Jostein Bjørnum
Magnus Urheim

# Predicting default on unsecured loans in the Norwegian market

## A study using a neural network for default predictions on real world application data

Master's thesis in Economics and Business Administration
Supervisor: Denis M. Becker
May 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

NTNU
Norwegian University of
Science and Technology

Jostein Bjørnum
Magnus Urheim

# Predicting default on unsecured loans in the Norwegian market

## A study using a neural network for default predictions on real world application data

Master's thesis in Economics and Business Administration
Supervisor: Denis M. Becker
May 2023

Norwegian University of Science and Technology
Faculty of Economics and Management
NTNU Business School

NTNU
Norwegian University of Science and Technology

# Preface

This thesis concludes our MsC within Business Analytics (MØA), at NTNU Handelshøyskolen. After five years of sweat, blood and tears, learning and friendships that will last forever, we look forward to take the big step into the working life. It has been a period we soon will forget, with stressful days and weeks, as well as funny moments and good coffee breaks.

We would like to extend a big thank you to our supervisor Denis M. Becker for the necessary assistance and guidance throughout this half a year. Even if you have been away from Trondheim, you have not been hard to get hold of. We also want to thank SpareBank 1 for providing access to a dataset of real world data. Hopefully our product is something you can use in the time to come.

Last, but not least, we want to thank eachother for great teamspirit and tough discussions, and our roommates Aleksander and Vegard for keeping us calm in desperate times, and for keeping our office exceptionally tidy.

We take full responsibility for the content of this thesis.

Jostein Bjørnum
josteinbj@yahoo.no

Magnus Urheim
murheim@ebnett.no

# Sammendrag

Mislighold av usikret gjeld medfører ikke bare tap for finansielle institusjoner, det øker også kostnadene for låntakere som gjennom en høyere rente må kompensere for andres mislighold. I denne oppgaven tar vi for oss bruken av et nevralt nettverk som en metode for å predikere mislighold av usikret gjeld. Hvor man i det norske markedet hovedsakelig bruker tradisjonelle metoder som logistisk regresjon, har man i andre land sakte, men sikkert økt fokuset mot maskinlæringsteknikker. Forskning på området viser til gode resultater, og aktualiteten er kanskje på sitt høyeste slik vi ser verden i dag.

På bakgrunn av problemstillingen til denne masteroppgaven: *"How do neural networks perform in predicting defaults on unsecured loans using application data?"*, har vi fått tilsendt et datasett med godkjente lånesøknader fra Sparebank 1, begrenset til tidsperioden fra midten av 2019 til og med 2022. Dette datasettet inneholder 93.039 observasjoner, hvorav 3,83% er definert som mislighold etter en periode på 12 måneder. Basert på dette, lager vi et multilayer feed-forward nevralt nettverk som skal fange opp så mange som mulig av de som misligholder. For å gjøre observasjonene mer tolkbare benyttes SHAP. Videre, for å evaluere ytelsen til nettverket, lager vi to ulike logistiske regresjoner. Ubalansen i datasettet vil håndteres ved å bruke en vektet tapsfunksjon.

Det nevrale nettverket og de logistiske regresjonene opptrer relativt likt på 2 av 4 evalueringsmål. Derimot ser vi en tendens til at det nevrale nettverket klarer å finne flere av de som misligholder, 68 av 503, enn det de logistiske metodene gjør, henholdsvis 28 av 1.068 og 25 av 534. Dermed ser vi at det kan være muligheter for markedet å dra nytte av et slikt nettverk. En implementering av liknende modeller hos långiveren kan gi reduserende effekt i risiko og usikkerheten knyttet til å gi ut lån, samtidig som det kan gi en økonomisk vinning for låntakeren. Men, selv om vårt nettverk viser gode prediksjonsevner, konkluderes det kun med at det er potensiale i bruken, og at det foreligger grunnlag for videre forskning på feltet.

# Abstract

Default on unsecured loans not only causes losses for financial institutions but also increases costs for borrowers who, through a higher interest rate, have to compensate for others' defaults. In this thesis, we consider using a neural network to predict default on unsecured loans. Where traditional methods such as logistic regression are mainly used in the Norwegian market, the focus has slowly but surely increased towards machine learning techniques in other countries. Research on the task shows good results, and the relevance is perhaps at its highest as we see the world today.

Based on the thesis problem: *"How do neural networks perform in predicting defaults on unsecured loans using application data?"*, we have been sent a dataset containing approved loan applications from Sparebank 1. It is restricted from the middle of 2019 and throughout 2022. The dataset contains 93.039 observations, whereas 3,83% are defined as defaults after 12 months. Based on this, we build a multilayer feed-forward neural network with the aim that it should find as many defaulters as possible. To make the observations interpretable, SHAP is used. We build two logistic regression models to evaluate the neural network's performance. The imbalance in the dataset will be handled with a weighted binary-cross-entropy loss function.

The neural network and the logistic regressions perform relatively equally on 2 out of 4 evaluation targets. However, there are signs that the neural network finds more of those who default, 68 out of 503, than the logistic regressions do, 28 out of 1.068 and 25 out of 534. Thus, there may be opportunities for the market to benefit from such a network. Implementation of similar models by the creditors can have a reducing effect on risk and the uncertainty associated with issuing loans. At the same time, it can provide financial gain for the borrower. However, even if our neural network shows good predictive capabilities, it is only concluded that there is potential in its use and that there is a basis for further research in the field.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

In this thesis, we want to evaluate neural networks as a tool for financial institutions. Specifically, we look at neural networks' capabilities to predict defaults in the Norwegian market for unsecured loans. This is important to both the customers and the financial institutions, as it can cut costs for both parties. Due to uncertain times for Norwegian debt holders, cutting customers' costs is highly topical. This also reflects in the risk creditors take, which again is compensated for by the customers.

The cost of living in Norway is increasing as a result of several factors. According to Statistics Norway (SSB), the average household had NOK 18.200 less available funds in 2022 compared to 2021. This is due to two crucial factors. The first factor is the increasing cost of necessities. The second factor is reduced purchasing power (lower salary increase compared to the beforementioned increased cost for necessities) (Tuv, 2022). In a time when the inflation rate in Norway is at a high level, and the financial uncertainty for families is high, we seek to improve the conditions for all parties involved (Trading-economics, 2023). The default rates for consumer loans are still high, even though these default rates have declined the recent years. For the first half of 2022, it was 9,9% (Finanstilsynet, 2022b).

Due to the relationship between risk, reward, and uncertain times, the companies offering these kinds of credit services will require a higher yield on their capital. The consequence will be that the consumers will pay more for their loans in already challenging times. By reducing risk, the consumers will pay less for their loans, and the financial institutions will meet their required return on investment. This will be theoretically possible since three parts influence the interest rate. One of these parts is compensation for risk, which we seek to lower (Visma, n.d.).

Loans, including unsecured loans, are regulated by "Låneforskriften", the legal boundaries financial institutions need to follow to approve a loan application (Finansdepartementet, 2022). In this thesis, we seek to build a neural network that follows these rules and gives predictions on whether the applicant will default on the loan or not. We have created two separate logistic regressions to evaluate the neural network, where the variables used are selected by a lasso feature selection technique.

The relevance of this topic these days is perhaps at its highest as we see not only increased prices but also more significant differences in people's private economy and less purchasing power (Schjetne, 2023). The last year's increase in living costs and the steady decline in unsecured loan defaults is a surprising fact, especially when purchasing power has decreased for many people in Norway.

To build models for the thesis' purpose, we have received real world data from a financial institution in Norway. By evaluating the provided information, there is a fear of higher default rates on unsecured loans in the near future. In times with this much uncertainty, we seek to reduce the risk of participating in the credit market. By that, we also want to lower the unsecured loans' interest rates by making default less likely on approved applications. Therefore, we seek to answer the following thesis statement:

*How do neural networks perform in predicting defaults on unsecured loans using application data?*

The thesis statement itself is broad, but by answering the three research questions, we aim to genuinely assess the neural networks' performance.

*Will the neural network be of support in the aim to reduce the risk on behalf of the financial institution?*

*Is it possible to create a shift in the credit market's uncertainty, which will benefit both the creditors and debtors?*

*Can the model be optimized for use in the creditors decision making?*

We seek to evaluate the model statistically and the repercussions of implementing it, both for creditors and debtors. Seeing as they are both affected by the uncertainty of the market. There are statistical evaluations of similar models built on data from different countries. Some of these results will be presented, as well as the differences in performance between the different types of models used within the field.

The structure of this thesis is as follows: In Chapter 2, we will go through the theoretical aspects that we see as most closely related to unsecured loans, as well as previous literature on the field. In Chapter 3, there will be a brief introduction to our data. In Chapter 4, we will introduce the methods used in this thesis and a final specification of our neural network. After this, we will in Chapter 5 introduce our results. Before we discuss the results and the practical use of the model in Chapter 6. At the very end, in Chapter 7, we will give our conclusions based on the problem and the final research questions and discuss weaknesses in the thesis and further research.

# 2 Theory and literature review

In this Chapter, we look at the theory that is relevant to this thesis. We also have a further look at previous research. As far as we know, the problem has yet to be specifically analyzed for the Norwegian market.

## 2.1 Unsecured loans

Unsecured loans are loans that do not require any type of collateral. Lenders approve unsecured loans based on the borrower's creditworthiness and put none of the borrower's property, car, or other assets as security. There are many types of unsecured loans. Some could be student loans, credit cards, and consumer loans (J. Chen, 2023a). As it does not include anything other than a credit assessment from the banks, a more significant risk will also be associated with this. This is something every bank must consider when approving applications.

### 2.1.1 Unsecured loans in the Norwegian market

The enterprises included in the Norwegian Financial Supervisory Authority's consumer loan market survey decreased consumer loan volume by 11,2% in 2021. From NOK 92,7 billion in 2020 to NOK 82,4 billion at the end of 2021. By the end of 2021, the default rate was 11,2%. The year before, it was 13,3%. In Norwegian banks who are specialized in consumer loans, the default rate was 15,9% in 2021 and 19,3% in 2020. It is essential to state that the development of the default rate is affected by the fact that banks sold non-performing loans in 2021 (Finanstilsynet, 2022a).

Co-borrowers are defined as two or more responsible for the debt, and the information is reproduced for each debtor (Gjeldsregisteret, n.d.-b). The total unsecured debt adjusted for co-borrowers was NOK 126,3 billion in December 2022, a decrease from NOK 153,9 billion in September 2019. Of this, consumer loans decreased 5,7% from NOK 76,2 to NOK 72,1 billion, while credit card debt was reduced from NOK 57,5 to NOK 44,2 billion (Gjeldsregisteret, n.d.-a). The reduction is caused by the decrease of non-interest-bearing debt, which essentially includes non-due credit card debt. Interest-bearing debt had an increase, caused by consumer loans and overdue credit card debt (Gjeldsregisteret, n.d.-b).

### 2.1.2 Default on loans

A default can be defined as "failure to make on-time payments on an amount owed" (Sumup, n.d.), and can occur for both secured and unsecured loans. If a borrower defaults on a secured loan, then the assets, which are the security of the borrower, will be at risk. In other words, the lender has a legal claim to the specific asset pledged as security for the loan. If the default occurs on unsecured debt, this will affect the borrower's credit rating and limit the borrower's ability to take out loans in the future. The lender still has a legal claim to this money. Usually, if they have yet to receive the money after six months, the lender will write off the unsecured debt as lost. Afterward, they sell the debt to a debt collection company, which will try to collect the money from the borrower (J. Chen, 2023b).

## 2.2 Market of unsecured loans in Norway

### 2.2.1 Inflation

Due to inflation being part of what makes up the costs of loans, it is essential to see how inflation has been over the years (Visma, n.d.). This is to understand our dataset, although it does not cover more than 30 months of loan requests. A lot changes in just one year, and trends in the market can change in a relatively short time.


Figure 1: Inflation in Norway between 2018 and 2022/2023 (Trading-economics, 2023)

As a result, the risk in the credit market is higher now than it was before. Therefore, financial institutions seek compensation for this additional risk and an adjustment for the depreciation of the loan value. The Norwegian central bank adjust the key interest to reach their inflation target. When inflation is high, they tighten the financial institution's access to loans by raising the key interest. The central bank says the financial institution's credit access is sufficient, even after the rise of the key interest rate. Still, it predicts a fall in the public's access to credit, resulting in lower growth in the credit market, as shown below (Norgesbank, 2022).

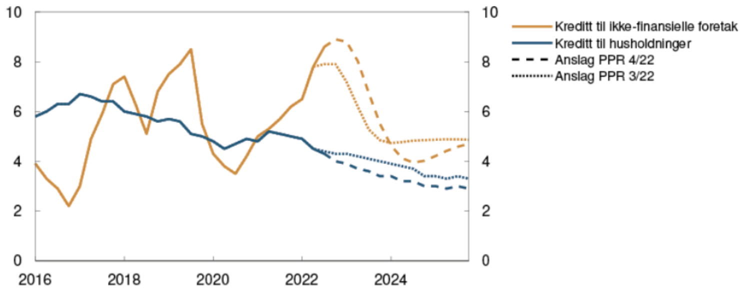


Figure 2: Prediction for the growth in the credit market for non-financial institutions (orange) and households (blue) (Norgesbank, 2022)

### 2.2.2 Interest rates

The financial institutions live off the difference between the interest rate and the interest they pay to borrow from the central bank. This includes interest on deposits, interest on market financing, and the return on equity that owners require (FinansNorge, n.d.). Several external factors affect the interest rate. Demand and supply are two, but perhaps primarily the key interest rate adopted by the Norwegian central bank (Lånemegleren, 2022).

In Norway, each bank sets its own interest rates. Within a not too large market, there is intense competition. In connection with the setting of the rates, it is not allowed for the bank to send out signals to the public about what they are supposed to do or not. This could not only affect other banks but also weaken the competition. The banks are also required by law to have a notice period from when the interest rate is decided to be changed until it is changed. This varies depending on the type of interest, e.g., whether it is a loan or a deposit. For loans, this is six weeks (FinansNorge, n.d.).

The interest rates of loans in Norway are regulated against the key interest rate. To calm the inflation, the central bank has already raised the key interest several times in 2022 and once in 2023, but it is likely to be raised even more (Norgesbank, 2023a).

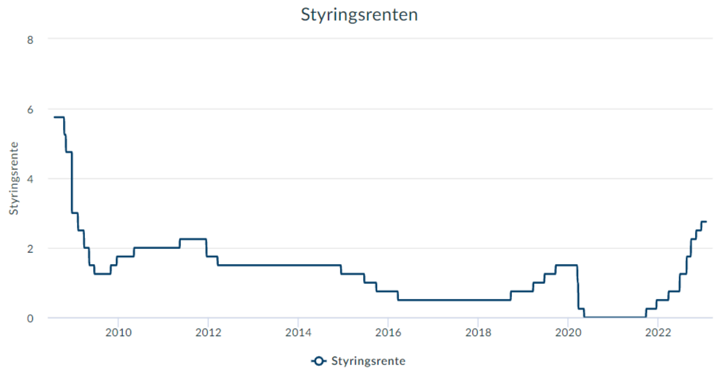


Figure 3: Key interest rate (Norgesbank, 2023b)

Figure 4 provides a history of the lending and deposit rates in Norway since 1900, which has a high correlation with the key interest rate. This overview also gives insights into other times of financial hardship.

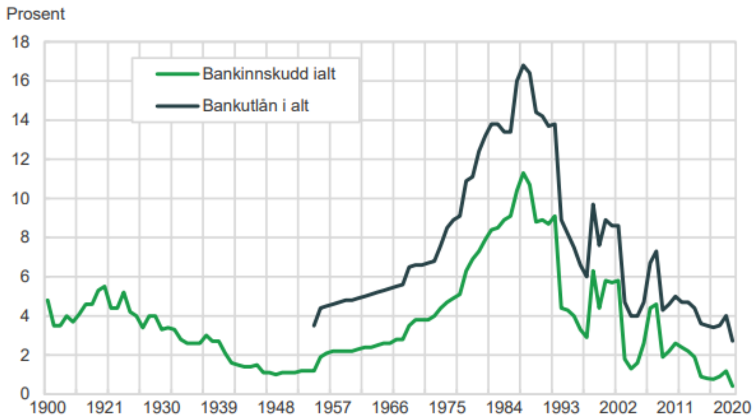


Figure 4: Deposit (green) and lending (blue) rates since 1900 in Norway (Wettre, 2021)

### 2.2.3 Expected loss

The implementation of the new write-off model in IFRS 9 set stricter requirements on how banks should deal with expected losses. In contrast to the previous model, IFRS 9 requires that the banks consider expected losses. Credit loss must be recognized over 12 months or for the rest of the term, depending on the risk. If there have been no changes in the risk, it is 12 months. IFRS 9 provides some discretionary assessments the bank makes, where the expected loss is considered the probability-weighted present value of all absent cash flows. Therefore, the banks must forecast expected losses based on past events, as well as the current situation, and use some time to predict what could happen in the future, both in the short and long term

(KPMG, 2017).

## 2.3 Literature review

For banks and other financial institutions, having an overall picture of the potential borrower before issuing loans is essential. First and foremost, not to lose money themselves, but also for the customer's financial situation. Therefore, assessing the credit risk is alpha and omega for the lender. There are plenty of aids for this today. But there has yet to be, and probably never will be, an unequivocal answer to which methods are most suitable. The use and result will depend on the type of loan, which data is available, which variables are available, and so on. In this thesis, we focus on neural networks. We look closer at how this classification method is suitable against other more traditional methods, such as logistic regression. To get an overall picture, this Section reviews literature in the field, where other classification methods also are mentioned.

