Norwegian School of Economics
Bergen, Spring 2020

Unsupervised Machine Learning on Tax Returns

Investigating Unsupervised and Semisupervised Machine Learning Methods to Uncover Anomalous Faulty Tax Returns

Nora Gedde and Ida-Sofie Sandvik

Supervisor: Jonas Andersson

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This thesis was written as a part of our master's degree in Economics and Business Administration, majoring in Business Analytics.

Working with tax research and having the opportunity to employ the skills we have acquired as part of our masters degree has been a rewarding, challenging and memorable experience. We would like to thank the Norwegian Tax Administration for providing the dataset used in this thesis. Furthermore, we would like to thank the IT Department at NHH, in particular Svein, for his patience setting up the server that enabled us to carry out the practical work in this thesis.

Last but not least, we also wish to extend our sincere gratitude to our supervisor, Jonas Andersson, for valuable guidance and helpful advice through the course of this thesis.

Norwegian School of Economics

Bergen, June 2020

Ida-Sofie Sandvik

Nora Gedde

Abstract

In this thesis we investigate whether unsupervised and semisupervised machine learning methods can be applied to detect undiscovered erroneous tax returns, and how the properties of the underlying data affect method performance. To do this we test the two fully unsupervised clustering algorithms K-means and DBSCAN, as well as the two semisupervised approaches One-Class Support Vector Machines and autoencoders. We use a sample of real anonymous tax returns, and evaluate model performance in situations where erroneous returns constitutes a minor percentage of the dataset.

Model performance suggest that our methods are not suited to serve as stand alone solutions for identifying faulty returns, with relatively low F1-scores between 0.1 and 0.15. Considering the resources needed to manually control a submitted tax return this would likely not be economically feasible. The underwhelming performance is especially clear when compared to a supervised boosted trees benchmark. However, a supervised approach would most likely not be able to detect undiscovered errors on its own.

To further study the less supervised methods behaviour we simulate new tax returns based on the original sample, where the differences between normal and faulty tax returns are exaggerated. We find that this improves model performance, but the most exaggerated differences would perhaps not occur in real life. The largest improvement did however stem from changes to the distribution of the tax return features, and this property might be more linked to what can be found in the data population.

If another data sample with these traits exist in the Tax administrations database, these methods would be promising. Even if that is not the case, the possibilities of utilizing the methods in combination with other approaches, to uncover new errors, is by itself worth researching further.

Keywords – Unsupervised Learning, Semisupervised learning, Tax returns, Anomaly detection, K-means clustering, DBSCAN, One-Class SVM, autoencoders

Contents

List of Figures

List of Tables

1 Introduction

Income and wealth tax accounts for approximately 20% of state revenues in Norway (Ministry of Finance, 2019), of which a considerable portion stems from individual taxpayers. Since personal taxes constitutes a significant amount of the National Budget, both deliberate and unintentional personal tax avoidance will have negative impacts on society and the state's welfare offerings if left unchecked. Therefore, active prevention of tax evasion is of great social importance.

The Norwegian Tax Administration is responsible for ensuring that all taxes and fees are paid in full and in due time - providing the financial basis for the welfare state. The Tax Administration use considerable resources on investigating and identifying incorrect tax returns. Since it is not possible to control all tax returns, one of the main problems is selecting returns for manual control. A survey from 2012 indicates that only about 5% of personal tax returns contain errors (Thorsager et al., 2016), meaning that one are in effect looking for needles in a haystack. Historically, this has been done by setting rules and thresholds which capture returns prone to mistakes (Berset et al., 2015). A drawback of this method is that far too many returns are flagged, and manual investigation is needed to select the final set of returns for further control. In the past years, however, the use of machine learning techniques for control selection has gained attention (Smedsvik and Christophersen, 2018).

Machine learning techniques are often grouped by degree of supervision. Briefly explained, supervision refers to whether a machine learning model must be explicitly instructed on what to look for. In the context of detecting incorrect tax returns, a *supervised* machine learning model would need to be presented with labelled examples of correct and faulty returns, and then be trained to best separate the two types from each other. So far, the Tax Administration has seemingly concentrated on supervised learning methods, investigating and grouping taxpayers according to the likelihood of errors in their tax returns. Developing supervised learning models requires preparatory work, since they are dependent on labelled data with known correct and faulty examples in order to predict the likelihood of errors in new filings. Supervised learning models generally perform better the more examples of correct and incorrect behaviour they have been fed (James et al.,

2017), causing this preparatory process to potentially consume a large amount of resources. Furthermore, as the models are generated from examples of past behaviour, they might struggle to detect new and unseen types of errors and methods of tax avoidance.

Techniques with less supervision are often referred to as *unsupervised* learning, and comprise multiple techniques used to group and detect patterns and relationships in data. Unsupervised learning algorithms do this based on properties of the data itself and without the need for an explicit response variable. They are not guided by examples of what to look for. The two approaches can also be mixed in a *semi-supervised* learning approach. These techniques use examples of normal returns without errors during training, but they do not need examples of the anomalous, faulty data (Chandola et al., 2009). This reduces the need for preparatory work and data labelling. It may also broaden the scope of which errors could be detected by the models since the types of anomalies to look for does not have to be decided on and labelled in advance.

We would like to investigate whether unsupervised and semisupervised learning methods can be used to address the challenge of faulty tax returns. Specifically, we focus our attention on methods which can be suited for anomaly detection, that is, methods capable of detecting tax returns that somehow deviates from the majority.

The flexibility of unsupervised and semisupervised approaches should make them suited to detect new types of tax return irregularities that might not be caught by existing models. A key reason to use unsupervised and semisupervised anomaly detection is that tax return errors will presumably change over time. As tax-forms and regulations change, the type of errors which occur might change as well. In addition, individuals who deliberately avoid taxes could evolve and find new methods of evasion in line with the Tax Administrations efforts. Thus, rule based and supervised learning methods that are dependent on characteristics of previous evasion strategies could struggle to detect these individuals. In order to detect these forms of tax irregularities we test a variety of methods aimed at anomaly detection. If the less supervised methods manages to correctly identify faulty tax returns, including them in the manual control selection could benefit the Tax Administration.

Both the access to real tax return data and the literature on unsupervised anomaly detection within the field is limited. Given that anomaly detection requires tailoring to

each field and problem (Chandola et al., 2009), the limited work on the subject means there are few studies that investigate suitable methods, and under which circumstances these generate useful results. Consequently, our focus during method selection will be to test a broad spectrum of available techniques. Even though we have gained access to real tax data, the selection of variables and predictors is limited compared to what is available for the Tax Administration. Because of this, we find that a useful contribution will be to conduct an initial study into which methods may be best suited for anomaly detection in tax type data, and the circumstances where these can be appropriate. Therefore, the work presented in this thesis would provide a proof of concept and starting point for further investigation of the subject.

In this thesis we use data on three different groups of tax returns provided by the Norwegian Tax Administration. The first group is a control group consisting of tax returns with no known errors. The second group contains returns that have been flagged by the Automatic Control Abroad (Automatisk Kontroll Utland, AKU) scheme. During the period we are studying the Tax Administration received information on foreign assets and income of Norwegian tax payers, whilst the same individuals had to self-report this information without knowledge of the automatic control. Individuals flagged in this process have had discrepancies between their self reported and externally reported assets or income. The third group of tax returns consists of filings selected for manual control based on the Tax Authorities internal selection process, and which were found to contain errors resulting in additional tax payments for the individuals in question (referred to as the TIL group). The distinction in detection method for the two last groups warrants the investigation into whether the models distinguish between the different types of anomalies. The two latter groups are used as examples of anomalous returns, whereas the control group represent the majority class of normal observations.

There have been previous studies into what separates and characterize different groups of taxpayers (Andersson and Lillestøl, 2017) and (Andersson et al., 2012). These studies suggest there might be some intrinsic differences between the returns of tax-payers with different types of errors, and that they can to some extent be identified by supervised learning algorithms. It is therefore interesting to see if these suggested differences also can be detected and utilized by less supervised machine learning algorithms, which perhaps

can be less rigid and capture a broader spectrum of anomalies than their supervised counterparts. If this proves to be the case the case, including less supervised learning methods in manual control selection may prove a useful addition for the Tax Administration detection efforts.

Specifically, we investigate four unsupervised and semisupervised methods to detect anomalous returns from the AKU and TIL observations amongst a larger group of control observations. The methods tested includes K-Means clustering, DBSCAN, One-Class Support Vector Machines, and autoencoders. The models' performance is benchmarked against a supervised Boosted Tree model, which was proven useful by Andersson et al. (2012) on this type of data. As the methods we employ are contingent on the properties of the data and different classes of observations, we further investigate some select transformations of the original dataset to see how changes in these properties affect model performance. Overall, this can be summarized into two research questions:

1. How well does the less supervised methods discussed in this thesis perform on detecting anomalous, faulty tax returns.

2. How does changing the properties of the underlying dataset influence the performance of the methods discussed in this thesis.

2 Background

2.1 Background

The first predictive model used by the Tax Administration was developed more than ten years ago (Smedsvik and Christophersen, 2018). It was a model used to predict the willingness to pay for people that owe tax, in order to target the Administration's use of resources. Since then, the adoption of, and focus on, predictive models has increased. The attention has seemingly been on supervised models, building on prior knowledge and finding statistical relationships in order to detect future incidents (Smedsvik and Christophersen, 2018). The Tax Administration possess vast amounts of financial data on Norwegian citizens, and they already allocate substantial resources towards utilizing this data to improve their operations with predictive models. Regardless, there is unused potential which unsupervised and semisupervised learning methods could exploit to reveal new information and relationships. One of the advantages of these methods is the fact that they do not need labelled errors in order to train the models. This means that the methods are not influenced by old bias.

One area in which the Tax Administration focus on employing statistical and predictive models is for selecting tax returns for manual control. The goal is for every individual to pay the correct amount of tax, however, not all tax-returns can be subjected for control after submission as this would demand far to many resources. Because of this, one of the main problems is to select the right individuals to receive extended controls. Due to the disproportional amount of correct compared to faulty tax returns, the choice of which tax returns to manually control becomes an anomaly detection problem.

2.2 Literature Review

The use of machine learning to detect errors and anomalies in tax data has been applied by Tax Administrations both domestically and abroad, including in the United States, Chile, Canada and Australia (González and Velasquez, 2013). This research is often performed in-house, and the results are not necessarily published. The Norwegian Tax administration does, however, report on some of their developments within the field

in their publication "Analysenytt". More recent developments include models which predict the likelihood of errors on reported tax deductions (Thorsager et al., 2016), and models which score the likelihood of errors in bi-monthly VAT statements (Berset et al., 2016). The Norwegian Tax Authorities do not report on the details of implementation, for instance which algorithms are used or how the models are trained. This means that it is not possible to compare their methods with other approaches directly. Even so, the models reportedly have impressive precision. Of the observations sent to control, 71% of observations in the tax deduction case contained errors (Thorsager et al., 2016), while 34-48% of observation in the VAT-statements contained errors (Berset et al., 2016).

A few published studies exist on unsupervised and semisupervised anomaly detection on tax data and other types of financial datasets. Roux et al. (2018) tackles the issue of underreported tax returns in Bogotà Colombia in "Tax Fraud Detection for Under-Reporting Declarations Using an Unsupervised Machine Learning Approach". Apriori auditing of tax returns in order to label data is time consuming and resource demanding, resulting in a lack of historic labelled data in Colombia. In order to address that challenge Roux et al. (2018) studied the performance of unsupervised learning techniques to detect underreporting of the Colombian Urban Delineation tax, consisting of 2,3% of a licensed construction project's budget. They assumed that similar tax declarations should pay similar taxes. By clustering the tax declarations and investigating how anomalies in the tax returns compared to the others in the same group they hoped to identify underreporting. After the analysis 10 declarations were presented to an expert auditor, where five had been marked normal and five was marked as suspicious. None of the five normal samples were flagged by the auditor, indicating that the model did not miss suspicious declarations. Of the five suspicious samples, the auditors flagged one as suspicious by their own methods. This indicates that the model flagged other reports than the expert. The dataset used in this study was fully unsupervised with no available labels, meaning that the tax returns had to go through a manual control afterwards in order to establish whether they actually contained errors or not. A long auditing period of up to six months hindered the ability to check if the flagged projects indeed were underreporting, and thus if the expert or the model had the correct classification. This highlights some of the challenges of unsupervised research, namely that it is difficult to evaluate the models' performance properly.

The use of Unsupervised Machine learning techniques to detect fraud has also been employed by Schreyer et al. (2017) in the article "Detection of Anomalies in Large-Scale Accounting Data using Deep Autoencoder Networks". Schreyer et al. (2017) use Autoencoders to detect fraud in accounting data as an alternative to today's common technique of hand-crafted rules derived from known fraud cases. Even though the rules are fairly successful, Schreyer et al. (2017) points out the limited ability of generalization and thus the fraudsters opportunity to gradually circumvent the rules. The Autoencoder models appeared to provide a highly adaptive anomaly assessment. The models were tested on two real world datasets extracted from SAP ERP instances, consisting of the entire population of journal entries for one fiscal year. They achieved effective models with high $f_1 - scores$ of 0.323 (dataset A) and 0.169 (Dataset B), and less false positives than benchmarked methods (PCA, One-Class SVM, Density based local outlier factor and DBSCAN). They also got qualitative feedback from auditors and forensic accountants on the observed anomalies that underpinned the models capacity of detecting journal entries with high relevance for a follow-up audit.

2.2.1 Characteristics of Norwegian Tax Payers

An extension of the dataset used in this thesis, with the addition of one extra group of tax payers, has previously been featured in a characteristics analysis by Andersson and Lillestøl (2017). The report was a follow up of a SNF-report by Andersson et al. (2012). The additional group of taxpayers consists of individuals who have voluntarily disclosed previously unreported foreign wealth (the FRIV group). In the characteristics analysis Andersson and Lillestøl (2017) identified select features which could serve to distinguish the different groups. The methods used for the analysis were simple categorization, multiple correspondence analysis and classification trees. In this description we focus on the findings for the AKU, TIL and control groups, since these are the groups we have access to in our analysis.

As part of the multiple correspondence analysis, Andersson and Lillestøl (2017) studied three different plots which highlighted different aspects of, and relationships in, the dataset. First, the different variables were plotted against the four most pronounced eigenvalue dimensions. A group variable with the four groups of taxpayers was included as a supplement variable. The plot indicated that the joint return code correlated with

dim 1 from the eigenvalue, the person code correlated with dim 1 and 2, debt and wealth correlated with dim 3, whilst income correlated with dim 4. The group variable did not seem to correlate with any of the four dimensions. However, given that it represent a mean of all the entries, and that it was only included as a supplementary variable, the low correlation did not come as a surprise (Andersson and Lillestøl, 2017).

Andersson and Lillestøl (2017) also made a bi-plot of the variables, where variables plotted close together tend to act together. The control group seemed to be mostly associated with mid-range income, and a centrality code indicating that they lived in more rural places compared to the other groups. The AKU and TIL groups themselves were plotted close together, making them hard to distinguish. Variables associated to AKU and TIL were: no wealth, a single joint return code, being the reference person and living in urban areas. Finally, Andersson and Lillestøl (2017) graphed a representative selection of individuals from each of the four groups. The individuals were shown to have a high degree of overlapping (Andersson and Lillestøl, 2017), suggesting that the groups might be difficult to distinguish from each other.

The last analysis conducted in Andersson and Lillestøl (2017) was classification trees used to study feature importance. When comparing TIL observations to the control group they found that no wealth and no debt, alongside the SKM-groups 10 or 18, were the most important features to distinguish the two groups from each other. With these splits Andersson and Lillestøl (2017) achieved correct classification for 66% of the returns that were classified as TIL observations (in a balanced population with the same number from the TIL and control groups). For AKU observations the tree picked SKM groups 14 or 13 (people with temporary residence end emigrated people), in addition to returns reporting income above 670 000 NOK and low to no debt as important for separating the classes. After the first split with the SKM-codes 83% of the observed returns were in the AKU group, while 61% in the other node was correctly classified as AKU.

3 Data

The data used in this thesis has been provided by the Norwegian Tax Administration to select researchers at the Norwegian School of Economics, and has previously been used for research into characteristics and categorization of different types of tax-payers with special interest for the Tax Authorities (Andersson and Lillestøl, 2017). The dataset contains panel data on personal tax returns from the years 2003 to 2012, with observations split across three different groups of taxpayers.

The first group of tax-returns includes individuals who have been subject to additional taxes due to errors in their filed returns. They have been selected for enhanced control by the Norwegian Tax Authorities, where the control uncovered incorrect information which had lead to a tax advantage, and where the errors were unlikely to be the result of simple mistakes or done in good faith (The Norwegian Tax Administration, 2020).

The second group of tax returns includes individuals who have been flagged with discrepancies in the Automatic Control Abroad (henceforth AKU) scheme. For the period in question, Norwegian tax-payers were required to self report relevant foreign assets and income along with their personal tax returns. At the same time, the Tax Administration received relevant tax information from other OECD countries on Norwegian residents, without this practice being common knowledge to the public. Individuals flagged in this process have failed to appropriately disclose foreign assets or income. Individuals with smaller discrepancies between their tax returns and AKU data are excluded, as this could likely be caused by exchange rate differences. Specifically, tax returns with less than a 10 000 NOK difference have been removed from the dataset. The third and final group consists of a reference group with no known errors (Andersson and Lillestøl, 2017).

Considering that the information in tax returns is highly sensitive personal data, several restrictions have been added to the dataset. First, social security numbers have been replaced by unique serial numbers. Address information have been replaced by a centrality indicator, all figures have been rounded to the nearest 1 000 and any taxpayer which would have been amongst the highest earning in the time period has been removed (Andersson and Lillestøl, 2017).

Furthermore, the number of available features has been reduced from that of a full tax

return. While the standard tax return can contain approximately 600 different attributes (Andersson et al., 2012), the dataset used for this analysis contains 16 variables. These 16 variables were selected as part of the aforementioned characterization analysis, based on previous studies into what characterize individuals who avoid taxes (Andersson and Lillestøl, 2017). In addition to these 16 variables, each observation is associated with an observation class. This is either "AKU" for those flagged in the AKU control, "TIL" for those who have received additional taxes and "CONT" for members of the random sample control group. A limited number of features can be beneficial as it reduce the computational load of performing machine learning analysis, as well as potentially removing non-relevant features which can negatively impact the analysis. However, as the variables were not selected specifically as part of this analysis, a risk is that features which either alone or in combination with others separate the three groups in a material way have been omitted. The discussion around the features will be revisited in section 3.4.

The panel data is not complete for each individual over the entire 2003-2012 period. Some observations are added to, and some are removed from, each of the three groups every year, resulting in a combined increase of 26% in the source data from 2003 to 2012 for all three tax-payer groups. For this thesis we concentrate on the year 2012, as this includes both the most recent observations as well as having the largest number of observations. This limitation center our analysis on within-year differences, which we consider as a sensible scope for our investigation. In addition, Andersson and Lillestøl (2017) found indications of some intrinsic differences between the groups in year 2012, and we are interested in investigating if they are pronounced enough to be identified by less supervised methods as well.

3.1 Variable Description

Of the 16 included explanatory variables, eight are categorical and eight are numerical. The categorical variables are presented in table 3.1. While some require little additional explanation, we would like to comment on others. The variable *SKM-Group* indicates the taxpayer status of the individual in question. *Person Code* indicates if the information regards the reference person themselves, their partner or a child. *Classification Code* refers to which sector the person in question is employed in. *Delivered Tax Return* refers to

whether or not the individual has actively filed their tax return, as opposed to submitting the pre-filed version. *History Code* indicates whether or not the person has had their tax returned manually controlled before, and finally *Joint Return Code* indicates if the tax return is filed independently, distinctively or jointly with another entity or person.

Categorical Variables	
English variable name (Norwegian)	Category
Marital status (Sivilstand)	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*
SKM-Group (SKM-Gruppe)	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*
Person code (Personkode)	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*
Classification code (Klassifikasjonskode)	*1 - Primary Industry* *2 - Agriculture & Industry* *3 - Industry* *4 - Service & Ind, Rural* *5 - Service & Ind, Urban* *6 - Service, Rural* *7 - Service, Urban*
Delivered tax return (Har levert SA)	*J - Yes* *N - No*
History code (Historiekode)	*J - Yes* *N - No*
Joint return code (Samskattkode)	*E - Distinctively* *F - Jointly* *_ - Independently*
Centrality code (Sentralitetskode)	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*

Table 3.1: Overview of the different categorical variables, including category levels. Adapted from Andersson and Lillestøl (2017)

The numerical attributes include the relatively self explanatory *Year of Birth (Fodselsaar)* featuring the tax payers birth year, as well as *Filing Year (Aar)* indicating which year the return in question was filed. In our case this will only be the year 2012. Numerical

attributes also include two variables related to income, two variables related to debt and two variables related to wealth, as summarized in table 3.2

Numerical Variables	
English variable name (Norwegian)	Description
Personal income (person_inntekt_lonn)	What the tax payer himself fill out as income on the tax return
Norwegian income after tax deductions (ib_alm_int_e_serf)	Income on the tax return after tax deductions
Net wealth (netto_form_stat) Domestic net wealth (ib_netto_formue)	Two variables which both describe the net wealth of the tax payer that fiscal year
Domestic Debt(gjeld_kun_ib)	Tax payers domestic debt
Domestic and foreign debt(gjeld_ib_ub)	Tax payers domestic and foreign debt

Table 3.2: Overview of the different numerical variables, including description. Adapted from Andersson and Lillestøl (2017)

3.2 Descriptive Statistics

The successfulness of unsupervised anomaly detection is largely contingent on there being some dimensions in which the faulty observations, that is tax returns from either the AKU or TIL group, tend to differ from those that are normal (Chandola et al., 2009). These differences can be apparent from looking at the data, or they can be more hidden and dependent on the combination of observed attributes. Either way, producing descriptive statistics can help give an overview of the data at hand, and will provide initial indication as to whether the three groups of tax payers differs in any material way. We would like to note that a similar summary and presentation of the dataset has been conducted by Andersson and Lillestøl (2017). However, for the sake of the reader, we include a similar presentation here.

The first categorical variable is marital status, presented in Figure 3.1. For individuals in the AKU group the most common is to be unmarried, and very few are widowed. For

individuals in the CONT and TIL groups being unmarried and married is equally common, and being widowed is the least frequent. Of the categorical this is one of the features where the AKU group stands out compared to the others.

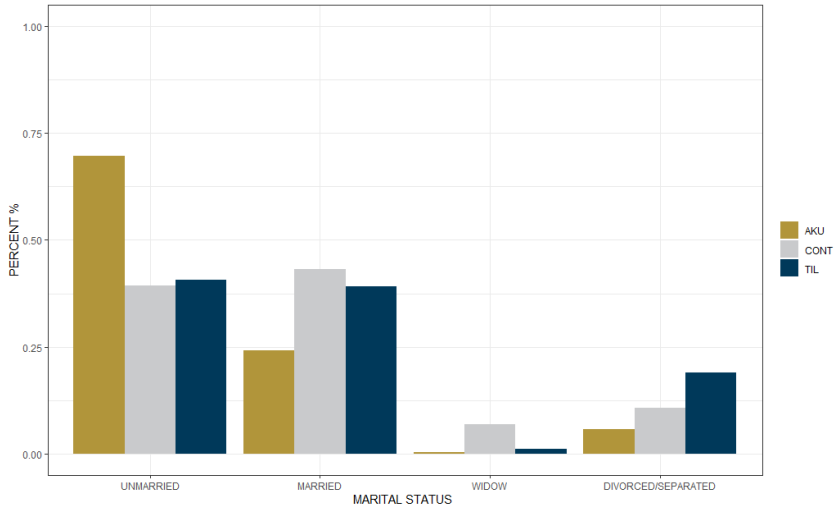

Figure 3.1: Marital status

The SKM-group is the second variable where the AKU group differ from the others. It is most common to be a temporary resident among the AKU individuals, with status as a fully taxable resident second. For individuals in the TIL and CONT groups it is by far most common to be a fully taxable resident. The other taxpayer statuses, as listed in table 3.1, are of negligible size.

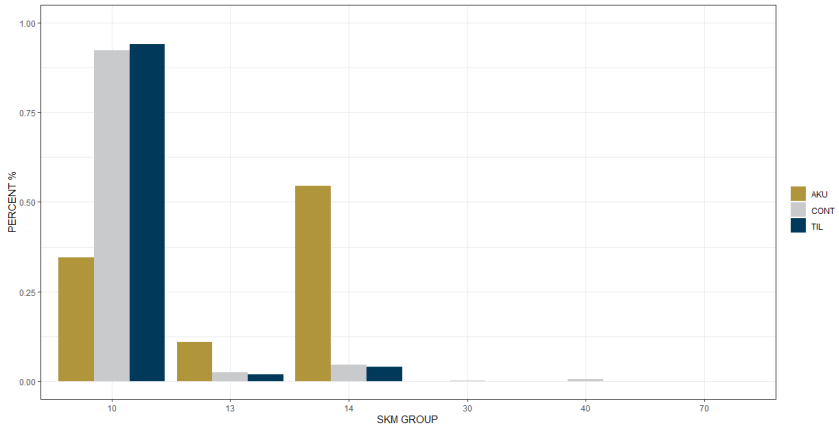

Figure 3.2: SKM-group. 10 - Fully tax liable resident, 13 - Emigrants, 14 - Temporary Resident, 20 - Local tax liable, resident, 30 - Local tax liable, non resident, 40 - Estate, 70 - Diplomat

For the Person Code variable presented in figure 3.3, the distribution across the three categories is similar for all the taxpayer groups, with it being most common to be the

reference person, then the married youngest partner and lastly a child. Being a child or the youngest partner in a marriage is however more common in the control group than the others.

