0.80
card type junior	0.02	0.98
region Prague	0.03	0.97
region central Bohemia	0.12	0.88
region east Bohemia	0.13	0.87
region north Bohemia	0.09	0.91
region north Moravia	0.17	0.83
region south Bohemia	0.09	0.91
region south Moravia	0.20	0.80
region west Bohemia	0.08	0.92
loan status A	0.30	0.70
loan status B	0.05	0.95
loan status C	0.60	0.40
loan status D	0.07	0.93
age group 1	0.02	0.98
age group 2	0.21	0.79
age group 3	0.23	0.77
age group 4	0.23	0.77
age group 5	0.23	0.77
age group 6	0.08	0.92
Average	0.19	0.81

Table 3.3: Categorical variable distribution

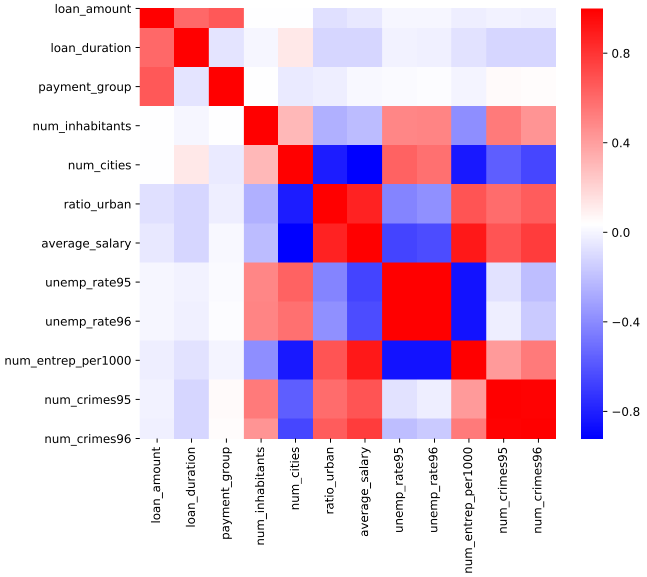
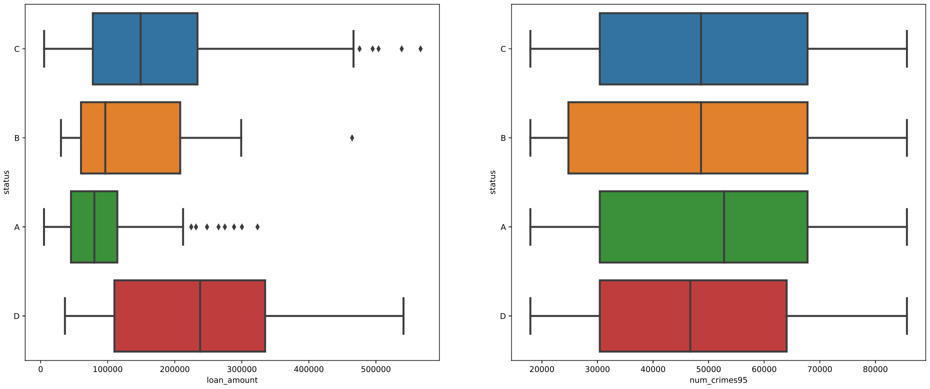


Figure 3.4: Correlation matrix for the numeric data



(a) Loan amount vs status

(b) Num crimes 95 vs status

Figure 3.6: Boxplots of variables vs loan status

Chapter 4

Results & Discussion

4.1 PCA

After having done some initial analysis, PCA is performed on the numeric data frame and Logistic PCA is performed on the categorical data frame. The reason for this is that PCA on binary variables might be influenced from variables with very skewed distribution. Each data frame is divided into training and test set and standardized beforehand, meaning test set validation is utilized as a result of there being a sufficient amount of objects to define a test set. Region and Loan status were displayed as sample groups in the score plot. The reason for not including them is to investigate if the the other variables directly could indicate differences without using them as categories as this is the objective with the classification and discrimination methods reported below.

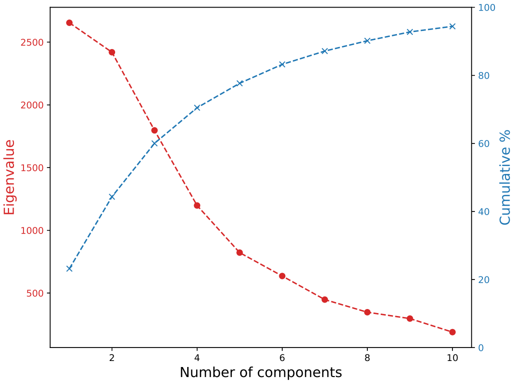


Figure 4.1: Scree plot and cumulative variance

The numeric set consists of 23 variables. In any data analytical procedure there is a risk of both overfitting and underfitting the model. Overfitting meaning that the closer the number of principal components are to the number of variables, the more of the variance is included. If the maximum possible number of components are calculated, the model is merely a rotation of the aces of the original variables. However, in most data sets, the underlying dimensionality is often (much) smaller due to redundancy between the variables and/or noisy variables. Underfitting on the other hand, means that the model is not able to capture the important information of the data set. A scree plot (Figure 4.1) showing the cumulative variance can be a good indication of the fit. Often it is desired to have at least 90% explained variance, however this is highly depending on the actual data. With 8 as the number of components, the total explained variance exceeds 90% for both the training and test set.