Predicting has been used for a long time, but in recent years and decades, machine learning and neural networks within economics and other areas have gained momentum. As an alternative to the traditional classification techniques, this has given financial institutions new opportunities. Building better and stronger models has, among other things, helped to achieve greater accuracy when predicting default. Even though neural networks often predict better (Albanesi & Vamossy, 2019), there is yet to be a reason to consider that it always does.

### 2.3.1 Predicting default on secured loans

To forecast any default, neural networks and other traditional classification methods are frequently used. In their study, Næss, Wahlstrøm, Helland, and Kjærland (2017) found that neural networks are the most suitable machine learning technique when predicting bankruptcy in Norwegian companies. They compared it with a generalized additive model, which gave a better result, but there was no significant difference. Bayraci and Susuz (2019) did much of the same when studying loan clients in a middle-sized Turkish commercial bank. They used many of the traditional classification techniques, including logistic regression, decision tree, naive bayes, support vector machine, and a deep neural network-based classification model. They discovered that the effectiveness of deep learning classification models increases as the complexity of the dataset increases. Results are shown in Table 1.

As mentioned, there are many different types of loans. Credit cards, mortgages, and consumer loans are just a few. These could be both secured and unsecured loans. Tsai, Lin, Cheng, and Lin (2009) used a dataset from a financial institution in Taiwan. By integrating the borrower's personal information and attitude towards money, they constructed four models: discriminant analysis, logistic regression, neural network, and DEA-discriminant analysis. They found DEA-discriminant analysis and neural networks had a better predicting capability. The accuracy rate of DEA-discriminant analysis was 100%, while the logistic regression predicted 92,75% correct and the neural network 98,55%. What is very interesting about this study is that they adopted historical information of borrowers, unlike previous studies, resulting in better predictions. But the high accuracy is likely part of a bigger problem with the dataset used. The other methods and results are shown in Table 1.

Baesens et al. (2005) studied a dataset from the UK, where they looked more closely at how neural networks work, with logistic regression and the Cox model as a basis for comparison. The neural network outperformed the logistic regression in three out of four attempts. First, they predicted default in the first 12 months of the dataset with eight hidden neurons. Logistic regression achieved an accuracy of 95,20%, Cox 95,20%, and neural network 95,28%. Secondly, they did the same but now for 12-24 months, resulting in an accuracy of 95,36% for logistic regression, Cox 95,59%, and the neural network 95,59%. Next, they changed from 8 to 16 hidden neurons and predicted default in the first 12 months on an oversampled dataset. It gave logistic regression an accuracy of 79,20%, Cox 79,00%, and the neural network 78,76%. Finally, they did the same for 12-24 months, which resulted in an accuracy of 78,24% for logistic regression, the Cox model 77,5%, and neural network 78,58%. This study emphasizes that neural networks appear to perform as well as or even better than the more traditional classification models like logistic regression. However, this is certainly not always true.

Table 1: Results predicting default on secured loans

| Authors | Type | Methods | Results | Period |
|---|---|---|---|---|
| Bayraci and Susuz (2019) | Credit (Loan Application Data) | DNN | 85,69% | 2 years and 2 months |
| | | Logistic Regression | 78,01% | |
| | | J48 Decision Tree | 82,34% | |
| | | Support Vector Machine | 77,93% | |
| | | Naive Bayes | 75,25% | |
| Tsai, Lin, Cheng and Lin (2009) | Consumer loan | Discriminant Analysis | 76,81% | 15 days |
| Odegua (2020) | Bank loan | Extreme Gradient Boosting | 79,00% | Unknown |
| Wang (2022) | Consumer loan | Logistic Regression | 86,11% | Unknown |
| Kvamme, Sellereite, Aas og Sjursen (2018) | Mortgage | CNN | 95,30% | 1 year |
| | | Logistic Regression | 94,30% | |
| | | Multilayer Perceptron | 92,70% | |
| | | Random Forest | 95,70% | |

### 2.3.2 Predict default on unsecured loans

Naik (2021) predicted credit risk for unsecured lending. He ran the research dataset through seven machine learning models, including a logistic regression classifier (all results in Table 2). The disparity in accuracy was significant. It was at 55,79% for logistic regression, which was the lowest out of the seven models. Light gradient boosting machine (LGBM) had the best accuracy with 95,53%. The ROC-plot in the study showed that LGBM outperformed the other models with an AUC of 0,99, while it was at 0,61 using logistic regression. Using multivariate logistic regression, Li et al. (2019) predicted default risk on unsecured consumer loans in online lending. The prediction result showed the same as Naik (2021). The accuracy was low, only 57,83% using logistic regression.

Malhotra and Malhotra (2003) used multiple discriminant analyses and neural networks in identifying potential problem loans. Using consumer loans, they found that neural networks performed the best. They used several small and large sample tests, where the neural networks overall performed respectively 66,73% and 71,98% while the multiple discriminant analysis performed 61,14% and 69,32%.

In his work, Duan (2019) used a multilayer perceptron deep neural network with three hidden layers trained by the backpropagation algorithm. He divided his loan data into three classes, safe, risky, and bad loans, where he first predicted all of them. Comparing his deep neural network model (DNN) with ten different methods, including logistic regression, DNN was the method predicting the very best with an accuracy of 93,2%. The logistic regression done by Feis et al. (2016) predicted the cut with an accuracy of 77,1%. Duan (2019) made the prediction again, combining the "non-safe loans", including the risky and bad loans. This time, the DNN model predicted better with an accuracy of 99,7%, while Feis et al. (2016) logistic regression had an accuracy of 88%.

Byanjankar et al. (2015) also predicted credit risk in peer-to-peer lending with a neural network approach. The results showed that the neural network performed more accurately than logistic regression when predicting default loans (74,38% vs. 61,03%). On the other hand, logistic regression performed more accurately than the neural network when predicting non-default loans (65,34% vs. 62,7%). However, they concluded that neural networks showed promising results in classifying credit applications in peer-to-peer lending. They also pointed out that neural networks enable lenders to make intelligent decisions when selecting loan applications.

Table 2: Results predicting default on unsecured loans

| **Authors** | **Type** | **Methods** | **Results** | **Period** |
|---|---|---|---|---|
| Naik (2021) | Unsecured lending | Logistic Regression | 55,79% | Unknown |
| | | Support Vector Machine | 69,47% | |
| | | K Nearest Neighbors | 76,77% | |
| | | Decision Tree | 81,45% | |
| | | Random Forest | 85,87% | |
| | | Extreme Gradient Boosting | 91,96% | |
| | | Light Gradient Boosting | 95,53% | |
| Feis, Mehta, Morris, Solitario and Graaf (2016) | P2P loans | Logistic Regression | 88% | 5 years |
| | | Linear Discriminant Analysis | 92% | |
| | | Support Vector Machines | 89% | |
| | | AdaBoost Classifier | 91,70% | |
| Zhu, Qiu, Ergu, Ying and Liu (2019) | P2P loans | Random Forest | 98,00% | 3 months |
| | | Decision Tree | 95,00% | |
| | | Support Vector Machines | 75,00% | |
| | | Logistic Regression | 73,00% | |

## 2.4 Who defaults on loans?

As we seek to find the defaulters of unsecured loans, getting a more accurate picture of them is essential. Our research shows this is not often the case in other studies. A report by Høie and SSB (2021), pointed out that unsecured debt increased by NOK 19,6 billion throughout the 2000s up to 2020. Out of approximately 5,5 million residents, about 2,057 million aged 17 years or more held on unsecured debt at the end of 2021 (Statistisk sentralbyrå, 2023). In Norway, there was NOK 65 billion in default in 2020. Out of this, NOK 9,5 billion was in default of debt in the age group 50-54 years. In the age group 18-25 years, the same numbers were at NOK 760 million (Kreditorforeningen, n.d.).

In their study of unsecured, short-term lending platforms, Chen et al. (2020) discovered that higher educated people had a lower default rate, and that older people defaulted more frequently than younger people, which refers to the same as Kreditorforeningen's (n.d.) report. Dynarski (1994) studied another type of unsecured loan, student loans, using data from the National Postsecondary Student Aid Study in the US. Dynarski divided his findings into three key results. First, he discovered that the ones from low-income families and members of minority groups, high school dropouts, and borrowers who attend private schools and two-year colleges had a higher likelihood of missing loan installments. Subsequently, the borrower's income after leaving school remained a clear factor for default, even after adjusting for several background factors. Lastly, if the borrower did not complete their program at postsecondary school, the probability of defaulting on the student loan increased significantly.

Silva et al. (2020) used a logistic regression model to evaluate the default risk of consumer loans in the Portuguese market. Their model gave an accuracy of 89,79% in predicting default correctly. More interesting, they found loan spread, loan term, and the customer's age as variables increasing the likelihood of default. If the customer received their salary in the same bank as the loan, the probability of default decreased. They also found the same as Dynarski (1994) did, those with the lowest salary income had a higher chance to default. On the other hand, they found that the more credit cards the customer owns, the risk of default decreased.

# 3 Data

It is essential to include the correct variables when predicting credit default. Those might be hard to find. Even though tools are available, one change might turn the results upside-down. In this Chapter, we go through the dataset, including the variables, and clarify the decisions made.

## 3.1 Data description

Our dataset contains 93.039 accepted applications for unsecured loans from Spare-Bank 1. This means the data have already gone through their data system and has been defined as loan applications that should be repayable. The dataset has been split into training- (70%), test- (15%), and validation set (15%), which provides both a large training set and a decent size of the test and validation set. Meng et al. (2020) argued there are no standard splitting strategies but that selecting a splitting strategy might strongly impact the results. The following split applies to the neural network sets, which will differ some from the logistic regression sets. This is primarily due to the models being coded in different languages. The splits in both languages are seeded with the same seed, yet some observations are too extreme for the logistic regressions and were removed. This means there are differences in the sizes of test, train, and validation, and which observations are present in the sets. The test set for our neural network consists of 503 defaults and 13.453 non-defaults of 13.956 observations.

## 3.2 Imbalanced data

The target variable consists of 3.561 defaults and 89.478 non-defaults. This means that of our 93.039 observations, we have a default rate of approximately 3,83%. Due to the nature of our data, it is imbalanced. This imbalance can cause problems when it comes to models dealing with the minority class, and we can expect the model to have high accuracy by putting every prediction to the majority class. This also goes for other measures as True Positive Rate (TPR) and False Positive Rate (FPR), and will impact the model's area under the curve (AUC). There are, however, methods of dealing with imbalanced data.

Since class imbalance impacts the performance of machine learning techniques, it is essential to address this issue by using oversampling/undersampling or adding weights to our loss function. Oversampling is a preprocessing technique that basically “creates” more observations of the minority class. This is done to balance the dataset, and it takes the already existing observations and adds noise to them. By doing this enough times, the different techniques seek to balance the set by having a 50/50 share of the classes. There are several different oversampling techniques, yet the essence remains the same. This, however, makes artificial data in the set, which takes away the principle of working with real world data (Gosain & Sardana, 2017).

Undersampling is another preprocessing technique. Instead of correcting the balance by creating new observations, the balance is achieved by "subsetting" the majority class. This means we remove or ignore a part of the majority class, balancing the dataset. However, this is problematic because important data might be removed. There are also many types and varieties here, but the essence is still the same (Liu et al., 2009).

We wish to keep the aspect of working with real world data. Therefore, oversampling the minority class is not a viable option. There is also a wish to keep as much information as possible for the neural network, meaning the data will not be undersampled either. The final dataset consists of 39 independent variables and one dependent variable. Our dependent variable, or target, "inkasso" is false (0) if the debtor has not defaulted on the loan within one year. Even if the debtor were to default after the 12-month period, "inkasso" would remain false. The neural network’s target will be one-hot-encoded due to syntax issues. The same dataset that goes through the network will be used in feature selection using lasso, except the one-hot encoded target.

## 3.3 Vital information

As mentioned, the dataset contains loans that have been granted, driven through a model, and reviewed. Given this information, the minority class is of interest. The dataset is already evaluated as non-defaults, making models that classify the entire set as the majority class useless/non-working. The data consists of:

i) Information on the loans
ii) Demographic information

The final data does not contain macroeconomic variables nor a time variable since the time variable’s effects are already working as a fixed effect on the independent variables provided in the data. The data contains three types of unsecured loans: Refinance loans, credit card loans, and consumer loans.

# 4 Method

In this Chapter, we present our research methods. This includes determining the network's parameters, topology, learning algorithms, and other important factors for the network's performance. Basic theory and history are also covered.

## 4.1 Machine learning and its importance

Machine learning is a type of artificial intelligence. It allows software applications to become more accurate in predictions without being programmed, but from experience. To predict new output values, the algorithm uses historical data as input. Using machine learning, the model can predict many years in the future, not only for the upcoming week, month, or year. However, it does not consider unanticipated events like those that have occurred in the previous years (Burns, n.d.).

There is no discussion of the importance of machine learning and what it can do today and into the future. Humans seem to be the limit to what it can achieve as it can handle larger problems and technical questions, which will be introduced (DataRobot, 2020). Generally, for many businesses, machine learning has emerged to be a key competitor differentiator and a field many companies are investing a lot of money (Burns, n.d.).

## 4.2 History of neural networks

In the early 1940s, neurophysiologist Warren McCulloch and mathematician Walter Pitts (1943) introduced the idea of neural networks. It all started with a model of how brain neurons work, known as connectionism (Jaspreet, 2016). The main objective was to use interconnected circuits to stimulate intelligent behavior (McCulloch & Pitts, 1943).

In the 1950s and 1960s, neural networks became increasingly advanced. Frank Rosenblatt created the first trainable neural network, the Perceptron, in 1957. The Perceptron's structure was quite like the modern neural network. The layer between the input and output layers with adjustable weights and thresholds made the difference (Hardesty, 2017).

In the late 1960s, the neural network hype was at its all-time high. Despite the hype and positivity surrounding it, neural networks lost almost everything in just a short time. Artificial intelligence was next up. There are many reasons why this happened. Primarily the need for more data. There were also limitations on computer technology. It was not advanced enough to handle the data that was available. In a 1969 demonstration, Minsky and Papert highlighted the drawbacks of simple Perceptron. They proved theoretically that the simple Perceptron model was computationally ineffective. As a result of this and other failed demonstrations, neural network research experienced a significant loss of financial support, which kick-started neural networks' demise (Bharath, 2020).

In 1986, neural networks returned to popularity with backpropagation (Kurenkov, 2020). Rumelhart et al. (1986) investigated how to teach neural networks to learn to represent data hierarchically by using the backpropagation algorithm. Their successful research was an important part of neural networks flourishment and paved the way for deep learning and artificial intelligence development.

Today, neural networks are used in everyday activities. It has the capability to predict the locations of natural disasters (DataRobot, 2020). Social media uses neural networks to study the behavior of the users. What we do are tracked and used to recommend other applications or advertise something we have searched for or are interested in. It is used in stock market predictions, risk assessments, and healthcare. For example, drug discovery. Generative neural networks have made it easier for healthcare personnel to find the correct combination of drugs, in addition to drug discovery. This means the time from when a medicine is needed until it is discovered is shortened drastically (Kaushik, 2021). More specifically, it is one helpful way healthcare personnel can classify sickness, for example, cancer. It is also an important tool when solving a crime or for people searching for their families because of the improvements in gene prediction (Krogh, 2008).

## 4.3 Practical use of neural networks

There are some disadvantages using neural networks. One is the "black box" problem. This means that understanding what the model has learned and what happens inside the network is hard, if not impossible. This becomes increasingly hard the more complex and deep the network is (Han et al., 2012). No one knows how complex neural networks reach their results, the only thing that can be observed is the input and output. Neural networks are also criticized for their long training time in search of the networks' optimal architecture (Han et al., 2012; Pacelli & Azzollini, 2011). The networks might be working too hard to find the best match for the training data. Overfitting is an issue that might arise as a result of this. Dietterich

(1995) points out that we could end up fitting the noise when memorizing various peculiarities of the training data rather than discovering a broad predictive rule.

On the other hand, there are many advantages in addition to the above regarding machine learning and neural networks. Pacelli and Azzollini (2011) pointed out that because of the ability to simulate intricate relationships between dependent and independent variables, neural networks are an excellent tool for credit scoring. Therefore, banks and other financial institutions widely use machine learning techniques. Another example of where it is used is during the development of self-driving cars and the optimization of routes, as we can see today in Google Maps. Customer service is also an important target group for machine learning. Be it virtual assistance or measuring customer intent. Moving on to farming and other subject areas that need a certain amount of labor, the development of robots has come a long way, and much of this development could not have been done had it not been for machine learning. Same with the utilization of the land, e.g., where is it best to plant, and where is it not suitable to plant. Machine learning is helping to make improvements more accessible for everyday activities (Microsoft, n.d.).

## 4.4 Multilayer feed-forward neural network

Multilayer feed-forward is a type of neural network. It consists of three parts: An input layer, an output layer, and one or more hidden layers. The input layer is fed information simultaneously with data and passes its information into the next network layer. This layer can again connect to another hidden layer or directly to the output. This depends on the network's depth. A network with one hidden layer is a two-layer network. In these models, each output takes a weighted input of the previous layer's output. This sum of inputs will then be subjected to an activation function. This makes the final output a non-linear regression from a statistical point of view. Thus, these models should be able to closely approximate any function, given enough training samples. It is called a feed-forward because the weights only work in a forward direction (Han et al., 2012). The illustration provided in Figure 5 may give clarity to the topological concept.