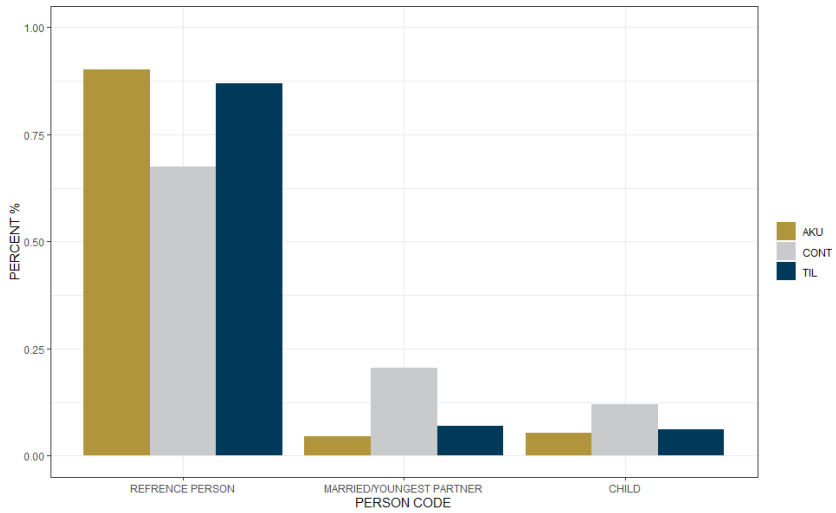

Figure 3.3: Person code

It is more common to actively deliver your tax return when you are in the AKU or TIL group, whilst the opposite is true for the control observation, presented in figure 3.4.

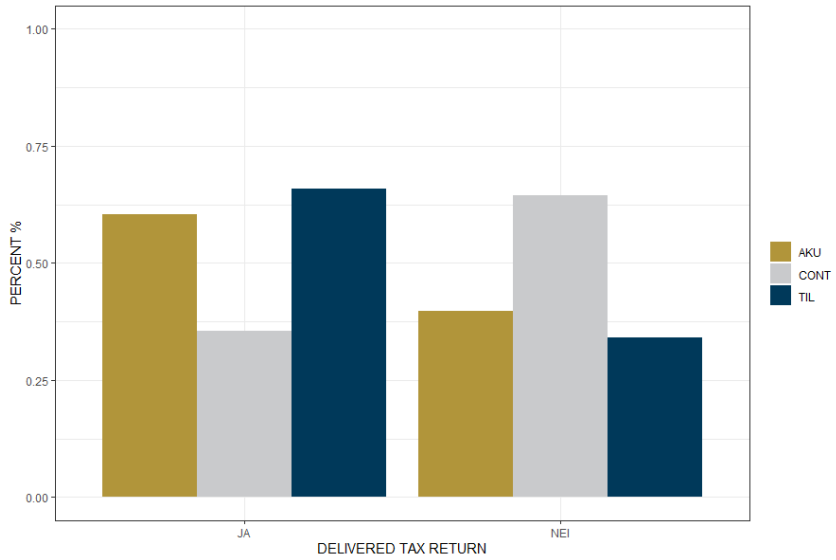

Figure 3.4: Delivered tax return

Having had your tax return manually controlled before is uncommon for all the groups, as presented in figure 3.5. However it is more common for AKU than control observations, and even common for individuals in the TIL group.

It is most common to fill out the tax return independently, and least common to fill it out

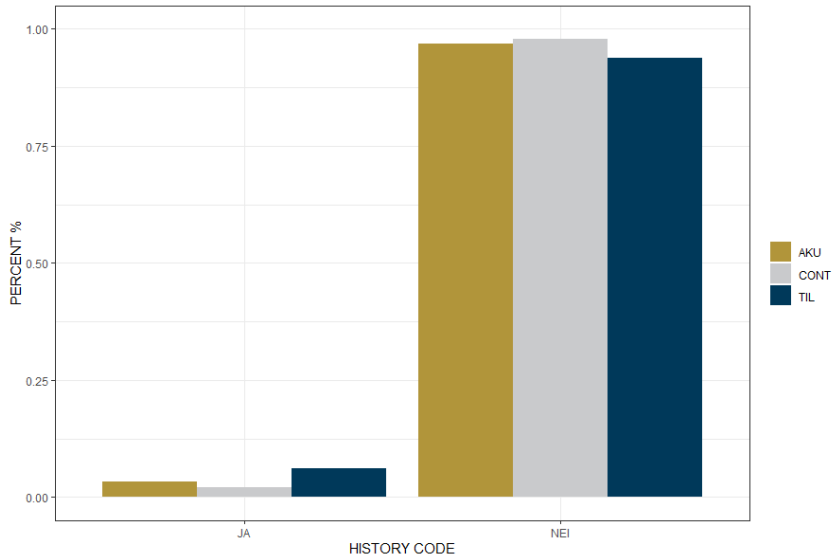

Figure 3.5: History code

jointly across all three groups, presented in figure 3.6.

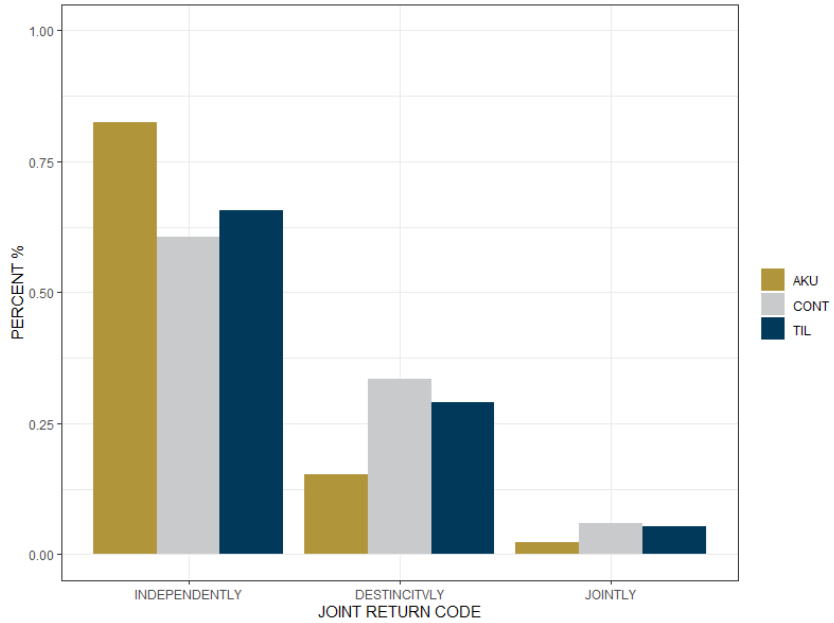

Figure 3.6: Joint Return code

Centrality is also similar across the groups. It is most common to live centrally, and the frequency drops the more rural the place. This feature is strongest for the AKU group, but it holds for the others as well, presented in figure 3.7.

The majority of individuals in all three groups have a registered income between 0 and 500.000 NOK, independent of whether we study the reported personal income, or the calculated income after tax deductions. As shown in Figure 3.8, the distribution of the

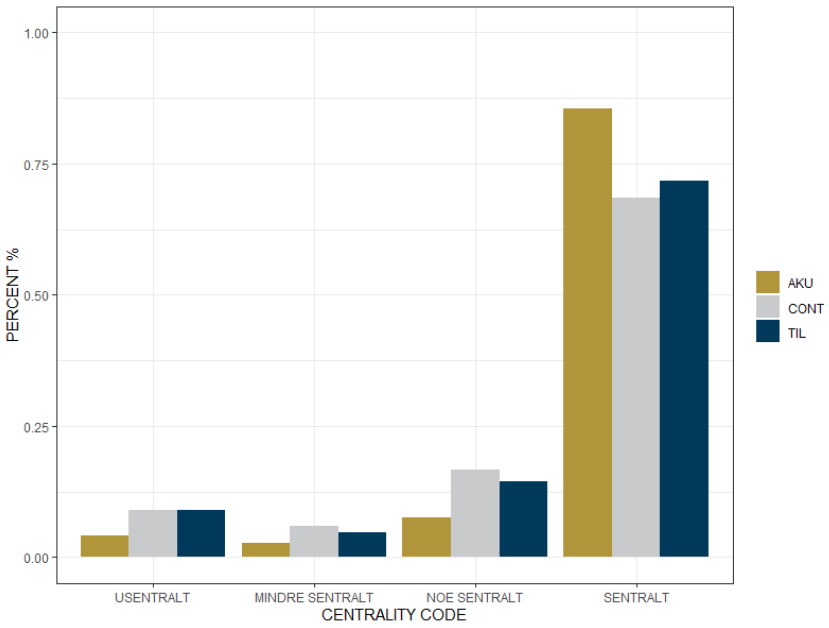

Figure 3.7: Centrality code

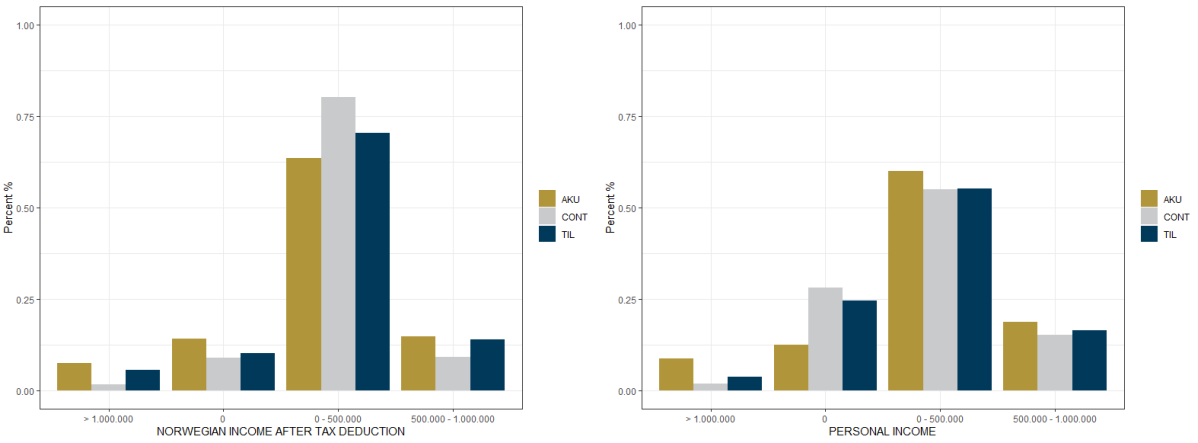

Figure 3.8: The distribution of *Norwegian Income After Tax Deduction* (left) and *Personal Income* (Right)

two different income variables are fairly equal. The distribution of income is also fairly equal across all three groups of taxpayers, although the AKU group has a somewhat higher percentage individuals with an income above NOK 1.000.000.

The two wealth variables have an almost identical distribution, as presented in Figure 3.9. In the TIL group there is a higher percentage of zero wealth individuals, while observations in the AKU-group are more likely to be in the 0 - 500.000 range. Very few individuals have a wealth above NOK 500.000, regardless of group affiliation.

The variables recording debt have more apparent differences, presented in figure 3.10 . All of the individuals in the AKU or TIL groups are registered with zero *Domestic and*

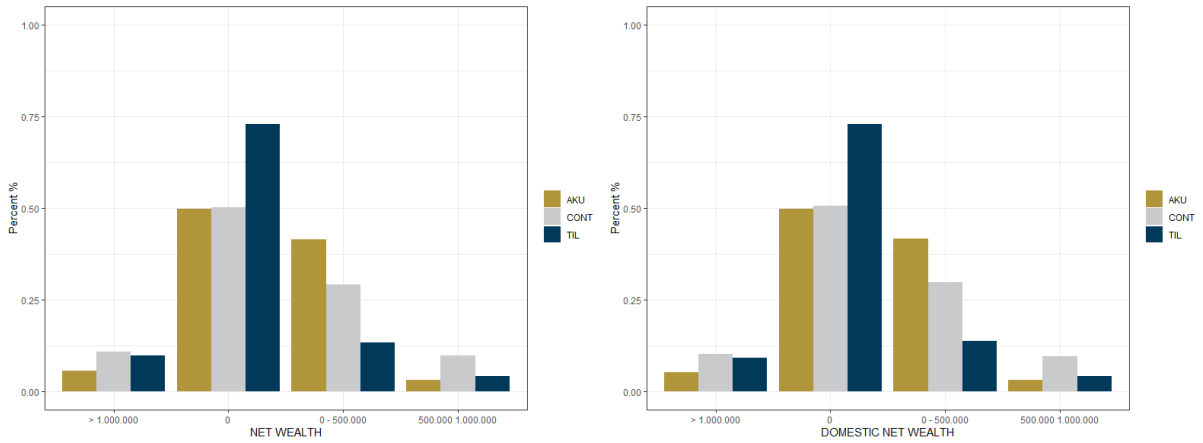

Figure 3.9: The distribution of *Net Wealth* (left) and *Domestic Net Wealth* (right)

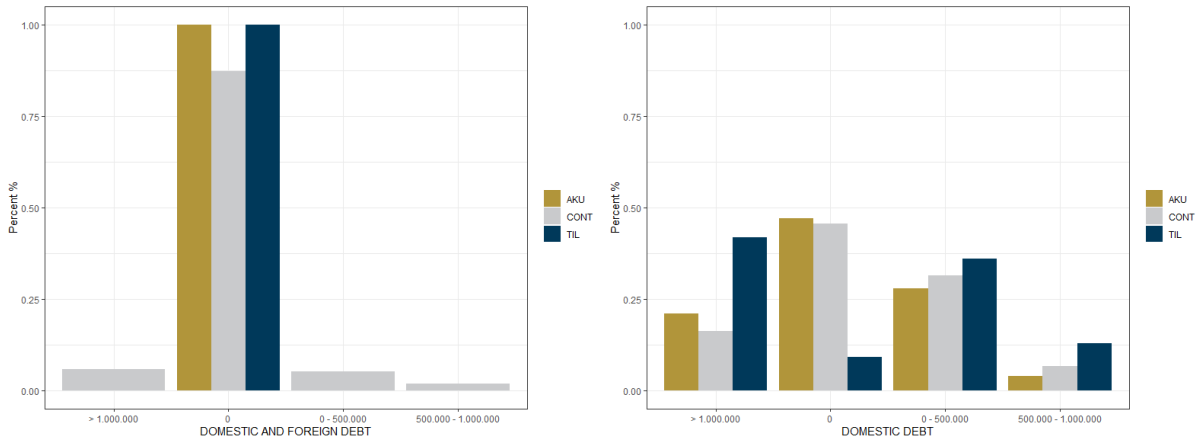

Figure 3.10: The distribution of *Domestic and Foreign Debt* (left) and *Domestic Debt* (right)

Foreign Debt for the year 2012. This is also the case for most of the observations in the control group. Looking at the variable which records domestic debt only, there seem to be some differences between the groups. Most notably are individuals in the TIL group more likely to have debt than the others.

Looking at the descriptive statistics, the three groups seem to follow a similar distribution across the different categories in several of the variables. For example, category 7 is the most common Classification Code in all three taxpayer groups, followed by category 5. This pattern is also apparent for the two income variables, as well as the the Centrality Code, History Code and Joint Return Code. However, for Marital Status it is more common to be unmarried amongst the AKU observations, and for the SKM-Group AKU observations are most likely to be in category 14, while category 10 is by far the most common for TIL and CONT observations. The models should learn to distinguish which

observations are anomalies that differ from the majority in some way. For our analysis the observations flagged as anomalies should be the ones in the AKU and TIL group. Considering that few variables separate the AKU and TIL observations from the control group in a very pronounced way, it will be interesting to study if the less supervised approaches manages to flag the correct observations as anomaly candidates. Although some differences exist, they might not be pronounced enough to distinguish the three groups alone. That being said, there might exist combinations of variables that better separate the different groups of taxpayers from each other, which are not apparent from this analysis.

3.3 Data Preprocessing

All data is subject to some initial preprocessing steps before it is included in the models. First, observations with missing values are removed from the sample. This includes 1142 observations with NA values, 47 observations with NULL values and 3 observations with a *Birth Year* value of 1000. This leaves 21 734 observations in total for the year 2012, split across 2962 AKU observations, 9588 TIL observations and 9184 CONT observations. All models studied in this thesis require numerical values only, while the tax dataset contains a mix of categorical and numerical data. In such cases, common practise is to transform categorical variables into dummy variables, converting each category value to a binary variable with value 0 or 1 depending on whether the category was present in the original observation or not. In addition to requiring numerical values, the methods are sensitive to the spread of the variables. Two incomes that differ with 100 000 NOK could be interpreted as 1000% further apart from each other than two people that have a 100 year age difference. In order to avoid this, the numerical variables are standardized. This is done by subtracting the mean, and dividing with the standard deviation, resulting in values with mean 0 and standard deviation of 1. The scaled values are now going to have the same relative effect on the models.

3.4 Feature Selection

As with all machine learning methods, the less supervised approaches discussed in this thesis can be crucially impacted by the choice of variables used in the models. For labelled

classification, sensible selection strategies includes and excludes variables in ways which contributes positively to class prediction. However, with truly semi- or unsupervised approaches, this is more difficult as there is no clear way to definitively determine the successfulness of any classification made from the models.

There are several reasons as to why limiting the number of features might benefit an unsupervised or semisupervised learning model. First, it can be desirable to remove features that are irrelevant for the problem in question. Simplified models with fewer variables can be preferable both in terms of run time and interpretation, provided the dimensionality reduction does not have an adverse impact on performance. Furthermore, some clustering algorithms risk collapsing with high dimensional data, giving an added benefit from removing less relevant variables (Dy and Brodley, 2004). Secondly, it might be beneficial to remove redundant features which contributes with little new information (Dy and Brodley, 2004). Not only do redundant features contribute to increased dimensionality, but they can also impact clustering results (Dy and Brodley, 2004). As an example, if two variables are highly correlated and refers to almost the same concept, they would with many algorithms give twice the weight of the other attributes to that particular aspect of the data. If this double weight does not reflect some natural aspect of the phenomena we try to model, reducing the redundancy could be considered.

Solorio-Fernández et al. (2019) outlines filtering features based on the properties of the data itself as an approach to unsupervised feature selection. This method can be used independently of which machine learning is applied later (Solorio-Fernández et al., 2019). As we explore a range of semi- and unsupervised algorithms in this thesis, we take a filtering approach to variable selection. This permits using the same dataset across all methods without bias towards the algorithms used in variable selection, which facilitate more straightforward model comparison. It should be noted that as the dataset do have labels these could be utilized to select the variables which yields best anomaly class predictions for each method. However, as we primarily employ semisupervised and unsupervised algorithms, we prefer to limit the use of information which would not be available in a less stylized application. Overall, we take a conservative approach to feature selection.

A visual inspection of the variable distributions in chapter 2.2 suggest some degree of

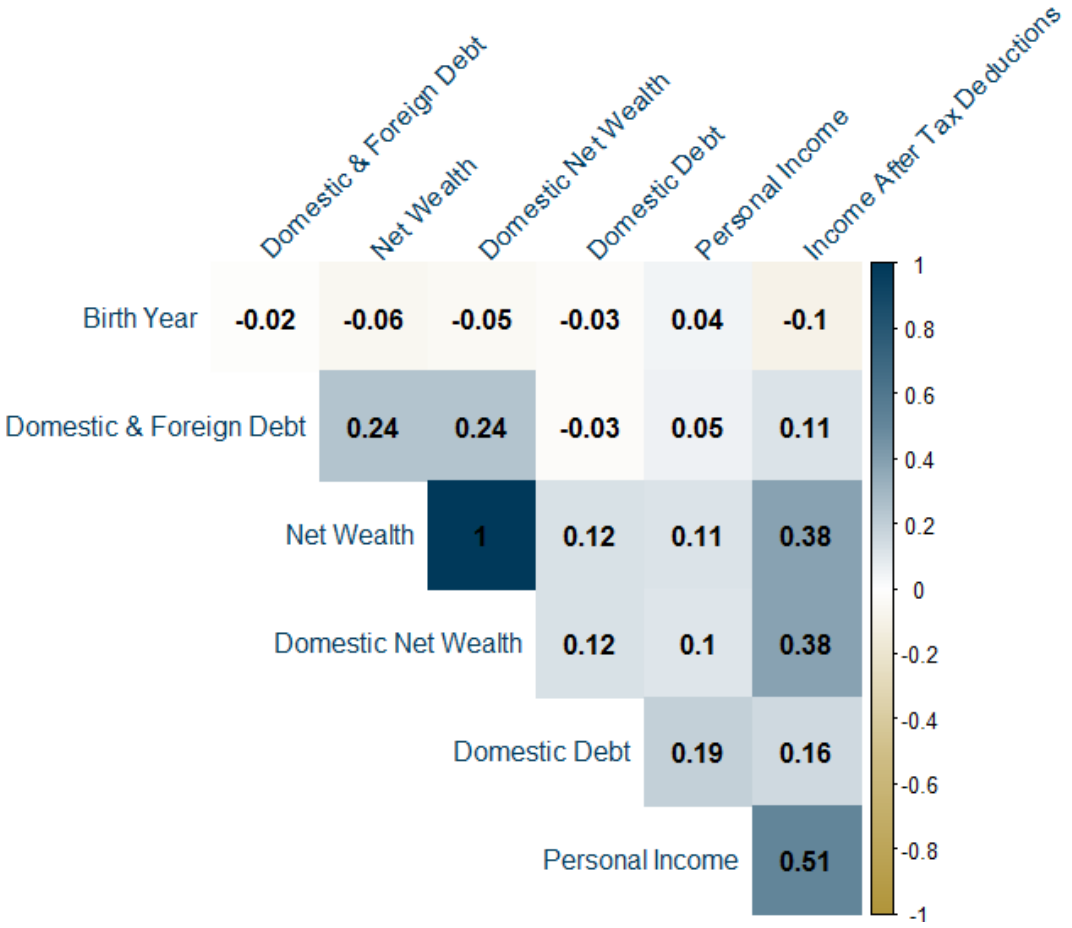

Figure 3.11: Pearson's correlation coefficients for numerical variables

correlation between multiple variables, for example *Net Wealth* and *Domestic Net Wealth*. Numerical variables are evaluated using Pearson correlation for continuous quantitative variables, and categorical variables are evaluated using Cramér's V. Correlation is only calculated for the 2012 observations. We note that almost all variables are considered to have a significant correlation with $p \approx 0$ both when evaluating Pearson's correlation test for quantitative variables and when using the Chi-Square test for categorical variables. This is also the case when the correlation coefficient is of a modest size. Correlation is calculated using a relatively large number of observations, $n = 21\ 734$, and we expect this to cause somewhat exaggerated estimates of significance.

As expected we observe from figure 3.11 that the two *net wealth* variables are close to perfectly correlated, with a correlation coefficient of ≈ 1. Furthermore, Personal Income and Norwegian Income After Tax Deductions are moderately to highly positively correlated with a score of ≈ 0.5. As a last remark, we note that the two net wealth variables are moderately positively correlated with *Norwegian Income After Tax deductions*, with a

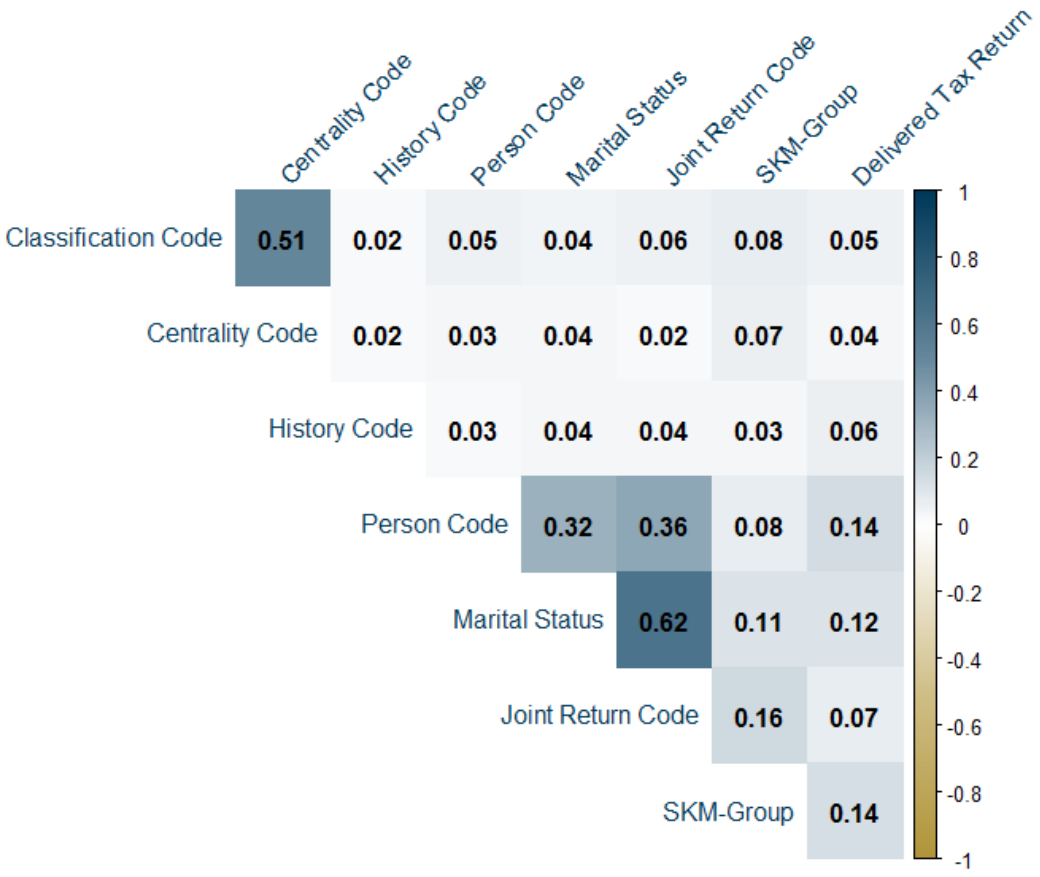

Figure 3.12: Cramers V' coefficients for categorical variables

score of $\approx$ 0.4.