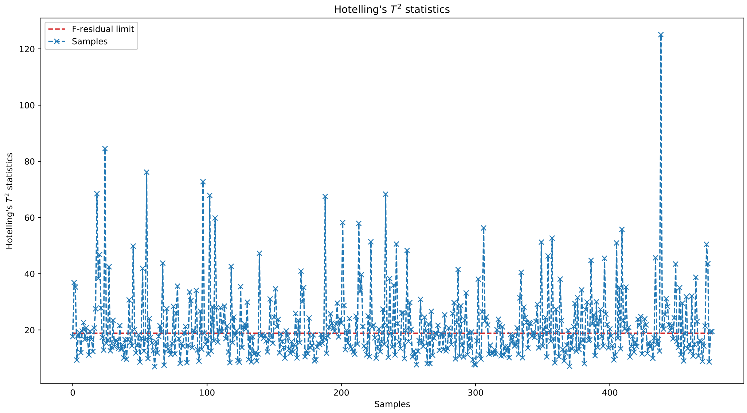
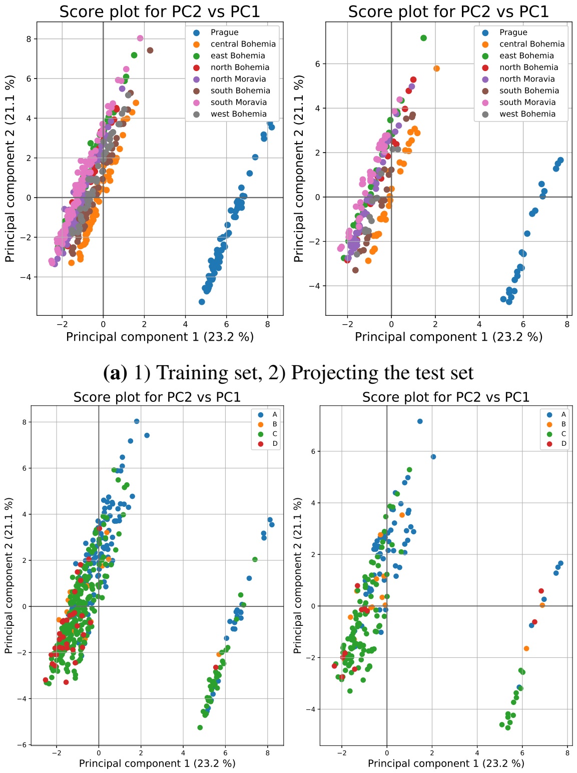


Figure 4.2: Hotelling's T^2 statistics

Figure (4.2) displays the Hotelling's T^2 statistics, a measure of the multivariate distance of each sample from the center of the model. Many samples look to be above the limit. This might be because the scores contain a number of groupings, thus the assumption that the data follow a multivariate t-distribution may not hold. Sample 438 distinguishes itself from the rest by having an abnormally high statistic. Looking more closely at sample 438, it is in payment group 9, meaning the group with the highest monthly loan payment. The transaction histogram indicates a lot of larger transactions and less smaller regular transactions.

Figure (4.3) shows PC2 vs PC1 for the training set to the left and the test set to the right grouped after Region above and Status below. The test set is projected onto the the principal components of the training set. The purpose of these plots is to compare the training scores with the test scores.

Observing Figure (4.3), part (a), Prague forms a cluster separated from the other re-



(a) 1) Training set, 2) Projecting the test set

(b) 1) Training set, 2) Projecting the test set

Figure 4.3: Score Plot PC2 vs PC1 with region and status visualized

gions, while the rest of the regions form clusters that are clearly connected but has the same linear pattern. The distinct shape each region cluster forms is not a surprise as the correlation plot in Figure (3.4) showed the region-related variables very correlated to each other and also clearly correlated to region itself. For the test set, Prague is still clearly distanced by the other regions. The test scores and training scores are shaped very similarly, and the axis values are more or less the same. This means the model generalizes well to new information.

The Status score plots (part (b)) show there does not seem to be any clear difference between the different status types. A and C looks to appear approximately on each side of the x-axis, however the 'bad' status types (B and D) appear on both sides. For principal components 1-8, only 1-5 have variables outside the 75% explained variance radii, such that the remaining three components will not be included in the plots and discussion.

Figure (4.4) displays the correlation loadings for PC2 vs PC1. By first glance, there are 10 variables outside the 75% radii, 'num cities', 'unemp rate95' on the upper left quadrant, 'average salary', 'ratio urban' and 'num entrep per1000' on the lower right and

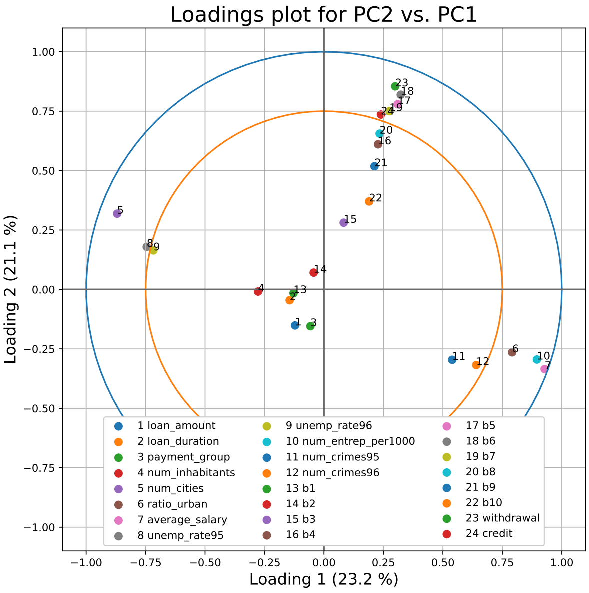


Figure 4.4: Correlation loadings for PC2 vs PC1

’withdrawal’, ’b5’, ’b6’, ’credit’, and ’b7’ on the upper right quadrant. ’num entrep per 1000’ and ’average salary’ sticks out as the variables closes to the 100% circle.

Figure (4.5) and (4.6) display 4 combinations of two principal components with corresponding correlation loadings with Region shown as a category. Figure (4.7) shows the four most interesting combinations with scores and loadings with Status as a category. The legend above represents the scores while the legend below represents the loadings.

Looking at the score plot and the corresponding loadings with points described in the previous paragraph, it is clear that there are a number of variables that contributes to Prague standing out as such in PC1. Although PC1 and PC2 are not showing a map with the variables directly on the PCs, the interpretation is nevertheless quite clear. PC2 reflects the transaction variables; withdrawal, credit and the bins describing the histogram of the client’s balance. This is more evident in the interpretation of PC3 vs PC2 below.