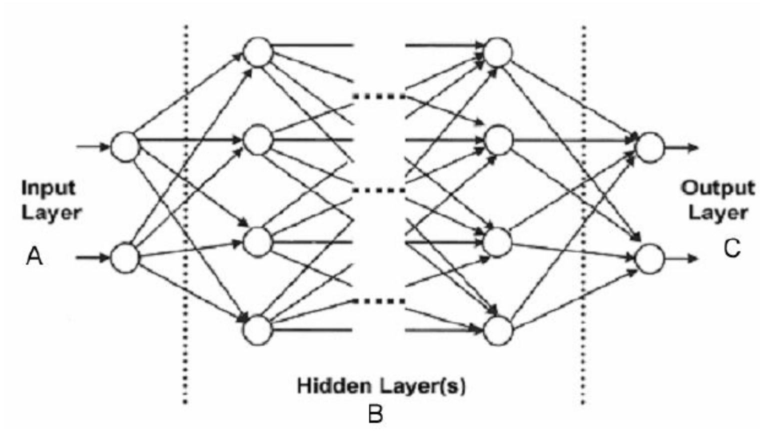


Figure 5: The concept of a feed-forward neural network (Hasan et al., 2011)

### 4.4.1 Network topology

Before training the model, the network topology needs to be set. There are, however, no clear rules for what makes the best performing network. Making the process of designing the network's topology, a process of trial and error. This means repeating the process of setting hyperparameters, such as the initial value of the weights or changing the topology entirely (Han et al., 2012).

### 4.4.2 Backpropagation

The expansion of the Widrow-Hoff learning rule to multiple-layer networks and non-linear differentiable transfer functions forms the basis of a backpropagation model with adaptive learning (Malhotra & Malhotra, 2003). In a neural network, it is one of the most fundamental building blocks. It was introduced in the 1960s, but it took almost three decades before it got popularized by Rumelhart, Hinton, and Williams in 1989 in their paper "Learning representations by back-propagating errors" (Kostadinov, 2019). The algorithm works by altering the weights inside the network. Since the training set contains known class labels, the network will seek to iterate through the observations and predict their known target value. For each iteration, the weights between the nodes are modified to minimize the loss function between the prediction and the true value. These adjustments work backward in the network, hence the name backpropagation. Each weight also has a bias associated with it, which also gets corrected by the learning algorithm. This technique, where the weights and biases are updated after every new observation, is called case updating. This differs from batch updating, where one whole iteration of the training set is an epoch. A third option, mini-batch updating, combines the two mentioned and will be further explained in Section 4.6. Due to the non-linearity in the loss function after each iteration, there is a possibility that the model converges towards a local minimum instead of a global one. However, some techniques increase the likelihood of convergence to a global minimum (Han et al., 2012).

### 4.4.3 Bias, weights, and learning rate

To better explain the adjustment processes of weights and biases, Figure 6 represents a hypothetical multilayer feed-forward neural network.

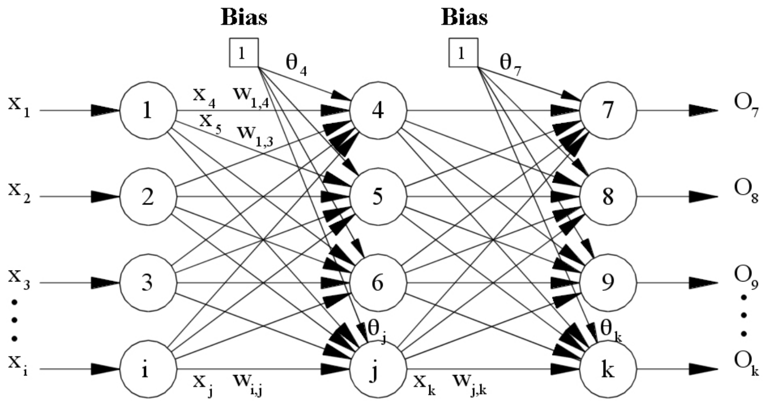


Figure 6: The concept of a backpropagation feed-forward neural network with bias and weights (UNSW Sydney, n.d)

Figure 6, shows that each node has a weight connected to its input (except for the input layer) and a bias associated with the same node. As previously explained, the learning algorithm propagates the error backward to adjust these weights and biases. For K in the output layer, the error is computed as:

$$(err_k) = O_k(1 - O_k)(T_k - O_k) \tag{1}$$

where $T_k$ is the known label from our training data. As this is propagated "backward" in the network, the error for the hidden layer's node j has another formula. $err_k$ means the error for node K, which in this case is the error for the output. For notations, see Figure 6. Every indexed O stand for the output of that index. This is, to sum up the weighted sum of errors from nodes connected to node j:

$$(err_j) = O_j(1 - O_j)\sum_k err_k W_{j,k} \tag{2}$$

$W_{j,k}$ is the weight between node j and k, as also shown in Figure 6. The same logic follows, and $err_j$ is the error in the node j's output. To reflect the error in the network, the network's biases and weights are updated to minimize its error. This provides an additional two equations. To understand the process, the network goes through to adjust its weights. These equations describe how the change depends on the learning rate $(l)$ and the output error of a previous node. In our example, the equations for updating the weights for $W_{i,j}$ is

$$\Delta W_{i,j} = (l) err_j O_i \tag{3}$$

$$W_{i,j} = W_{i,j} + \Delta W_{i,j} \tag{4}$$

$\Delta W_{i,j}$ represents the change in $W_{i,j}$. This effect will work its way through the weights of the network. However, the bias is also subject to change in this learning algorithm. In our example, the biases for nodes k and j are updated similarly. The indexed Os means that it is an output from its indexed node.

$$\Delta \Theta_j = (l) err_j O_i \tag{5}$$

$$\Theta_j = \Theta_j + \Delta \Theta_j \tag{6}$$

$\Delta \Theta_j$ represents the change in bias for node j, and $\Theta_j$ is the bias of node j. For the indexing of the nodes, see Figure 6. The learning rate, which repeats through the correction of both weights and biases, are hyperparameters set to a number between 0 and 1. This controls how fast the model learns. This means that a low learning rate makes the time consumption of teaching the model higher than if the rate was high. However, this needs to be balanced out since a too high learning rate may cause oscillations between non-optimal solutions. The standard for this hyperparameter is 1/t, where t is the number of observations that have gone through the network thus far (Han et al., 2012), but it varies depending on the optimizer.

## 4.5 Activation functions

Every node in the feed-forward neural network has an attached activation function. This is not a necessity, but if no activation functions are present in the network, the output will simply be a linear function. This makes the model overly complex for its output, which would be relatively limited and lack the ability to mimic/extract functions effectively. Since very few relationships in the real world are linear, we seek to make a model with one or more degrees of curvature (non-linear functions). The non-linearity is achieved by taking the node's net input, applying an activation function to the node and sending it down the network. This means that the network can consist of several activation functions, all giving different shapes to the input (Sharma et al., 2017).

#### 4.5.1 Leaky ReLU

To get a better understanding of what these activation functions look like, a couple will be presented. ReLU is a popular and widely used activation function. The activation function is, however, not optimal for every problem. This is why there have been developed different variations of ReLU, such as Leaky ReLU. What makes Leaky ReLU different from ReLU is the non-zero slope. In a ReLU, this slope would be zero and flat instead of negative. This means that the slope of the Leaky ReLU can have negative values instead of only non-negative values (Dubey & Jain, 2019). This is shown in Figure 7.

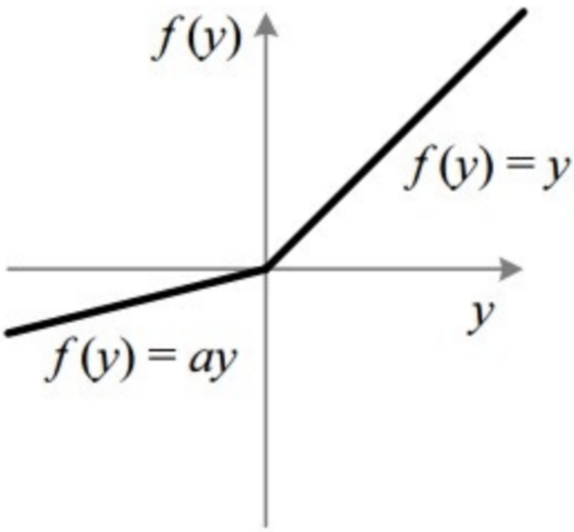


Figure 7: Leaky ReLU (Paperswithcode, n.d.)

#### 4.5.2 Tanh

Hyperbolic tangent function (Tanh) is another type of activation function, which has an "s" shape. This is due to the function being a logistic activation function. This function is a reshaped Sigmoid function, which is further explained in Section

4.9. The shape of the functions are similar, yet Tanh has a larger range and can be considered as a "biased" Sigmoid (Lau & Hann Lim, 2018). Figure 8 illustrates this and consists of both a Tanh and a Sigmoid function.

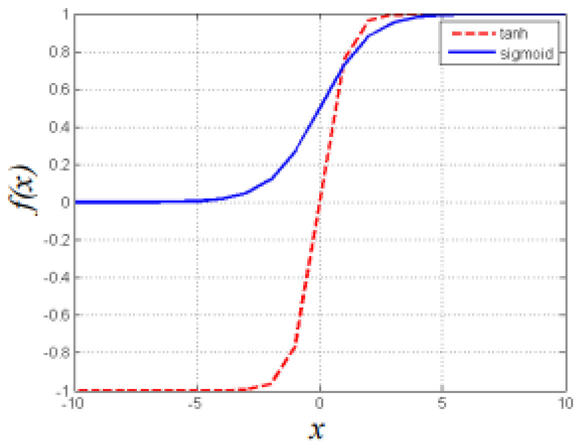


Figure 8: Tanh and Sigmoid comparison (Lau & Hann Lim, 2018)

## 4.6 Optimization algorithms: Gradient decent

Gradient descent algorithms are the most used type of algorithms to optimize a neural network. These kinds of algorithms are used to optimize "black boxes". The algorithms minimize an objective function, which we will call E(x). The algorithms do the same thing but in different ways, to update E(x)'s parameters in the opposite direction of their gradient. The algorithms seek the "lowest" point on the non-linear objective function. If the objective function is a highly non-convex error function, the function will have several "valleys", or local minimum. Getting stuck in a local minimum is, however, a non-optimal solution. The distance we move for each update in the network is determined by the learning rate. This means that a low learning rate, as previously mentioned, tends to get stuck in the closest minimum and sheds light on why the objective function fluctuates a lot more with a higher learning rate (Ruder, 2016).

There are three gradient descent variants:

1. Batch gradient descent
   The batch gradient descent (BGD) iterates through the entire dataset to calculate E(x)'s gradient and perform one single update. This variant is very slow and uses a lot of memory. This gradient descent is not guaranteed to converge to a global minimum unless the objective function is convex (Ruder, 2016).

2. Stochastic gradient descent
   Stochastic gradient descent (SGD) is earlier referred to as "case updating". The network updates itself after each observation, and by starting at high learning rates, which decrease for each update, allows the model to jump into other "valleys". This means that it will converge to a minimum, but it is not guaranteed that it is the global minimum. It is, however, guaranteed to land in a global minimum when the objective function is convex and has the possibility to find the global minimum for non-convex as well (Ruder, 2016).

3. Mini-batch gradient descent
   This variant is a combination of BGD and SGD and performs updates after every set amount of observation. The term SGD is often employed for mini-batch gradient descent as well and is not guaranteed to find the global minimum of E(x) (when it is non-convex) (Ruder, 2016).

All three variants have the same main problem if we do not take computing power into account. Which is that none of the variants has a guarantee for finding the global minimum of E(x), which is why we have optimization algorithms for gradient descent to better the chances of hitting a global minimum. There are a large variety of algorithms within this area. The following three are given examples of what optimization algorithms do (Ruder, 2016).

### 4.6.1 Adagrad

Adagrad has an adaptive learning rate and performs small updates for frequently updated parameters and larger updates for less frequent parameters. This makes the algorithm well-suited for sparse datasets. Adagrad uses a different learning rate for all parameters. The main weakness of the algorithm is that it changes its learning rate automatically and only in a decreasing direction, making the learning rate infinitesimally small on larger datasets (Ruder, 2016).

### 4.6.2 Adadelta

Adadelta is an extension of Adagrad and seeks to reduce the effect of the decreasing learning rate. Another difference is that with this algorithm, there is no need to set a default learning rate, as it has been eliminated from the update rule, and the rule uses an RMS of the parameter as an update rule instead (Ruder, 2016).

### 4.6.3 Adam

Adaptive Movement Estimation (Adam) is considered favorable to other adaptive learning-method algorithms. This algorithm also has proposed default values for its hyperparameters, making it easier to work with. Adam is a kind of combination between Adadelta and another algorithm called Momentum. This algorithm, in contrast to Adadelta, uses an adaptive learning rate for each parameter (Ruder, 2016).

## 4.7 Dropout as a method to avoid overfitting

One problem in machine learning, including neural networks, is overfitting. However, there are methods to deal with this problem, one is dropout. Dropout reduces overfitting, and that way improves the neural network. It is possible to see from Figure 9, that the technique momentarily drops units randomly, both hidden and visible, during training of the network. Therefore, units are prevented from coadapting too much. As the testing now takes place, the effect is easier approximated. It is accomplished by averaging the predictions made over the remaining thin network. This way, we are only left with one thin network (Srivastava et al., 2014).

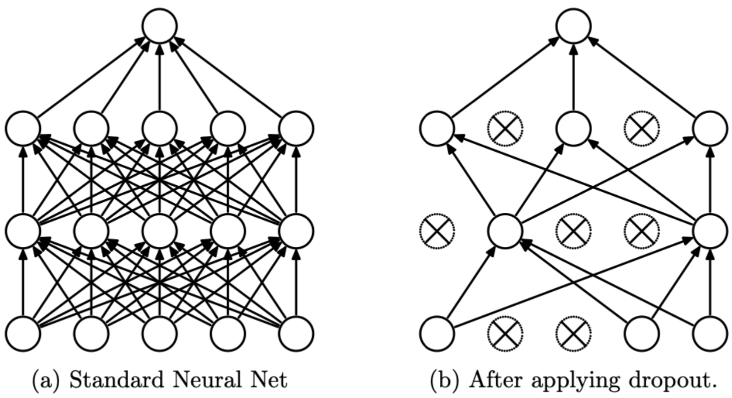


Figure 9: Dropout - one way to avoid overfitting

(Srivastava et al., 2014)

## 4.8 SHAP

SHapley Additive exPlanations (SHAP), introduced by Lundberg and Lee (2017), is an idea derived from game theory. It is one way to clarify the importance and the necessity of the different variables in a dataset. SHAP values give a better picture of which variables are more crucial when predicting, e.g., credit default (Ozturkal & Wahlstrøm, 2022). There are many ways to measure variables using SHAP. In Section 5.3, we will introduce three of them. They show how the different variables impact the results, both positively and negatively (Wahlstrøm, 2023). By that, we will be able to understand the importance of each one of them.

## 4.9 Logistic regression

To see how our neural network performs, it will be put up against a logistic regression model for comparison. Logistic regression is a popular method for predicting default. Frequently used by financial institutions themselves and those who study the relevant topics. At the same time, it is a tool which is considered simple to comprehend. Logistic regression is estimated by Maximum Likelihood estimation. The method is usually used when $Y$ is a dummy variable. There are several ways to interpret logistic regression, either by the logistic regression coefficients, odds ratio, or predicted probabilities (Hammervold, 2022). Logistic regression forecasts the probability that a binary outcome will occur. The output is in a Sigmoid shape, to get an "s" shape in the model's output. The Sigmoid function converts any value to a number between 0 and 1 (Bhor, 2022). The formula of the Sigmoid function is given by:

$$f(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{7}$$

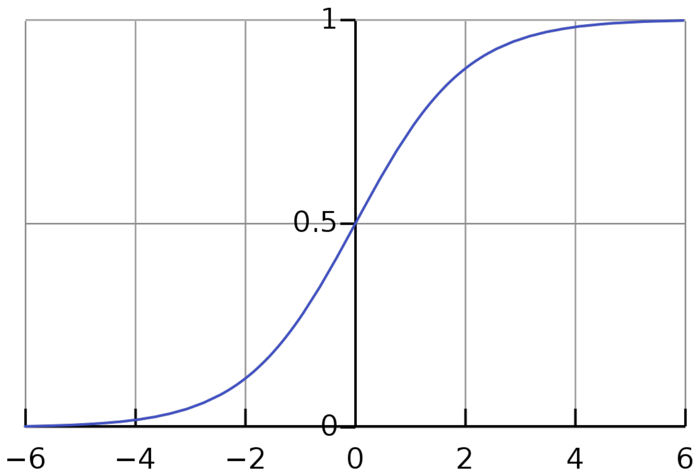


Figure 10: Sigmoid function (Chakraborty et al., 2019)

To get the classification as 1 or 0, there is a threshold. This is vital to the model's performance and will also impact the model's predictions (Handoyo et al., 2021). This is further discussed in Section 6.4.3.