For the categorical variables illustrated in figure 3.12, Marital Status and Joint Return Code have a correlation coefficient of $\approx$ 0.6. Secondly, Classification Code and Centrality Code have a fairly high correlation of $\approx$ 0.5. This is not unexpected as they both include some measure of centrality. Furthermore, the variable pairs Marital Status and Person Code, and Person Code and Joint Return Code have modest correlation coefficients of $\approx$ 0.3

In order to remedy potential issues related to redundancy, and to minimize the concern of double weighted features, we exclude some of the most correlated variables from further analysis. Specifically, we remove the three variables *Domestic Net Wealth*, *Classification Code* and *Joint Return Code*. With this selection most of the available features are intact and used in the models.

In addition, a smaller dataset with fewer features is included and tested as well. This

dataset further removes correlated variables, leaving the seven features Delivered Tax Return, Marital Status, Centrality Code, Domestic Debt, Norwegian Income After Tax Deductions, Net Wealth and finally Birth Year. This reduction also leans on the analysis in chapter 3.2, where most of the selected variables for the small dataset seems to, in some extent, be able to separate the groups from each other. In a practical application of the methods discussed in this in paper, supporting feature selection by using domain knowledge into what specifically separates the groups, might yield better separation of normal an anomalous observations. A discussion surrounding the SKM-group variable can be found in chapter 6.6.

3.5 Use of Labels

Even though this thesis is about unsupervised machine learning, the dataset we use does contain labels. Labels are used in order to validate the results, and get a definitive answer on the performance of the methods for detecting anomalies in these tax returns. As this thesis is primarily aimed as an initial investigation, we are not only interested in how the models perform, but also the considerations which has to be made in a practical application where the anomalies will not be labelled. Because of this, we disregard the labels for a portion of the thesis, and rather use them for in the final validation stages in model selection and evaluation.

4 Methodology

4.1 Machine Learning and Anomaly Detection

In this thesis we explore machine learning and anomaly detection. Common for all machine learning methods is that they use existing observations of a specific phenomena in combination with statistical theory, pattern recognition and inference to produce new information, estimates or predictions(Murphy, 2013).

Chandola et al. (2009) defines anomaly detection as the problem of finding patterns in data that do not conform to expected or normal behaviour. Anomaly detection can be tackled using a wide variety of techniques, employing concepts from fields such as statistics, machine learning, data mining and information theory. The need for such a wide range of techniques stems from the fact that anomaly detection is a non-trivial problem. Both defining what constitutes a normal region of behavior and drawing the line for anomalous observations can be challenging. In addition, and depending on the context of the problem, true anomalous behaviour might not be discernible from the normal region of behaviour, or not be separable from noise in the dataset (Chandola et al., 2009).

Machine learning methods are typically grouped according to degree of supervision. *Supervised* learning methods consist of algorithms which can predict or classify a specific feature of an observation, based on that observation's other characteristics (Hastie et al., 2017). In order to make sensible predictions, supervised learning models must get some idea of which explanatory variable values are associated with different values of the response variable. For the tax data this would be training the algorithm on a set of observations where the group affiliation is presented as a label feature. In other words, one needs to *supervise* the learning by instructing the model of what is correct behaviour. If the training data is representative for the phenomena we are exploring, supervised learning methods can yield highly useful information. It is, however, not always plausible or practical to use supervised learning methods, mainly due to the need for labeled data.

Unsupervised learning methods are often used for finding structures and patterns in data (Hastie et al., 2017), without guidance on what specific features to focus on. Since we do not instruct the model on what we specifically want to find, the learning is considered

unsupervised. The fact that unsupervised learning methods do not need a response variable makes them more accessible to use, and applicable to a wide range of domains and datasets. However, unsupervised methods demand more prior knowledge about the underlying data and domain in order to identify useful outputs from the models, when the possibility for standard validation with labels disappears. Since the Tax Administration are domain experts they have good prerequisites for this preparatory work.

It is also possible to take a *semisupervised* approach. Semisupervised learning algorithms do need some labelled data, but only from the normal, non-anomalous class of observations. They do not need to be provided examples of previous anomalies. This can be preferable to the fully supervised case, as it can be difficult to obtain good examples of all types of anomalies, and the type of anomalies present in the dataset may change over time (Chandola et al., 2009). This makes semisupervised learning suitable for the Norwegian Tax Administration, as they have great access to labeled data, but not exhaustive knowledge on the anomalies that might occur.

Based on the works of Chandola et al. (2009) and other applications of unsupervised anomaly detection on financial data we have selected four anomaly detection approaches to implement on our dataset. These are selected based on performance in other financial applications, in addition to representing a range of different approaches towards anomaly detection, and allows us to tackle the challenge from multiple angles. As we take an exploratory approach to tax data anomaly detection, it is preferable to investigate a range of approaches. The methods we have selected are K-Means Clustering, DBSCAN, One-Class Support Vector Machines and autoencoders. The methods range from pure clustering algorithms with k-means, to clustering for noise with DBSCAN, and transition to separation using hyperplanes with One-Class Support Vector Machines and lastly deep learning with autoencoders.

4.2 K-means Clustering

In general, clustering involves grouping data together so observations within the same cluster are more similar than observations in different clusters. There exists a wide range of clustering algorithms, with K-means being one of the most well-known. K-means clustering has the benefit of being relatively straightforward. This makes it an interesting starting

point for both clustering and anomaly detection analyses. With K-means clustering we partition the observations into a predefined number of clusters, $C = \{C_1, C_2, ..., C_k\}$, with the goal that every observations in each cluster are as similar as possible to each other. Cluster assignment is performed in a way that ensures that every observation belongs to one and only one cluster and that no clusters overlap (James et al., 2017). Formally, K-means clustering seek to minimize the within-cluster variation - the aggregate dissimilarity between each observation in each cluster (James et al., 2017):

$$minimize \sum_{k=1}^{K} W(C_k) \tag{4.1}$$

Dissimilarity is measured by the squared Euclidean distance, giving the following definition of the within-cluster variation:

$$W(C_k) = \frac{1}{|C_k|} \sum_{x_1, x_2 \in C_k} \sum_{j=1}^{n} (x_{1j} - x_{2j})^2 \tag{4.2}$$

Where x_1 and x_2 is a pair of observations assigned to cluster C_k, and $j = \{1, 2, ..., n\}$ represent the different attributes associated with each observation. The Euclidean distance is then calculated as the sum of squared attribute differences. $|C_k|$ is the number of observations assigned to cluster C_k, leading to $W(C_k)$ expressing the average squared Euclidean distance between each pair of observations in cluster k (James et al., 2017).

Finding the global minima when minimizing the within-cluster distance is rarely feasible as the number of possible combinations explodes with the number of clusters and the size of the dataset. Therefore, the K-means clustering algorithm utilizes random initialization of the cluster centers and use this initial position to locate a local minimum instead (James et al., 2017). The process of is done in an iterative fashion:

1. Set a number of clusters k and initialize the clusters with randomly selected observations as the cluster centers.
2. Calculate the squared Euclidean distance from every observation to each cluster centre and assign each observation to the nearest cluster.
3. Recalculate the position of the cluster centres to be the mean value of all observations assigned to the cluster.
4. Repeat steps 2 and 3 until the algorithm converge and there is no change in cluster assignment from one iteration to the next

While this approach gives highly efficient computation, there is no guarantee that the best partition of the dataset is found (James et al., 2017). We therefore follow the recommended strategy of creating multiple clusters with random initialization and selecting the best one, to reduce the risk of randomly selecting a disproportionately bad partition.

4.2.1 Cluster Validation

Given that clustering is an unsupervised learning technique it can be difficult to evaluate the quality of the clusters produced. For K-means, one of the key decisions is determining the number of clusters k. To determine k we use an internal validation criteria known as the *Elbow Method*. The elbow method studies the average within-cluster distance to determine optimal k. The average within-cluster distance (AWD) can be defined as the sum of all within-cluster pairwise distances divided by all the possible within cluster pair combinations:

$$AWD = \sum_{k=1}^{K} \sum_{\substack{x_1 \in C_k, \\ x_2 \in C_k, \\ x_1 < x_2}} d(x_1, x_2) \; / \; \sum_{k=1}^{K} \frac{|C_k|(|C_k| - 1)}{2} \tag{4.3}$$

Where $d(x_1, x_2)$ is the distance method used to generate the cluster. The Elbow method is based on an intuitive principle of trying to determine k in a fashion that best reflects the underlying structure of the data: Assuming that there exists an unknown number of K^* different groupings naturally in the dataset, a k which is smaller than K^* would result in clusters which contains observations from multiple natural groups and thus clusters with a relatively high average within-cluster distance (Hastie et al., 2017). As k increase

towards K^* one would expect the average within-cluster distance to decline sharply as the natural groups becomes increasingly separated by the clusters. As k passes K^*, the "excess" clusters will consist of smaller partitions of the already separated natural groups. This should further reduce the average within-cluster distance, but the reduction from each successive increase in k would be smaller, as the clusters already consists of naturally similar observations (Hastie et al., 2017).

Using this principle, we generate multiple k-means partitions, with k ranging from 1 to a sufficiently large number to hopefully allow us to identify a k close to K^*. Plotting the AWD of each K-means, the elbow principle indicates that a k which corresponds to sharp decline in the AWD curve (an elbow point) would be a sensible choice of k.

4.2.2 Cluster Analysis for Anomaly Detection

The use of K-means clustering for anomaly detection is based on two key assumptions. First, normal observations form large and dense clusters, and anomalous observations form smaller and sparser clusters. Second, normal data should lie closer to the cluster centers, while anomalies should lie towards the outer edges (Chandola et al., 2009). In our approach we have implemented two filters which flag anomaly candidates in the K-means clusters based on the two assumptions. First, all observation that belongs to a cluster containing less than a given percentage of all observations are flagged. This threshold must be set by the user. For example, if the threshold for classifying a cluster as a *small* cluster is 10%, then all observations that belong to cluster with less than 10% of total observations are flagged as anomalies. This approach introduce a new tuning parameter, which requires active consideration of what cluster size should be considered so small that they might contain anomalies. Since cluster sizes varies with each solution the size threshold should be considered in light of the actual cluster structure at hand in combination to prior domain knowledge and assumptions.

Second, all observations which can be considered outliers within their own clusters are flagged as anomalies. This evaluation is based on the Euclidean distance from each observation to their cluster center. Formally, within every cluster an observation is considered an outlier if their distance to the center is more than 1.5 times the interquartile range above the third quartile of all the distances to the center in that cluster. The 1.5

times interquartile range above the third quartile is a common definition of an outlier (Keller, 2014). The outlier evaluation does not introduce any new tuning parameters.

4.3 DBSCAN

Density Based Spatial Clustering of Applications with Noise, or DBSCAN, is another widely used clustering method. As opposed to K-means which assigns every observation to a cluster, density based clustering methods assumes that clusters are continuous dense regions in the data space, which are separated by low density areas (Hahsler et al., 2019). Therefore, observations with many close neighbours are assigned to clusters, while observations that lie far away from others with few neighbours are treated as noise and are not assigned to any cluster. DBSCAN has the added benefit of being able to capture clusters with arbitrary shapes, without assuming convex or hyper-spherical shapes such as k-means. This allows for cluster structures which might be more representative of the actual data (Hahsler et al., 2019).

To cluster using the DBSCAN algorithm we need to specify two parameters which determines how the clusters, data points and noise are defined. The first is neighboring distance, ϵ, and the second is minimum numbers of neighbors, *MinPts*. Data points with more than *MinPts* neighbors (including the observation itself) within the neighboring distance ϵ are defined as *core points*. These observation are considered to be in the densest areas of the data space, and determine the location of the clusters. Every data point within a ϵ distance from a core point is considered to be part of the same cluster. Observations which are assigned to a cluster but are not core points themselves are defined as border points. This is the case when an observation's number of neighbors is smaller than *MinPts*, but the observation is within ϵ of a core point. If an observation is not a core point nor a border point it is defined as noise, and is not assigned to any cluster. This is the case when the observation's number of neighbours is below *MinPts*, and the observation is longer than ϵ from any other core point (Schubert et al., 2017b). The concept and definitions is illustrated in Figure 4.1. ϵ is defined by circles, and *MinPts=4*.

DBSCAN can be defined for a number of different distance measures. In our implementation we use the squared Euclidean distance, as defined for K-means clustering in equation 4.2.

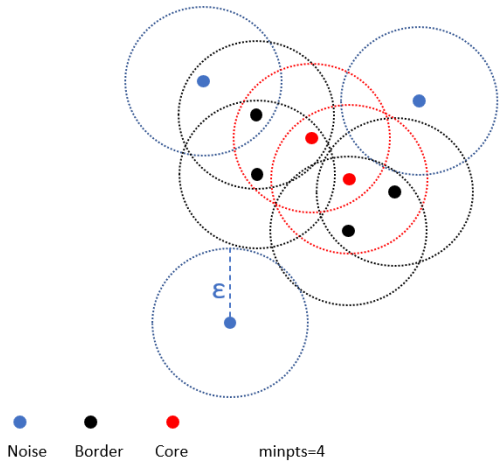

Figure 4.1: DBCSAN with MinPts = 4. ϵ is represented by the circles. The Figure is adapted from (Hahsler et al., 2019)

The core points and noise are deterministic, and the DBSCAN algorithm will only produce *one* solution for these points given the combination of *MinPts* and ϵ provided . Cluster labels and border points are, however, assigned on a first come first serve basis, and may change if the dataset is permuted(Schubert et al., 2017b). The original DBSCAN algorithm assigns border points to the first cluster they appear in, and then jump over that observation in later iterations. This is done because users often desire unique cluster assignment, as well as requiring less memory. In our implementation of DBSCAN, we treat observations with no cluster assignment (the noise) as anomaly candidates, and the instability of boarder point assignment will not affect our analysis. Using this approach, the DBSCAN algorithm offers a rather different perspective than k-means clustering. However it is possible to attack anomaly detection with DBSCAN in the same way as at k-means, looking at the clusters and their properties to find anomalies. If that is the approach you use, the distribution of border points can be crucial, and this is something one should be aware of. Given that we try to find different ways of discovering anomalies and already have implemented k-means we are not going to go further with that approach.

4.4 One-Class Support Vector Machines

Support Vector Machines (SVM) are originally from the field of supervised learning, and is normally used for two-class classification (James et al., 2017). The One-Class SVM (OC-SVM) is an adaptation of this approach, where the goal is to distinguish one minority class from the majority of observations, as opposed to distinguishing two classes from each other. This is done by modelling the region where the majority group of observations reside, and separating observations which falls within or outside this region. Since we only model the region of majority observations, the OC-SVM can operate semisupervised or even fully unsupervised (Brownlee, 2020). In this thesis, we explore the semisupervised approach, where we model normal tax returns, and use the established boundaries to classify new observations as either normal or anomalous.

SVMs use the notion of separating hyperplanes to distinguish two classes from each other. Given n observations in a p-dimensional feature space, the goal is to establish a hyperplane which separates most of the observations from each class at either side of the plane (James et al., 2017). For the OC-SVM, we do not try to distinguish two classes from each other. Instead, we try to model observations so that a hyperplane separates the majority class observations from the origin in the feature space (Schölkopf et al., 2001). In our case the majority class consists of the control group. The OC-SVM can be though of as a normal SVM where all the majority data represent the first class, while the origin becomes the only member of the second class during training (Maglaras and Jiang, 2015).

The general idea is to determine the hyperplane in a way where most observations from the majority class of observations falls on *one* side of this plane, defining the region of normal behaviour. When new observations are introduced, observations which fall inside the boundaries of the plane are defined as normal, while observations which falls outside are classified as anomalous observations (Schölkopf et al., 2001). Our goal is that the AKU and TIL observations fall outside of the hyperplane, while the control observations in the test set fall inside.

There are an infinite number of different planes which could separate the anomalies from the majority. Thus, in order to determine the exact position of the separating hyperplane, an additional requirement is included. The hyperplane in the OC-SVM is determined

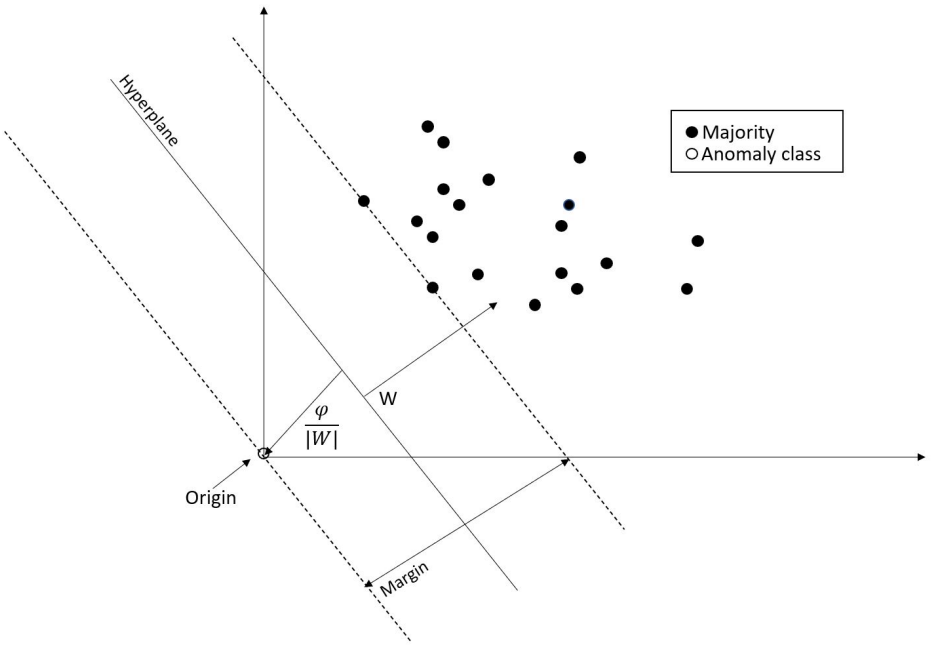

Figure 4.2: The training of an OC-SVM. The Figure is adapted from (Maglaras and Jiang, 2015)

to maximize the margin between the origin in the feature space and the observations from the majority class, as this will define the smallest possible segment as the normal region (Schölkopf et al., 2001). An illustration of the training process for the OC-SVM is presented in figure 4.2.

A strict boundary would likely lead to some degree of overfitting to the training data. Therefore, the OC-SVM introduce the parameter ν which dictates the allowance for observations that cross the margin or the plane itself. ν is a real valued parameter which can take values ranging from 0 to 1, bounding the percentage of observations in the training set which are allowed to be classified as belonging to the minority class (Schölkopf et al., 1999). Depending on the application, some suggest setting ν to the expected ratio of outliers in the data (Brownlee, 2020). A smaller ν corresponds to a harder margin, and vice versa. In the illustration in figure 4.2, the margin is strict, $\nu = 0$. With ν one also introduces a slack variable, ξ_i, which determines how much a particular observation x_i violates the margin or the plane. The full problem is defined by Schölkopf et al. (1999):

$$\min_{w,\rho,\xi_i} \frac{1}{2}||w||^2 + \frac{1}{\nu n}\sum_{i=1}^{N}\xi_i - \rho, \text{ subject to} \tag{4.4}$$

$$(h(x_i)w) \geq \rho - \xi_i \ \forall i, \quad \xi_i \geq 0 \ \forall i \tag{4.5}$$

Where w and ρ define the position of the separating hyperplane, and n denotes the number of observations in the training set. $h(x_i)$ represents a linear or non-linear transformation of observation x_i, that will be discussed in more detail shortly. A detailed explanation of the representation above is outside the scope of this discussion, but we would like to highlight some key aspects. As with the traditional SVMs, only variables which violate the margin contributes to the position of the hyperplane. These observations *support* the separating hyperplane, and are referred to as support vectors. It is possible to deduct that the value of ν corresponds to a upper bound on the number of outliers permitted, as well as a lower bounds on the fraction of support vectors used (Schölkopf et al., 1999). Theoretically, it is possible to solve the minimization problem using the Lagrangian function. Following the Lagrangian solution, class prediction for observation x is given by the decision function $f(x)$ (Schölkopf et al., 1999):

$$f(x) \ = \ sign(\sum_{i=1}^{N} \alpha_i \langle h(x), h(x_i) \rangle - \rho) \tag{4.6}$$

Where x is the new observation to be classified, $\langle h(x), h(x_i) \rangle$ is the inner product between x and the other observations x_i and α_i is a Lagrangian multiplier. One of the factors which makes SVMs applicable to a wide range of problems is that the separating hyperplane need not be determined in the original feature space, where linearly separability is unlikely. Instead, both traditional SVMs and the OC-SVM utilize the principle of enlarging the feature space, where linear boundaries will generally achieve better separation of the training data (Hastie et al., 2017). This is feasible as the computation only requires knowledge of the inner product of each pair of observations, and the transformation of $h(x)$ need not even be specified, but only requires knowledge of the kernel function K (Hastie et al., 2017):

$$K(x_1, x_2) = \langle h(x_1), h(x_2) \rangle \tag{4.7}$$

A kernel is a function that quantifies similarities between two observations. However, the

kernel function can represent similarities which are non-linear, and expand the possible feature space almost indefinitely without having to actually work with the data in the enlarged feature space (James et al., 2017). Popular choices for kernel functions are the Radial Basis kernel and the Neural Network Sigmoid kernel (Hastie et al., 2017):

$$\text{Radial Basis: } \exp(-\gamma \sum_{j=1}^{p}(x_1 j - x_2 j)^2) \tag{4.8}$$

$$\text{Sigmoid: } tanh(\gamma \ \langle x_1, x_2 \rangle \ + \ k) \tag{4.9}$$

Although the observations are separated by a hyperplane in the enlarged feature space formed from the kernel, this does not necessarily translate to a linear decision boundary in the original feature space. Consequently, the introduction of non-linear kernels allows SVMs to express non-linear decision boundaries in all shapes and sizes.

As stated earlier, the OC-SVM can be implemented both as a semisupervised and fully unsupervised method (Brownlee, 2020). In the semisupervised case, the separating hyperplane is defined using tax returns which are known to be normal and without errors. The hyperplane then defines a region of normal behaviour, and ν regulates potential overfitting to the normal data in the training sample. In the fully unsupervised approach the OC-SVM exploit the fact that anomalies are rare occurrences, which do not affect the overall structure of the dataset. However, as one expects the anomalies to be different from the normal observations, these might play a larger role in determining the hyperplane, and in a greater extent become support vectors. In this case, ν might be a bit higher than for the semi-supervised case, as we would expect more anomalies in the training set to lie on the outside of the hyperplane in order to achieve a good class separation, as well as still controlling for overfitting to the normal data. This of course depends on where the anomalous observations lie in the enlarged feature space, and to which extent they differ from the majority class. Given that this is not known in advance the unsupervised approach might add more uncertainty to the output of the model. As it has to be examined whether the anomalies actually end up on the right side of the hyperplane in order for this approach to work, the fully unsupervised might require even more validation afterwards. Considering that the Tax Administration do have access to a fair amount of normally

labelled data from their manual controls, we believe that the semisupervised approach could be an attractive option for them, and use this in further analyses.

4.5 Autoencoders

An autoencoder is a feed forward neural network that attempts to copy its input to its output. If the autoencoder simply copied the input to the output it would not be very useful. Instead, autoencoders are designed in a way which restricts them from perfectly copying the input, forcing the model to prioritize which aspects to copy. This results in a model which learns useful properties of the data, while disregarding less important information (Goodfellow et al., 2016).

The autoencoder consists of two parts, an encoder and a decoder. Most commonly the encoder and decoder are symmetrical architectures consisting of several layers of neurons, each layer followed by a nonlinear function and shared parameters (Schreyer et al., 2017). Both the input and output layer have the same number of nodes as the number of features in the training data (Hawkins et al., 2002). Internally the autoencoder has hidden layers that are compressed in the encoder towards a latent space, before being expanded again in the decoder. This gives the autoencoder a funnel shape, also referred to as "bottleneck" architecture. An example of this structure is shown in figure 4.3. The funnel is what forces the autoencoder to not just copy the input to the output. When it is not possible to transfer all the information through the shrinking layers in the encoder, only the most important characteristics are pushed forward and used in the decoder in order to try and reconstruct the input in the output layer, with as low reconstruction error as possible. During training the autoencoder tries to determine what information to push forward in order to minimise this error.