PC3 vs PC2 is seemingly one big pile, where the only mildly interesting observation is how the data points from Prague is situated slightly to the left of the other regions. The loadings for PC2 vs PC3 is on the other hand maybe the most interesting of the combinations. 10-11 of the variables form an almost half circle beyond the 75% circle, that includes ’b1’-’b9’ and ’withdrawal’ as well as ’credit’. This can be interpreted as loading 3 and loading 2 describing mainly the transactions data frame. Expect for Prague, this does not seem to be region specific. Bins 5 and 6 in the histogram are the ones that distinguish

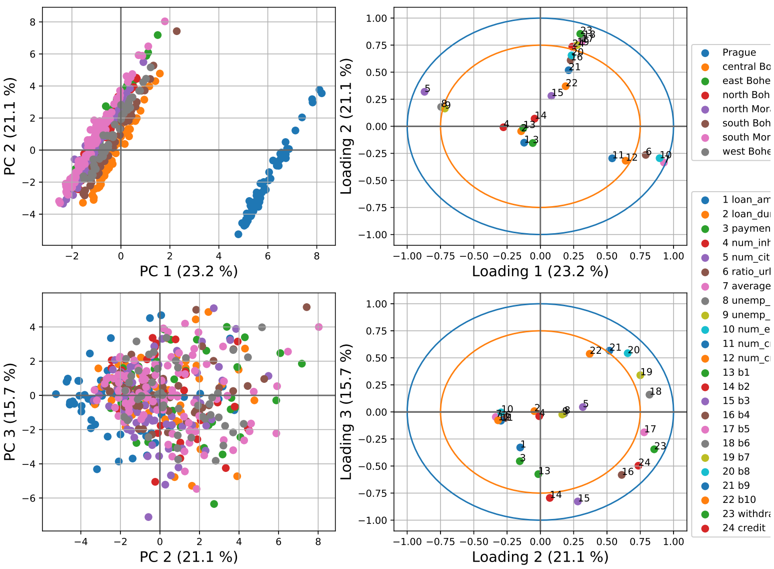


Figure 4.5: Score and Loading Plot, Region visualized, part 1

mostly between Prague and the other regions.

Moving to Figure (4.6), PC3 vs PC4 finds common ground for Prague and the rest of the regions, while they are roughly linear on the PC3 axis. For the loadings, only 4 variables are outside the 75% circle, 'num inhabitants' and 'num crime95' for Loading 3 and 'b2' and 'b3' for Loading 4.

PC4 vs PC5 almost completely separates the regions except for 'north Bohemia' and 'south Moravia'. This effect can be largely caused by the loan related variables by looking at the loadings. 'loan amount', 'loan duration' and 'payment group' are the only variables correlated above 0.5 for Loading 5, however this is not related to Region.

Separating the four Status categories is as mentioned earlier seemingly more difficult than separating the regions, visualized in Figure (4.7). PC1 vs PC2 is recognizable as Prague vs the rest (remember the same model is used for both responses), since most of B and D show up on the negative PC1-axis. This repeats for all four, but PC1 vs PC4 manages to mostly separate both status on the negative PC1-axis and the regions at the same time. PC5 is on the other hand the only component where the loan related variables are above the 50% explained variance line. There are no clear groups related to loan status,

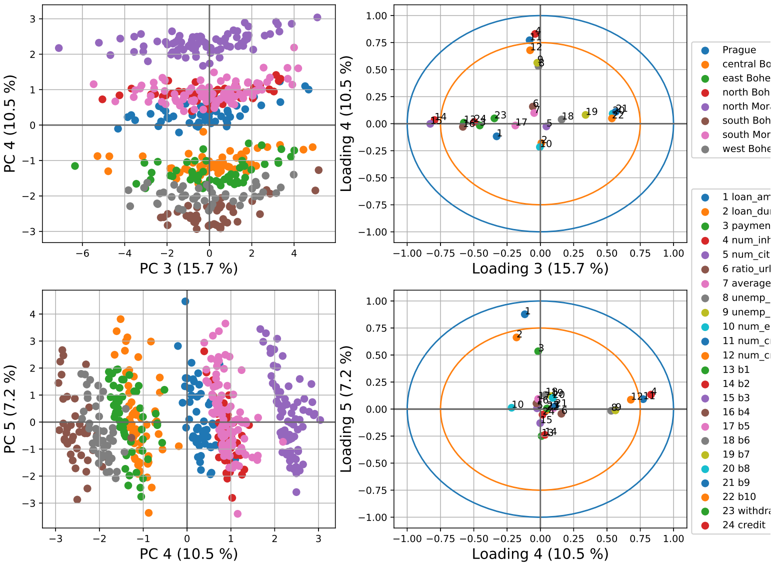


Figure 4.6: Score and Loading Plot, Region visualized, part 2

thus establishing a good model with classification and discrimination methods does not seem so promising.

4.2 LPCA

Although PCA may also be applied to binary data, Logistic-PCA (LPCA) is a better alternative. It is therefore implemented and performed on the categorical data frame. The categorical data frame loan status, which cannot be a part of the data frame and this therefore removed from the data frame.

Observable from the cumulative variance plot for LPCA (Figure 4.8) is a slower growing cumulative variance compared to the numeric data frame. However from 80% the model catches up with the numeric PCA model. This is because the PCA model components are sorted in a decreasing order yielding a logarithmic growth, while the LPCA models has a almost linear growth where the component with the largest explained variance is not necessarily the first component. The explained variance for the LPCA model is given in Table (4.1).