## 4.10 Lasso regression

The popular model selection and shrinking estimation method introduced by Tibshirani (1996), called lasso regression, was originally proposed for linear regression models (Meier et al., 2008). It is often used to find the variables and corresponding regression coefficients that result in a model with the least prediction error. It also has the advantage of reducing overfitting without limiting parts of the dataset to internal validation only, as is done in a neural network. Practically, lasso regression aims to constrain the model's complexity by restricting the model parameters. The slope value decreases as a result. In other words, reducing the absolute value of the regression coefficients to below a fixed value, λ . After this shrinking, values of 0 are removed from the model. The most popular way of finding the best λ is by using k-fold cross-validation. The dataset is randomly divided into k sub-samples of the same size. k-1 sub-samples are used to develop the prediction model, while what remains is used for validation. The procedure executes k times. The result is clear as the k separate validation results are combined for a whole range of λ -values, and the choice of λ which ultimately helps to determine the final model (Ranstam & Cook, 2018).

## 4.11 Model evaluation

There are many ways to evaluate a neural network, depending on the suitability of the dataset. In this Section, we go through the ones we use in this thesis. There will also be a practical evaluation of the model, in the form of risk-adjusted returns. This evaluation originate from the Sharpe ratio given by the following formula:

$$Sharperatio = \frac{R_p - R_f}{\sigma_p} \tag{8}$$

Since the Sharpe ratio also accounts for the concept of uncertainty (Zvi Bodie, 2021), it will be found in Chapter 6.

### 4.11.1 Receiver operator characteristic (ROC) curve

The Receiver operator characteristic (ROC) curve, shown in Figure 11, is a probability curve used as an evaluation matrix for binary classification problems (Bhandari, 2023). The first step in producing a ROC curve is to put the sensitivity- and specificity values for different values of continuous test measures in tabulates. Mainly, this results in a list of different test values together with the test's sensitivity and specificity for each value. This again results in a graphic ROC curve with TPR (sensitivity) on the y-axis and FPR (specificity) on the x-axis for the various values (Hoo et al., 2017). This distinguishes the "signal" from the "noise" (Bhandari, 2023). The main idea with the ROC curve is to make the area under the curve as large as possible so that it approaches 1.

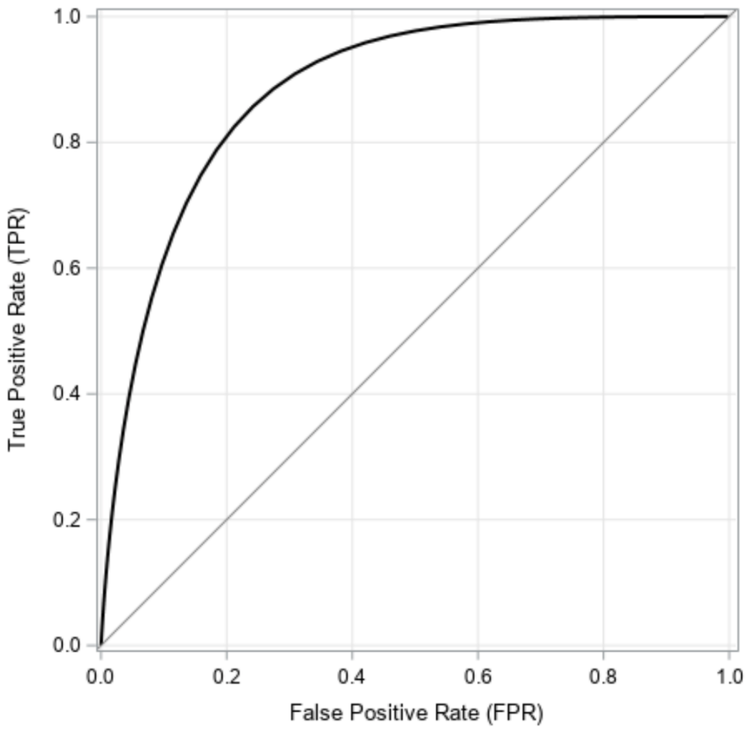


Figure 11: ROC curve (Wicklin, 2020)

As a measure of a classifier's performance, Bradley (1997) discussed the use of AUC. It is a method often used if there is a problem with imbalance in the dataset, which is the case in this thesis. It is used as a summary of the ROC curve. AUC measures how well a model distinguishes positive and negative observations and ranks them correctly (Rosset, 2004). A good AUC value is debatable, but in general, an AUC of 0,5 suggest no discrimination, from 0,7 to 0,8 are considered acceptable, values from 0,8 to 0,9 are considered excellent, while values above 0,9 are outstanding (Mandrekar, 2010). On the other hand, it would be unrealistic with an AUC equal to 1.

### 4.11.2 Confusion matrix

Another possible evaluation method when predicting credit default is the True Negative (TN), False Negative (FN), False Positive (FP), and True Positive (TP) values found in a confusion matrix. These give an insight into what values the model picks, both right and wrong. FP is often referred to as a Type I error as it finds TN observations to be FP. FN, on the other side, is often referred to as a Type II error as it finds TP observations to be FN. One way to evaluate these values is to find the difference between TN and FN and a model that finds many TN and not as many FN. Generally, Type I errors are more tolerated than Type II (Beauxis-Aussalet & Hardman, 2014). Therefore, one wishes the relationship between TP and TP + FN (TPR) to be as close to 1 as possible, while the relationship between FP and FP + TN (FPR) to be as close to 0 because the model can then remove incorrect predictions (Shrivastav, 2020).

$$TPR = \frac{TP}{TP+FN} \tag{9}$$

$$FPR = \frac{FP}{FP+TN} \tag{10}$$

### 4.11.3 Accuracy

Accuracy is probably the most used evaluation method when predicting default. It points out the model's accuracy and is a excellent aid for comparing several models. Previous work has shown that imbalance can significantly impact on the value of accuracy, e.g., Tsai, Lin, Cheng, and Lin (2009), but also the results meaning (Luque et al., 2019). Therefore, it might be a problem using accuracy as a measure of our dataset. The formula of accuracy is given by:

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{11}$$

## 4.12 Our network specifications

As mentioned in Section 4.4.1, there are no clear rules for what makes the best performing network. It is a process of trial and error (Han et al., 2012). The final network consists of one input layer, three dense linear layers, and one output layer. The first layer has a batch normalization and a Leaky ReLu activation function. The next two layers use a Tanh activation function, and the output uses a Sigmoid. The network's topology is respectively: 39, 512, 840, 950, 2. The last layer before the output is a dropout layer, with $p = 0,25$.

The loss function is a weighted binary-cross-entropy function modified to reflect the imbalance of the minority and majority classes. The network has been trained for 1000 epochs, using a mini-batch approach to the training loader, where each batch fed is 128. The validation frequency has been one, meaning it also iterates through the validation set once per epoch. The optimization algorithm is ADAM, with an initial learning rate of 0,0001. Other initial parameters, such as weights and biases, are set by PyTorch automatically. The hyperparameter for the classification threshold is further discussed in Section 6.4.3. For initial information given on the networks performance, as well as the code for both the loss function and the network's attributes, see Appendix H.

# 5 Results

In this Chapter, we go through the results of our research. First, we take the reader through our results using a neural network to predict credit default before the more traditional logistic regressions is shown. At the end of the Chapter, we go through the SHAP values to see which variables affect the neural network the most.

## 5.1 Results using neural network

It has been important throughout this thesis to follow the thesis statement: *How do neural networks preform in predicting default on unsecured loans using application data?* Therefore, we have focused on understanding how neural networks work in practice and how they have performed in previous articles and research.

As mentioned in Section 4.11.1, an imbalance can significantly impact on the value of accuracy (Luque et al., 2019). The accuracy calculated according to Formula (11) is extremely high at 96,72%, and it has been between 96% and 98% for every neural network trained. This makes it hard to use accuracy to interpret the performance, due to low variations in high values. The accuracy is also highly dependent on the threshold hyperparameter, and will change as the threshold changes. Therefore, we cannot, in good faith, consider this alone when predicting credit default on unsecured loans. There is a major imbalance problem in our dataset. However, when looking at the ROC in Figure 12, we observe that the AUC is 0,79, which is within what is defined as acceptable in Section 4.11.1.

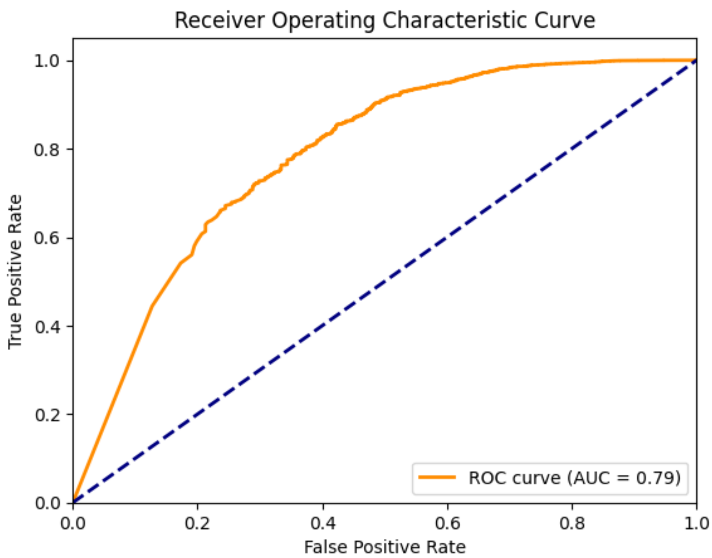


Figure 12: ROC neural network

The beforementioned accuracy problem is also confirmed by the confusion matrix in Table 3, where we can see that there is a significant imbalance in our dataset. 503 observations are default, while 13.453 are not. Our model finds 68 TN out of 503 default observations and 23 FN out of 13.453 non-default observations. This means that we have a "positive difference" in predicting more right observations than what was the case when approving the applications. Even if the model had a "negative difference", it does not mean that the model is too bad as we seek to minimize the incorrectly predicted defaults. Looking at the confusion matrix in Table 3 and using the formulas in Section 4.11.2, we found that the TPR is 0,998 while the FPR is 0,865. TPR is close to the goal of 1, while FPR is further from its goal of 0. The same applies here. It is unlikely to search for a value of 0 in the real world. Results using different thresholds on the confusion matrix are shown in Appendix B.

Table 3: Confusion matrix neural network

| | | Actual | | |
|---|---|---|---|---|
| | | True | False | Total |
| Predicted | True | 13 430 | 435 | 13 865 |
| | False | 23 | 68 | 91 |
| | Sum | 13 453 | 503 | 13 956 |

## 5.2 Lasso and logistic regression

As logistic regressions appear to be the most used prediction technique in financial institutions, based on what we have from the literature review in Section 2.3, it will be natural to use the method as a supplement in this thesis. Based on the variables used in the neural network, lasso regression has helped to find the most relevant independent variables. This way, our assumptions are as similar as possible while we build the most suitable logistic regressions. We have created two different logistic regressions. One that includes a validation set and one that is not. Both are introduced below.

### 5.2.1 Logistic regression 70-30 split

Since there is usually no validation set in a logistic regression, and we use the same split as for the neural network, the number of observations in the test set is doubled.

This can constitute a difference in the lasso regression, which further changes the results of both the AUC and the values of TN and FN. Our model finds an AUC of 0,8003, which is better than what the neural network gives, and just within the definition of excellent from Section 4.11.1. The ROC plot is shown in Figure 13:

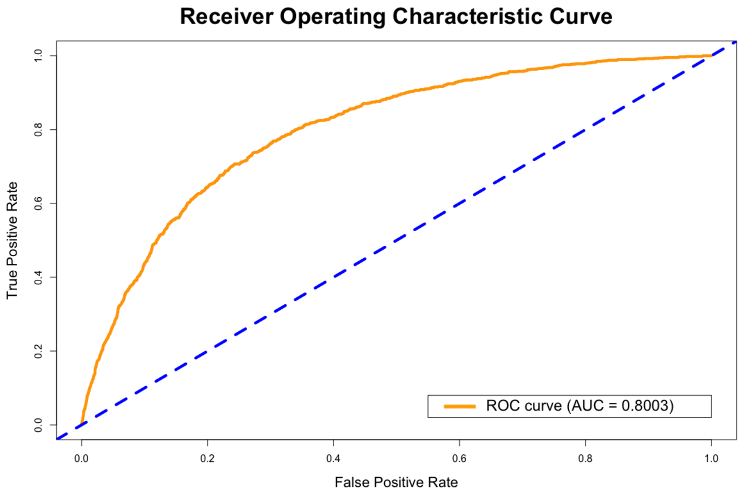


Figure 13: ROC logistic regression 70-30 split

Although there is no reason to jump to a conclusion. The lasso regression picks out the variables that have further been used in the logistic regression to find TN, FN, FP, TP, TPR and FPR values. As shown in Appendix E, there are 21 independent variables used to find the best model. Comparing the results from the confusion matrix in Table 4 with the one in Table 3, there are fewer predicted TN. The model correctly predicts 28 TN out of 1.068. On the other hand, out of 26.842 correct predicted non-defaults, it finds 79 as default. Looking at it the same way as we did with the neural network, this indicates a "negative difference" of 51 observations, as the model predicts more wrong observations than was the case when approving the applications. Further, we see from these values that TPR is very high at 0,997 and the same for FPR with 0,974.

Table 4: Confusion matrix logistic regression 70-30

| | | Actual | | |
|---|---|---|---|---|
| | | True | False | Total |
| Predicted | True | 26 763 | 1 040 | 27 803 |
| | False | 79 | 28 | 107 |
| | Sum | 26 842 | 1 068 | 27 910 |

### 5.2.2 Logistic regression, including validation-set

Including validation set when using logistic regression is not common practice. What the validation set does to our dataset is removing 15%, which means we are only using 85% of our data. At the same time, it is easier to compare with the neural network as the number of observations is almost the same. The model finds an AUC at 0,791, which as for the neural network is within what is defined as acceptable. The ROC plot is show in Figure 14:

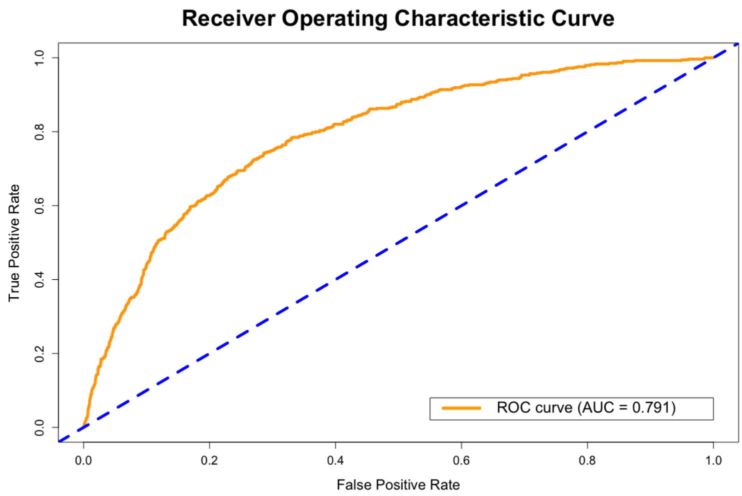


Figure 14: ROC logistic regression including validation-set

As is possible to see in Appendix F, the lasso regression uses 22 independent variables to find the best model. Comparing the confusion matrix in Table 5 with the one in Table 4, the differences between TN and FN differ. Considering that there are half as many observations for the logistic regression including the validation set, as it is for the one with a 70-30 split. The model finds 25 TN and 95 FN, which again indicates a "negative difference", and finally, the model does not pick up enough of the TN observations compared to the neural network. The low TN is also illustrated by an FPR of 0,953, while the TPR is 0,993. This means that for both the logistic regressions, TPR is very good, while the FPR is bad as we seek to have a value as close to 0 as possible.

Table 5: Confusion matrix logistic regression validation (70-15-15)

| | | Actual: True | Actual: False | Total |
|---|---|---|---|---|
| Predicted | True | 13 326 | 509 | 13 835 |
| | False | 95 | 25 | 120 |
| | Sum | 13 421 | 534 | 13 955 |

## 5.3 SHAP

As was mentioned in Section 4.8, it is essential to have a greater understanding of what variables affect our neural network. In what follows, we will present three alternative SHAP plots that can be used to do so. To see which variable a feature is and its description see Table 6. For the full list of all 39 variables; Appendix A.

Table 6: Decoding of features

| **Variable name** | **Feature** | **Description** |
|---|---|---|
| refin | Feature 1 | 1 if refinancing, 0 otherwise |
| uføre | Feature 10 | 1 if disabled, 0 otherwise |
| ansatt | Feature 11 | 1 if employee, 0 otherwise |
| student | Feature 16 | 1 if student, 0 otherwise |
| Midlertidig_ansatt | Feature 17 | 1 if temporary employee, 0 otherwise |
| maksgrense | Feature 20 | Loan amount |
| alder | Feature 21 | Age of customer |
| total_gjeld | Feature 35 | Total debt |
| total_innskudd | Feature 36 | Total deposit |
| rentebærende_gjeldsregister | Feature 38 | Loan-amount with interest registered at Gjeldsregisteret |

The SHAP barplot shows the independent variables' average impact on the model's output. Figure 15 shows that whether or not the applicant is employed has the highest impact when classifying defaults and non-defaults.