For training the autoencoder we use a set of N observations, $X = x^1, x^2, .., x^n$. Each observation x^i consists of a vector of K attributes $(x_1^i, x_2^i, .., x_k^i)$. x_j^i denotes the j^{th} attribute of the i^{th} observation. The encoder mapping $f_\theta(\cdot)$ maps an input vector x^i to the compressed representation z^i in the latent space Z (Schreyer et al., 2017). The compressed representation should contain the most salient properties of the data, which in turn enable the decoder to reconstruct the input with as low reconstruction error as possible. The latent representation is mapped back by the decoder $g_\theta(\cdot)$ to a reconstructed

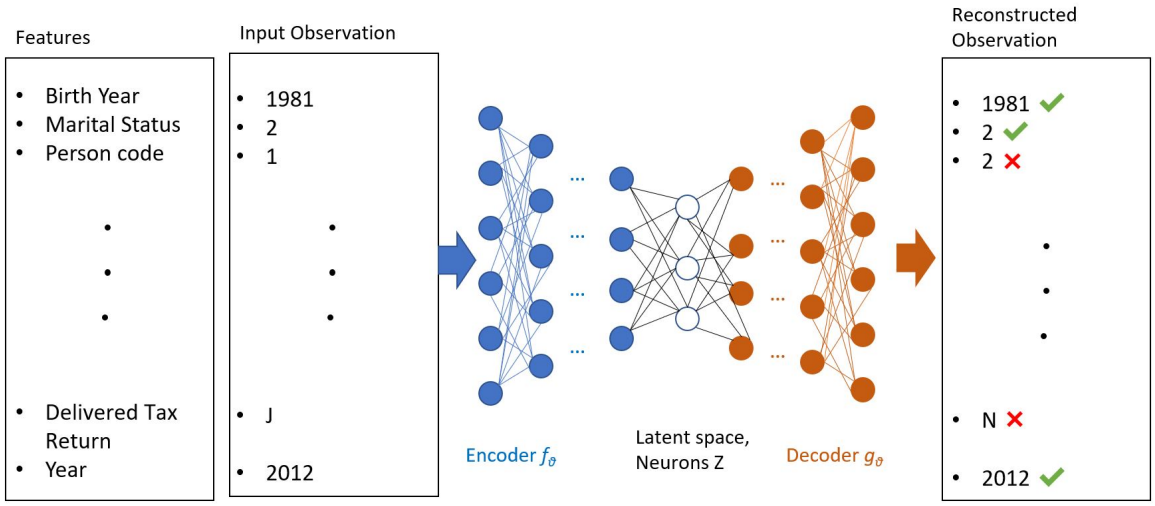

Figure 4.3: Illustration of the autoencoder adapted from Schreyer et al. (2017)

vector $\hat{x}^i$, as presented in figure 4.3 The nonlinear encoder and decoder of an autoencoder with several layers can be defined by:

$$f_\theta^l(\cdot) = \sigma^l(W^l(f_\theta^{l-1}(\cdot) + b^l), \; and \; g_\theta^l(\cdot) = \sigma^{'l}(W^{'l}(g_\theta^{l-1}(\cdot) + d^l) \tag{4.10}$$

Here σ and $\sigma^{'}$ denote the non-linear activation function. θ denote the model parameters $\{W, b, W^{'}, d\}$, $W \in R^{d_x X d_z}$ and $W^{'} \in R^{d_z X d_y}$ are the weight matrices, $b \in R^{d_z}, d \in R^{d_y}$ are offset bias vectors and l denotes the number of hidden layers. For interested readers a closer description of activation functions and neural network architectures can be found in Goodfellow (2016).

In order to minimize the dissimilarity of a given observation x^i and the reconstructed $\hat{x}^i = g_\theta(f_\theta(x^i))$ the autoencoder try to learn a set of optimal model parameters θ^* during training, resulting in $x^i \approx \hat{x}^i$. The training objective can be expressed as:

$$\underset{\theta}{argmin}||X - g_\theta(f_\theta(X))||, \tag{4.11}$$

for every observation X (Schreyer et al., 2017). During the training a loss function is minimized. In our case we minimize the mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x^i - \hat{x}^i) \tag{4.12}$$

As we do not want the autoencoder to simply learn the identity function, the number of neurons in each consecutive hidden layer is reduced giving $R^{d_x} > R^{d_z}$. The restriction forces the model to learn the optimal parameters θ^* for creating a "compressed" version of the most common attribute distributions and their dependencies(Schreyer et al., 2017).

Detecting anomalies with autoencoders can be done in a semisupervised or fully unsupervised way. For the semisupervised approach, the training set contains data we know or presume is only normal data. During training the weights are updated in order to minimize the mean square error, or mean reconstruction error for the training patterns. Properties that are common in the input are more likely to be reproduced well by the trained autoencoders. When we later test the data, the anomalous observations are in theory going to differ from the normal data. The autoencoder should to struggle to reconstruct the anomalous data because it has never seen it before, resulting in a high reconstruction error. This reconstruction error is what is used to measure the outlierness of the observations and find the outliers or anomalies in the data (Hawkins et al., 2002). In the fully unsupervised method we exploit the fact that anomalies occur very rarely in the dataset. Infrequent outliers will be worse reproduced by the autoencoder, resulting in a higher reconstruction error, because the model is going to focus on reconstructing the properties from the majority class i order to minimise the overall reconstruction error.

Rare occurrence of faulty tax returns is the case in the real world as well. Thus, the fully unsupervised approach would be an applicable alternative. However, as with the OC-SVM, we use the semi-supervised approach in our analysis.

4.6 Machine Learning with Mixed Data Types

Many Machine Learning Algorithms are unable to handle mixed data types in their raw form without any form of conversion. This stems from the fact that most algorithms are implemented using similarity or distance measures which are dependent on numerical data only. One example of this is the Euclidean distance, used as the distance measure in this thesis. Common practise is to transform categorical variables into dummy variables.

However, this approach does have some drawbacks. First, it can greatly increase the size of the dataset, causing poor computational performance to the point of infeasibility (Huang, 1998). Considering the limited size of our dataset this is not of great concern, but the dataset does grow from 13 features to 35 after dummifying. Using dummy variables can also affect the interpretation of distances between observations, as the weight of numeric or categorical variables might be misleading or skewed. As an example, consider two observations with different Martial Status. Using dummy encoding in combination with the squared Euclidean distance gives these two observations an added distance of 2 from this feature, since they now differ by 1 in two separate dummy encoded variables. On the other hand, scaled numerical variables with a mean of 0 and standard deviation of 1, will perhaps have a smaller contribution to the Euclidean distance because they often occur in a relatively small interval, and are only counted once. Consequently, using dummy variables introduce some assumptions to the importance of categorical versus numerical variables. This could potentially affect the analysis if the weighting does not represent the underlying relationships in the dataset. Alternative measures such as Gower's similarity coefficient can in such cases be more suited as they better balance between categorical and numerical attributes. Even so, testing the performance of different algorithms using non-standard distance measures would be a topic for a separate thesis, and is not something which will be considered here. Using dummy variables is common practice in most applications, and will suffice for the purpose of our investigation.

4.7 Estimation, Validation and Evaluation

Before we can evaluate the method's performance several considerations have to be made to get a concise evaluation. Some considerations are what performance measures to use and how the methods tuning parameters should be determined. These decisions critically affect outcome and evaluation.

4.7.1 Measuring Model Performance

In order to evaluate and compare the models we need a way to summarize the models' performance. Performance metrics must be considered in terms of what the goal of the analysis is, the structure of the underlying dataset and so forth. In anomaly detection a

positive prediction corresponds to an anomaly, while a negative class prediction represent a normal observation. The errors can then be divided in false positives and false negatives. False positives are objects the model thinks are anomalies, that in fact are normal, and false negatives are objects the model think are normal but in fact are anomalies. The counterparts are true positives and true negatives, that are correctly classified by the model (Müller and Guido, 2016). Measures that use these terms to evaluate performance are dependent on labels in order to identify if a positive or negative prediction is true or false. In a practical application the Tax Administration cannot use these measures until after a manual control is conducted, and they would then probably only prioritize identifying true and false positives. In the thesis we use the measures to evaluate the actual performance of the models, to be able to give an opinion on the fit of the models on our data. But other measures might be used in a true unsupervised approach.

A comprehensive way to evaluate binary classification models is through a confusion matrix. This is a matrix where the rows correspond to the true classes, and columns correspond to the predicted class. The result is a matrix showing the number of samples that belong to the four divisions true positive, true negative, false positive, false negative (Müller and Guido, 2016).

	predicted negative	predicted positive
negative class	TRUE NEGATIVE	FALSE POSITIVE
positive class	FALSE NEGATIVE	TRUE POSITIVE

Since the confusion matrix gives a good overview of the distributions of the different classes and error, but is cumbersome to inspect, it is common to use measures that summarize the information rather than the confusion matrix itself. (Müller and Guido, 2016).

4.7.1.1 Precision, Recall and F-score

Precision, recall and F1-score are ways to summarize the errors of a model that take the difference between a false positive error and false negative error into account. In contrast, an accuracy score would simply divide the number of correctly classified observations by the whole dataset. Precision measures the amount of objects predicted as positive that actually are positive. If the goal is to limit the number of false positives, then precision is a suitable measure. Recall, on the other hand, measure the amount of positive

objects captured by the model. When it is important to find all the positive cases, recall is a good measure. In most cases, however, both precision and recall are important (Müller and Guido, 2016). When determining how to score the models' performance, the consequences of false positives and false negatives must be taken into account. For the Tax Administration, the manual control needed in order to determine if a tax return actually contain errors is a time consuming and costly affair. This translates to a high cost for false positive results. On the other hand, many false negatives can accumulate to substantial missed tax incomes for the government. These considerations must be balanced. A way of combining the precision and recall in a harmonious way is with the F1-score:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.13}$$

The F1-score is a good measure for binary classification on imbalanced datasets as it takes both precision and recall into account. This makes it suited for our tax return dataset. If we were to classify every tax return as normal we would get an accuracy of 95%, which seems very impressive, but that model would not be very helpful. When is comes to understanding and explaining the score, $F_1 - score$ is less intuitive than accuracy, but it is more suited for this kinds of unbalanced cases (Müller and Guido, 2016).

4.7.2 Generalizing on New Data

In this thesis we study different types of machine learning methods that require different considerations when estimating model parameters and validating the results. The two unsupervised clustering methods try to find the natural clusters in the data, which can be used later to detect anomalies. The models are validated based on properties of the clusters' structure, and not the anomaly prediction of each observation,. This means that they use all available data, instead of splitting it into a training and test set. The fact that we validate against the structure of the data, might reduce the risk of overfitting to what we are trying to predict. It is possible to fit new data to these methods, but if a lot of new data is provided, it is likely that new natural clusters occur, as errors and regulations evolve, calling for the method to be run again in order to fully represent this new dataset. When deciding to use a clustering method you should therefore try to find a "recipe" on how to cluster your data, in order to make this re-run of the method as

streamlined as possible, without the need of a full process every time there is a small change in the underlying data.

The two semisupervised methods are intended to be fitted with new data. They should be able to make accurate predictions on new unseen data given that it contains the same characteristics as the training data. We want models that generalize well on new data. It is therefore important to test the models using both a training and test set, and to use a validation method to investigate how well the models predict on seen and unseen data. Particularly in supervised learning, where labels on all classes are included both in the training and testing phase, it is easy to make overly complex models which overfit to the training data and generalize poorly on unseen data (Müller and Guido, 2016). When it comes to less supervised models, however, there is limited access to labels on the different classes. This reduces the ability of making complex models to capture the peculiarities of the different classes, and thus overfitting on them. However, it is still possible to overfit on the normal class of observations during training for the semisupervised methods. This would result in the models not recognising the characteristics of the normal data in the test set, and marking a lot of normal data as anomalies. This can happen if, for example, the hyperplane in OC-SVM follows the training data to closely. If this is the case, many control observations in the test set could fall on the other side of the plane and be classified as anomalies. Another example is if the autoencoder learns the training set to well, and the new control observations get a high reconstruction error. The models could also be too simple and lead to underfitting, resulting in a high false negative rate.

4.7.3 Validation

Statistical models' performance can be sensitive to which data is included in the training and test phases, and estimates done on the basis of one sample can be highly variable (James et al., 2017). Considering this, only selecting and testing one sample can lead to highly uncertain estimates and more uncertainty around model selection and evaluation. In standard supervised classification approaches, a common practice is to implement cross-validation strategies to get more precise and less variable estimates of model performance (James et al., 2017). One of the most widely used approaches is K-folds cross validation where the full dataset is randomly split into k separate parts. One of the k folds is used to test the model, while the remaining *k-1* folds are used to train the model. Model

performance is then calculated as the average performance across the k different folds. Averaging model performance across multiple random samples leads to less variability and more certain estimates (James et al., 2017). In this thesis, we validate the performance from each split using the F1-score. It is, however, possible to use other validation measurse, depending on the data and labels available. A discussion on alternative validation of our methods when labels are not available is presented in chapter 5.1.

The 2012 tax-returns featured in our dataset are split across 9184 control observations, 9588 observations from the TIL group and 2962 observations from the AKU group. Due to this, we need to restructure the data in order for observations from the AKU and TIL groups to in fact be *anomalous*. In our analysis we include observations from both the AKU and TIL groups as anomaly candidates in each model. The restructuring is done by undersampling observations from the AKU and TIL groups. Inspired by Thorsager et al. (2016), we set the percentage of anomalous observations to approximately 5% of the control data. We acknowledge that this portion might not reflect the true rate of anomalous tax returns, but for the purposes of this analysis we consider it a sensible fraction. This, however, leaves very few observations from the AKU and TIL groups. When selecting only a couple of hundred observations, we are more vulnerable to 'lucky' or 'unlucky' draws of anomaly observations which consequently increase variability in estimates of model performance. To accommodate for this, anomaly observation selection is folded. In order to make use of all of the available AKU data, and have the same amount of AKU as TIL in the test sets, we loop through the AKU and TIL observations between 13 and 18 times, depending on the method. The proportion of AKU and TIL is 5% of the control data in the test set. Ideally, the control data is included for training and testing the models should also be varied using a k-folds approach as this would reduce uncertainty even further. Given the number of models and the number of tuning parameters we want to test, it is not feasible for us to have a k-fold on the control data in combination with the undersampling. This would expand the computational load extensively. We prioritize to utilize all of the AKU data, and have a simple train test split on the control data. However, in order to reduce the risk of a strange split of control data, we check that the two data sets have a similar ratio of observations. This evaluation can be found in appendix A3. We do acknowledge that this approach increase variability and uncertainty of model performance, and that selection of model parameters could benefit from being

validated against a larger sample. Still, as our primary goal is to get an indication of suitable approaches we find it a fair compromise considering the data an resources at hand.

5 Analysis

We want to investigate how well the four methods discussed in this thesis are able to detect anomalous, faulty tax returns in our provided data sample. Before evaluating model performance we discuss how the models' tuning parameters are determined. After which we evaluate model performance on the full dataset, discussed in chapter 3.4. A smaller dataset is also tested to see if the methods perform better with fewer, and maybe less redundant variables. We also investigate which observations are flagged as anomalies, in order to get a more detailed image of model performance. Lastly, the unsupervised models are benchmarked against supervised boosted trees, in order to put model performance into perspective. After we have investigated how well our chosen methods perform on the real tax return samples, we study how the two top performing methods fare when the properties of the dataset change. This way we are able to examine how differences in properties effect the methods.

5.1 Parameter Tuning

All methods discussed in chapter four contains one or more tuning parameters which need to be determined. A tuning parameter is a parameter where the value has to be chosen before running the model. Because the choice of values for the tuning parameter is going to have a considerable impact on the end performance of the model, this must be determined carefully. There are different approaches for different methods and types of models. In supervised machine learning it is common to look at how different tuning parameters affect the end predictions of the model, and compare that to the labels in the training and test set. In a real unsupervised learning case one would not have these labels when tuning the models. It would therefore be necessary to use other methods to validate the tuning parameters. This could be done by using different plots or characteristics of the models giving an indication of the fit of the model to the data, supported by use of domain knowledge. When we validate our methods, we first try to determine how the models could be validated using an unsupervised approach, before evaluating model fit using the actual class labels.

5.1.1 Autoencoder

There are a number of tuning parameters and structures of deep neural networks which needs to be decided upon before training. This range from the depth of the network and number of neurons in each layer, to choice of activation function, optimizer, learning rate, batch size and so on. To add to the complexity, the parameters influence each other, meaning how they are combined also impact the end result. This makes it quite infeasible to test every single combination of parameters to find the ultimate values. One approach is to select a range of values for each parameter based on how they have performed on similar tasks, and test a subset of these combinations. We started with the parameters used in Schreyer et al. (2017), and added well known parameters to this. The parameters tested are presented in table A5.5.

Activation function	Learning rate	Batch size
relu	0.1	15
tanh	0.01	30
selu	0.001	45
sigmoid	0.0001	60
		128

Table 5.1: Hyperparamters tested for the Autoencoders

The search for the best combination of tuning parameters was done with random search, meaning that a subsection of the different number of combination was tested. We ended up with an initial selection of 16 models, with different combination of the parameters. The six best performing models were tested with both 2000 ephocs and 3000 ephocs. We also experimented with different depths in autoencoder. The input- and output layers of an autoencoder consist of the same number of neurons as the number of parameters in the dataset, which is 35 after one-hot encoding the variables. The latent space consist of 3 neurons. When deciding on the depth of the autoencoder we explored a number of hidden layers where the number or neurons was doubled from one new layer to another. For example, the simplest structure became 35-4-3-4-35 number of neurons in each layer. The second contained 35-8-4-3-4-8-35, and so on up to 35. This resulted in four models with varying depth.

The metric we used to determine the best performing models is the MSE of the reconstruction error. This gives an indication of how well the decoder reconstruct the

input data. In a real unsupervised application without labels, the loss metrics are the only metrics available. Since MSE is also what is being used to determine the anomalies, this approach runs the risk of overfitting on the data. One way to adjust for this would be to split some of the training data in a validation set, and use this to validate model fit. If the MSE on the validation set was high, and the MSE on the training set was low, it would indicate that the model was overfitting. This approach however requires that you have enough data in the training set. In our case we already have less data than we would hope for for a deep neural network. Because we actually do have labels we choose to use all available data in the training set, and rather use the F1, precision and recall values to find the best model, amongst the models with the lowest MSE.

Because the results from the autoencoder are not fully deterministic we ran the models five times in order to get a stable mean of the results. After evaluating the six best performing models in terms of MSE, we made the final selection on the basis of the models' F1-score. Notably, the best performing model in terms of F1-score was not the model with the lowest MSE loss. This might indicate some degree of overfitting when solely looking at the MSE measure, underscoring the importance of validating the results from the training data with different methods before choosing the end model. The model with the highest F1-score consists of three hidden layers. The architecture of the model is Encoder: 35-16-8-4, latent space: 3, Decoder:4-8-16-35. The model is trained for 2000 ephocs, with activation function tanh, learning rate 0.01 and batch size 30.

In contrast to the OC-SVM and the clustering methods where the amount of anomalies predicted is decided automatically by the models, the number of anomalies the autoencoder pick must be decided upon after the model is run. There are several approaches that can be used in order to decide number of anomalies. In our analysis we know the dataset contains approximately 5% anomalies. As threshold we choose to set $|\text{number of anomalies}| \times 2$. This is further discussed in chapter 5.2.

5.1.2 K-means

Since pure clustering algorithms generally do not produce any predictions, they are often validated based on requirements to their internal structure. Considering that the internal structure of the clusters depend on the entirety of the input data, it is less common to

use cross validation techniques such as k-folds and train-test-validation splits to estimate the fit of the models. The K-means algorithm only has one tuning parameter, k, which is estimated using the internal validation method discussed in chapter 4.2.1. AWD values are calculated for *k's* in the range of 1:60 for a sample of dataset combinations. Looking at the AWD values, we find that k's in the region of 6 to 9 seems to be good choices. We select $k = 9$. A plot of the AWD values can be found in appendix A4.

After k is selected, final K-means models are generated for each of the anomaly fold and control observation combinations. In order to minimize the risk of finding a particularly bad local solution we generate 100 K-means solutions for each fold and select the best one. Anomaly selection is done as described in section 4.2.2.

Finding the optimal threshold parameter for cluster size can be difficult. In our solution, we select a subset of possible threshold values based on how many anomalies are predicted. Where one would need further domain knowledge to separate the different solutions, we select the final, best performing model based on the F1-score. We find that, in our case, an optimal threshold lie around 10%. This means that observations in clusters which contain less than ten per cent of all observations are flagged as anomaly candidates. As discussed in chapter 4.2.2, the number of anomalies flagged from the K-means clusters are a function of two factors: cluster size and distance from the cluster center. The number of anomalies flagged due to the distance from their cluster center is independent of the cluster size threshold parameter, and might vary from solution to solution. In the winning model, an average of 810 observations (8%) are flagged as anomalies. Of these 570 stem from small clusters and the remaining 240 are outliers in larger clusters.

5.1.3 DBSCAN

As the second fully unsupervised clustering method, DBSCAN, is also evaluated based on requirements to the internal cluster structure. As discussed in chapter 4.3 the final cluster structure of the DBSCAN algorithm is decided by parameters *MinPts* and ϵ, which determines how many observations must be within a given distance from each other in order to constitute a cluster. There are multiple evaluation methods and heuristics for selecting *MinPts* and *epsilon*. A standard heuristic is selecting $MinPts = 2 \times dim$, where dim is the dimensionality of the dataset (Schubert et al., 2017a). For our tax return dataset

this is $2 \times 35 = 70$ with one hot encoded categorical variables. After *MinPts* is found, the recommended procedure is to produce a K-nearest neighbours distance plot to get an overview of the average distance from each observation to its *MinPts* nearest neighbours (Schubert et al., 2017a). The idea is that for any given *MinPts* the KNN-distance plot will give an indication of a sensible value of ϵ, where the elbow point of the graph represent an ϵ which combined with *MinPts* work well together in generating clusters.

Using the recommended *MinPts* of 70, the models break down in our case, regardless of ϵ value. Here a break down can mean two things. Either the model only outputs one cluster besides from noise, indicating that it is not capable of finding the underlying structures in the dataset. Secondly the number of anomalies is much higher than the expected number of anomalies. If this happens you have to try a different combinations of *MinPts* and ϵ, and validate the output based on the number of anomalies and clusters in combination with domain knowledge. Because we have labels, we do this validation based on the F1-score. Seeing that the tuning parameter selection heuristics did not produce usable clusters, we test a wider range of *MinPts* and ϵ values, as presented in table 5.2.

MinPts	2, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80
ϵ	0.1, 0.5, 1.0, 1.15, 1.25, 1.4, 1.5, 1.75, 2.0, 3.0, 5.0, 7.0

Table 5.2: Tuning parameters tested for DBSCAN

Using this approach, we find that *MinPts* values ranging from 3-20 in combination with an ϵ of 1.0-1.5 seems to generate useful cluster structures data. The best results is produced with a *MinPts* and ϵ of 10 and 1.0 respectively.

5.1.4 OC-SVM

OC-SVM is tuned searching for parameters in the range presented in 5.3. As with the other models we limit our search to solutions which finds up to 10% anomalies. In a real application without labels the choice between the remaining models would come from a mix of domain knowledge and a validation set passed to the model in order to control for overfitting, as discussed for the autoencoders. If more than a predetermined threshold of observations from the validation set lands on the anomaly side of the hyperplane, this could indicate the the models overfits to the training data. When that threshold is fulfilled, domain knowledge, reasonable parameters and number of anomalies have to dictate model

choice. In this analysis we instead exploit our labels and choose the model with highest F1-score. The two best performing models do in fact have a ν of 0.05, following the suggestion of using a ν equivalent to expected anomalies. The best model combines a ν of 0.05 with a γ of 10 using a Sigmoid Kernel.

Kernel Function	ν	Degree (For Polynomial)	γ (For Radial & Sigmoid)
Linear	0.001	2	0.000001
Polynomial	0.005	3	0.00001
Radial	0.01	4	0.0001
Sigmoid	0.05		0.001
	0.1		0.01
	0.15		0.1
	0.20		1
	0.30		10
	0.40		100
			1000

Table 5.3: Hyperparamters tested for the One Class Support Vector Machine

All tuning parameters used in all models discussed in this analysis can be found in Appendix A5.

5.2 Results

When investigating model performance, we have chosen to include precision and recall, as well as the F1-score which represents a balanced measure of the two. While the F1-score does weigh the consideration between false positives and false negatives, it is affected by the number of anomalies predicted. As an example of this, consider a dataset with 5% anomalies. Randomly predicting 5% of observations to be anomalies would yield an expected F1-score of 0.05. However, selecting 100% of observations and classifying them as anomalies would almost double the expected F1-score to 0.095. As a result of this, the F1-score might provide a slightly skewed representation of model performance in situations where the cost of false positives is particularly high. Knowing this, studying recall and precision, as well as F1-score, provide valuable information which aids model assessment and balancing catching enough errors while not getting to many false positives.