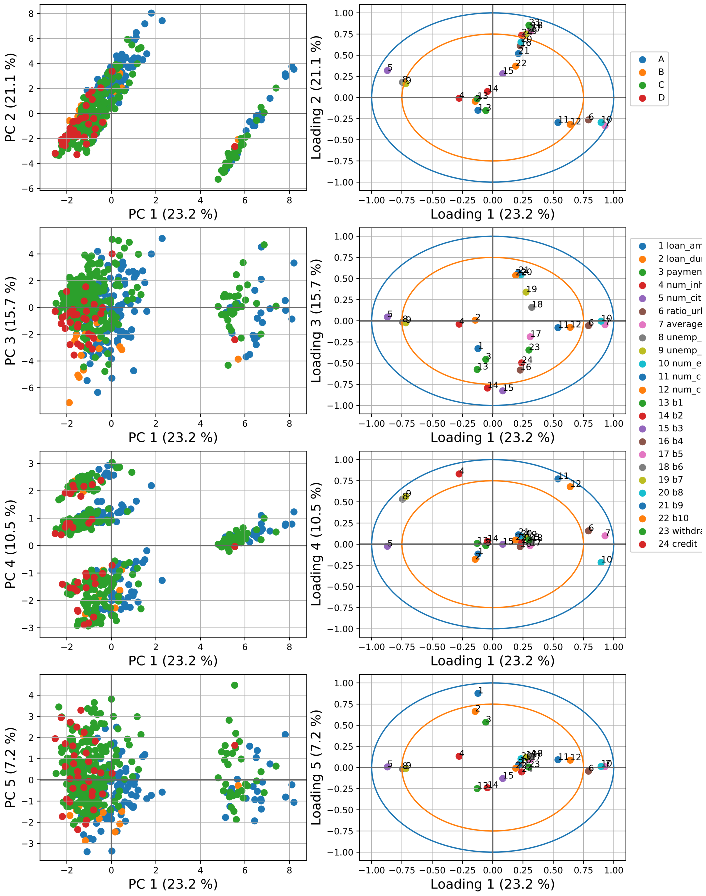


Figure 4.7: Score Plot interesting PC combinations, Status visualized

To get a cumulative variance above 90%, 9 components is needed. This is not a too bad result, however there are a lot of components with very similar amount explained variance. This often results in having to attain a larger amount of components than if the first two components covered 70% of the variance and the remaining 5 covered 20%.

Scores and loadings for the categorical data frame is visualized in two different types

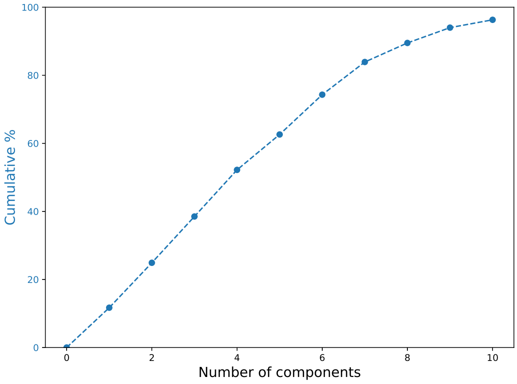


Figure 4.8: Cumulative variance for LPCA for Region and Status model

Component number	Status
1	0.117
2	0.132
3	0.136
4	0.137
5	0.104
6	0.117
7	0.096
8	0.056
9	0.045
10	0.023

Table 4.1: Explained variance for the LPCA model

of plots. Firstly is a score plot (Figure 4.9) of PC1 vs PC2 displayed with Status visualized, as well as three other variables. These were included in the X set for reference and insight on which variables contributed the most to the components. There is clearly no way to differentiate the status types. As the other categorical variables include frequency, age group, gender, card type and region, it might seem the status variable is very little correlated to these variables. Mapping the data samples to their corresponding variable values for age group, frequency and card type yields the other 3 visualizations. It looks as if there is no clear indication on what the three lines represent, but divide the points into three lines following the PC2 axis, and each line represent a frequency. For each line, there are somewhat distinguishable age groups. It is also interesting how for the samples on the positive PC1 axis, very few own cards.

Figure (4.10) display scores and corresponding loadings for four combinations with Status visualized. The legend above represent the scores while the legend below repre-

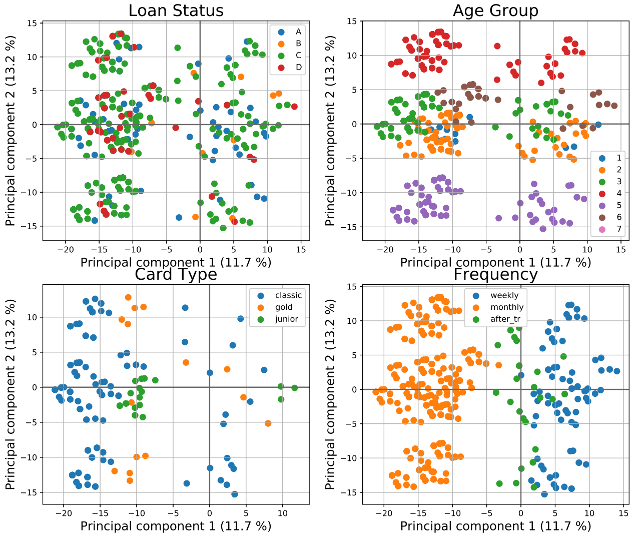


Figure 4.9: Score plot

sent the loadings. Observing the figure, most of the variables in the loading plot are close to the origin and the rest forms a skewed rectangle. Scores for PC3 vs PC4 form a pattern which is almost symmetric about both axis'. Which variables are the most correlated seems somewhat sporadic. Age group 6 is very correlated for loading 4, while age group 5 and 4 is correlated for loading 2, age group 2 and 3 is correlated for loading 3, and age group 1 is not very correlated for any of the first four components (or the next two). The same sporadic behavior can be spotted for the other variable groups except for the card types which are not very correlated for any of the components. This might be because a very small part of the samples have cards at all.

4.3 Classification and discrimination results

Since the data is highly imbalanced as there is a lot more non-default customers than default customers, bear in mind all models will be biased towards the non-default customers.