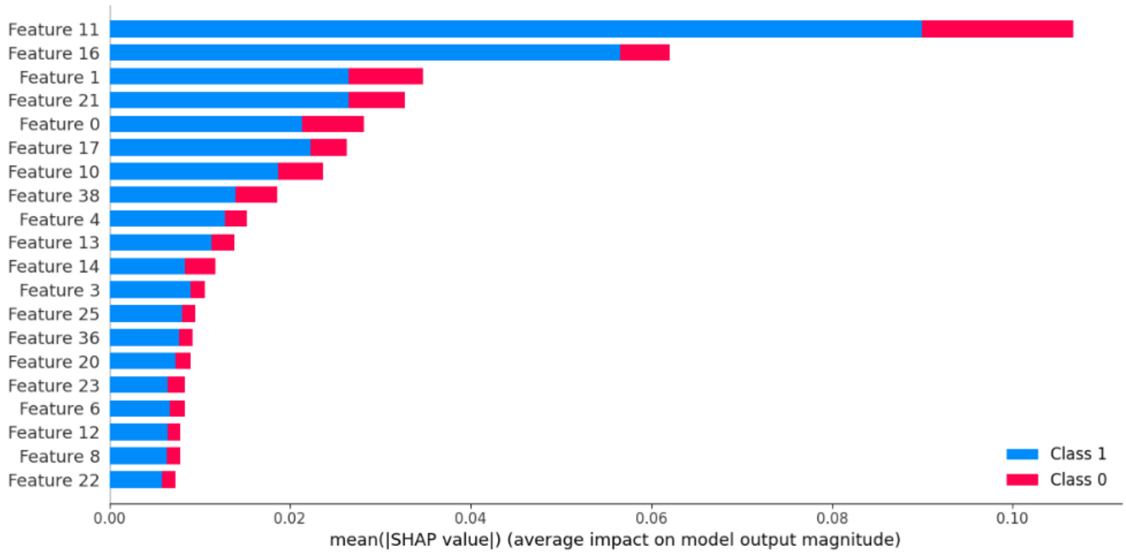


Figure 15: SHAP: average impact on output

The plot also shows that being a student is important for classifying non-defaults and significantly impact the classification more than the loan being a refinanced loan and the applicant's age. On the other hand, the plot shows these two last mentioned features as more important when classifying a default than being a student.

The beeswarm plot in Figure 16 shows how the variables impact the specific movement within a node and gives more detail to the bigger picture. Due to the output being one-hot encoded, the model has two output nodes. The plot will be taken from the "default node". The color encoding of the dots tells whether the value of the feature is high or low. The SHAP value on the x-axis shows the observation's impact on its prediction. The plot also shows clear non-linearity when it comes to age and other registered loans with interest. Being temporarily employed has the most significant impact on the minority class in one of the cases, and normal employment has the same towards the majority class.

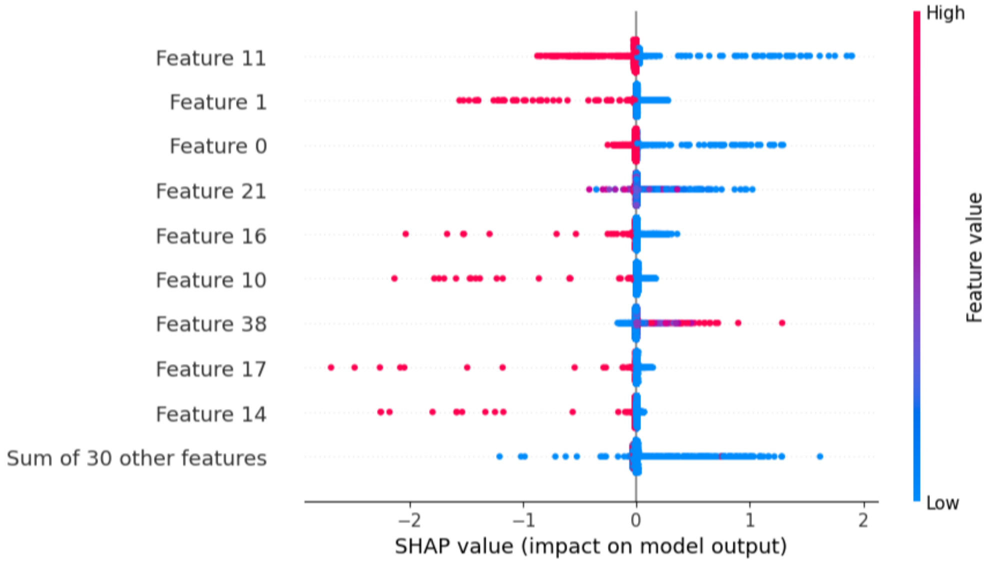


Figure 16: SHAP: Impact on the model for each case

It is important to remember that the SHAP values itself does not say anything about how much the prediction move, just the impact it has on the overall prediction of the model. It is a way to see what is important and in which direction it pushes the model.

A deeper dive into the workings of the model is done by waterfall plots. The waterfall plot shows the features' impact on the result. The node we look at in the following is the output node where 1 = default and 0 = non-default. In other words, the following plot is from the same node as the beeswarm plot. Figure 17 shows a correct default prediction and which factors push the model to make that prediction. The same logic follows for Figure 18, a correctly classified non-default.

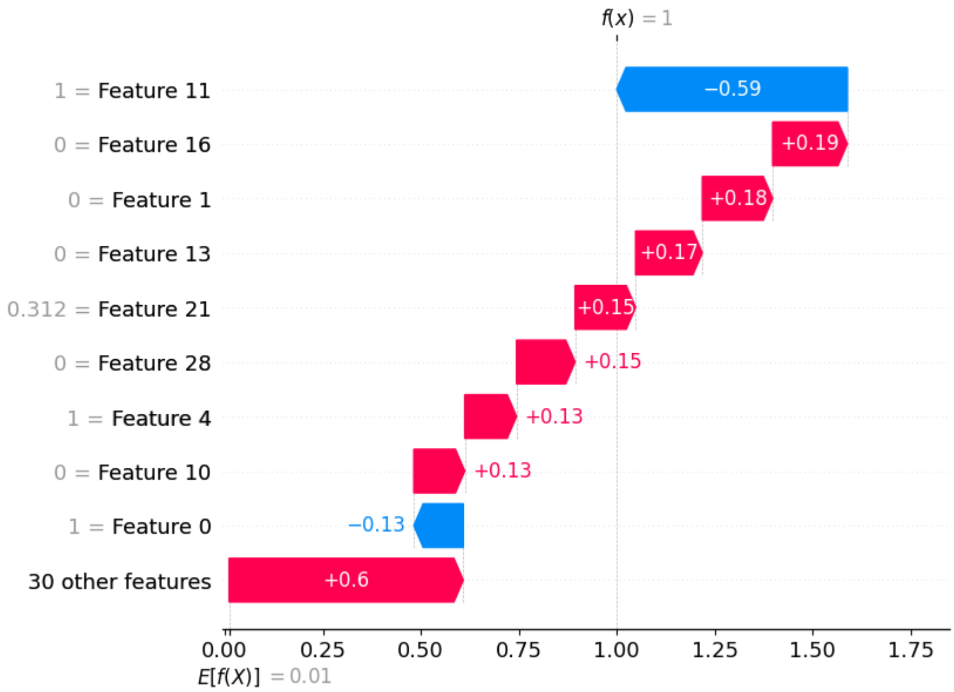


Figure 17: SHAP: Impact on the model's prediction, one individual default case

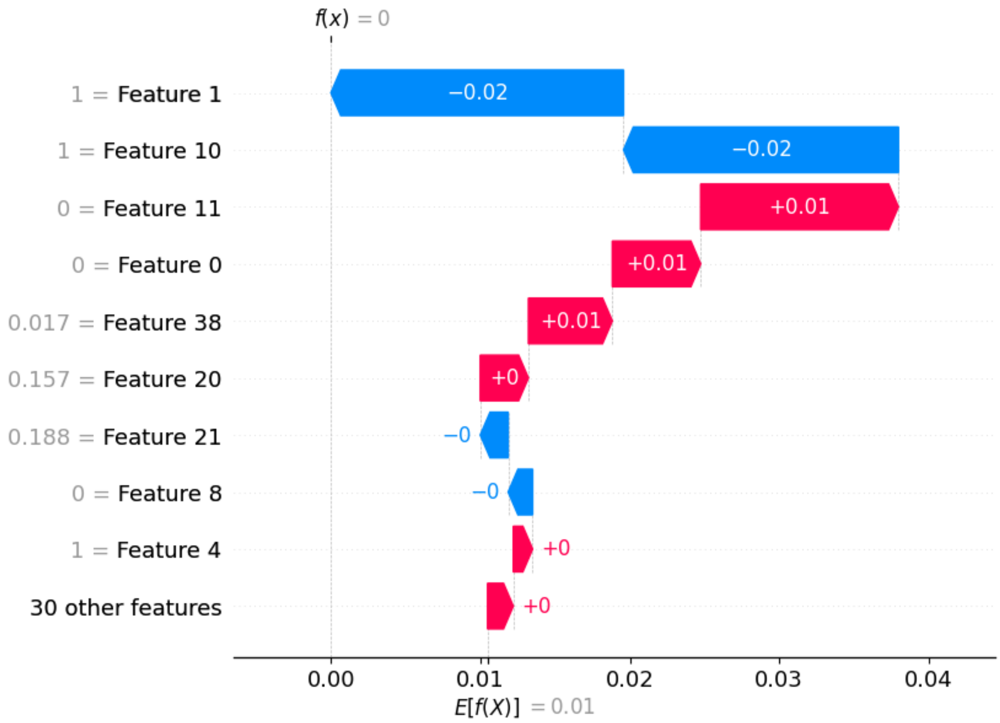


Figure 18: SHAP: Impact on the model's prediction, one individual non-default case

This illustrates that the model weighs its input differently for each case, but the prediction would still be the same, if all the input is equal. How much a feature moves the model in a direction also changes from case to case. Showcasing how hard it is to understand the inner workings of these "black boxes".

# 6 Discussion

In this Chapter, we discuss both the use and the results of our models and how neural networks can be implemented in a financial institution. We discuss the overall possibilities and potential effects of using the neural network and how it can be further developed for the better, with other variables or prerequisites included.

## 6.1 Discussion of our results

In this thesis, we have worked towards finding a neural network that identifies more customers with a high potential to default on their unsecured loans. As we saw in Section 5.1, the accuracy of the neural network is artificially high. Therefore, we had to find other measures that could give us a better insight into how the neural network performs, like Naik (2021) did for LGBM and logistic regression. The ROC plots showed that there are no significant differences when it comes to the AUC between the neural network and the two different logistic regression models. As mentioned, the AUC appears to be slightly better using logistic regression, but this alone gives no reason to conclude.

As the AUC did not shed light on the objective itself, we had to look closer at the confusion matrix values, as described in Section 4.11.2. We have worked towards finding the best TN, FN, FP, and TP in 3 different confusion matrices. Unlike the logistic regressions, there is what we defined as a "positive difference" between the TN and FN of 45 in our neural network. This means that the model finds 45 more TN than FN. Using a neural network appears to be an advantage based on the four different values.

The TPR and FPR described in Section 4.11.2 are also values we have looked at more closely. The difference to be discussed is the FPR from the neural network and the logistic regression models. We have worked towards finding a value as close as possible to 0. What was mentioned in Chapter 5 and is essential to understand, is that the target has never been finding an FPR equal to 0. The discussion should instead be about which number is good enough. What is good about our neural network is that it finds considerably more TN than the logistic regressions do and a lower number of FN. The TPR is slightly better for the neural network, as well as the FPR being better.

One of the most central problems with evaluating the model is how the default is defined in our data. Over the last years, the default rates have been declining, and the last data indicates a default rate of 9,9% on consumer loans (Finanstilsynet, 2022b). Meanwhile, our dataset has a default rate of 3,83%. This may indicate that some of the loans default after 12 months. If that is the case, these are still not classified as defaults, meaning that some of the wrongly predicted defaults may indeed be correctly predicted defaults. We did not possess data on this, but it may mean that the network performs better than the evaluation at first sight.

## 6.2 Most suited benchmark

This thesis presented two logistic regressions, where the difference consisted of the training, validation, and test split. As neither the regression nor the feature selection uses a validation set, we ended up hiding 15% of the data from the logistic regression. However, comparing models built on equal splits is easier and more intuitive. Due to both logistic regressions performing poorly, we gave the logistic regression with 70-30 split the benefit of the doubt. This is primarily due to the void the validation set gives our logistic regression. The intuitive aspect is not as important due to the logistic regressions not finding and identifying the minority class correctly, which is of interest to the creditor and debtor. Therefore, the most suitable benchmark would be the one using the most data. This illustrates the lacking ability of the logistic regression models to determine which observation belongs to the minority class.

When it comes to other factors, such as the finding of the lasso hyperparameter lambda, cross-validation with five folds has been used in both models. There is, however, one extreme observation where a debtor has 15 payment remarks and must be removed if it is in the test split, which causes the difference between observations in the logistic regression models and the neural network. These extreme observations can cause problems for linear models, due to unique factor observations in the test set. This is not an issue with non-linear models, such as neural networks. The difference between linear and non-linear models is also why we used the lasso as a feature selection for the logistic regressions. The essence is that linear models will never perform worse when going from z numbers of independent variables to z + u numbers of independent variables. For this to be true, z and u have to be positive integers. Lasso essentially punish the model for adding new variables, and at some point, this punishment is greater than the return of adding a new variable. This is because adding new variables has diminishing returns, as shown in Figure 19.

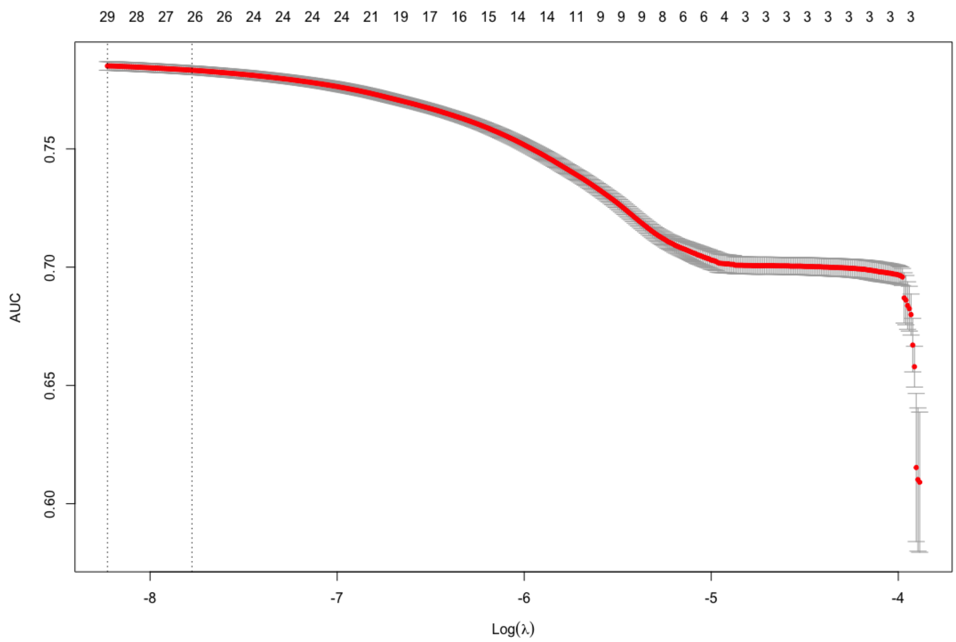


Figure 19: AUC-plot: Shows the diminishing returns of adding new variables. The greater the lambda, the fewer variables

The plot shows which lambda to use. This hyperparameter is a scalar that decides how much the parameters are punished. Thus, punishing some variables so much that their effect is zero. When these variables hit zero, they remain zero. Therefore, the features that still provides an effect with the given lambda are selected. This is as illustrated in Figure 20.

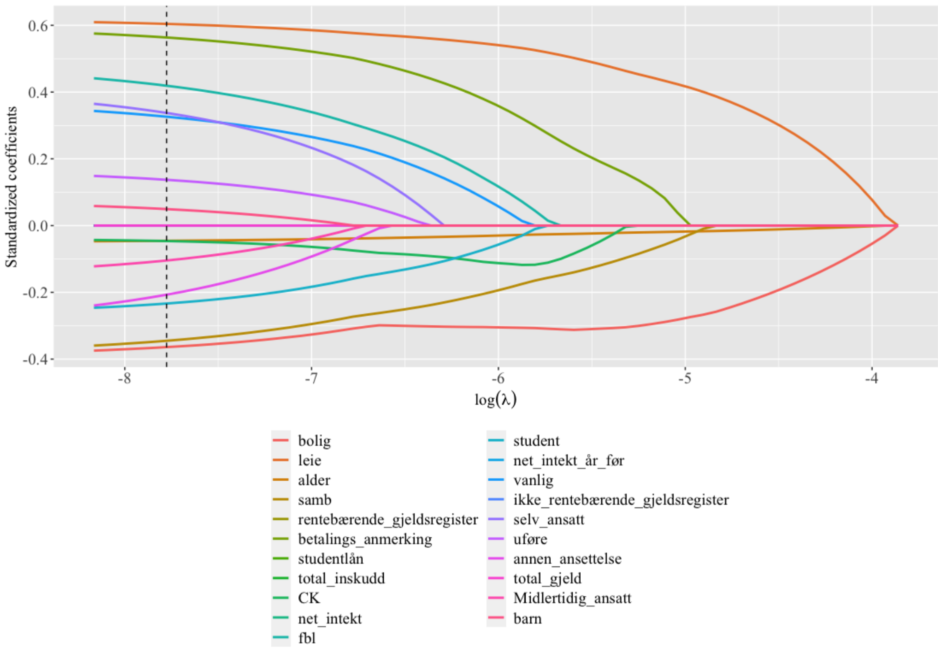


Figure 20: Lasso Regression-plot: Variables that are not zero at the dotted line are kept as features. From the original set of 39, only 21 remain

This is one of the points where non-linear models differ from linear models. In the case of neural networks, they adjust themselves, and if one input value is the same, the impact it has on the model is not necessarily the same. This is due to the model's complexity and how it regulates its evaluation of a variable, and its impact based on the other independent variables. Thus, making the model non-linear. Features that have little to add are automatically weighted as such by the model itself. Due to this, there is no reason to select the variables fed to the network before training starts. If the variables are selected beforehand, there is a risk of leaving out vital information for the model. This could also cause the model to not adjust properly compared to its potential.