In the case of selecting returns for manual control, the inclusion of many false positives could prove particularly costly, as model predictions must be verified by manual investigation by the Tax Administration. Manual investigation is time consuming and expensive, and

the target should be to uncover as many discrepancies as possible during this process. Considering this, we impose an added constraint on model selection. Given that the dataset only contains approximately 5% anomalous observations, we rule out solutions which produce anomaly predictions on more than 10% of all observations, as mentioned in section 5.1. More than 10% anomalies would cause what we consider to be an unreasonable amount of false positives. This allows us to still consider the F1-score which provides an important balance on precision and recall, whilst limiting the sheer number of false positives. The 5% anomaly share and doubled 10% limit is, to some degree, selected for illustrative purposes. In an actual application of the methods discusses in this thesis, the "double the expected amount of anomaly"-threshold might be altered to better suit the false-positive false-negative trade off. Within the 10% limit, model selection is performed on the basis of F1-score.

The best performing model for each method is displayed in table 5.4. The figures presented represent the average score of each of the folds which are tested.

Model	Precision	Recall	F1 Score	Number of anomalies predicted
K-Means	0.103	0.162	0.119	8.1%
DBSCAN	0.111	0.229	0.149	9.6%
OC-SVM	0.126	0.179	0.147	7.1%
Autoencoder	0.092	0.264	0.138	10.0%

Table 5.4: Comparison on the best performing models using both AKU and TIL observations as anomaly candidates

The models have quite similar performance, with DBSCAN as the best model and K-means in last place. K-means is on par with the other models in terms of precision, but capture very few of the total anomalous observations. The OC-SVM has the highest precision of all models, but just a slightly higher recall than K-means. The low recall is at least partially influenced by the fact that the two models flag fewer observations as anomalies than the DBSCAN and autoencoder. Still, the relatively high precision cause the OC-SVM to have the second highest F1-score, just 0.002 points behind DBSCAN. Not flagging to many anomalies whilst having a relatively high precision could be beneficial considering the high cost of false positives. Even though the DBSCAN has a slightly higher F1-score, the OC-SVM can be considered a better method by the fact that the higher precision makes it more cost-effective compared to the others.

The models perform approximately twice as good as random selection. This cannot be considered very impressive, especially if the goal is for the methods to be a stand alone solution for manual control selection. Only between 16,2% and 26,4% of anomalies in the test data are flagged. In other words, most errors evade detection. Furthermore, if the Tax Administration were to employ these methods directly on this data, they would have to manually control between 9.1 and 8.7 normal tax returns for every return with errors.

So far, it does not seem like the methods are adequately capable of capturing the subtle, but present, differences between the control observations and the AKU and TIL group. One explanation for this might be that there is too much noise in the dataset caused by redundant or correlated variables. We therefore test the same four methods on a smaller dataset with fewer features, as discussed in chapter 3.4. Considering that there exists domain knowledge into what differs between the three groups, narrowing feature selection to variables which shows potential for class distinction might prove beneficial. Still, in a practical application where the specific distinctions between the anomalous and normal observation are less studied this might be more difficult. Specifically, one risks removing variables that are important for error detection because the type of errors which the models could detect is not known beforehand. The results of the analysis on the smaller dataset is presented in table 5.5. The models' tuning parameters have been re-estimated for this dataset, and model selection follows the same guidelines as previously.

Model	Precision	Recall	F1 Score	Number of anomalies predicted
K-Means	0.170	0.116	0.134	3.8 %
DBSCAN	0.124	0.194	0.150	8.4 %
OC-SVM	0.094	0.183	0.124	10.0 %
Autoencoder	0.066	0.187	0.127	10.0 %

Table 5.5: Comparison on the best performing models using both AKU and TIL observations as anomaly candidates on a dataset with a reduced number of features

The effects of reducing the number of features varies from model to model. K-Means sees an increased average F1-score from 0.119 to 0.134. Estimated F1-score also increases marginally for DBSCAN from 0.149 to 0.150. The OC-SVM does not see the same increased performance, with an F1 reduction from 0.147 to 0.124. This is also the case for the Autoencoder, with a reduction in F1 from 0.138 to 0.127. Interestingly, K-means' precision of 0.17 on this dataset is the highest precision of all methods, regardless of whether we compare it to the results from the original or reduced dataset. The increase

in precision is probably related to the fact that only 3.8% of observations are flagged as anomalies. In fact, the high precision is combined with the lowest recall we have seen so far. This illustrates that high precision alone is not enough for a model to be helpful, and highlight the importance of recall when evaluating performance.

The clustering methods seems to benefit from reducing the number of variables, while the OC-SVM and autoencoders report poorer performance. A possible explanation for the improved performance for the clustering methods could be that they use the Euclidean distance to compare the objects. The Euclidean distance might become more unclear with additional features because the differences between objects fades. This explanation is uncertain, given that we have a somewhat limited number of features to begin with, but the influence might still be there. Furthermore, using the Euclidean distance in combination with one-hot encoded categorical features might also explain some of performance increase from reducing the number of variables. As mentioned in section 4.6, there is a risk that the difference in how the numerical and categorical features are evaluated the with Euclidean distance might influence these methods in a negative way, which is alleviated when the number of features is reduced. It seems like the OC-SVM and autoencoder handle the higher dimensional of the dataset in a better way.

The removal of additional variables from the smaller dataset might not have been as well-founded as variable selection for the original dataset. In addition the OC-SVM and autoencoder see a more pronounced decline in performance, than the incline for the clustering methods. More importantly, the incline in clustering performance is still not high enough for us to conclude that it would be economically feasible to use it in a real application given the cost of low precision, and lack of results with such a low recall. Based on this we are going to continue with the larger dataset for the remainder of the thesis.

5.2.1 Investigating Flagged Anomalies

Going forward, we are primarily concerned with the two methods showing the greatest potential, DBSCAN and OC-SVM. These two methods are based on different approaches to detect anomalies, and are both designed to handle unbalanced datasets and anomaly detection. Seeing that the two perform the best out of all the methods, but still not

very satisfactory, we are interested in seeing what types of observations the two flag as anomalies.

In total for all folds DBSCAN flags 2508 observations as anomalies. The OC-SVM flags 1306 observations. The difference between the two is partly caused by the fact that DBSCAN on average flags a higher percentage of observations as anomalous. In addition, DBSCAN makes predictions on a somewhat larger dataset as it does not require a train-test split. For DBSCAN, 40% of all observations flagged as anomalies are control observations, while approximately 30% are observations from the AKU group, and 30% are observations from the TIL group. This indicates that on average, DBSCAN does not prefer one type of anomaly over the other. We would like to emphasize that these results are aggregated for all folds, and that the control observations are present in multiple folds and can be predicted as anomalies several times. Every observation that has been flagged as an anomaly one or more times is counted. Looking at the total percentage of observations, we find that while 40 % of all flagged observations are from the control group, these only constitute 11 % of the total control observations. This means that the same control observations are flagged in multiple iterations. Combined for all folds, the DBSCAN manages to pick up 25% of all AKU observations and 25% of all TIL observations.

For the OC-SVM, 20% of all observations flagged as anomalies are from the control group, while approximately 37% are observations from the AKU group, and 43% are observations from the TIL group. Out of total observations, however, only 6% from the control group are flagged. Again, this means that the same control observations are flagged as anomaly candidates over and over again. Across all folds, the OC-SVM manages to pick up a 16% of all AKU observations combined, and 18% of all TIL observations.

When looking closer at the observations that are flagged, four features stand out. These are Income After Tax Deductibles, Debt, Delivered Tax Return and Marital status. These are the features where the flagged observations differ the most from the full dataset. The plots in figure 5.1 to 5.4 illustrate the anomalies that was flagged by both DBSCAN and OC-SVM.

In the full dataset the largest income group is the one with an income after tax between 0 and 500.000 as presented on the right hand side in figure 5.1. Among the observations flagged as anomalies, the percentage earning more than 500.000 is larger than the

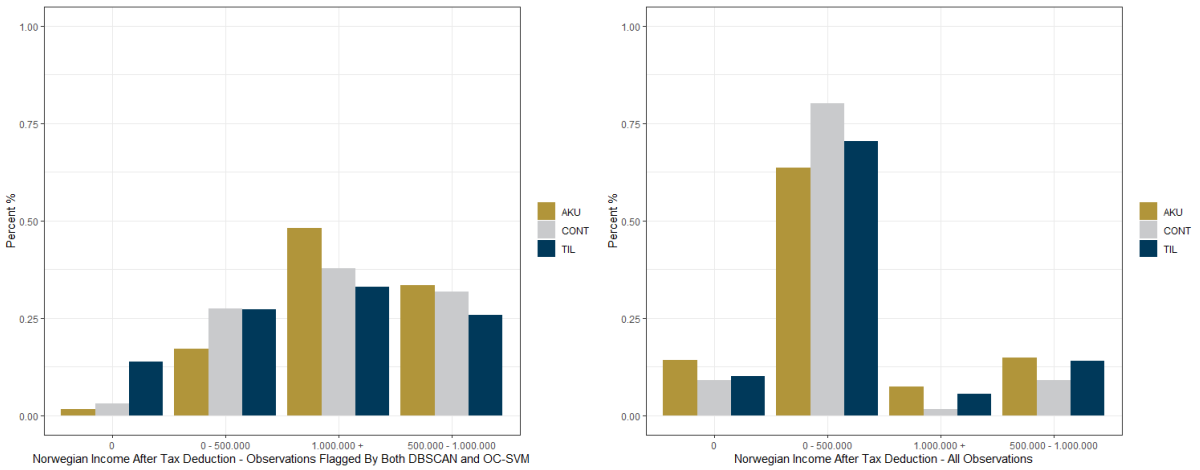

Figure 5.1: Distribution of Income for Observations Flagged as Anomalies *(Left)*, Compared to the Overall Distribution of Income in the Dataset *(Right)*

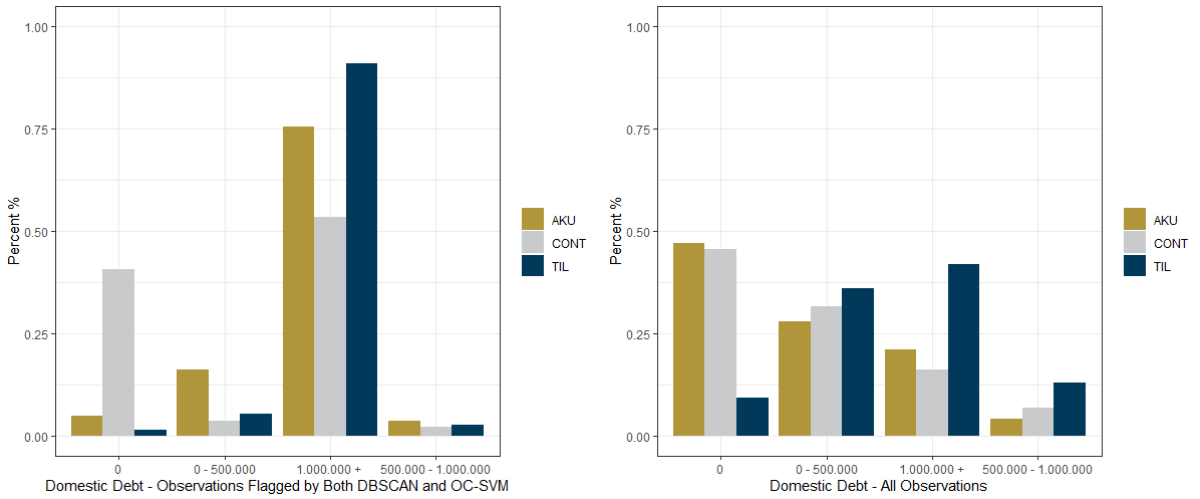

Figure 5.2: Distribution of Domestic Debt for Observations Flagged as Anomalies *(Left)*, Compared to the Overall Distribution in the Dataset *(Right)*

percentage earning between 0 to 500.000, shown on the left side in figure 5.1. The distribution of income is similar across the AKU, TIL and CONT observations which are flagged as anomalies. This indicates that high income is one traits that the models notice and that affects the anomaly prediction. However, the models does not seem to be able to distinguish the control observations with high income from AKU and TIL, even though there are slightly more AKU observations with an income above 1.000.000, than in the two other groups.

Looking at the original distribution of domestic debt, shown in the right panel in figure 5.2, the TIL group stands out as having more debt than the others. The TIL group also has the largest percentage of observations with more than NOK 1 million in debt.

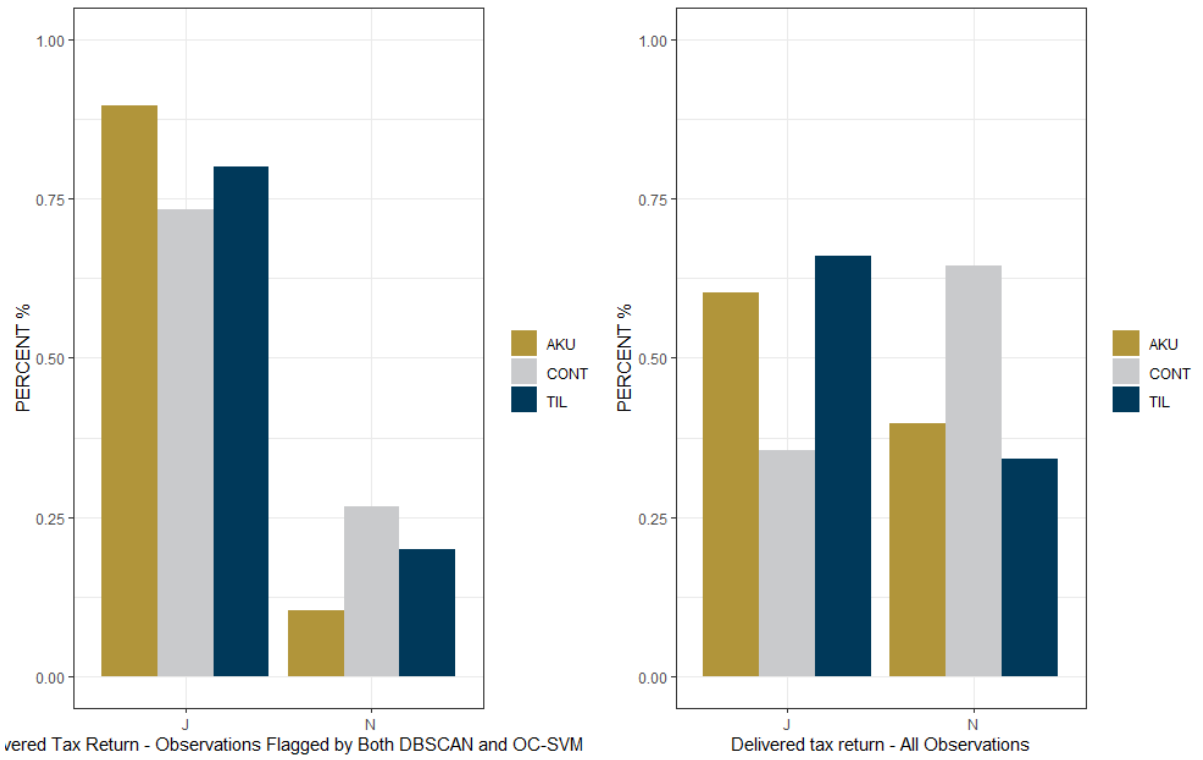

Figure 5.3: Distribution of the Delivered Tax Return variable for Observations Flagged as Anomalies *(Left)*, Compared to the Overall Distribution in the Dataset *(Right)*

From the left plot of the predicted anomalies it can seem like the models were able to capture this, since the most common debt across all three groups is over 1 million. This is particularly true for the AKU and TIL observations, with more than 75% of observations belonging to that category. The control group does however differ somewhat, with almost as many with 0 debt as 1 million. This illustrates that not one feature is important in itself, but that it is the sum of features which plays a part, and the control observations noted here probably have some other features that influenced the anomaly prediction.

The next feature we want to comment on is delivered tax return, illustrated in figure A4.2. Approximately 75% of observations flagged as anomalies had actively delivered their tax return. In the full dataset this proportion is approximately 60% in the AKU and TIL groups, and almost 40% in the control group. This indicates that Delivered Tax Return is an important variable for marking observations as anomalies. However, it is seemingly not combined with other variables by the models in a way where it effectively manages to separate the true erroneous observations from control observations which also have delivered their tax returns.

The last feature we want to comment on is Marital status, presented in figure 5.4. As

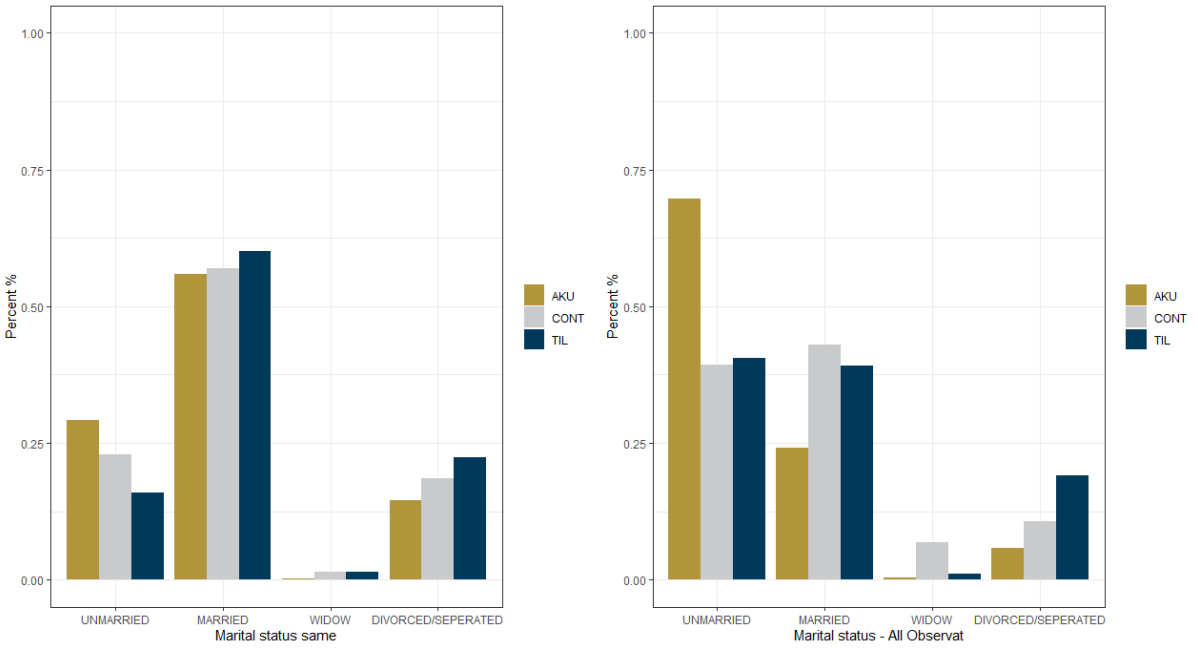

Figure 5.4: Distribution of the marital status variable for Observations Flagged as Anomalies *(Left)*, Compared to the Overall Distribution in the Dataset *(Right)*

mentioned in section 3.2, this is one of the features that most clearly separates AKU observations from the other groups of tax returns, with more observations belonging the the unmarried category. Because of this, it might seem like an important variable to identify AKU observations. Looking at the anomalies that are flagged by DBSCAN and OC-SVM, it does however not come across that they emphasize this distinction. A higher percentage of flagged AKU observations are unmarried, but the most common category for all three groups of flagged observations is being married. This is an example of a category which that seem promising for distinguishing the groups, but that the less supervised methods are not able to detect.

After evaluating performance on our tax return data, we find that none of the models can be said to be very impressive, at least not if the goal is for the methods to be a stand alone solution for manual control selection. The methods do not seem to identify and utilize the underlying differences in the dataset in a way which separates observations from the AKU and TIL groups from the control observations. One hypothesis was that this might be due to noise in the data, thus an additional, smaller dataset was tested with the same methods. Using a smaller dataset did improve performance for the clustering methods, but had a negative impact on the OC-SVM's and autoencoder's performance. Furthermore, reducing the number of features did not improve the clustering based methods' performance to

a point where they could be considered satisfactory. Looking at the observations that were flagged as anomalies by DBSCAN and OC-SVM, we find that the models are able to identify some categories which separates the three groups, such as having delivered your tax return. Still, many control observation that had also delivered their tax return got picked as well. Other variables, such as Marital Status, did not seem to be emphasised in anomaly selection to the extent we perhaps had expected based on the analysis in chapter 3.2, where the most common category in the predicted anomaly observations was married and not unmarried

That being said, it is difficult to fully evaluate model performance on their own, without comparison to other possible approaches. In order to better evaluate the performance of the less supervised approaches discussed in this thesis, we test the same sample of tax returns with a supervised method benchmark. This gives a better understanding of how well the performance can be when the models know what to look for, and where the models might be better at using the important subtitles in the data to distinguish the groups.

5.3 Boosted Trees Benchmark

As mentioned in chapter 2.2.1, a version of this dataset has previously been examined in two studies where supervised methods were used as part of characteristic analyses. Andersson et al. (2012) found that boosted trees made good predictions on this type of data. This analysis was based on the same type of tax returns, but the "anomaly group" was different, consisting of tax returns from individuals who had voluntarily disclosed foreign assets. From Andersson and Lillestøl (2017) and Andersson et al. (2012) we observe that this group differs from the control observations to a greater extent than the AKU and TIL groups do. Andersson et al. (2012) also used additional features, which may affect model performance. The datasets are still similar enough that the applicability of boosted trees should hold for us as well. We therefore use boosted trees as a benchmark for the unsupervised methods discussed in this analysis, in order to compare and discuss model performance. All the trees have been through a tuning process to insure good performance.

The trees have been tested under different circumstances in order to better compare

the results to the unsupervised models. First, the boosted tree model was computed on a balanced dataset to see how well boosted trees perform with the best possible conditions. This resulted in an F1-score of 0.778, precision of 0.763 and recall of 0.795, clearly outperforming the unsupervised models. That being said, it is not surprising that the trees do better. First, they have labels during training, meaning they have the ability to learn what to look for to efficiently separate the classes. Second, in this scenario the three types tax returns are balanced, meaning that the model get an equal amount of data to train on and does not have to balance how much weight to put on each class.

Next, the boosted trees are tested on an unbalanced dataset. Unbalanced classes are, after all, most often the case in the real world, and that added twist would level the ground a bit more between the trees and our methods. The package we use to make the boosted trees is called Caret. This has a sampling parameter that can be set to "smote", which rebalances the dataset when executing the method. Smote weighs two methods that can be used to rebalance data. The first is downsizing the dataset by reducing the control group. The second is up-scaling the AKU and TIL group by resampling those observations. This helps the model make the right predictions also in an unbalanced case. The reason to resample inside the method, instead of doing it outside as above, is that the first method might lead to overly optimistic estimates of the performance. In addition, it introduce more model uncertainty as you do not know if the models could handle the unbalanced real world (Kuhn, 2019). With this approach the model achieved a F1-score of 0.34, precision of 0.269 and recall of 0.466.

Finally, we emulate a scenario which is even more similar to what the less supervised methods would be used for. Now the difference in how the AKU and TIL observations were discovered is important when designing the train and test set. The TIL errors was discovered based on the Tax Administrations own controls. This means that they would have prior labels on the same groups of error that they could use in their supervised learning models or to make rule based controls. The AKU group, however, was discovered due to the cooperation with OECD. This entails that these errors might have gone undiscovered, were it not for the information sharing. AKU is also a group the Tax administration would not have any prior labels on. In the last model the AKU group is labelled as normal data in the training set, because in this scenario the Tax administration

does not have labels on them. Only the TIL group is labelled as anomalous in the training set. In the test set however both AKU and TIL are labelled as anomalies, since both groups actually are faulty regardless of the label information Tax administration posses. Now only the control group is labelled as normal in the test set. This resulted in a F1-score of 0.188, precision of 0.137 and recall of 0.302.

In addition to looking at the performance measures, it is interesting to see what proportion of the three groups is captured in the three scenarios. It is especially interesting to see how many AKU observations the model is able to capture in the last scenario compared to the first. An overview of the proportion of groups among the observations flagged as anomalies is presented in table 5.6

Group	Scenario 1	Scenario 2	Scenario 3
AKU	43%	18%	4%
Control	21%	73%	86%
TIL	36%	9%	10%

Table 5.6: Proportion of the three groups predicted as anomalies

As presented in table 5.6 the tree-based models seem to be better at capturing AKU than TIL when both labels are available, especially when the dataset provided to the model is unbalanced (Scenario 2). When the AKU label is removed from the training set, the model has problems with capturing the AKU group. This is exactly what to expect from a supervised model. Even though the third scenario is rather stylized and perhaps far from reality, it illustrate the difficulties supervised models can have on catching "new" anomalies in the data. In scenario 1 where everything is balanced and all labels are provided, the trees show promising performance. As we make the scenarios more realistic in terms of class balancing and available labels, the performance drop rapidly. These scenarios might not be entirely accurate to a real application, but point to some of the advantages of less supervised methods compared to the fully supervised.