4.3.1 PCR, PLSR and LPCR

Based on the theory in Section (2.3.1) and (2.3.2) we want to predict Status (either default or non-default) using Principal Component Regression and Partial Least Squares Regression. The scores from PCA and LPCA are therefore fed to a 5-fold cross-validation for

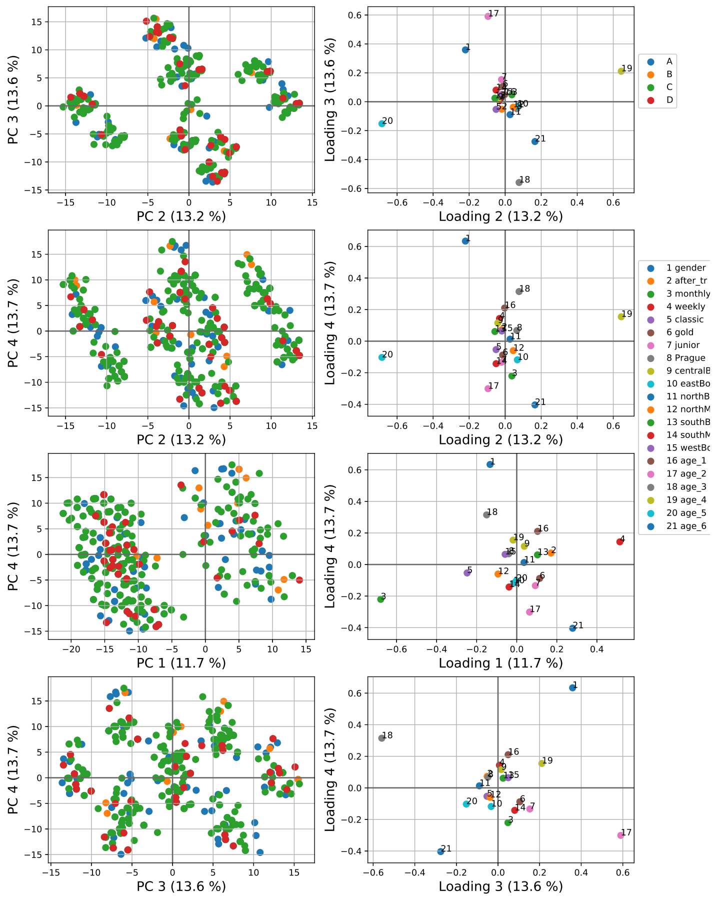


Figure 4.10: Score and loading plot

a conservative assessment of the model dimensionality to see how it influences the MSE. Figure (4.11) displays that the smallest cross-validation error occurs when ncomp=6 for PCR and ncomp=8 for PLSR for the numeric data set. The same is done for LPCR with the categorical data set. The Y-set for all three sets consists of either $[0, 1]$, meaning respectively default and non-default. In comparison to the status variable, 0 is the combination of both B and D, while 1 is the combination of both A and C.

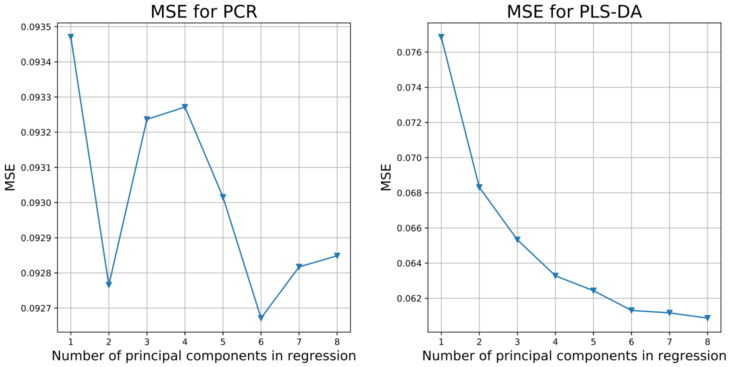


Figure 4.11: Cross Validation for PCR and PLS

Features	Model	Precision	Recall	F_1
Numeric	PLSR (ncomp=8)	0.88	0.98	0.93
	PCR (ncomp=6)	0.88	0.99	0.93
Categorical	LPCR (ncomp=9)	0.88	1.0	0.94

Table 4.2: PCR/PLSR/LPCR scores

The training data is then fed to the Scikit-Learn `PLSRegression` library to create a model and then trained using `fit`. Prediction of $\hat{Y}$ from X is done with the `predict` function on the test set.

For PCR, the precomputed model scores from LPCA and PCA is trained using the `fit` function from the Scikit-Learn library `LinearRegression`, followed by prediction on the test set with `predict`. Finally for all three models, precision, recall, F_1-score and confusion matrices are calculated and displayed in Table (4.2) and Table (4.3) respectively.

The precision, recall and F_1-scores (Table 4.2) for all three models are seemingly very good. However looking at the confusion matrices (Table 4.3), none of the bad clients are correctly classified for any of the models, with 25 false positives. Because of the poor result, it did not make sense to include the regression coefficients. In a practical context, this is not a usable result.

4.3.2 SVM

Implementation of Support Vector Machines (SVM's) is done using the feature set, the numerical score set for A components as chosen based on the explained variance and interpretation of the scores and loadings and the categorical score set. The data is split into

		Predicted	
		1	0
True	1	177	3
	0	25	0

(a) PLSR

		Predicted	
		1	0
True	1	179	1
	0	25	0

(b) PCR

		Predicted	
		1	0
True	1	180	0
	0	25	0

(c) LPCR

Table 4.3: PCR/PLSR Confusion Matrices

training and test data. Then the support vector classifier (SVC) in the Scikit-Learn `svm` library is used as the chosen built-in classifier. It takes one parameter, the kernel type. The `fit` function is then called to train the algorithm on the training set. Finally the `predict` function is called, which uses the trained model on a separate test set. Precision, recall and F_1-score is then calculated as explained in Section (2.3.6), and displayed in Table (4.8). There are four main kernels: linear, polynomial, gaussian and sigmoid. Table (4.4) shows results for the scores numeric set for the different kernels. Since all kernels show very similar results, the rest of the SVM classification in this thesis will be based on linear, as it is the only kernel that offers weight coefficients. These are important to interpret which variables contribute the most to the classification and can be found in Figure (4.12). Table (4.5) display the confusion matrix. See Section (4.3.5) for more detailed interpretation of the weights and coefficients from SVM.