## 6.3 SHAP and interpretability

Due to the non-linearity of neural networks, interpreting the effects of the variables is challenging. In comparison to the logistic regression that provides a coefficient for each variable, the neural network does not. That makes SHAP a valuable tool for understanding the impact of a variable in the neural network. The SHAP values indicate which variables are important in the classification process. This is essentially the same as what lasso does.

SHAP plots are meant to make "black boxes" more interpretable. But, there are still problems in explaining cause and effect. The bar plot shows which variables that have the most impact on average, and the beeswarm gives insight into which values of what variables contribute to what degree. However, when looking at the waterfall plot, chaos emerges. From the beeswarm, it is possible to see that a feature with the same value can differ in its impact. This inconsistency in the features' impact makes the interpretation and expectation more uncertain. This effect illustrates the issues with "black boxes", as their complexity is high. With a neural network of this size, explaining the cause and effect is impossible, as shown by our SHAP plot in Section 5.3.

Contrary to Chen et al. (2020), our model's SHAP indicates that younger people tend to impact the classification in the default node towards default. Meaning that our model provides a picture where younger adults pose a higher default risk. This also leaves room for speculation on the amount lost by the financial institution due to Kreditorforeningen's (n.d.) report on amounts defaulted for different age groups. On the other hand, being a student pulls in the opposite direction, towards a non-default. This is indicated by the SHAP beeswarm plot shown in Figure 16. There is a possibility that student, and younger age, together, negatively impact the default classification. This can be derived from the mean impact shown by the SHAP barplot (Figure 15). This hypothesis makes sense when seen in the light of Dynarski's (1994) study.

## 6.4 Importance of the minority class

As mentioned in Section 3.3, the minority class is of interest. Our neural network finds about 13,5% of the default cases using these variables and the specified architecture. This indicates a pattern not picked up by the model(s) used by the financial institution. This interests the participants in the debt market, as it will create shifts in the uncertainty associated with these kinds of loans.

As the data already has been passed through models and, in many cases, also human evaluation, it is fair to assume that no loans are given to see default. Therefore, the entire dataset received can be considered as classified as the majority class when it was given. This is what makes the majority class of almost no interest. Thus, making some of the normally used evaluation metrics less important and artificially high compared to what we wish to measure.

Therefore, we seeked models that deal with the minority class and do not just "save itself" by classifying everything as the majority class. Giving logistic regressions a disadvantage for these problems. Especially seeing how the data likely have been fed through a logistic regression before.

### 6.4.1 Changes to the independent variables

There are several changes to the data that can be tried out to amplify the pattern of default. One concern is that our neural network does not take the data as a time series. The reason for this is that the model does not need it if implemented as a part of the exciting procedures for approving or denying loans. Since the dependent variable only accounts for a maximum of one year after the application is granted, and the data already have a couple of lagged variables, time itself does not contribute to anything. However, if it exists for the customers, this could change by introducing macroeconomic variables, behavioral data, and/or changing the network's topology. This can be done by building the model for time series and using other layers, like recurrent layers, in the network. This is possible, but we have no reason to say it would be better or worse. However, Tsai, Lin, Cheng, and Lin (2009) showed the potential of using behavioral data to improve machine learning models. The neural network's behavior may change and, ultimately, its performance.

The optimal use for a neural network would be to have a trained network on all existing data. The "test" set, which is to be predicted, consists of new application data for the latest week or month. If there were no legal issues (e.g., GDPR, "Utlånsforskriften") with this kind of practice, then time as a variable itself would not be necessary if the independent variables could reflect time as a fixed effect. However, there are possibilities for using lagged variables when it comes to income and changes for real wages in time t, t-1 etc.

### 6.4.2 Exemption of time in our model

Whether the network should take a time series depends on how the model will be used and to what degree the independent variables already have time as a fixed effect in their development. It is also conditioned to the necessity given the interval for the "default" measurement. As mentioned, the actual default rate may be higher than the data suggests because this measurement's time horizon is one year. Due to our dataset only containing data from the middle of 2019 and throughout 2022, the possibilities for expanding this horizon will improve in the years to come. This is essentially due to "Gjeldsinformasjonsloven", which imposes financial institutions to hand over this kind of data to debt registers for unsecured loans (Gjeldsinformasjonsloven, 2017).

Given this short time interval in our data, we decided not to treat the data as time series due to having several lagged variables in our set. We concluded that the time of the applications would not make any difference for the pattern leaking through the already exciting models, which again probably uses time as an input. Seeing as an independent variable of time itself in the current format would not change anything, it has been exempted from our neural network.

### 6.4.3 Threshold

Another critical decision is setting a threshold for the classification in the logistic regression and the neural network. This is because all our models' outputs are continuous with values [0,1]. The nature of the classification problem is to either set the value to 0 or 1, thereby making the classification. This is done by the threshold. The threshold forces the continuous output in either direction by saying any output over a set value is 1, and anything else is 0 (Handoyo et al., 2021).

Changes to this hyperparameter can drastically change the model's predictions and performance. This is illustrated in our Appendix, sections; B, C, and D. Due to the minority class being of interest, we can see that the neural network outperforms the logistic regression. This is based on the network predicting more TN than FN. The practicality of this is that the model identifies mostly true defaults in its prediction, minimizing the loss for the financial institution by denying the loans.

Due to the neural network being one-hot encoded, the output needs to be reduced to one dimension. This leads to the difference in the behavior of the threshold values for the neural network compared to the logistic regression. The optimal value differs from model to model because different models will give different outputs for the same observation, meaning that the threshold needs to be set accordingly. The final threshold is set based on the classification's performance and objective, which in this case, prioritizes the minority class. The final threshold for the neural network was set to 0,28. For the logistic regression (70-30 split), the threshold was set to 0,20.

## 6.5 Economic effects

From the aspect of the financial institution, the economic effects of using a neural network to decide which applications will be approved are hard to pinpoint. Our data provides no information on whether the loan defaults within the first or eleventh month. Another issue is that there is no data on the economic gain on the loans from the interest rate due to the time before default being unknown. There are also most likely loans that default after the dataset's 12 months default definition, which causes more problems regarding an economic evaluation. This is, however, easy for the creditor with this data to assess.

When it comes to the debtor, the repercussions of defaulting on a loan are huge. In some cases, this can even have effects that hurt the individuals. By having defaulters participate in the market, we raise the uncertainty in the credit market and the cost for non-defaulters to participate in it. Identifying these cases before the loans are approved will lower the cost of participating for debtors that do not default. At the same time, the expected return for the financial institution could be higher. Another effect of this is that the defaulter will have fewer problems, and a better economic situation by being denied the loan instead of defaulting it.

Since the Norwegian central bank dictates the access to credit for the public, lending to non-defaulters would be more sustainable. Not only would there be more freed-up credit in the market, but the cost would be reduced for everyone due to the cost of collecting what is owed being reduced.

When looking at returns and economic effects, uncertainty goes hand in hand. Without uncertainty, there would be hard to capitalize off of the markets. When investing money, the investor always seeks to maximize their risk-adjusted return. Even in credit markets, you could use the Sharpe ratio to describe this (Correia et al., 2012).

$$Sharperatio = \frac{R_p - R_f}{\sigma_p} \tag{12}$$

The Sharpe ratio describes the ratio of reward given a certain risk ($\sigma_p$). The formula describes the risk-relative return of an investment. $R_p$ and $R_f$, respectively, represents the return on their portfolio (in this case their portfolio consists of loans) and the risk-free rate (Zvi Bodie, 2021). A way to increase $R_p$ would be to decrease the cost of the portfolio. This is as long as the costs cut are greater than the loss of income. However, there are important to remember the paid rates and repaid portion of the loans. If the savings of using a neural network to determine defaults exceed the income loss, then the portfolio's return would increase. The same goes for risk-adjusted returns.

However, as mentioned, the loan's interest rate reflects the market's uncertainty. If defaulters were not participating in the market, the risk of lending money would not exist. Therefore, it follows by logic that a reduction of defaulters would give a reduction of risk in the market. This means that $\sigma_p$ would be lowered, as well as the $\boldsymbol{R_p}$, due to interest rates making up the return on loans. Assuming an unchanged Sharpe ratio, would mean that the uncertainty of the market has a negative shift, which would also benefit debtors when it comes to the cost of having loans.

Compared to the logistic regression, the neural network performs better when classifying the minority class. However, as mentioned, it is highly unlikely that this network is optimal regarding hyperparameters and topology. This means that there is a potential to use these models to shift the uncertainty. This would better the terms for debtors that would not default on their loan, and increase the Sharpe ratio for the creditors, indicating better returns for the risk they are taking if the $\boldsymbol{R_p}$ stays the same. By identifying more than one-eighth of all defaults leaking through an already well-integrated system, the network shows potential to reduce risk on behalf of the creditor. If the reduction of risk is real, it will, as shown, also improve the conditions for the debtors. This means the model can be used as part of the creditor's decision-making, to better the market.

## 6.6 Options for implementation

Implementing new and more correctly predicting models allows the possibility to save the customers and the financial institution the loss a default gives. But it is also essential to look at the implementation considering ethics. Whether it is unethical to refuse a loan application because a machine tells the institution to do so remains to be discussed. The same goes for whether the implementation is right or even more relevant to what extent. Ethically the results should be used as a conversation tool between the creditor and debtor.

Another ethical option mentioned but not discussed is using these neural networks to group existing debtors. Financial institutions already monitor their customers to some degree. This is due to the risk of default and the possibility of minimizing the cost of their portfolio. For example, they could allocate more resources to the customers classified as default by a machine learning model, like a neural network. This would not change anything on the debtor side of the loan. Thus, making the creditor better suited for stepping in before a default and making the existing agreement more suitable for all parties involved. If the model performs well, this will also save man-hours for the creditors. There are also possibilities for opening the network for existing debtors and applicants. Let them review the prediction from the model and make the necessary changes not to default. By using the network this way, the model would act more as a tool for making behavioral changes in the debtors. From the creditor's side, it is also possible to use it to measure how well their traditional models work.

At this point, there are several ways machine learning models could be implemented into the vital parts of the creditor's process. It is great potential in different financial institutions sectors, which can save customers from unnecessarily high costs. However, legality issues exist in replacing the current decision-making models and processes. The "black box" problem mentioned in Section 4.3 is one of them, as an auditor cannot easily observe anything but the input or output. Therefore, the "black box" problem must be seen in the light of the ethical problem, as one cannot easily observe why the answer to the application is as it is.

Another question is whether neural networks are generally appropriate for usage in financial institutions. There are both advantages and disadvantages of using neural networks. First, what has been mentioned, is the noise and that the model can often be sensitive to changes in market conditions. On the other hand, neural networks help analyze big data better than traditional logistic regression does. It is also a better tool for finding patterns and trends to help make better decisions. More specifically, to help make a better decision when loan applications are to be processed.

# 7 Conclusion

This thesis aimed to build a neural network that finds more defaulters on unsecured loans than the traditional logistic regression. We built a multilayer feed-forward neural network that shows promising results. Due to a larger imbalance problem, a weighted binary-cross-entropy loss function was created to reflect the problem of the minority and majority classes. As a result of the imbalance, we analyzed the results using AUC and confusion matrices, with a heightened focus on the minority class and a lesser focus on the majority class. This means that the negative predictions and the difference between TN and FN are valued the most. This is because this difference and the TN make the model valuable from an economic point of view.

Our neural network has the potential to outperform logistic regression when identifying defaulters. At the same time, it is a positive difference between TN and FN. The results presented in this thesis are promising, yet the subject requires further research. There are, however, some thoughts about different data. As there are strict rules on what variables and information we have been able to get and use in this thesis, using behavioral variables has been impossible. For future work, we recommend this to be included. This may have an even stronger impact on the results. It would also be interesting to see what results the model would have had with a larger dataset available. Finally, we recommend changes to the definition of default concerning the time horizon.

Even though our neural network is likely sub-optimal, it classifies more than one-eighth of the defaulters. Thus, it shows the potential to create shifts in the market's uncertainty and the ability to reduce risk on behalf of financial institutions. Since our results may be anomalous, there is hard to conclude on the research question, but rather the potential. However, the scale of the shift in uncertainty depends on the scale of implementation. It will lead to lower customer costs, benefiting the main parties involved. At the same time, to experience these changes on a market scale, more than one financial institution must adapt neural networks into its production line. This could further improve the conditions for debtors since increased competition could likely result in lower consumer costs.

Making optimal neural networks is time-consuming. Whether this requires changes to the input variables, hyperparameters, or topology is impossible to say. However, the possibility exists. Financial institutions must develop their networks and assess models with acceptable performance. Each financial institution must determine what is acceptable performance due to its individuality. Another question is if neural networks fit into an already existing process. This is due to these processes' secrecy and their differences. We believe that this ultimately depends on the network's performance and must be assessed considering current procedures.

The ethical evaluation of whether using a neural network is the correct way to go must still be discussed. The same goes for changes in the legal practices and laws regarding the usage of these models. As every aspect of our society adapts to new technology and advances in its way, these practices for giving loans are bound to change, but if this is the correct way remains unanswered. However, it is a tool with high potential which deserves more space in the financial institutions' everyday life and development. It is a tool that can be implemented if it stays within the legislation.

# Bibliography

Albanesi, S., & Vamossy, D. F. (2019). *Predicting consumer default: A deep learning approach* (Working Paper No. 26165). National Bureau of Economic Research. https://doi.org/10.3386/w26165

Baesens, B., Gestel, T. V., Stepanova, M., Poel, D. V. d., & Vanthienen, J. (2005). Neural network survival analysis for personal loan data. *Journal of the Operational Research Society*, *56*(9), 1089–1098. https://doi.org/10.1057/palgrave.jors.2601990

Bayraci, S., & Susuz, O. (2019). A Deep Neural Network (DNN) based classification model in application to loan default prediction. *Theoretical and Applied Economics*, *0*(4(621), W), 75–84. https://ideas.repec.org/a/agr/journl/vxxviy2019i4(621)p75-84.html

Beauxis-Aussalet, E., & Hardman, L. (2014). Simplifying the visualization of confusion matrix.