Even after the AKU group is labelled as belonging to the normal group, the supervised method had a higher performance than the unsupervised. We never suspected that the less supervised methods would be able to measure up to the performance of the supervised on this data sample where the labels are available. However, we had thought that the performance would be more similar in scenario 3 for the supervised, compared to the less supervised methods. One reason for the supervised continued higher performance might

be that the TIL and AKU groups have quite similar features, resulting in the supervised model still being able to predict some AKU observations as anomalies just because they were similar to the TIL observations. On the other hand, there was a sharp decline in predicted anomalous AKU observation from scenario 2 to scenario 3. In other words, most AKU observations does not get predicted as an anomaly by the tree model in scenario 3. Instead, it seems the model is that much better at predicting the TIL observations compared to the less supervised models in terms of F1-score, that this weighs up for the loss of AKU predictions.

5.4 Simulation

As stated, we theorize that the observed performance in the less supervised approaches stems from the fact that they are unable to clearly identify the variables which most effectively separate the different groups of tax returns from each other, described in section 5.2.1. In this thesis we use AKU and TIL observations as examples of the type of faulty tax returns which can be found. These are, of course, not representative for all types of erroneous tax returns. This makes it interesting to study how changing the properties of the underlying dataset influence the performance of the less supervised methods. To investigate how changing the properties of the tax returns affect model performance, we simulate seven datasets based on the tax returns provided in the original sample. The goal of the simulation is to exaggerate the differences that already exist in the dataset, in order to see how much the anomalous observations should differ from the majority class for the models to distinguish them from each other.

The number of tax return features the Tax Administration possess is much larger than what is used for this thesis. It is therefore possible that some of these features have larger differences, or better separates erroneous tax returns, than what is the case in our sample. If these kinds of features exist, it would be interesting to investigate the effect larger class differences have on the methods. Furthermore, simulating data with differing structures might help us uncover useful information about the inner workings of the models. It is important to note that altering the underlying data would affect *all* machine learning methods, independent of degree of supervision. In this section we study how the performance of the DBSCAN and OC-SVM are affected by increased class

differences. This means that we focus on change in model performance in itself, and not in relation to the supervised benchmark.

In order to get a cleaner comparison of the effects of changing the characteristics of normal and anomalous tax returns we choose *one* group to represent the anomalous observations. Since AKU seems to separate slightly more from the control observations than those in the TIL group, presented in section 3.2, we use AKU observations as the baseline for simulating new anomalous observations. To precisely study the effects of changing the underlying data, the models are estimated and tested again on data where only AKU tax returns are included as anomaly observations. To ensure the best starting point for the comparison, the tuning parameters for the models are re-estimated in the same manner as described in section 5.1. The results are presented in table 5.7.

Model	Precision	Recall	F1 Score	Number of anomalies predicted
DBSCAN	0.120	0.204	0.151	9.1 %
OC-SVM	0.100	0.193	0.132	9.8 %

Table 5.7: Comparison on the best performing models using *only* AKU observations as anomaly candidates

Illustrated in table 5.7, the performance of the DBSCAN model with only AKU observations as anomalies is slightly better than when the TIL group was included. This is due to an increase in precision, while there is a minor decline in the recall. This might be caused by model capturing fewer anomalies over all. The OC-SVM is subject to the opposite effect. The OC-SVM predict more anomalies, resulting in a higher recall, but the decrease in precision is bigger resulting in a decrease in the F1-score compared to the first model.

The simulated dataset contains the same number of features as the large dataset discussed in section 3.4, where 6 are numerical an 6 are categorical. Since we theorize that the sub-par performance is due to an inability to identify more subtle differences, we focus our attention on how the methods perform when the differences between control observations and erroneous returns becomes more pronounced overall, as opposed to studying the effects of altering single variables. Considering this, we increase the differences between the two groups of tax returns across all features. In other words, we exaggerate the differences in the numerical variables as well as altering the distribution of the categorical variables.

The numerical variables are simulated from a Gaussian distribution using the sample mean and variance for the AKU and control observations. In the simulated dataset new variables are drawn from this distribution. The motivation behind the simulation, was to be able to investigate how different properties in the underlying dataset influence the methods. The number of features used in this thesis is much smaller than what the Tax Administration actually posses. The hope is that the simulation analysis might give some indications of how other variable might be suited for use in the methods in a future feature selection process. As we do not know the distributions of any of the other variables, a simple assumption is to use Gaussian distribution.

Looking at the initial data analysis in section 3.2, the true distributions of our numerical variables are, however, most likely not Gaussian. A distinction from Gaussian distribution is that the original variables' distributions seems to be skewed and have a longer right tail. This indicates that the original variables might have gamma or log-normal distributions. This difference in distributions between the simulation and the original dataset may have a practical impact on the predictions.

In order to investigate the effects of using the Gaussian distribution, we first simulate a dataset with the same mean and variance as estimated for the original sample. In this dataset the categorical features also have the same probabilities as in the original sample. This allows us to better isolate the effect the difference in distributions might have on the methods.

The categorical features are simulated from a categorical distribution, where the probability of each category is based on the original samples of control and AKU observations. When we simulate changes to the distributions of the categorical variables we alter them in a way which makes the distribution of each categorical variable more different between the two types of observations. Based on our prior data analysis, and the analysis performed by Andersson and Lillestøl (2017), we changed the distribution of the features that seems to occur most frequently in AKU, and made the opposite adjustment on these features in the control group.

To best study model performance under changing circumstances, we simulate seven datasets. We start with a baseline simulation with no increase in variable difference. This can later be compared to the six next simulated datasets.

We continue with simulations with four gradual increases in the difference between control and AKU observations. The increase in difference in the mean for the numeric variables is 200%, 100%, 50% and 10% respectively. We keep the same variance throughout all the simulated datasets. For example, in the simulated dataset with the most exaggerated differences (200%), we calculate the difference between the mean in the control group and AKU group, and multiply this difference with 200%. We then add half this difference to the mean in one group and subtract it from the mean in the other, increasing the distance between the mean of the two groups. The adjustments made to the categorical variables were to move 20 percentage points (pp), 10 pp, 5 pp or 1 pp, respectively, from one category to one or two other categories in the same variable. For example, in the simulated dataset with the most exaggerated differences (20pp), the probabilities for the variable Delivered Tax Return are changed from the original distribution of P(J) = 35% and P(N) = 65% to P(J) = 15% and P(N) = 85% for the control observations, and from P(J) = 60% and P(N) = 40% to P(J) = 80% and P(N) = 20% for the anomaly candidates. All the changes to the categorical variables can be found in appendix A2. The last simulation we do is to alter numeric and categorical variables separately to study the extent to which the two types of variables effect the methods.

Model	Dataset	Precision	Recall	F1 Score	% anomalies predicted
DBSCAN	base simulation	0.213	0.316	0.267	6,5%
DBSCAN	10% num and 1 pp cat	0.233	0.317	0.269	6,5%
DBSCAN	50% num and 5 pp cat	0.214	0.331	0.260	7,4%
DBSCAN	100% num and 10 pp cat	0.138	0.202	0.164	6,9%
DBSCAN	200% num and 20 pp cat	0.171	0.236	0.199	6,6%
DBSCAN	50% num	0.241	0.327	0.277	6,5%
DBSCAN	5 pp cat	0.241	0.361	0.289	7,2%

Table 5.8: Comparison on DBSCAN on simulated dataset

As presented in table 5.8, the baseline prediction with no changes in values has a F1-score of 0.267 for the DBSCAN model. This represent a sharp increase from the real data tested in table 5.7. This change is also present when comparing the baseline simulation on the OC-SVM with the results on the real observations in table 5.7. This is perhaps most likely caused by the transition to the Gaussian distribution for the numerical variables. Considering that the mean and variance are estimated from the original data sample, this may also support our suspicion that the original variables have a different distributions.

Looking further at the results from DBSCAN simulations we find that model performance improve the most by using the baseline simulation. Performance is further improved in the 10% change simulation, although not by much. Following this, DBSCAN's performance decrease sharply from the 50% to 100% change simulations. We first thought this decline in performance was caused by the simulated AKU observations making their own clusters as the difference between them and the simulated control data increased. After investigating the clustering structure we found that this was not the case. In fact, the anomalies seemingly only moved from being noise points to belonging to the largest cluster, and not to their own cluster. We are not completely sure why this happens, but there seems to be some contradictory effects when calculating the difference.

We also want to get an overview of how changing only one type of variable affect the models. We therefore test increasing the difference in the numerical variables by 50% and changing the probabilities of the categorical variables by 5pp separately. This way we can study how the increased differences affect the models when introduced separately rather than together. Interestingly, only changing the categorical variables gives better results than only changing the numerical ones, or changing them both together for the DBSCAN method. While the differences are marginal, this could be partially caused by the Euclidean distance favouring the categorical variables as discussed earlier.

As illustrated in table 5.9, F1-score improvement follows the increase in differences more closely for the One-Class SVM. From the baseline simulation to the 200% increase in change simulation, there is a steady increase in F1-score from 0.404 to 0.668. Again, we observe that changing the categorical variables with a 5pp shift in the probability distribution has a better impact on performance than increasing the differences in the means of the numerical variables with 50%.

Model	Dataset	Precision	Recall	F1 Score	% anomalies predicted
OC-SVM	base simulation	0.405	0.402	0.404	5,5%
OC-SVM	10% num and 1pp cat	0.426	0.429	0.428	5,6%
OC-SVM	50% num and 5pp cat	0.441	0.462	0.452	5,8%
OC-SVM	100% num and 10pp cat	0.419	0.601	0.494	8,0%
OC-SVM	200% num and 20pp cat	0.548	0.855	0.668	8,7%
OC-SVM	50% num	0.414	0.462	0.437	5,5%
OC-SVM	5pp cat	0.462	0.455	0.458	7,2%

Table 5.9: Comparison on DBSCAN and OC_SVM on simulated dataset

Changing the properties of the underlying dataset influence the methods in a notable way. The sharpest incline in performance for both methods was caused by simulating new variables under the assumption of a Gaussian distribution. The OC-SVM also saw a sharp incline from increasing the difference in the variables as well. This increase in performance is interesting, and give an indication of how the methods operate. While the OC-SVM had an intuitive performance increase, in line with what we expected beforehand, the DBSCAN's behavior is more difficult to explain. We have not managed to pinpoint why it exhibits this behaviour, but it would be an interesting topic for further research. If the Tax administration expect errors with features exhibiting the same properties investigated in the simulation, they can use this analysis if they were to investigate these methods on other variables or data samples.

6 Discussion

Four unsupervised and semisupervised methods have been tested on their ability to correctly identify and separate anomalous faulty tax returns from normal observations without errors. In the analysis in the previous chapter we discovered that model performance was not as strong as we had hoped, and we speculate that this might be caused by an inability to distinguish the three groups of tax payers from each other. In other words, the less supervised methods are not able to detect the differences between the three groups, and thus not able to separate them.

6.1 Predicted Anomalies by our Methods

When investigating the observations flagged by the models to be anomalies, we did notice that they were more extreme compared to the full dataset. By extreme we refer to observation with uncommon feature values, compared to the rest of the observations. The plots in section 5.2.1 indicates that the flagged anomalies have feature values which occur less often than in the original sample. That being said, they do not separate the values common for AKU and TIL, but those that are extreme or uncommon over all, resulting in the control group occurring in much the same proportion as AKU and TIL. This indicates that the models are not able to find the features where observations from the AKU and TIL group stands out compared to the control observations

Given that the observations that are marked as anomalies by the models often are the more extreme values, some of these observations could also be detected by simpler methods, for instance rule based selection. A benefit our less supervised approaches have over rule based methods are that they do not need instructions on which extreme values to look for. Furthermore, the more advanced methods, such as the One-Class SVM and autoencoders, may be better apt at identifying local anomalies, where features are unusual in relation to other observed features of that observation, but not in relation to the entire dataset. While we do not uncover clear evidence of this in our analysis, it would be an interesting topic for further investigation.

6.2 Supervised Learning Benchmark

In addition to the less supervised methods, we tried identifying faulty tax returns with boosted trees as a supervised machine learning benchmark. Not surprisingly, the supervised approach had better performance in terms of F1-score, precision and recall. As discussed, the supervised method uses labels to pinpoint the differences between each group, enabling it to capture specific characteristics of the erroneous tax returns. As unsupervised methods look more at the properties of the entire observation, they might struggle to separate the classes when only a few dimensions are relevant. Particularly the fully unsupervised clustering algorithms, K-means and DBSCAN, consider each observation as a whole, and does not make any considerations to distinct values of particular features. Practically this means that two observations which differs slightly across all features might be evaluated equally to two observations which are identical except for one feature with larger differences. Considering that the data sample used in this thesis only seems to vary in some dimensions, this might contribute to the performance differences we observe between the approaches.

The semisupervised methods might be better than the fully unsupervised at separating classes based on specific traits, because they can learn how to discover the normal observations, and separate based on those. On the other hand, the semisupervised methods are only capable of finding these subtle traits in the normal observations they learn during training, while the fully supervised might find important traits in the other group as well, as they train with labels on all the groups. In our case the supervised method can find subtle, but important traits in the AKU and TIL group, and not only the control group.

The performance of the boosted tree method varied based on the scenarios it was put trough. In the best scenario we used a balanced dataset and provided the correct labels on all observations, resulting in an F1-score of 0.795. This score is quadruple that of the best unsupervised and semisupervised methods tested in this thesis. The third scenario we developed was a stylized example illustrating how a supervised model would fare in a more realistic "anomaly detection scenario". Here, the supervised model's performance dropped to an F1-score of 0.188. While this was still an improvement on our less supervised models, it illustrates how sensitive performance can be to the circumstances in which the model is employed.

Testing the boosted trees under a best case scenario with balanced classes and all labels illustrates that there, at least to a reasonable extent, exists differences between the groups which can be used to separate faulty returns from the majority class of normal observations. Even so, the F1-score of 0.795 for the boosted trees represent a best-case scenario, which is not very applicable to an anomaly detection case. The primary argument for including less supervised learning algorithms for the purposes of anomaly detection is that they are capable of identifying errors which does not have to be specified in advance. When looking for new types of errors in tax returns, which you do not know what looks like, and do not have labelled observation on, balancing classes is impossible. Consequently, while the performance of the boosted trees in the best case scenario serve to illustrate that there exists differences between the majority class of normal observations and the observations we have used as examples of erroneous tax returns, it cannot be considered representative for the performance a supervised approach would have in the situations we consider appropriate for our less supervised methods.

It is important to note that less supervised approaches seems to primarily be appropriate in situations where you need the flexibility which comes with not having to specify exactly what to look for. For tax return errors which are recurrent year after year, and where the Tax Administration already have a solid collection of labelled observation, such as those discussed in (Thorsager et al., 2016), supervised approaches seemingly have a superior performance, as illustrated by our comparison to the "best case" boosted trees.

6.3 Simulation

Machine learning methods are often sensitive to the input data, and it is often difficult to generalize the result across multiple data samples. Since the data sample used in this thesis is very small compared to the population of data samples in the Tax Administration possess, we wanted to test the methods on data with slight different properties. This might give an indication as to how the methods could perform on one of the other data samples in the Tax administrations database. The simulation is based on the tax returns we use throughout the thesis. This is done in order to have a basis in tax type data, as well as enabling comparison to the other results. The simulated dataset differ both in terms of a different underlying distribution of the numerical values, as well as an increased

difference between the variables in the control and AKU group. The simulated dataset was tested on the two best performing methods, DBSCAN and OC-SVM.

The simulation study gave mixed results. The OC-SVM performed as expected, with increased prediction accuracy as class differences increased. This was not the case for DBSCAN. Performance improved for small increases in the class differences, but then the performance declined as the difference between the variables in the control group and AKU group increased. This seems to be the opposite of what to expect. When the difference between the variables in the two groups increase, one should expect the Euclidean distance between the observations to increase as well, resulting the the anomalous AKU observations to fall out of the control group clusters, and become noise. We hypothesised that this was caused by the AKU data making their own clusters when the differences between the groups exceeded a certain point, instead of becoming noise. This, however, did not seem to be the case. Rather, some effects seems to work in opposite directions making it more difficult for the DBSCAN to distinguish AKU data as noise when class differences increase. We have not managed to uncover specifically why this happens. As mentioned earlier, we suspect that the use of Euclidean distance, and particularly in combination with dummy encoded categorical variables, influence DBSCAN's ability to predict on datasets with more features, which might become more pronounced as class differences increase. We do not know that this for a fact happens, and an investigation into different distance measures might therefor be an interesting topic for further research. This might uncover more information on the behavior we see in the simulation as well. The OC-SVM could have an advantage as group differences increase. Given that the model learns to identify normal behaviour, it seems to be better able to separate observations as normal and anomalous when the differences increase.

From the simulation exercise we also observe an increase in the models' performance between the original data analysis in table 5.4 and the simulated datasets with no changes made to the estimated parameters, presented in the first rows of table 5.8 and 5.9. We postulate that the original data sample have another distribution than Gaussian, causing the performance differences from the base case. Instead, most of the numerical variables shows signs of having a gamma- or log-normal distribution. These distributions can be more centered around the mean than the standard Gaussian distribution. They also

have a one-sided tail, as compared to the Gaussian distribution's symmetrical shape. Estimating the mean and variance parameters from a non-Gaussian distribution, and generating data from Gaussian distribution defined by said parameters, might skew the data in a way which increase performance. Specifically, we hypothesize that the transition to Gaussian distributions with symmetrical tails, as opposed to one-sided gamma or log-normal distributions, will cause some observations from each group to drift further apart by altering how the variance affect the distance from the mean. Thus, increasing the models' ability to separate the two classes. If the Tax Administration try to fit the methods to other features, this might indicate that the models gets increased performance if the distribution is more dispersed than what is the case in our original data sample.

6.4 Practical Application

When determining how the methods discussed in this thesis perform on detecting anomalous, faulty tax returns, more than just F1 score, precision and recall should be considered. This is especially true if the end goal is to determine whether the methods are suitable to determine if the Tax administration should send an observation to manual control or not. We are not going to go look into what constitutes if it is economically feasible for the Tax administration to actually use a method. We do however have a brief discussion of important aspects to include in an analysis investigating this. Aspects such as ease of implementation, prepossessing requirements, how to validate the results, how to maintain the methods and uncertainty of the results should all be considered. All of these aspects effect if the method is economically viable, or if the cure is worse than the disease.

Some degree of prepossessing is needed for all four methods, including scaling, cleaning and one-hot encoding. This is however standard practise and what is expected to some degree by every machine learning method. For the unsupervised clustering methods there is limited need to use resources on labeling the data. On the other hand, unsupervised approaches requires prior knowledge of the field and the data. This is needed as the methods require the user to have some idea of how many anomalies are expected and in what extent they might differ from the normal data, in order to generate useful cluster structures, set reasonable thresholds and determine parameters. The last tuning can be done in combination with rules of thumb and validation techniques, but the final

decision should be done with some linkage to prior knowledge. Considering that the Tax Administration in fact are domain experts, we expect that the requirements to prior knowledge will not be prohibitive for them to utilize unsupervised methods.

One drawback when it comes to maintaining the clustering models compared to methods with training and test data, built to be able to predict on new data, is that the clusters might need re-estimating in a greater extent. This is due to a rising uncertainty of the underlying structure of the data that determine the clusters when huge amounts of new data is introduced. They can handle new data, but this rise in uncertainty is another point where prior knowledge can be used to determine when the cluster structure should be re-checked. This means that the clustering methods might need a relatively small amount of resources prior to clustering, given that you have some knowledge about the data, but they need more resources when maintaining the models. Another benefit of this approach, where there is no train and test set, is that the risk on overfitting on the data is smaller, because the data is validating against structures of the data, and not on how the methods perform on the data. An additional factor influencing the uncertainty of the results is how deterministic the final results are. Here DBSCAN have an advantage over k-means. In DBSCAN, for every given tuning parameter, the noise and core points are deterministic. For K-means on the other hand, the clusters can change for each iteration, demanding more resources on running the method several times in order to get a more stable result.

The semisupervised methods require additional resources and preparatory work prior to implementation. First and foremost, the methods require normal, non-erroneous, data, or what is expected to be normal data. This is information the The Tax Administration posses, as the manual control process itself generates vast amounts of labelled normal tax returns, or at least, tax returns where they have not identified any errors. This last aspect is important, and illustrates why training data should be selected carefully. The normal tax returns are what is used to teach the models what is normal behavior, and what to separate new observations on. If the data they use for this purpose does actually contain a lot of errors, the model is most likely going to have a harder time detecting those errors. Even so, the models have the ability to be used fully unsupervised if the errors actually are anomalous as explained in section 4.4. This makes the methods robust

if some errors still ends up in the supposedly normal training data. Training the models with presumed normal data is however going to inflict less uncertainty to the end result. If normal labels are available this would be preferable to use, but a lack of labels does not necessarily prevent the use of these methods.

As mentioned earlier, the semisupervised methods run a risk of overfitting to the normal data. This entails that you might need to use a cross validation method when training to reduce this risk, and the model you end up with should be validated in manner that can detect overfitting. How deterministic the end result is varies between the methods. For the OC-SVM, once the parameters are found, the result is deterministic. Since the autoencoder is initialised with random weights, the end result might vary. This entails that you should run the autoencoder several times in order to get a result stable enough for further investigation.

6.4.1 Tax Administrations Evaluation Prior to Manual Control Selection

In a practical application, observations flagged as anomalies must be subjected to manual control in order to determine if they in fact are erroneous. The Tax Administration apply domain knowledge when evaluating the output of predictive models, and observations that are flagged might not be subjected to actual control (Berset et al., 2016). Although Berset et al. (2016) reference a qualitative evaluation of valued added tax controls flagged by their predictive models, it is not unthinkable for this approach to be combined with unsupervised error detection in personal tax statements as well. This might effect the upper threshold for false positive from the model, and allow for a lower precision in exchange for higher recall, because the predicted outputs goes through a second check before they are sent to a full investigative manual control. This must however be evaluated in light of economic feasibility.

6.5 Method Performance Comparison

Even though none of the methods we have applied does exceedingly well as is, we find that the OC-SVM do have some traits which might make it favourable above DBSCAN and the other methods discussed in this thesis. Looking at the results in table 5.4, we

find that performance in terms of the F1-score is almost identical between DBSCAN and the OC-SVM, although the OC-SVM model flags relatively fewer anomalies in total. It also has a higher precision score. Given the previous discussion on the F1 measure and sensitivity to the number of positive predictions made, this indicates that the OC-SVM in fact is the best performing model. We do see that the OC-SVM has the worst performance on the reduced size dataset. As previously discussed, this might indicate that the OC-SVM suffers when information is reduced, and that it to some extent is more robust to higher dimensional data. Considering that the use-case for unsupervised error detection is aimed at discovering rare and unknown types of anomalies, a reduced need for feature engineering and selection could be an extra benefit.

From the simulation exercises we also experience that the OC-SVM performs more intuitively and exhibits more stable behaviour than the DBSCAN. While it would have been interesting to compare this performance to the autoencoders, a simulation experiment on autoencoders was prohibited both by their highly unstable and variable performance combined with the sheer resources needed in order to successfully fit an autoencoder model. In itself this is also a reason why we consider the OC-SVM to have the most promise of all methods explored in this thesis. Of the two methods that should handle higher dimensional data exploration and anomaly detection the best, the OC-SVM shows a robustness and ease of use which instinctively seems more attractive. As the OC-SVM evaluates observations in relation to a non-linear decision boundary, discussed in chapter 4.4, it is also perhaps better equipped to flag observations which are only anomalous in some dimensions, or in relation to other feature values in that observation, than the clustering methods.

Even though the OC-SVMs shows the most promise, the performance of the models on the original dataset is not good enough for us to recommend using them as a stand alone solution for detecting errors in tax returns. If however, there exist tax return features with different properties and more predictive powers in the Tax Administrations databases, unsupervised machine learning have many benefits which can provide a good addition to the other supervised methods. Investigating unsupervised anomaly detection on other samples of tax data, or employing the methods in combination with other approaches, would help further evaluating the performance of these methods

6.6 Method Critique

This thesis is written as an initial investigation into the use of unsupervised and semisupervised machine learning to detect anomalous faulty tax returns. We have therefore investigated multiple methods from a number of different angels and kept a broad scope. A drawback of this approach is that every method is investigated rather briefly. This means that while we get an initial indication of model performance on our tax return dataset, we risk overlooking solutions which could have yielded better results. One example of this is using the standard, Euclidean distance as the dissimilarity measure in our clustering methods, although other measures might be better suited for mixed data types. Another example is that we only test one method of selecting anomalies from the K-means clustering structure. The strategy for selecting observations from a K-means cluster as anomalies is based on two simple assumptions of anomaly behaviour, which we have tried to reflect in our measures. However, including different weighting on the two assumptions or exploring different ways of capturing these might have been beneficial.

Another area where our analysis might have benefited from a more extensive approach is the simulation exercise. This analysis is meant as a supplement to the analyses done on the original tax data, in order to get a better understanding of the models' performance, and test a scenario with other features. After investigating the variables, the assumption of Gaussian distribution does not seem to be true on our data. We tried to accommodate for this by first having a base simulation, and isolating the effect of differing distribution. Another approach could be to try other distributions, for example log-normal or gamma.