Features	Kernel	Precision	Recall	F_1
Scores PCA (ncomp=5)	Linear	0.88	0.99	0.93
	Polynomial	0.89	0.87	0.88
	Gaussian	0.88	0.99	0.93
	Sigmoid	0.90	0.96	0.93

Table 4.4: SVM Kernels

		Predicted	
		1	0
True	1	180	0
	0	25	0

(a) Feature Set

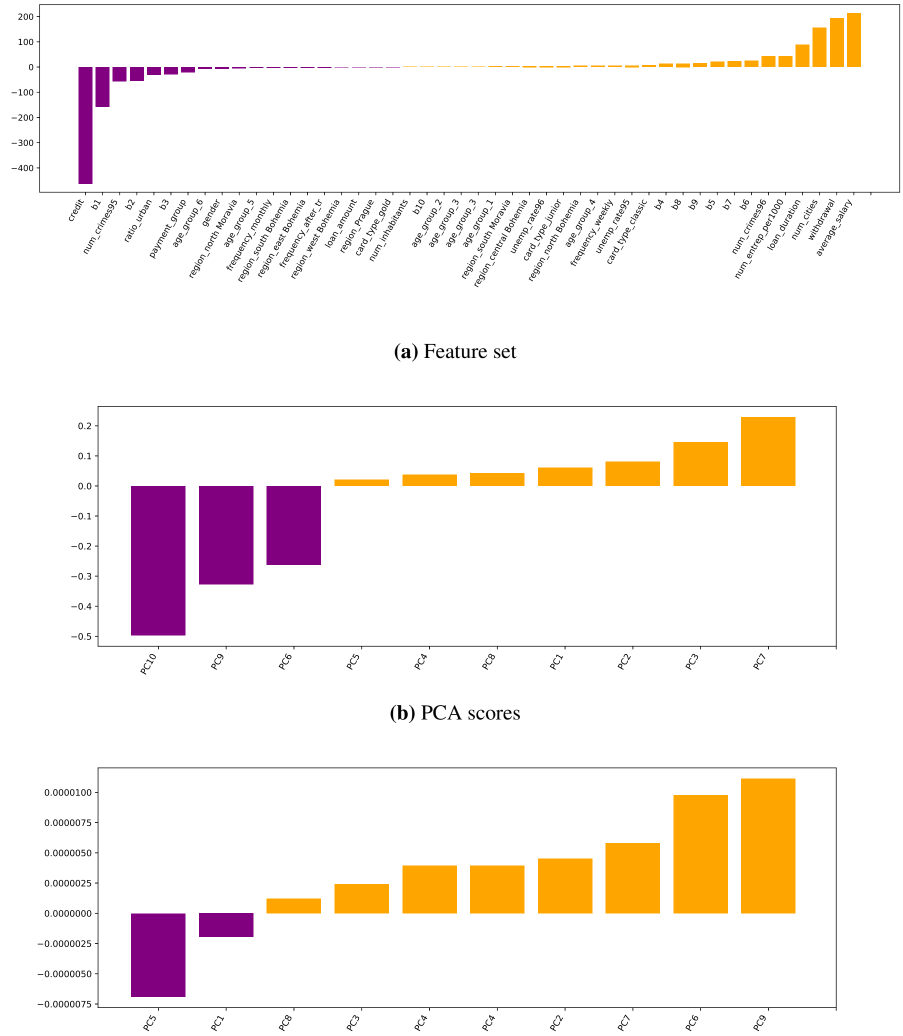
		Predicted	
		1	0
True	1	177	3
	0	25	0

(b) PCA

		Predicted	
		1	0
True	1	180	0
	0	25	0

(c) LPCA

Table 4.5: SVM Confusion Matrices



(a) Feature set

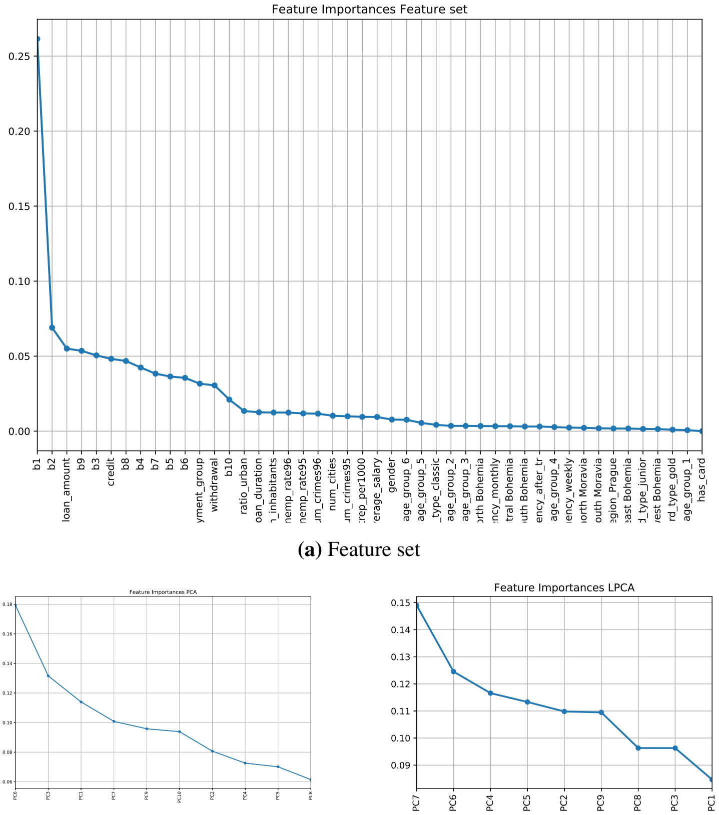
(b) PCA scores

(c) LPCA scores

Figure 4.12: SVM coefficients

4.3.3 Random Forest

Similar to section (4.3.2), the same data sets are used for Random Forest. Then the `RandomForestClassifier` from Scikit-Learn is applied to the sets, then trained using `fit` on the training data, followed by predicting class probabilities and finally predict classes on the test set. Finally, confusion matrix (Table 4.6), precision, recall and F_1-score (Table 4.8) and feature weight coefficients (Figure 4.13) are calculated. See Section (4.3.5) for more detailed interpretation of the weights and coefficients from RF.