Bhandari, A. (2023). Guide to auc roc curve in machine learning: What is specificity? *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Bharath, K. (2020). The complete interesting and convoluted history of neural networks! *Towards Data Science*. https://towardsdatascience.com/the-complete-interesting-and-convoluted-history-of-neural-networks-2764a54e9e76

Bhor, Y. (2022). Guide for building an end-to-end logistic regression model. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/09/guide-for-building-an-end-to-end-logistic-regression-model/

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Burns, E. (n.d.). Machine learning. *TechTarget*. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

Byanjankar, A., Heikkilä, M., & Mezei, J. (2015). Predicting credit risk in peer-to-peer lending: A neural network approach. *2015 IEEE Symposium Series on Computational Intelligence*, 719–725. https://doi.org/10.1109/SSCI.2015.109

Chakraborty, K., Bhattacharyya, S., Bag, R., & Hassanien, A. A. (2019). 7 - sentiment analysis on a set of movie reviews using deep learning techniques. In N. Dey, S. Borah, R. Babo, & A. S. Ashour (Eds.), *Social network analytics* (pp. 127–147). Academic Press. https://doi.org/10.1016/B978-0-12-815458-8.00007-4

Chen, J. (2023a). Unsecured loans: Borrowing without collateral. *Investopedia*. https://www.investopedia.com/terms/u/unsecuredloan.asp

Chen, J. (2023b). Default: What it means, what happens when you default, examples. *Investopedia*. https://www.investopedia.com/terms/d/default2.asp

Chen, S., Gu, Y., Liu, Q., & Tse, Y. (2020). How do lenders evaluate borrowers in peer-to-peer lending in china? *International Review of Economics  Finance*, *69*, 651–662. https://doi.org/10.1016/j.iref.2020.06.038

Correia, M., Richardson, S., & Tuna, İ. (2012). Value investing in credit markets. *Rev Account Stud 17*, 572–609. https://doi.org/10.1007/s11142-012-9191-x

DataRobot. (2020). The importance of machine learning data. *DataRobot*. https://www.datarobot.com/blog/the-importance-of-machine-learning-data/

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM Comput. Surv.*, *27*(3), 326–327. https://doi.org/10.1145/212094.212114

Duan, J. (2019). Financial system modeling using deep neural networks (dnns) for effective risk assessment and prediction. *Journal of the Franklin Institute*, *356*(8), 4716–4731. https://doi.org/10.1016/j.jfranklin.2019.01.046

Dubey, A. K., & Jain, V. (2019). Comparative study of convolution neural network's relu and leaky-relu activation functions. In S. Mishra, Y. R. Sood, & A. Tomar (Eds.), *Applications of computing, automation and wireless systems in electrical engineering* (pp. 873–880). Springer Singapore. https://doi.org/10.1007/978-981-13-6772-4_76

Dynarski, M. (1994). Who default on student loans? findings from the national postsecondary student aid study. *Economics of Education Review*, *13*(1), 55–68. https://doi.org/10.1016/0272-7757(94)90023-X

e Silva, E. C., Lopes, I. C., Correia, A., & Faria, S. (2020). A logistic regression model for consumer default risk. *Journal of Applied Statistics*, *47*(13-15), 2879–2894. https://doi.org/10.1080/02664763.2020.1759030

Feis, A., Mehta, V., Morris, S., Solitario, J., & Graaf, C. V. d. (2016). P2p loan selection. *Standford University.*

Finansdepartementet. (2022). Utlånsforskriften. *Regjeringen.* https://www.regjeringen.no/no/tema/okonomibudsjett/finansmarkedene/utlansforskriften/id2950571/

FinansNorge. (n.d.). Slik setter bankene boliglånsrenten. *FinansNorge.* https://www.finansnorge.no/tema/bank/slik-setter-bankene-boliglansrenten/

Finanstilsynet. (2022a). Redusert utlånsvolum i foretak som tilbyr forbrukslån, men fortsatt høy andel misligholdte forbrukslån. *Finanstilsynet.* https://web.archive.org/web/20221109001936/https://finanstilsynet.no/nyhetsarkiv/nyheter/2022/redusert-utlansvolum-i-foretak-som-tilbyr-forbrukslan-men-fortsatt-hoy-andel-misligholdte/

Finanstilsynet. (2022b, October). *Utviklingen i forbruksgjeld.* https://www.finanstilsynet.no/contentassets/e238d68d2f0d4cfb8e412683307ec20f/utviklingen-i-forbruksgjeld-oktober-2022.pdf

Gjeldsinformasjonsloven. (2017). *Lov om gjeldsinformasjon ved kredittvurdering av privatpersoner* [LOV-2017-06-16-47]. Lovdata. https://lovdata.no/dokument/NL/lov/2017-06-16-47?fbclid=IwAR07gRdcddSMrNkyN_Wq_taJBU--ul0XEGKLgwfS/DGLseu2BrmcONhKOD9w#KAPITTEL_1

Gjeldsregisteret. (n.d.-a). *Nøkkeltall.* https://www.gjeldsregisteret.com/pages/nokkeltall?fbclid=IwAR274uRlVbU-lMeHIz55s_cpWm1HUyQ7DigxfyWBEz3R/u53z6Oj_dM9LfuE

Gjeldsregisteret. (n.d.-b). Samlet forbruksgjeld gikk ned i desember, men vi må ikke la oss lure. *Gjeldsregisteret.* https://www.gjeldsregisteret.com/blog/post/samlet-forbruksgjeld-gikk-ned-i-desember-men-vi-ma-ikke-la-oss-lure?fbclid=IwAR1jXhCXICEt_zHfGPnqXMvDS3Kz4plMxnHZ5QoX2peSM2lSVakuGMYCX/YA

Gosain, A., & Sardana, S. (2017). Handling class imbalance problem using oversampling techniques: A review. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 79–85). https://doi.org/10.1109/ICACCI.2017.8125820

Hammervold, R. (2022, March 28). Logistisk regresjon.

Han, J., Kamber, M., & Pei, J. (2012). 9 - classification: Advanced methods. In *Data mining: Concepts and techniques* (Third Edition, pp. 393–442). Elsevier Inc.

Handoyo, S., Chen, Y.-P., Irianto, G., & Widodo, A. (2021). The varying threshold values of logistic regression and linear discriminant for classifying fraudulent firm. *Mathematics and Statistics*, *9*(2), 135–143. https://doi.org/https://doi.org/10.13189/ms.2021.090207

Hardesty, L. (2017). Explained: Neural networks. *Massachusetts Institute of Technology*. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

Hasan, A., Al-Assadi, H., & Azlan, A. (2011). Neural networks based inverse kinematics solution for serial robot manipulators passing through singularities. In K. Suzuki (Ed.), *Artificial neural networks*. IntechOpen. https://doi.org/10.5772/14977

Høie, H. (2021, December). *Usikret gjeld – omfang og kjennetegn ved låntakerne*. Statistisk Sentralbyrå. https://www.ssb.no/inntekt-og-forbruk/artikler-og-publikasjoner/_attachment/449421?_ts=1786315b3f0&fbclid=IwAR3Dk-h__ST4CQ0w8EKOBKBLevtaob/goXJQamugpdKZ3115du2VA7anSzzw

Hoo, Z. H., Candlish, J., & Teare, D. (2017). What is an roc curve? *Emergency Medicine Journal*, *34*(6), 357–359. https://doi.org/10.1136/emermed-2017-206735

Jaspreet. (2016). A concise history of neural networks. *Towards Data Science*. https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec

Kaushik, V. (2021). 8 applications of neural networks. *Analyticssteps*. https://www.analyticssteps.com/blogs/8-applications-neural-networks

Kostadinov, S. (2019). Understanding backpropagation algorithm. *Towards Data Science*. https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

KPMG. (2017). Ifrs 9 - den nye nedskrivningsmodellen for utlån i banker. *KPMG*. https://kpmg.com/no/nb/home/nyheter-og-innsikt/2017/01/ifrs-9-den-nye-nedskrivningsmodellen-for-utlan-i-banker.html

Kreditorforeningen. (n.d.). Denne gruppen har mest misligholdt gjeld. *Kreditorforeningen*. https://www.kred.no/bedrift/nyheter/denne-gruppen-har-mest-misligholdt-gjeld/

Krogh, A. (2008). What are artificial neural networks? *Nat Biotechnol*, *26*, 195–197. https://doi.org/10.1038/nbt1386

Kurenkov, A. (2020). A brief history of neural nets and deep learning. *Skynet Today*. https://www.skynettoday.com/overviews/neural-net-history

Kvamme, H., Sellereite, N., Aas, K., & Sjursen, S. (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications*, *102*, 207–217. https://doi.org/10.1016/j.eswa.2018.02.029

Lånemegleren. (2022). Hvordan bestemmes renten på et lån? *Lånemegleren*. https://www.xn--lnemegleren-x8a.no/nyheter/hvordan-bestemmes-renten-pa-et-lan/

Lau, M. M., & Hann Lim, K. (2018). Review of adaptive activation function in deep neural network. In *2018 ieee-embs conference on biomedical engineering and sciences (iecbes)* (pp. 686–690). https://doi.org/10.1109/IECBES.2018.8626714

Li, Z., Li, K., Yao, X., & Wen, Q. (2019). Predicting prepayment and default risks of unsecured consumer loans in online lending. *Emerging Markets Finance and Trade*, *55*(1), 118–132. https://doi.org/10.1080/1540496X.2018.1479251

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(2), 539–550. https://doi.org/10.1109/TSMCB.2008.2007853

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Luque, A., Carrasco, A., Martn, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231. https://doi.org/10.1016/j.patcog.2019.02.023

Malhotra, R., & Malhotra, D. (2003). Evaluating consumer loans using neural networks. *Omega*, *31*(2), 83–96. https://doi.org/10.1016/S0305-0483(03)00016-1

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316. https://doi.org/10.1097/JTO.0b013e3181ec173d

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*, 115–133. https://doi.org/10.1007/BF02478259

Meier, L., Van De Geer, S., & Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*(1), 53–71. https://doi.org/10.1111/j.1467-9868.2007.00627.x

Meng, Z., McCreadie, R., Macdonald, C., & Ounis, I. (2020). Exploring data splitting strategies for the evaluation of recommendation models. In *Proceedings of the 14th acm conference on recommender systems* (pp. 681–686). Association for Computing Machinery. https://doi.org/10.1145/3383313.3418479

Microsoft. (n.d.). Hva er maskinlæring? *Microsoft*. https://azure.microsoft.com/nb-no/resources/cloud-computing-dictionary/what-is-machine-learning-platform

Næss, A. B., Wahlstrøm, R. R., Helland, F. F., & Kjærland, F. (2017). Konkursprediksjon for norske selskaper - en sammenlikning av regresjonsmodeller og maskinlæringsteknikker.

Naik, K. S. (2021). Predicting credit risk for unsecured lending: A machine learning approach. https://arxiv.org/pdf/2110.02206.pdf

Norgesbank. (2022, December 15). *Pengepolitisk rapport med vurdering av finansiell stabilitet 4/2022*. https://www.norges-bank.no/aktuelt/nyheter-og-hendelser/Publikasjoner/Pengepolitisk-rapport-med-vurdering-av-finansiell-stabilitet/2022/ppr-42022/innhold/

Norgesbank. (2023a). Styringsrenten. *Norgesbank*. https://www.norges-bank.no/tema/pengepolitikk/Styringsrenten/

Norgesbank. (2023b, January 19). *Rentebeslutning januar 2023*. https://www.norges-bank.no/tema/pengepolitikk/Rentemoter/2023/januar-2023/?tab=126950

Odegua, R. (2020). Predicting bank loan default with extreme gradient boosting. https://doi.org/10.48550/ARXIV.2002.02011

Ozturkal, B., & Wahlstrøm, R. R. (2022). Explaining mortgage defaults using shap and lasso. https://doi.org/10.2139/ssrn.4212836

Pacelli, V., & Azzollini, M. (2011). An artificial neural network approach for credit risk management. *Journal of Intelligent Learning Systems and Applications*, *3*(2), 103–112. https://doi.org/10.4236/jilsa.2011.32012

Paperswithcode. (n.d.). *Leaky relu*. https://paperswithcode.com/method/leaky-relu

Ranstam, J., & Cook, J. A. (2018). LASSO regression. *British Journal of Surgery*, *105*(10), 1348. https://doi.org/10.1002/bjs.10895

Rosset, S. (2004). Model selection via the auc. In *Proceedings of the twenty-first international conference on machine learning* (p. 89). https://doi.org/10.1145/1015330.1015400

Ruder, S. (2016). An overview of gradient descent optimization algorithms. https://doi.org/10.48550/ARXIV.1609.04747

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature 323*, 533–536. https://doi.org/10.1038/323533a0

Schjetne, S. (2023). Fagforbundet krever økt kjøpekraft til alle. *E24*. https://e24.no/naeringsliv/i/EQBP2j/fagforbundet-krever-oekt-kjoepekraft-til-alle

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Science*, *6*(12), 310–316.

Shrivastav, N. (2020). Confusion matric (tpr,fpr,fnr,tnr), precision, recall, f1-score. *Medium*. https://medium.datadriveninvestor.com/confusion-matric-tpr-fpr-fnr-tnr-precision-recall-f1-score-73efa162a25f

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Statistisk sentralbyrå. (2023). *Befolkning - hvor mange bor det i norge?* https://www.ssb.no/befolkning/faktaside/befolkningen

Sumup. (n.d.). Default - what is default? *Sumup*. https://www.sumup.com/en-gb/invoices/dictionary/default/

Trading-economics. (2023). *Norway inflation rate*. https://tradingeconomics.com/norway/inflation-cpi

Tsai, M.-C., Lin, S.-P., Cheng, C.-C., & Lin, Y.-P. (2009). The consumer loan default predicting model - an application of dea-da and neural network. *Expert Systems with Applications*, *36*(9), 11682–11690. https://doi.org/10.1016/j.eswa.2009.03.009

Tuv, N. (2022). Husholdningene har i snitt fått 18 000 kroner mindre å rutte med i 2022. *Statistisk sentralbyrå*. https://www.ssb.no/inntekt-og-forbruk/inntekt-og-formue/artikler/husholdningene-har-i-snitt-fatt-18-000-kroner-mindre-a-rutte-med-i-2022

UNSW Sydney. (n.d). Backpropagation. https://www.cse.unsw.edu.au/~cs9417ml/MLP2/BackPropagation.html

Visma. (n.d.). Hva er rente? *Visma*. https://www.visma.no/eaccounting/regnskapsordbok/r/rente/

Wahlstrøm, R. R. (2023). Explainable artificial intelligence (xai) for interpreting machine learning methods and their individual predictions. http://dx.doi.org/10.2139/ssrn.4321303

Wang, Q. (2022). Research on the method of predicting consumer financial loan default based on the big data model. *Wireless Communications and Mobile Computing*, *2022*. https://doi.org/10.1155/2022/3786707

Wettre, S. (2021). Rentestatistikk 1820 - 2020. *Statistisk sentralbyrå*. https://www.ssb.no/bank-og-finansmarked/artikler-og-publikasjoner/_attachment/455774?_ts=17a13ed4e08

Wicklin, R. (2020). Roc curves for a binormal sample. *SAS*. https://blogs.sas.com/content/iml/2020/03/09/roc-curve-binormal-sample.html

Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, *162*, 503–513. https://doi.org/10.1016/j.procs.2019.12.017

Zvi Bodie, A. M., Alex Kane. (2021). The sharpe ratio. In *Ise essentials of investments* (Twelfth Edition, p. 125). McGraw-Hill Education.

# A Variable description

Table 7: Description of the final variables

| Variable name | Feature | Description |
|---|---|---|
| CK | Feature 0 | 1 if credit card, 0 otherwise |
| refin | Feature 1 | 1 if refinancing, 0 otherwise |
| fbl | Feature 2 | 1 if consumer loans, 0 otherwise |
| ung | Feature 3 | Type of loan agreement for young debitors |
| vanlig | Feature 4 | Normal type of loan agreement |
| platinum | Feature 5 | Type of loan agreement, differing from the others |
| samb | Feature 6 | 1 if cohabitant, 0 otherwise |
| bolig | Feature 7 | 1 if home owner, 0 otherwise |
| leie | Feature 8 | 1 if tentant, 0 otherwise |
| hjemmeværende | Feature 9 | 1 if homemaker, 0 otherwise |
| uføre | Feature 10 | 1 if disabled, 0 otherwise |
| ansatt | Feature 11 | 1 if employee, 0 otherwise |
| annen_ansettelse | Feature 12 | 1 if other employment, 0 otherwise |
| pensjonert | Feature 13 | 1 if retired, 0 otherwise |
| selv_ansatt | Feature 14 | 1 if self employed, 0 otherwise |
| trygd | Feature 15 | 1 if social security, 0 otherwise |
| student | Feature 16 | 1 if student, 0 otherwise |
| Midlertidig_ansatt | Feature 17 | 1 if temporary employee, 0 otherwise |
| arbeidsløs | Feature 18 | 1 if jobless, 0 otherwise |
| betalings_anmerkning | Feature 19 | Number of payment note |
| maksgrense | Feature 20 | Loan amount |
| alder | Feature 21 | Age of customer |
| barn | Feature 22 | Number of children |
| bil | Feature 23 | Number of cars |
| ikke_bil | Feature 24 | Number of other vehicles |
| net_inntekt | Feature 25 | Net income |
| net_inntekt_måned | Feature 26 | Net income per month |

| | | |
|---|---|---|
| net_inntekt_år_før | Feature 27 | Net income last year |
| net_inntekt_år_før_før | Feature 28 | Net income two years ago |
| formue | Feature 29 | Fortune |
| leieutgifter | Feature 30 | Rental cost |
| leieinntekter | Feature 31 | Rental income |
| studentlån | Feature 32 | Student loans |
| huslån | Feature 33 | Mortage |
| billån | Feature 34 | Car loan |
| total_gjeld | Feature 35 | Total debt |
| total_innskudd | Feature 36 | Total deposit |
| ikke_rentebærende_gjeldsregister | Feature 37 | Loan-amount without interest registered at Gjeldsregisteret |
| rentebærende_gjeldsregister | Feature 38 | Loan-amount with interest registered at Gjeldsregisteret |
| inkasso | | 1 if default, 0 otherwise |

# B Different thresholds neural network

Table 8: Threshold comparison

| Threshold | False negatives | True negatives | Difference (TN - FN) |
|---|---|---|---|
| 0,74 | 13453 | 503 | -12950 |
| 0,70 | 534 | 186 | -348 |
| 0,69 | 485 | 181 | -304 |
| 0,68 | 437 | 177 | -260 |
| 0,67 | 386 | 173 | -213 |
| 0,66 | 357 | 168 | -189 |
| 0,65 | 324 | 164 | -160 |
| 0,64 | 301 | 161 | -140 |
| 0,63 | 278 | 160 | -118 |
| 0,62 | 257 | 155 | -102 |
| 0,61 | 239 | 151 | -88 |
| 0,60 | 227 | 151 | -76 |
| 0,55 | 161 | 134 | -27 |
| 0,54 | 147 | 130 | -17 |
| 0,53 | 136 | 123 | -13 |
| 0,52 | 126 | 121 | -5 |
| 0,51 | 108 | 113 | 5 |
| 0,50 | 90 | 104 | 14 |
| 0,49 | 84 | 99 | 15 |
| 0,48 | 78 | 95 | 17 |
| 0,47 | 76 | 94 | 18 |
| 0,46 | 71 | 91 | 20 |
| 0,45 | 69 | 89 | 20 |
| 0,44 | 66 | 87 | 21 |
| 0,43 | 64 | 84 | 20 |
| 0,42 | 63 | 84 | 21 |
| 0,41 | 61 | 83 | 22 |
| 0,40 | 60 | 82 | 22 |
| 0,39 | 57 | 82 | 25 |
| 0,38 | 55 | 82 | 27 |
| 0,37 | 51 | 82 | 31 |
| 0,36 | 50 | 80 | 30 |
| 0,35 | 50 | 80 | 30 |

| | | | |
|---|---|---|---|
| 0,34 | 47 | 78 | 31 |
| 0,33 | 46 | 78 | 32 |
| 0,32 | 39 | 76 | 37 |
| 0,31 | 33 | 75 | 42 |
| 0,30 | 28 | 73 | 45 |
| 0,29 | 27 | 70 | 43 |
| 0,28 | 23 | 68 | 45 |
| 0,27 | 11 | 52 | 41 |
| 0,26 | 0 | 0 | 0 |