Throughout the thesis we have tried to keep the unsupervised and semisupervised use-cases in mind. Considering that the primary application of our less supervised methods would be to detect new and unknown forms of errors, validating the models fit by using labels would be impossible in a real scenario. We therefore try to make active considerations as to how these methods would be employed in a real application. However, as we do not posses the required domain knowledge to evaluate our less supervised models, labels are used to some extent both for estimating the models' tuning parameters and evaluating performance. One could both argue that labels should have been used to a greater or lesser extent. Utilizing labels more actively had allowed us to further customize our models to the data sample. This could have yielded better performance scores and perhaps clearer

recommendations of tuning parameters' values. On the other hand, relying less on labels had allowed us to better investigate a real application of the methods, dedicating more attention to aspects such as unsupervised model validation.

As a final note we would like to discuss a more specific weakness in our analysis. When selecting features for the dataset with fewer variables (discussed in chapter 3.4 and presented in table 5.5) the variable SKM-group was removed by mistake. This was done on the basis of an error in the correlation analysis, where the SKM-group was incorrectly shown to have a high correlation with Centrality Code and Classification Code. In an attempt to investigate if removing correlated and possibly redundant features would aid model performance, the variable was excluded. SKM-Group is in fact *not* highly correlated with any other variable, and one of the variables where we see a clear distinction for the AKU observations. Consequently, the SKM-group should have been included in the smaller dataset as well. As the inclusion of SKM-group could have improved performance, the analysis should ideally be done again. The mistake was unfortunately discovered rather late. After assessing the situation, with the knowledge that the small dataset was not used for the rest of the thesis, we concluded that the best use of resources was to focus on other improvements of the thesis. We still chose to include the analysis in table 5.5 as part of this thesis. Although the inclusion of SKM-group would have impacted the results, we find that the smaller dataset still highlight interesting aspects of the four methods.

6.7 Further Research

As stated, this thesis aims to serve as a basis for further research into unsupervised anomaly detection on tax return data. The methods discussed have F1-performances which could be viewed as rather unimpressive, at least in comparison to the fully supervised benchmark. Even so, detecting unspecified errors is difficult, and models with seemingly low F1-scores might still prove useful either alone or in combination with other evaluation methods. Considering this, a further application of the concepts discussed here could benefit from quantifying what exactly constitutes adequate performance. And in extension, what determines if the models are economically feasible.

Based on our initial analysis, several interesting topics for further investigation emerged. First, as we evaluate the semisupervised approaches to be particularly suited for the Tax

Authorities, looking further into these could be considered. The OC-SVM is highlighted as the most promising method in this analysis, but it would be interesting to study how the autoencoders fare on a larger data sample, where they might show increased stability and improved performance. In addition, the semisupervised method Self Organizing Maps was considered for this thesis, but not included in the final analysis as we had to limit the scope of our investigation. Still, the method showed some promise and should be investigated further.

While we experienced that the semisupervised approach performed best in our analysis, it would be interesting to continue investigating the clustering approach as well. In a further investigation of clustering based methods we would exploring other and perhaps more suited distance measures for mixed data types. The performance of DBSCAN seemed relatively promising on the original data sample. If it is desirable to utilize clustering approaches, we would prioritize a further investigation into the counter intuitive performance the DBSCAN displayed in the simulation analysis.

Another aspect that would be an interesting topic for further research is the assumption of Gaussian distributed variables in the simulation. Specifically, it could be interesting to study how increased variable differences affect model performance when the numerical variables are more similar to those found in the original tax return sample. This could be done by using other variable distributions, for example log-normal or gamma that looks suitable based on the initial data analysis in section 3.2. As a starting point of an analysis into other distributions, we wanted to quantify the difference in distribution of the original dataset and our Gaussian distributed simulation. The dataset used as the simulation dataset is the base case.

In order to estimate the difference between the original variable distributions and the distributions in the simulated data we calculate the Kullback-Leibler divergence, or KL-divergence. The KL-divergence is an asymmetrical measure on the difference between two distributions over the same event space. If the distribution of the original dataset is P(X) and distribution of the simulated dataset is Q(X), then the KL divergence on a finite set of χ is defined as (Bigi, 2003):

$$D(P||Q) = \sum_{x \in \chi} P(x) log \frac{P(X)}{Q(X)} \tag{6.1}$$

The KL divergence comes from information theory, and can be interpreted as the average number of bits of information wasted by encoding the events from a distribution P with a code based on the imprecise distribution Q. The smaller the relative size, the more similar is the two distributions of the two variables, with the smallest size being 0 if the two are identical (Bigi, 2003). The KL-divergence by using the simulated distribution to explain the original dataset for all the control and AKU variables is presented in table 6.1.

Feature	KL Divergence Control	KL-Divergence AKU
Personal income	3.48	3.57
Birth Year	1.70	1.79
Net Wealth	4.36	2.42
Domestic Debt	3.66	2.60
Income after tax deduction	3.02	2.20

Table 6.1: KL divergence on each feature from original dataset to simulated

The feature with the smallest distance to the simulated dataset is the Birth Year. Overall, it seems like the simulated AKU features better resemble the original variable distributions than those in the simulated control group. This might indicate that they are a bit wider dispersed than the control group, where more of the observations gather in the middle. To investigate this further, a density plot of the features comparing the original and simulated data is created, in addition to plotting the KL divergence for the finite space. The plots for the Income After Tax Deductions variable are presented in figure 6.1. Equivalent plots for the other simulated numerical variables can be found in appendix A1.

The density plots support some of the suspicions about the distributions. The original data is exclusively positive values, while the simulated also include negative values. The original variables have a right tail,while the Gaussian simulation is symmetric. The original dataset also seems to be more concentrated around the mean, while there is a more dispersed distribution in the simulated dataset. This might make it easier to distinguish the AKU data in either tail in the simulated dataset, than in the denser original dataset. The KL divergence plot illustrate that the difference between the two distributions is biggest in the tail areas, which seems reasonable when looking at the density plot.

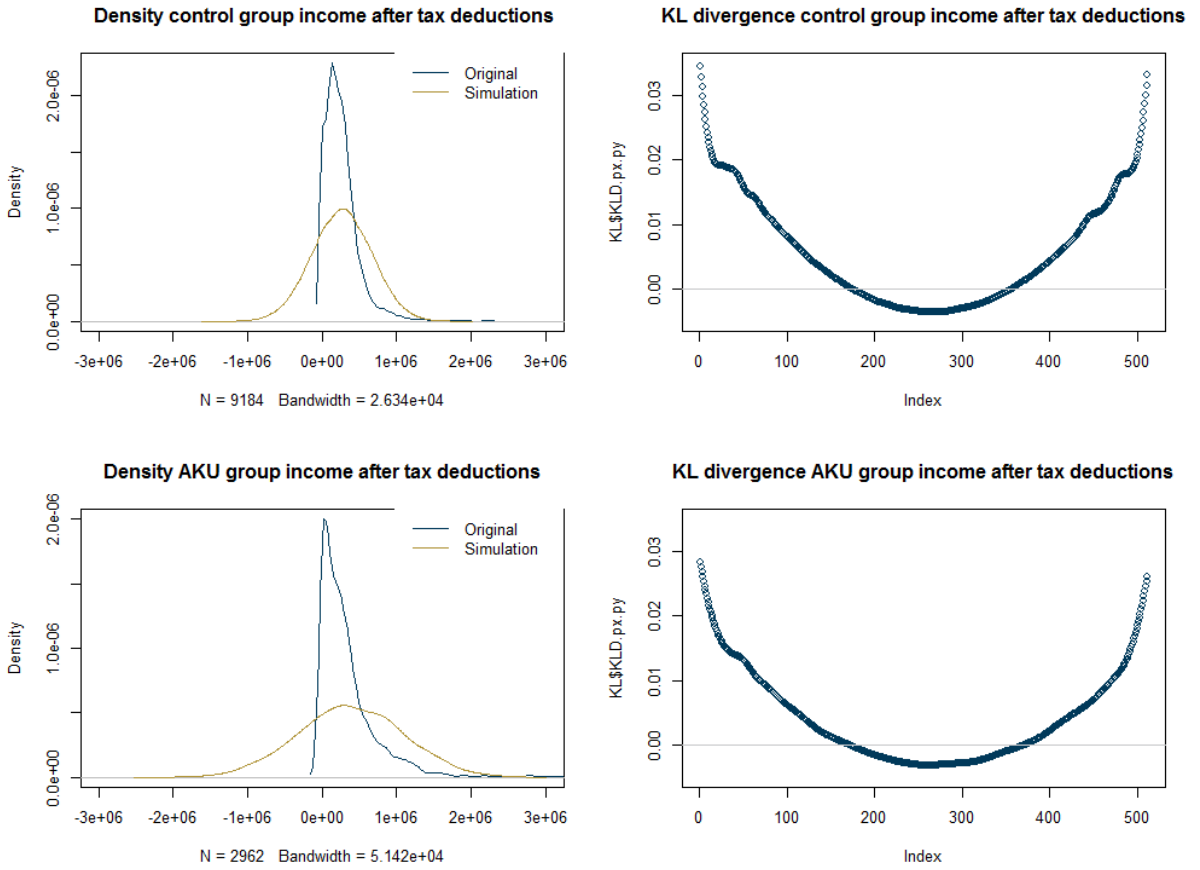

Figure 6.1: *Topleft* Density plot control group original and simulated income after tax deductions. *Bottomleft* Density plot AKU group original and simulated income after tax deductions. *Topright* KL-divergence plot control group original vs simulated income after tax deductions. *Bottomright* KL-divergence plot AKU group original vs simulated income after tax deductions.

7 Conclusion

In this thesis we have investigated how select unsupervised and semisupervised machine learning methods perform on detecting anomalous faulty tax returns for the Norwegian Tax Administration. In extension we have also investigated what the methods are capable of detecting, and how different properties of the underlying dataset influence their performance. The key motivation was to study whether these are suitable approaches that are applicable for the Tax Administration in order to detect new forms of errors. The end goal is that these methods might result in less tax errors influencing tax revenue and the National budget in the future, while not being to resource demanding.

In our analysis we use tax returns of individuals who have been flagged via the AKU scheme and individuals who have been subject to an additional tax payment as representatives of faulty tax returns which should be discovered by our less supervised models. Overall, we find that while the models are able to separate the most extreme returns regardless of group affiliation, they struggle to separate the true erroneous returns from the control observations. If the AKU and TIL observations used in this thesis are representative of the types of new and unknown forms of tax return errors we theorized that our models would discover, our initial analysis suggest that the less supervised methods are unable to effectively identify erroneous tax returns on their own on our data.

The supervised methods used as a benchmark in this thesis had better performance than our less supervised approaches. This was however expected as they have access to labels. We experience that class differences must be present to a greater extent with less supervised approaches than with supervised models in order to correctly separate classes. It is important to note that the methods discussed might prove useful in other contexts. In this analysis we have evaluated model performance as if they were used directly to select observations for manual control. For this application we consider the models to have too poor performance to be considered economically viable on our data sample. However, the methods could also be employed for more traditional data discovery purposes, where the findings would be used as the basis for supervised selection models, or be subject to further investigation before being used as a basis for manual control selection. Considering that the models do identify observations which stand out in some

respect, such applications might still prove a useful addition to the supervised methods.

Of the methods discussed the OC-SVM stands out with a good and intuitive performance. The fact that they are more stable and intuitive, and perhaps better able to pinpoint the observations that differ from the majority, reduces the uncertainty of the methods. Their intuitive nature might enable the Tax Authorities to transfer their domain knowledge to best utilize the methods in an effective way. Furthermore, the Tax Authorities are in a unique position in terms of the quantity and quality of their labels. In addition, the resources available to the Tax administration allows them to investigate the output from the semisupervised models to determine if they are successful at finding errors.

Utilizing less supervised methods to detect faulty tax returns is not an out of the box solution. Further investigation into whether they have the potential of helping the Tax Administration find new errors in the future is needed. Still, it shows promise as an approach to detect new information in the data that you did not now you were looking for.

References

Andersson, J. and Lillestøl, J. (2017). Tre grupper skatteytere i søkelyset: Har de ulike kjennetegn? Discussion paper 10, Norwegian School of Economics(NHH).

Andersson, J., Lillestøl, J., and Støve, B. (2012). Kjennetegnsanalyser av skattytere som unndrar skatt ved å skjule formuer og inntekter i utlandet. *SNF Report Nr 10/12*.

Berset, A., Hussain, S., and Paulsen, P. A. (2015). Modeller for effektiv utvelgelse av omsetningsoppgaver til kontroll. *Analysenytt*, 1.

Berset, A., Hussain, S., and Paulsen, P. A. (2016). Prediktiv modell har økt treffprosenten på mva-oppgavekontroll. *Analysenytt*, 1.

Bigi, B. (2003). Using kullback-leibler distance for text categorization. *Lecture Notes in Computer Science*, pages 305–319.

Brownlee, J. (2020). One-class classification algorithms for imbalanced datasets. Retrieved June 8. 2020 from https://machinelearningmastery.com/one-class-classification-algorithms/?fbclid=IwAR3sB4eZukQyDGbFQ-LW87vEysYMN52AzEF2gAueoS1HFVYr5ooZikByuFM.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41.

Dy, J. and Brodley, C. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research (JMLR)*, 5:845–889.

González, P. and Velasquez, J. (2013). Characterization and detection of taxpayers with false invoices using data mining techniques. *Expert Systems with Applications*, 40:1427–1436.

Goodfellow, I., B. Y. . C. A. (2016). *Deep Learning*. The MIT Press, Cambridge, Massachusetts.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. retreieved June 10 from http://www.deeplearningbook.org.

Hahsler, M., Piekenbrock, M., and Doran, D. (2019). dbscan: Fast density-based clustering with r. *Journal of Statistical Software*, 91(1).

Hastie, T., Tibshirani, R., and Friedman, J. (2017). *The Elements of Statistical Learning*, volume 1. Springer.

Hawkins, S., He, H., Williams, G., and Baxter, R. (2002). Outlier detection using replicator neural networks. *Data Warehousing and Knowledge Discovery*, pages 170–180.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2:283–304.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2017). *An Introduction to Statistical Learning*, volume 8. Springer.

Keller, G. (2014). *Statistics for management and economics*. Cengage Learning.

Kuhn, M. (2019). The caret package. retreieved May 30 2020 from, https://topepo.github.io/caret/subsampling-for-class-imbalances.html#subsampling-during-resampling.

Maglaras, L. A. and Jiang, J. (2015). A novel intrusion detection method based on ocsvm and k-means recursive clustering. *ICST Transactions on Security and Safety*, 2(3):e5.

Ministry of Finance (2019). Statens inntekter og utgifter. Retrieved April 13. 2020 from https://www.statsbudsjettet.no/Statsbudsjettet-2019/Satsinger/?pid=89003.

Murphy, K. P. (2013). *Machine Learning: A Probabilistic Perspective*, volume 1. MIT Press.

Müller, A. C. and Guido, S. (2016). *Introduction to Machine learning with Python*. O'Reilly Media, first edition.

Roux, D., Pérez Gutiérrez, B., Moreno, A., Villamil, P., and Figueroa, C. (2018). Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach. pages 215–222.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471.

Schreyer, M., Sattarov, T., Borth, D., Dengel, A., and Reimer, B. (2017). Detection of anomalies in large scale accounting data using deep autoencoder networks. *CoRR*, abs/1709.05254.

Schubert, E., Sander, J., Ester, M., Kriegel, H., and Xu, X. (2017a). Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, 42:1–21.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017b). Dbscan revisited, revisited. *ACM Transactions on Database Systems*, 42(3):1–21.

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J. (1999). Support vector method for novelty detection. volume 12, pages 582–588.

Smedsvik, V. and Christophersen, K. (2018). Skatteetatens erfaringer med bruk av prediktive modeller. *Skatteetatens Analysenytt*, 1.

Solorio-Fernández, S., Carrasco-Ochoa, J., and Martínez-Trinidad, J. F. (2019). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, pages 53:907–948.

The Norwegian Tax Administration (2020). Additional tax. Retrieved May 10. 2020 from https://www.skatteetaten.no/en/business-and-organisation/start-and-run/deadlines-certificates-and-accounting/additional-tax.

Thorsager, M., Olsen, , and Foss, C. (2016). Prediktiv modell gir høy treffprosent på kontroll av selvangivelser. *Analysenytt*, 1.

Appendix

A1 KL Divergence plots

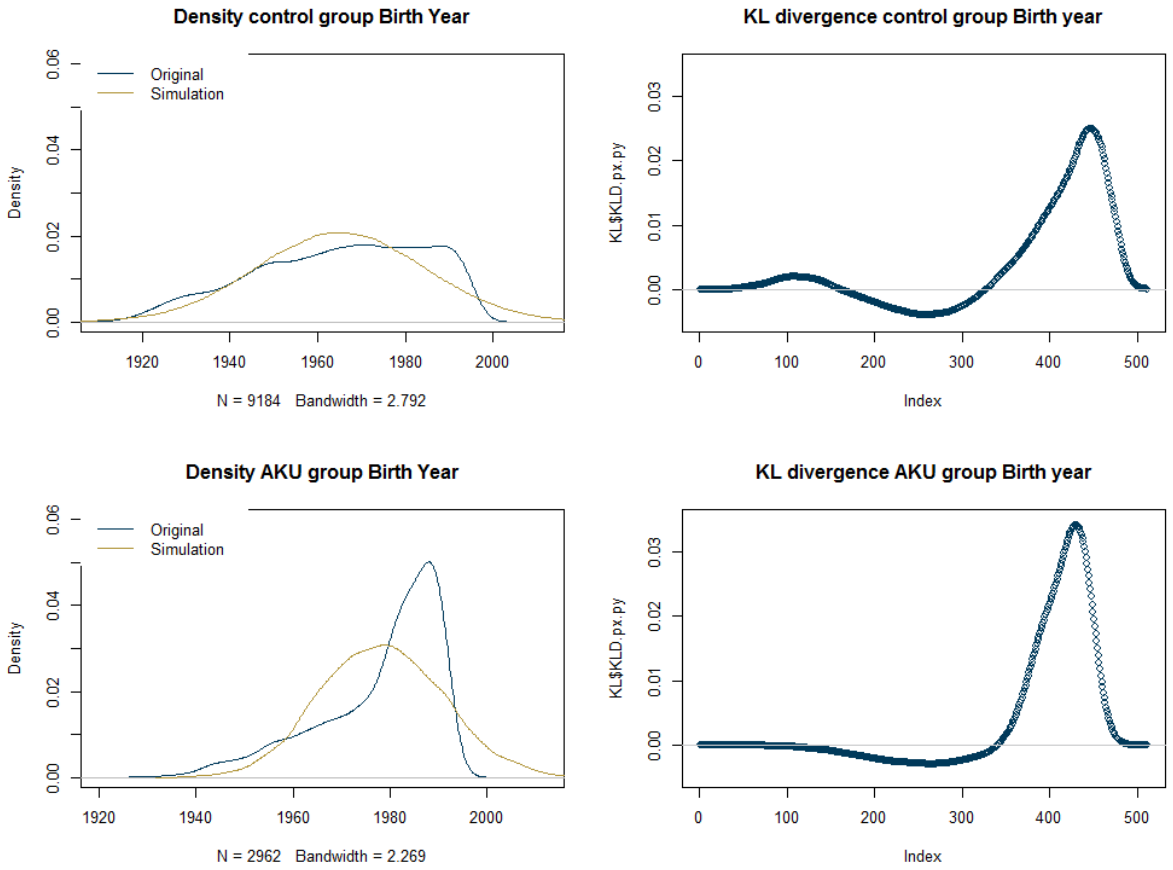

Figure A1.1: *Topleft* Density plot control group original and simulated birth year. *Bottomleft* Density plot AKU group original and simulated birth year. *Topright* KL-divergence plot control group original vs simulated birth year. *Bottomright* KL-divergence plot AKU group original vs simulated birth year.

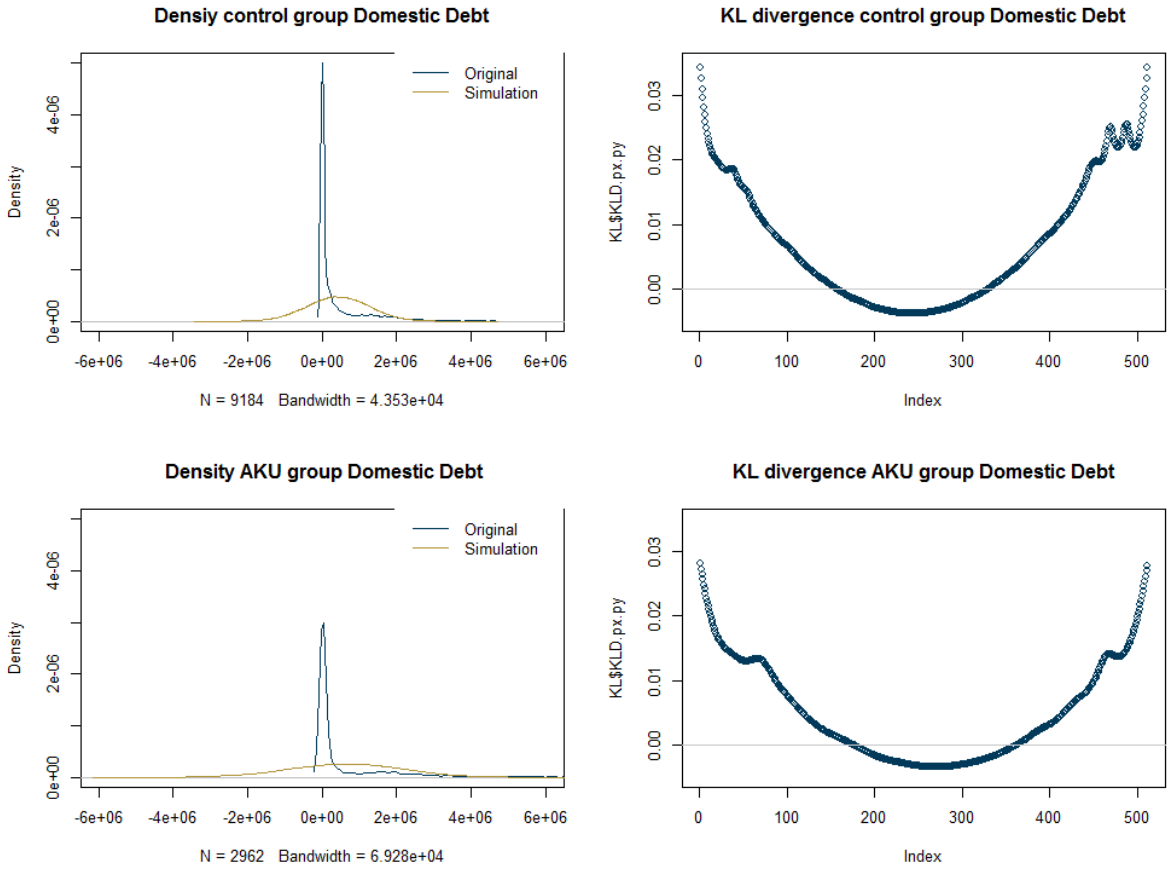

Figure A1.2: *Topleft* Density plot control group original and simulated Domestic Debt. *Bottomleft* Density plot AKU group original and simulated Domestic Debt. *Topright* KL-divergence plot control group original vs simulated Domestic Debt. *Bottomright* KL-divergence plot AKU group original vs simulated Domestic Debt.

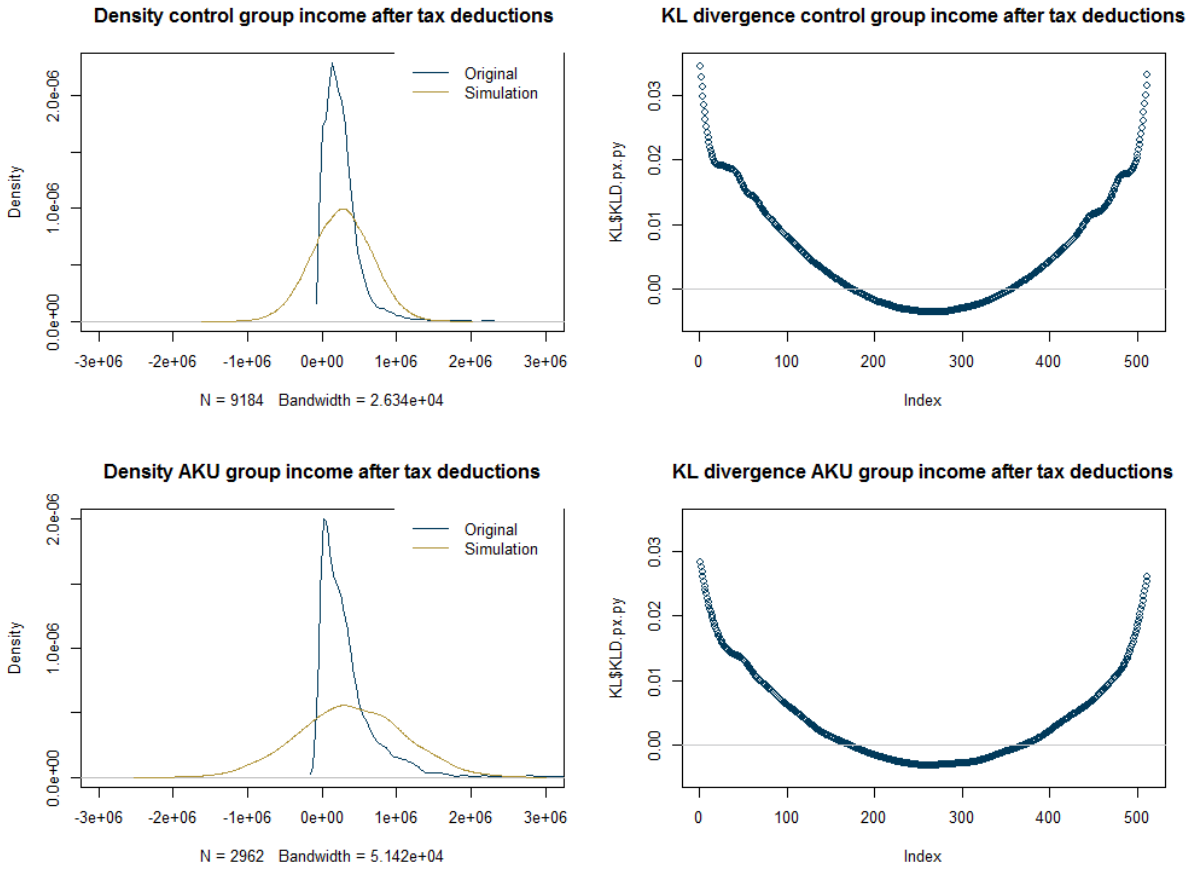

Figure A1.3: *Topleft* Density plot control group original and simulated income after tax deductions. *Bottomleft* Density plot AKU group original and simulated income after tax deductions. *Topright* KL-divergence plot control group original vs simulated income after tax deductions. *Bottomright* KL-divergence plot AKU group original vs simulated income after tax deductions.