(a) Feature set

(b) PCA scores

(c) LPCA scores

Figure 4.13: Random Forest Coefficients

		Predicted	
		1	0
True	1	179	3
	0	15	8

(a) Feature set

		Predicted	
		1	0
True	1	175	5
	0	24	1

(b) PCA

		Predicted	
		1	0
True	1	174	8
	0	19	4

(c) LPCA

Table 4.6: Random Forest Confusion Matrices

		Predicted	
		1	0
True	1	180	0
	0	25	0

(a) Feature Set

		Predicted	
		1	0
True	1	178	2
	0	24	1

(b) PCA

		Predicted	
		1	0
True	1	180	0
	0	25	0

(c) LPCA

Table 4.7: Logistic Regression Confusion Matrices

4.3.4 Logistic Regression

The `LogisticRegression` method from Scikit-Learn is very similar to both the `SVC` and the `RandomForestClassifier`. For the regressor a solver needs to be chosen, where the alternatives are ’liblinear’, ’lbfgs’, ’newton-cg’, ’sag’ and ’saga’. The main differences are how they handle L2, L1 or no penalty. ’liblinear’ is a good choice for small data sets and it turned out the best fit for our purpose.

The model is trained on training data followed by predicting class probabilities on the test set. The confusion matrix is computed and given in Table (4.7). Precision, recall and F_1-score is displayed in Table (4.8).

4.3.5 Comparison of results from classification methods

The precision, recall and F_1-scores for SVM, RF and LR can be found in Table (4.8).

Features	Model	Precision	Recall	F_1
Feature set	SVM	0.88	1.0	0.94
	RF	0.92	0.98	0.95
	LR	0.88	1.0	0.94
Scores PCA	SVM	0.88	0.98	0.93
	RF	0.88	0.97	0.92
	LR	0.88	0.99	0.93
Scores LPCA	SVM	0.88	1.0	0.94
	RF	0.90	0.96	0.94
	LR	0.88	1.0	0.94

Table 4.8: Classification scores

Comparing the methods, a series of measures can be compared. Firstly, what was run time for the different methods? Measuring run time can be an important factor if two methods give roughly the same results, but one method requires vastly more computational resources. If the chosen method will be used frequently further on and time is a restriction, choosing a slightly worse performing method with a tenth of the run time might be a better choice. For these methods, all of them ran under 1 second and their time is very similar.

Next up is how difficult the different methods were to implement. In the case that the work will be taken further or adopted by another similar project, it is important to be aware of which methods are difficult to implement and which methods are easy. All methods in this thesis were relatively easy to implement given a basic programming understanding and some experience with `scikit-learn` and `pandas/numpy`.

Another important factor is what type of misclassification is done. Using a medical example, falsely diagnosing a healthy patient as sick is nowhere near as bad as falsely diagnosing a sick patient as healthy. In the context of a confusion matrix, a false positive corresponds to falsely classifying a bad client as a good client, and a false negative corresponds to falsely classifying a good client as a bad client. Assuming all predicted bad clients will be manually reviewed for validation, then false positives is the misclassification to avoid. It is better to find a small group of potentials and then eliminate or verify one by one instead of not finding potential bad clients at all.

In what ways can the results be improved? The maybe most common problem when applying classification methods is a lack of data. The final data table included 682 samples, with 477 samples in the training set and 205 in the test set. This is not a very big data set and increasing the number of samples might very well improve the results. The same can be said for the number of variables, which varied a lot depending on if the whole feature set was used, or the numeric or categorical one. Ultimately the main problem is most likely how the data set is very unbalanced (11%/89%). Realistically, the amount of default customers in a bank will always be severely outnumbered by the number of non-default, meaning any data set with similar characteristics will be unbalanced unless the ratio is manually altered.

Random Forest, Logistic Regression & SVM

When comparing the results for Random Forest, Logistic Regression and Support Vector Machines, it involves mainly classification scores and confusion matrices. Looking at the scores in Table (4.8), Random Forest with Feature set as input scores marginally better than the rest. Since the data set is very biased towards Non-Default clients, it is important to consider the confusion matrices. Looking at the SVM confusion matrices (Table 4.5), none of the input sets are able to predict any True Negatives, also known as default clients. Logistic regression (Table 4.7) manages to predict one correct default client. Finally Random Forest (Table 4.6) does quite a lot better, especially for the future set. 8/25 correctly classified default clients is not however a great result.

Feature Importance

To quantify the usefulness of all the variables, a feature importance plot can be interpreted (Figure 4.13). The by far top feature in Figure (4.13c) is $b1$, followed by $b2$ and $loan$ $amount$. $b1$ is the first bin of the transaction histogram, meaning the absolute lowest balance. The coefficient value itself represent how much including $b1$ will improve the prediction results. The top three features are pretty intuitive as it makes sense that defaulting is very related to how little the bank account balance is and the amount of loan is acquired. Figure (4.14) display a histogram of the distribution for ’b1’ for default and non-default clients. Disregarding the obvious imbalance between the amount of clients in each category it is clear that a significantly larger portion of the default clients’ balance can be found in ’b1’ than for the non-default clients. This confirms the observation from the feature importance plot.

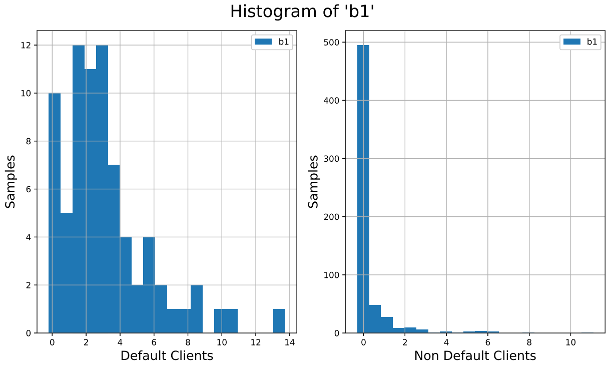


Figure 4.14: Histogram of ’b1’ for default and non-default clients

Comparing the most important features in random forest to the most important features in SVM (Figure 4.12). Since the chosen kernel is linear, the result is a hyperplane (Figure 2.7). This hyperplane separates the classes. The absolute value of each feature indicates how important the feature was for the separation. The features with the largest absolute value is $credit$, $b1$, $average\ salary$ and $withdrawal$. $b1$ is recognizable from the RF coefficients, and $credit$ and $withdrawal$ is related to the transactions. $credit$ is a number of how many transactions in total of the type credit. Credit transactions is intuitively related to financial trouble, particularly in the case of large or frequent credit transactions.