# C Different thresholds logistic regression 70-30 split

Table 9: Threshold comparison

| Threshold | False negatives | True negatives | Difference (TN - FN) |
|---|---|---|---|
| 0,03 | 10.697 | 890 | -9.807 |
| 0,0383 | 8.673 | 836 | -7.837 |
| 0,04 | 8.331 | 824 | -7.507 |
| 0,05 | 6.702 | 755 | -5.947 |
| 0,06 | 5.490 | 696 | -4.794 |
| 0,07 | 4.505 | 641 | -3.864 |
| 0,8 | 3.636 | 571 | -3.065 |
| 0,9 | 2.835 | 484 | -2.351 |
| 0,1 | 2.144 | 405 | -1.739 |
| 0,11 | 1.654 | 347 | -1.307 |
| 0,12 | 1.236 | 270 | -966 |
| 0,13 | 944 | 232 | -712 |
| 0,14 | 724 | 188 | -536 |
| 0,15 | 526 | 142 | -384 |
| 0,16 | 352 | 108 | -244 |
| 0,17 | 246 | 85 | -161 |
| 0,18 | 185 | 62 | -123 |
| 0,19 | 133 | 45 | -88 |
| 0,2 | 79 | 28 | -51 |
| 0,25 | 26 | 11 | -15 |
| 0,30 | 10 | 5 | -5 |
| 0,35 | 6 | 5 | -1 |
| 0,38 | 5 | 4 | -1 |
| 0,39 | 4 | 4 | 0 |
| 0,4 | 4 | 4 | 0 |
| 0,41 | 4 | 4 | 0 |
| 0,42 | 4 | 4 | 0 |
| 0,43 | 4 | 4 | 0 |
| 0,45 | 4 | 4 | 0 |
| 0,46 | 4 | 3 | -1 |
| 0,5 | 4 | 3 | -1 |

# D Different thresholds logistic regression validation

Table 10: Threshold comparison

| Threshold | False negatives | True negatives | Difference (TN - FN) |
|---|---|---|---|
| 0,03 | 5.352 | 438 | -4.914 |
| 0,0383 | 4.311 | 410 | -3.901 |
| 0,04 | 4.138 | 404 | -3.734 |
| 0,05 | 3.299 | 371 | -2.928 |
| 0,06 | 2.678 | 335 | -2.343 |
| 0,07 | 2.204 | 311 | -1.893 |
| 0,08 | 1.778 | 283 | -1.495 |
| 0,09 | 1.388 | 240 | -1.148 |
| 0,1 | 1.034 | 188 | -846 |
| 0,11 | 796 | 162 | -634 |
| 0,12 | 595 | 131 | -464 |
| 0,13 | 474 | 107 | -367 |
| 0,14 | 356 | 88 | -268 |
| 0,15 | 258 | 71 | -187 |
| 0,16 | 171 | 51 | -120 |
| 0,17 | 126 | 34 | -92 |
| 0,18 | 95 | 25 | -70 |
| 0,19 | 59 | 13 | -46 |
| 0,2 | 43 | 9 | -34 |
| 0,25 | 12 | 3 | -9 |
| 0,3 | 6 | 1 | -5 |
| 0,35 | 5 | 1 | -4 |
| 0,4 | 3 | 1 | -2 |
| 0,45 | 3 | 1 | -2 |
| 0,5 | 3 | 1 | -2 |
| 0,6 | 1 | 1 | 0 |

# E AUC- and lasso regression-plot 70-30 split

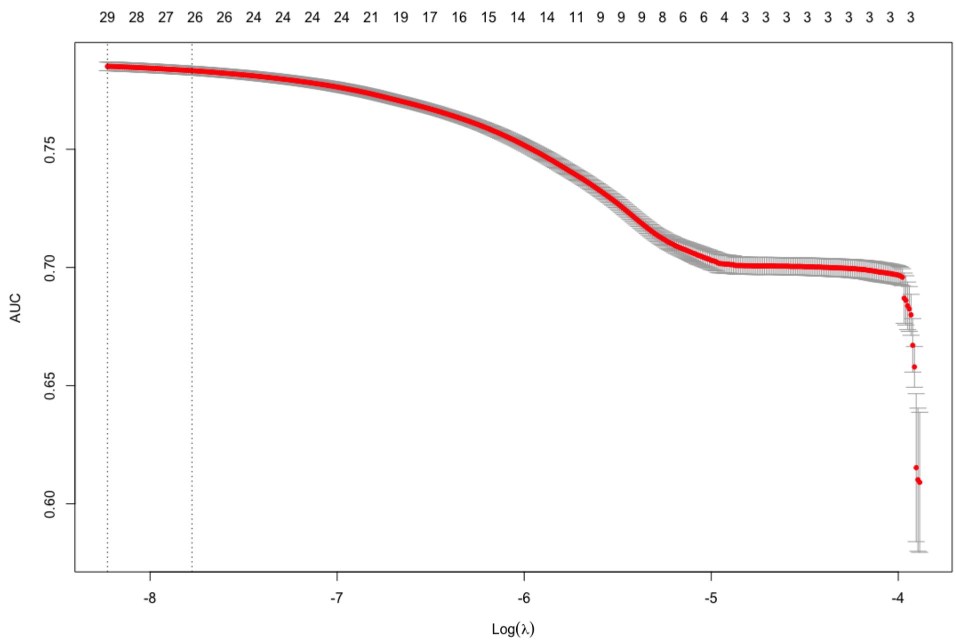


Figure 21: AUC-plot: Shows the diminishing returns of adding new variables. The greater the lambda, the fewer variables

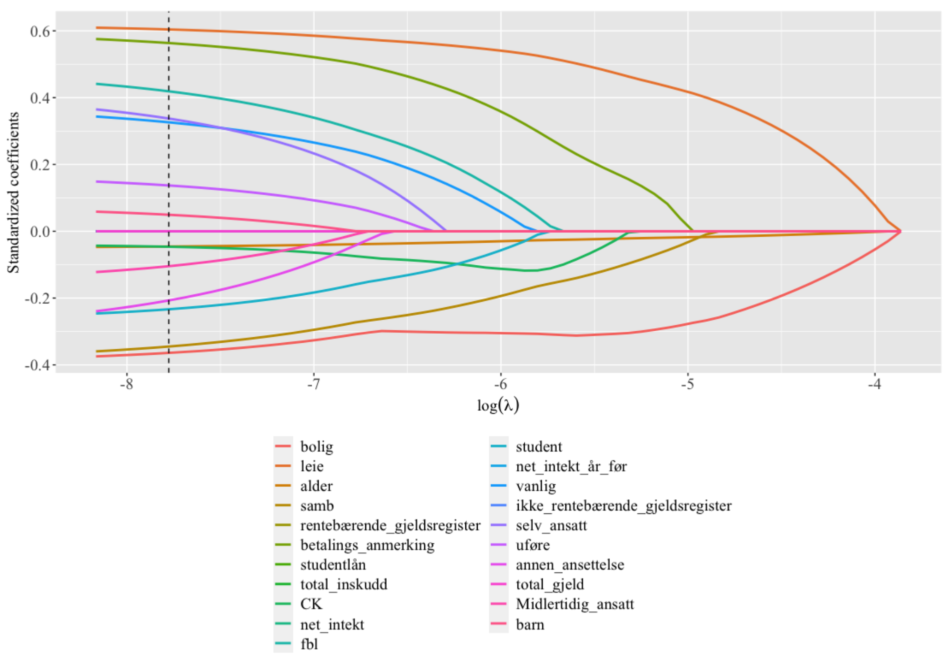


Figure 22: Lasso Regression-plot: Variables that are not zero at the dotted line are kept as features. From the original set of 39, only 21 remain

# F AUC- and lasso regression-plot validation

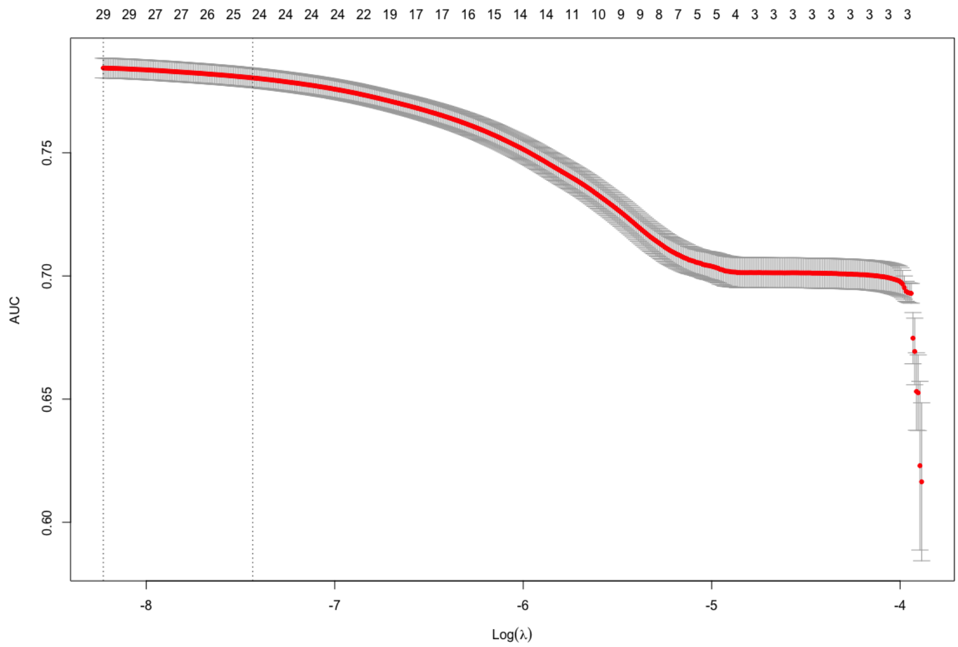


Figure 23: AUC-plot: Shows the diminishing returns of adding new variables. The greater the lambda, the fewer variables

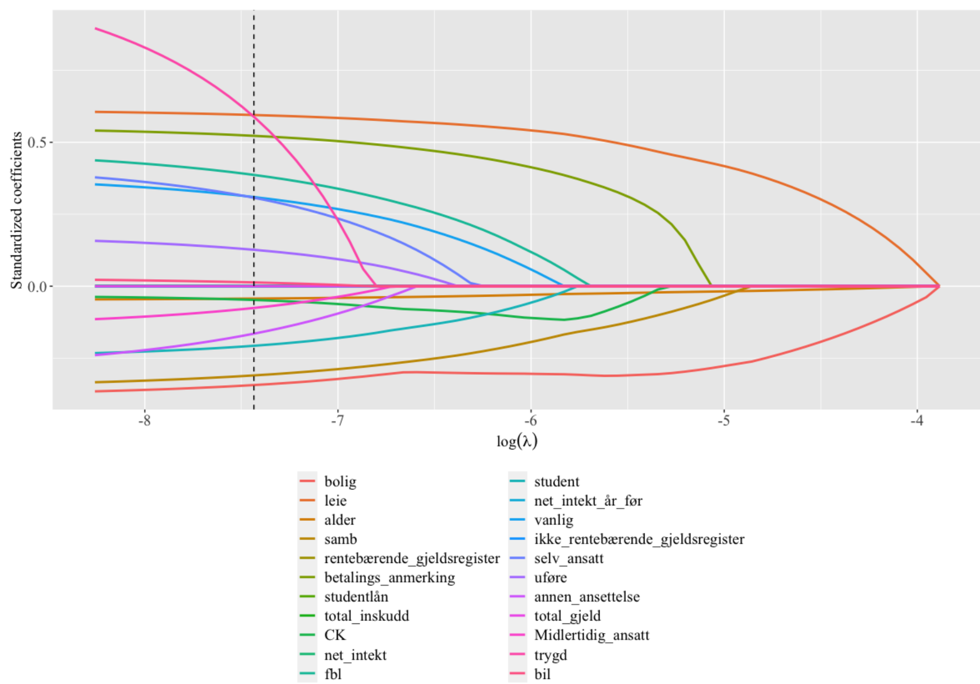


Figure 24: Lasso Regression-plot: Variables that are not zero at the dotted line are kept as features. From the original set of 39, only 22 remain

# G Variable description

Table 11: Estimated LR-coefficients

| Variable name | Coef. 70-30 split | Coef. validation |
|---|---|---|
| (Intercept) | -1,0754 | -1,1738 |
| CK | -0,0429 | -0,0294 |
| fbl | 0,4895 | 0,4829 |
| vanlig | 0,3907 | 0,3877 |
| samb | -0,3679 | -0,3531 |
| bolig | -0,3769 | -0,3831 |
| leie | 0,6201 | 0,6156 |
| uføre | 0,1755 | 0,1835 |
| annen_ansettelse | -0,3171 | -0,3033 |
| selv_ansatt | 0,4216 | 0,4283 |
| trygd | | 1,1136 |
| student | -0,2707 | -0,2489 |
| Midlertidig_ansatt | -0,1576 | -0,1433 |
| betalings_anmerkning 1.0 | 1,9385 | 1,9457 |
| betalings_anmerkning 2.0 | 3,1923 | 2,9355 |
| betalings_anmerkning 3.0 | -9,6669 | -8,199 |
| betalings_anmerkning 5.0 | 1,5446 | 1,6675 |
| betalings_anmerkning 8.0 | 15,913 | 15,2721 |
| betalings_anmerkning 18.0 | 13,6585 | 12,6817 |
| alder | -0,0494 | -0,0479 |
| bil | | 0,0291 |
| barn1 | -0,046 | |
| barn2 | -0,0001 | |
| barn3 | 0,1827 | |
| barn4 | 0,4664 | |
| barn5 | 0,9368 | |
| barn6 | -1,0519 | |
| barn7 | 1,439 | |
| barn8 | 1,2719 | |
| barn9 | -8,6528 | |
| barn10 | -9,0022 | |
| barn11 | -9,1226 | |
| net_inntekt | -5,0853e-07 | -5,3259e-07 |
| net_inntekt_år_før | -4,8326e-07 | -4,9086e-07 |

| | | |
|---|---|---|
| studentlån | -2,9499e-06 | -2,9542e-06 |
| total_gjeld | 1,2621e-07 | 1,268e-07 |
| total_innskudd | -1,1234e-05 | -1,1266e-05 |
| ikke_rentebærende_gjeldsregister | 1,1185e-05 | 1,1121e-05 |
| rentebærende_gjeldsregister | 1,1982e-06 | 1,2065e-06 |

# H Raw information, neural network

Loss function:

```
positive_class_proportion = np.mean(y_train[:,1])
pos_weight = 1/positive_class_proportion
alpha = torch.tensor([1 / 0.0386015, 1 / 0.9613985]).to(device)
class FocalLoss(nn.Module):
def__init__(self, gamma=2, alpha=None, pos_weight=None):
super(FocalLoss, self).__init__()
self.gamma = gamma
self.alpha = alpha
self.pos_weight = pos_weight

def forward(self, inputs, targets):
BCE_loss = F.binary_cross_entropy_with_logits(inputs, targets, reduction='none',
pos_weight=self.pos_weight)
targets = targets.type(torch.float)
pt = torch.exp(-BCE_loss)
if self.alpha is not None:
alpha = self.alpha.to(device)
alpha = alpha[targets.data.view(-1).long()]
focal_loss = ((alpha * (1-pt)**self.gamma * BCE_loss).mean())
else:
focal_loss = (((1-pt)**self.gamma * BCE_loss).mean())*20
return focal_loss
```

```
model = nn.Sequential(
nn.Linear (39, 700),
nn.BatchNorm1d(700),
nn.LeakyReLU(),
nn.Linear(700, 950),
nn.Tanh(),
nn.Linear(950, 1000),
nn.Tanh(),
nn.Dropout(0.5),
nn.Linear(1000,2),
nn.Sigmoid()
```

```
EPOCHS = 1000
```

validation_freq = 1
alpha = torch.tensor([1 / 0.0386015, 1 / 0.9613985]).to(device)
loss = FocalLoss(gamma=2.5, alpha=None)
optimizer = Adam(model.parameters(), lr=0.0001)
scheduler = ExponentialLR(optimizer, gamma=1)
train(model, train_data=train_loader, val_data=val_loader, epochs=EPOCHS, verbose=True, optimizer=optimizer, criterion=loss, validation_freq=validation_freq, scheduler=scheduler, device= device)

Threshold 0.5:
True Negatives: 104
False Positives: 399
False Negatives: 90
True Positives: 13363
Accuracy: 0.9649613069647464
Precision: 0.9710071210579858
Recall: 0.9933100423697316
F1 Score: 0.9820319676648906
ROC AUC: 0.7857379620293552

[Epoch:1000] Loss: 1.441 | Train Accuracy: 0.989% | Val Loss: 1.532 | Val Accuracy: 0.981% | lr: 0.0001 | Time: 21.2 s

Test accuracy: 96.50%
Test average loss: 1.5256

NTNU

Norwegian University of
Science and Technology