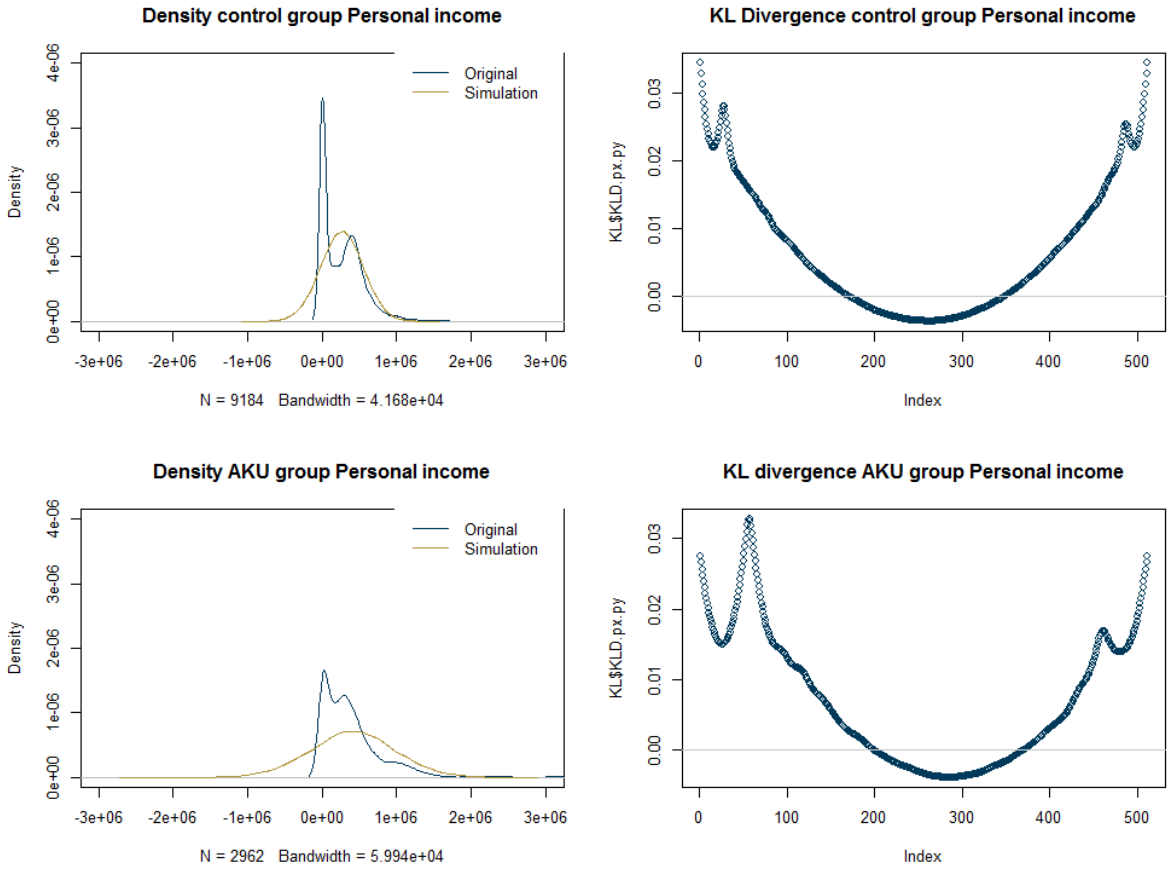

Figure A1.4: *Topleft* Density plot control group original and simulated personal income. *Bottomleft* Density plot AKU group original and simulated personal income. *Topright* KL-divergence plot control group original vs simulated personal income. *Bottomright* KL-divergence plot AKU group original vs simulated personal income.

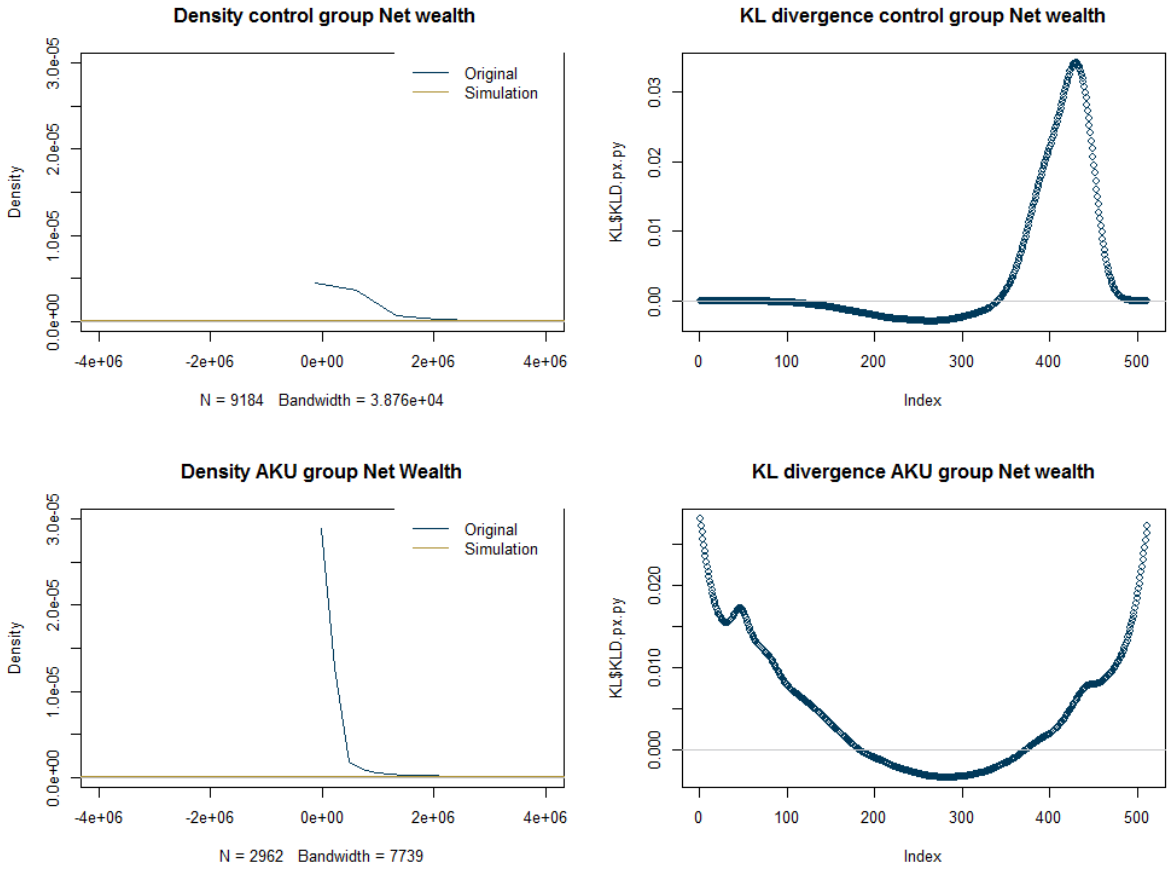

Figure A1.5: *Topleft* Density plot control group original and simulated net wealth. *Bottomleft* Density plot AKU group original and simulated net wealth. *Topright* KL-divergence plot control group original vs simulated net wealth. *Bottomright* KL-divergence plot AKU group original vs simulated net wealth.

A2 Changes to the Dataset in the Simulations

Base case Control group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.39* *2 - 0.43* *3 - 0.07* *4 - 0.1*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.92* *13 - 0.02* *14 - 0.05* *20 - 0* *30 - 0.01* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.68* *2 - 0.2* *3 - 0.12*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.35* *N - 0.65*
History code	*J - Yes* *N - No*	*J - 0.02* *N - 0.98*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.09* *1 - 0.06* *2 - 0.17* *3 - 0.68*

Table A2.1: Base Case Simulation Categorical features on Control group

Base case AKU group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.70* *2 - 0.24* *3 - 0.05* *4 - 0.01*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.345* *13 - 0.11* *14 - 0.545* *20 - 0* *30 - 0* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.90* *2 - 0.045* *3 - 0.055*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.60* *N - 0.40*
History code	*J - Yes* *N - No*	*J - 0.03* *N - 0.97*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.04* *1 - 0.03* *2 - 0.08* *3 - 0.85*

Table A2.2: 20% change simulation Categorical features on control group

20% change control group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.19* *2 - 0.63* *3 - 0.07* *4 - 0.11*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.92* *13 - 0.02* *14 - 0.05* *20 - 0* *30 - 0.01* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.48* *2 - 0.30* *3 - 0.22*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.15* *N - 0.85*
History code	*J - Yes* *N - No*	*J - 0.02* *N - 0.98*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.19* *1 - 0.16* *2 - 0.17* *3 - 0.48*

Table A2.3: 20% change Simulation Categorical features on control group

20% change AKU group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.90* *2 - 0.04* *3 - 0.05* *4 - 0.01*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.145* *13 - 0.21* *14 - 0.645* *20 - 0* *30 - 0* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.90* *2 - 0.045* *3 - 0.055*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.80* *N - 0.20*
History code	*J - Yes* *N - No*	*J - 0.23* *N - 0.77*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.04* *1 - 0.03* *2 - 0.08* *3 - 0.85*

Table A2.4: 20% change Simulation Categorical features on AKU group

10% change control group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.29* *2 - 0.53* *3 - 0.07* *4 - 0.11*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.92* *13 - 0.02* *14 - 0.05* *20 - 0* *30 - 0* *40 - 0.01* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.58* *2 - 0.25* *3 - 0.17*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.25* *N - 0.75*
History code	*J - Yes* *N - No*	*J - 0.02* *N - 0.98*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.14* *1 - 0.11* *2 - 0.17* *3 - 0.58*

Table A2.5: 10% change Simulation Categorical features on control group

10% change AKU group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.80* *2 - 0.14* *3 - 0.05* *4 - 0.01*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.245* *13 - 0.16* *14 - 0.595* *20 - 0* *30 - 0* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.90* *2 - 0.045* *3 - 0.055*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.70* *N - 0.30*
History code	*J - Yes* *N - No*	*J - 0.13* *N - 0.87*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.04* *1 - 0.03* *2 - 0.08* *3 - 0.85*

Table A2.6: 10% change Simulation Categorical features on AKU group

5% change control group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.34* *2 - 0.47* *3 - 0.07* *4 - 0.11*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.92* *13 - 0.02* *14 - 0.05* *20 - 0* *30 - 0* *40 - 0.01* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.62* *2 - 0.225* *3 - 0.155*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.70* *N - 0.30*
History code	*J - Yes* *N - No*	*J - 0.02* *N - 0.98*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.115* *1 - 0.085* *2 - 0.17* *3 - 0.63*

Table A2.7: 5% change Simulation Categorical features on control group

5% change AKU group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.75* *2 - 0.19* *3 - 0.05* *4 - 0.01*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.295* *13 - 0.145* *14 - 0.56* *20 - 0* *30 - 0* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.90* *2 - 0.045* *3 - 0.055*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.65* *N - 0.35*
History code	*J - Yes* *N - No*	*J - 0.08* *N - 0.92*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.04* *1 - 0.03* *2 - 0.08* *3 - 0.85*

Table A2.8: 5% change Simulation Categorical features on AKU group

1% change control group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.38* *2 - 0.44* *3 - 0.07* *4 - 0.11*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.92* *13 - 0.02* *14 - 0.05* *20 - 0* *30 - 0* *40 - 0.01* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.67* *2 - 0.20* *3 - 0.13*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.34* *N - 0.66*
History code	*J - Yes* *N - No*	*J - 0.03* *N - 0.97*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.095* *1 - 0.065* *2 - 0.17* *3 - 0.67*

Table A2.9: 1% change Simulation Categorical features on control group

1% AKU group		
English variable name	Category	Values
Marital status	*1 - Unmarried* *2 - Married* *3 - Widowed* *4 - Divorced/separated*	*1 - 0.71* *2 - 0.23* *3 - 0.05* *4 - 0.01*
SKM-Group	*10 - Fully tax liable resident* *13 - Emigrants* *14 - Temporary Resident* *20 - Local tax liable, resident* *30 - Local tax liable, non resident* *40 - Estate* *70 - Diplomat*	*10 - 0.33* *13 - 0.12* *14 - 0.55* *20 - 0* *30 - 0* *40 - 0* *70 - 0*
Person code	*1 - Reference Person* *2 - Married, youngest partner* *3 - Child*	*1 - 0.90* *2 - 0.045* *3 - 0.055*
Delivered tax reurn	*J - Yes* *N - No*	*J - 0.61* *N - 0.39*
History code	*J - Yes* *N - No*	*J - 0.04* *N - 0.96*
Centrality code	*0 - Rural* *1 - Less Rural* *2 - Less Urban* *3 - Urban*	*0 - 0.04* *1 - 0.03* *2 - 0.08* *3 - 0.85*

Table A2.10: 1% Simulation Categorical features on AKU group

A3 Variable Distribution in Train-Test split of Control Observations

	Birth Year		Personal Income		Income After Tax Deductions	
	Train	Test	Train	Test	Train	Test
Min	1914	1912	0	0	0	0
1^{st} Quart.	1952	1950	0	0	102 000	103 750
Median	1967	1966	216 000	204 500	213 000	213 000
Mean	1965	1965	266 276	260 455	269 510	259 825
3^{rd} Quart.	1981	1981	433 000	433 000	345 000	346 000
Max	1995	1995	3 589 000	3 493 000	21 846 000	12 218 000
	Net Wealth		Domestic Debt		Domestic & Foreign Debt	
	Train	Test	Train	Test	Train	Test
Min	0	0	0	0	0	0
1^{st} Quart.	0	0	0	0	0	0
Median	0	0	10 000	9 000	0	0
Mean	513 685	389 482	419 004	420 689	185 032	157 758
3^{rd} Quart.	352 500	366 250	410 000	388 000	0	0
Max	370 148 000	28 925 000	13 784 000	11 859 000	99 107 000	10 363 000

Table A3.1: Variable Distribution Test-Train Split of Control Observations - Numerical Variables

Marital Status			SKM-Group		
Category	Train	Test	Category	Train	Test
1	39.76 %	38.55 %	10	92.20 %	92.4 %
2	43.14 %	42.90 %	13	2.30 %	2.65 %
3	6.24 %	7.62 %	14	4.88 %	4.23 %
4	10.78 %	10.78 %	Other	0.62 %	0.72 %
Other	0.08 %	0.14 %			
Person Code			**Centrality Code**		
Category	Train	Test	Category	Train	Test
1	67.64 %	67.50 %	0	8.96 %	8.99 %
2	20.02 %	20.91 %	1	5.86 %	6.00 %
3	12.34 %	11.57 %	2	16.52 %	16.75 %
Other	0.00 %	0.02 %	3	68.66 %	68.26 %
Delivered Tax Return			**History Code**		
Category	Train	Test	Category	Train	Test
J	35.56 %	35.59 %	J	2.32 %	1.91 %
N	64.44 %	64.41 %	N	97.68 %	98.09 %

Table A3.2: Variable Distribution Test-Train Split of Control Observations

A4 Elbow Plot for K-means Cluster

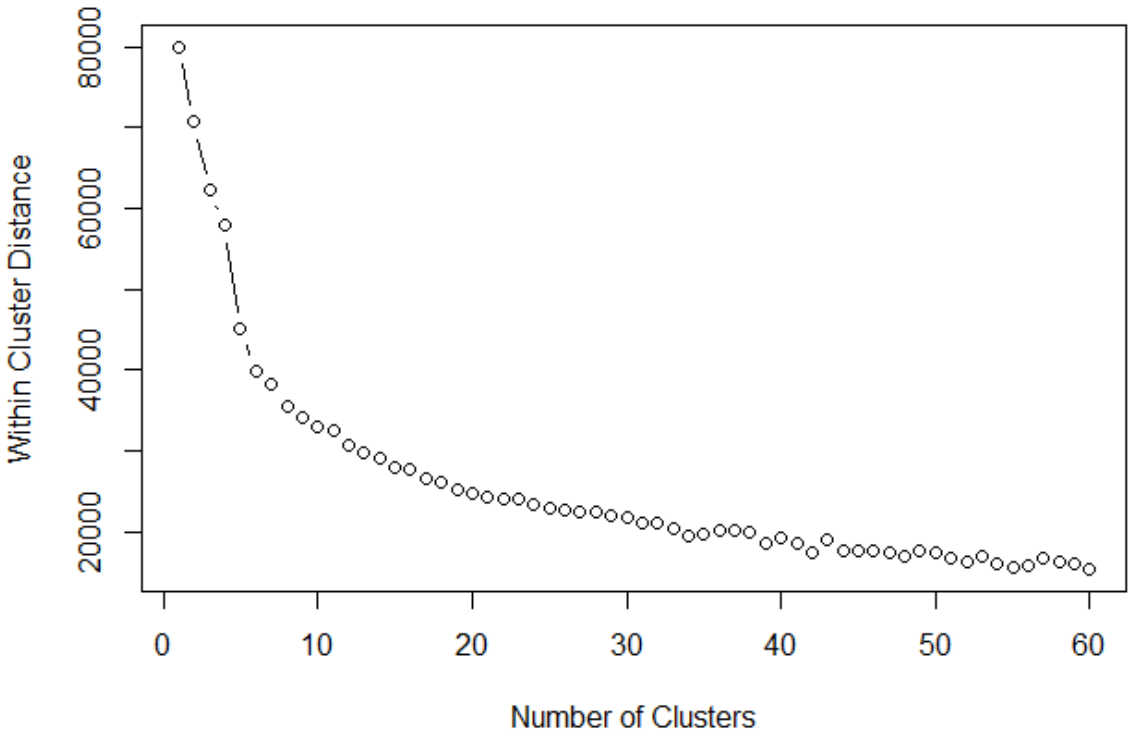

Figure A4.1: Elbow plot for the K-means clusters using the large dataset and both AKU and TIL observations as anomaly candidates (as discussed in table 5.4)

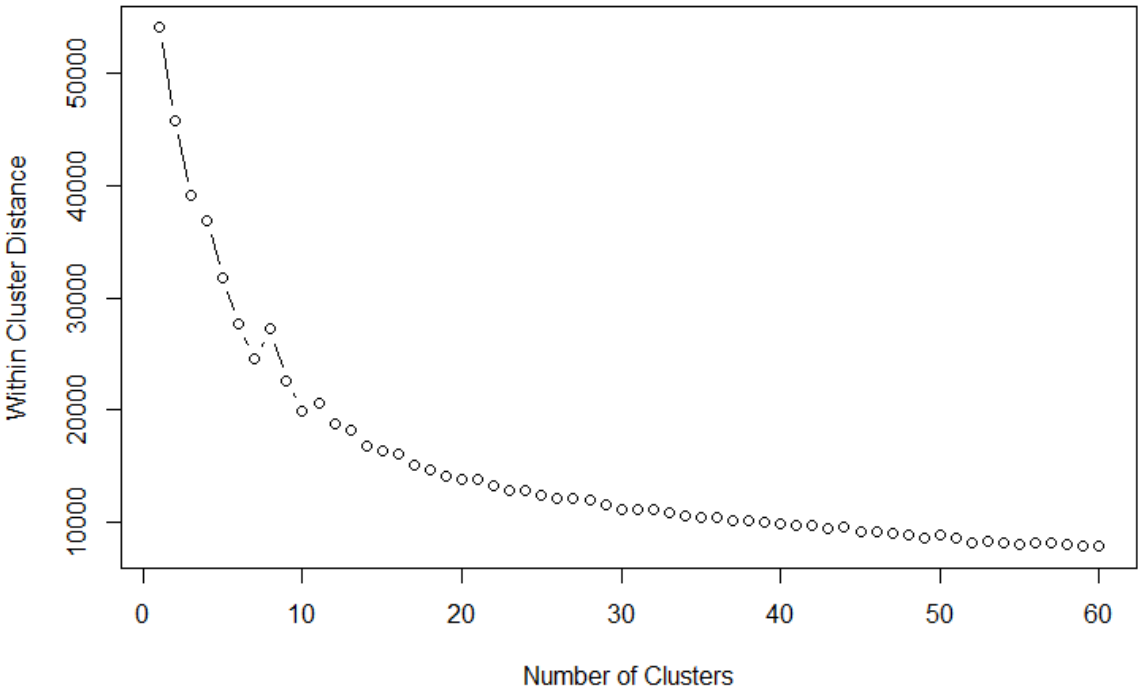

Figure A4.2: Elbow plot for the K-means clusters using the small dataset and both AKU and TIL observations as anomaly candidates (as discussed in table 5.5)

A5 Tuning Parameters

Selected Tuning Parameters for the Models Presented in Table 5.4	
K-Means	k = 9 Cluster Size Threshold = 10%
DBSCAN	$\epsilon = 1.5$ MinPts = 10
OC-SVM	$\nu = 0.05$ $\gamma = 10$ Kernel = Sigmoid
Autoencoder	Epochs = 2000 Batch Size = 30 Learning Rate = 0.01 Activation Function = Tanh

Table A5.1: The tuning parameters used on the models whose results are presented in table 5.4. This is for the large dataset with 12 variables, and using both AKU and TIL observations as anomalies.

Selected Tuning Parameters for the Models Presented in Table 5.5	
K-Means	k = 11 Cluster Size Threshold = 5%
DBSCAN	$\epsilon = 0.5$ MinPts = 5
OC-SVM	$\nu = 0.1$ $\gamma = 1$ Kernel = Radial
Autoencoder	Epochs = 2000 Batch Size = 30 Learning Rate = 0.0001 Activation Function = RELU

Table A5.2: The tuning parameters used on the models whose results are presented in table 5.5. This is for the small dataset with 7 variables, and using both AKU and TIL observations as anomalies.

Selected Tuning Parameters for the Models Presented in Table 5.5	
DBSCAN	$\epsilon = 1.4$ MinPts = 10
OC-SVM	$\nu = 0.1$ $\gamma = 0.01$ Kernel = Sigmoid

Table A5.3: The tuning parameters used on the models whose results are presented in table 5.7. This is for the large dataset with 12 variables, and using only AKU observations as anomalies.

Selected Tuning Parameters for the Models Presented in Table 5.5 This refers to DBSCAN models on the simulated datasets	
base simulation	$\epsilon = 1.75$ MinPts = 3
10% num and 1pp cat	$\epsilon = 1.75$ MinPts = 3
50% num and 5pp cat	$\epsilon = 1.75$ MinPts = 3
100% num and 10pp cat	$\epsilon = 1.75$ MinPts = 3
200% num and 20pp cat	$\epsilon = 1.75$ MinPts = 3
50% num	$\epsilon = 1.75$ MinPts = 3
50pp cat	$\epsilon = 1.75$ MinPts = 3

Table A5.4: The tuning parameters used on the models whose results are presented in table 5.8

Selected Tuning Parameters for the Models Presented in Table 5.5 This refers to OC-SVM models on the simulated datasets	
base simulation	$\nu = 0.005$ $\gamma = 0.000001$ Kernel = Radial
10% num and 1pp cat	$\nu = 0.05$ $\gamma = 0.1$ Kernel = Radial
50% num and 5pp cat	$\nu = 0.05$ $\gamma = 0.1$ Kernel = Radial
100% num and 10pp cat	$\nu = 0.005$ $\gamma = 0.000001$ Kernel = Sigmoid
200% num and 20pp cat	$\nu = 0.05$ $\gamma = 0.1$ Kernel = Sigmoid
50% num	$\nu = 0.05$ $\gamma = 0.1$ Kernel = Radial
50pp cat	$\nu = 0.05$ $\gamma = 0.1$ Kernel = Radial

Table A5.5: The tuning parameters used on the models whose results are presented in table 5.9

Figure A5.1: NHH logo