For both Random Forest and SVM, coefficients for PCA scores and LPCA scores are included, however both cases provide ambiguous results such as PC1 for PCA having very little importance despite having the most explained variance. Running SVM and Random Forest several times with PCA and LPCA scores gave very similar scores and confusion matrix, however the coefficients kept changing.

Chapter 5

Conclusion and Further Work

5.1 Conclusion

This thesis has investigated what primary differences can be found for default and non-default clients as well as explored different classification methods in order to predict default clients. PCA and LPCA were two model reduction methods utilized with numeric and categorical variables, respectively. These methods also served the purpose of exploratory analysis of the two sets of variables. In addition, a comparison of the performance of 5 different classification methods was conducted.

The PCA was validated by projecting the test set onto the model based on the training set confirming the common structure in both. Variables subject to each region were highly correlated and it was therefore possible to almost completely separate the different regions in principal components. Furthermore, the transaction related variables and the region related variables contribute looking at the loadings, while the loan related variables do not. Visualizing status gave very little clear information about the differences between good and bad clients, however quite a lot of the variables were very correlated which is a good basis for classification.

The LPCA model lack the ability to separate the status types, and the loadings have a very sporadic pattern for variable correlation. In combination with a lower explained variance per component, gender, region, card type, region and age show very little underlying correlation, causing the results to be fairly difficult to interpret.

Classifying default clients proved to be difficult, largely caused by a very imbalanced data set. The non-default clients are preferred over the default, and both SVM and LR finds it more profitable to classify most clients as non-default. Random Forest does slightly better by classifying 8/25 default clients correctly. This is however not satisfactory for practical usage.

Earlier on, a research question was formulated: 'What variables contribute the most in the classification decision process?'. The answer can be found in the feature importance plots. Both SVM and Random Forest showed that the $b1$ bin from the transaction balance histogram is wildly important for prediction success. It is also intuitive that having a low balance over a longer period of time may cause some financial trouble. Another important variables include $b2$, $loan\ amount$ and $credit$ (amount of credit transactions), also logical factors.

The results are subject to improvement, particularly the classification scores. The combined training and test set make up 682 samples or clients, which in a data mining context is not a lot of data. In addition, having less than 25 variables in each data frame might not reflect the information needed to predict default. A good foundation of information with a sufficient number of samples is crucial to succeed in this type of problem. To conclude, the results are not good enough to use in a practical decision process with such inability to classify the default clients.

5.2 Future Work

This thesis has some limitations when it comes to the data set as mentioned in Section (5.1), and these are definitely subject to future improvement. In addition to the variables that were available in this study, further client-related and derived variables can be included which would enhance the information foundation. For example, including more of the time series effect such as how long the loans last, how long has it been for (so far) successful loans and (so far) unsuccessful loans. In addition the distribution of how many credit and withdrawal transactions were carried out over time. Another data table, Orders, was not included in the analysis, to safely scope the thesis within the time limit given. Orders make up the second of in total two time series tables.

Another suggestion for further work is to implement higher order methods such as PARAFAC by including a third dimension in the data set, for example representing the time series for each client, which in a coherent way might give some meaningful insight.

Lastly, since both PCA, LPCA, PCR, LPCR and PLSR were implemented, the only method missing is Logistic PLSR. Since PLSR computes the principal components by use of the covariance to Y, Logistic PLSR is not so straight forward as for Logistic PCR. Doing so might be a part of a potential Ph.D. project.

Bibliography

Bennett, K.P., Campbell, C., 2000. Support vector machines: hype or hallelujah? Acm Sigkdd Explorations Newsletter 2, 1–13.

Brereton, R.G., 2016. Hotelling's t squared distribution, its relationship to the f distribution and its use in multivariate space. Journal of Chemometrics 30, 18–21. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.2763, doi:10.1002/cem.2763, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.2763.

Carlson, R., 2019. Financial definition of default. URL: https://www.thebalance.com/what-does-it-mean-to-default-on-a-loan-4684116.

Esbensen, K.H., 2001. Multivariate data analysis - in practice : an introduction to multivariate data analysis and experimental design.

Investopedia, 2018. Loan-definition. URL: https://www.investopedia.com/terms/l/loan.asp.

James, G., W.D.H.T., Tibshirani, R., 2013. An Introduction to Statistical Learning. Springer. URL: http://www-bcf.usc.edu/~gareth/ISL/.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P., 2006. Data preprocessing for supervised learning. International Journal of Computer Science 1, 111–117.

Landgraf, A.J., 2013. An introduction to the logisticpca r package. URL: https://cran.r-project.org/web/packages/logisticPCA/vignettes/logisticPCA.html.

Marius Lorentzen, E.A.N., 2018. Kraftig økning i lånetapene for bank norwegian. URL: https://e24.no/boers-og-finans/i/WLPBMg/kraftig-oekning-i-laanetapene-for-bank-norwegian.

PKDD, . Ecml/pkdd discovery challenges 1999 - 2005. URL: https://sorry.vse.cz/~berka/challenge/PAST/index.html.

PWC, . Fintech. URL: https://www.pwc.no/no/teknologi-omstilling/digitalisering-pa-1-2-3/fintech.html.

Russell, S., 2016. Artificial Intelligence: A Modern Approach, Global Edition. Pearson. URL: http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1419715&site=ehost-live.

NTNU

Norwegian University of
Science and Technology