LUND UNIVERSITY

SCHOOL OF ECONOMICS AND MANAGEMENT

Imbalanced Predictions

An Evaluation of Classification Techniques for Imbalanced Data

Stella Säfström

STAN40
Master's Thesis (15 credits ECTS)
June 2022
Supervisor: Jakob Bergman
Examiner: Björn Holmquist

Abstract

The aim of the thesis is to evaluate solutions to the class imbalance problem using real world data sets with varying degrees of class imbalance. The analysis is limited to binary classification. Three large data sets relating to credit card fraud, vehicle insurance and heart disease are used for the analysis.

Several methods are compared and evaluated. Logistic regression, SVC and decision trees are used as benchmark classifiers in order to compare these to imbalanced learning techniques. Random undersampling and SMOTE are used to to evaluate resampling techniques. Cost-sensitive versions of logistic regression, SVC and decision trees are used to evaluate cost-sensitive algorithms. The resampling techniques are also used in combination with the cost-sensitive algorithms. The results are evaluated using six measures: accuracy, recall, precision, F-measure, G-mean and AUC.

The conclusion of the thesis is that none of the methods evaluated outperforms all others. Depending on the data set used for analysis, the methods produced varying scores for the different evaluation measures. As an example of this, the method used to produce the highest precision score was not the same for the credit card fraud detection data and for the heart disease data. The analysis further showed that which evaluation measure to use depends on the goal of the analysis.

This shows that none of the evaluated techniques are optimal for all data sets. Depending on the data set used and the goals of the analysis, different methods and evaluation measures may be applied.

Keywords: Imbalanced data, cost-sensitive learning, SMOTE, random undersampling

Contents

List of Figures

List of Tables

Introduction

In the binary class setting, data sets with an imbalanced class distribution is characterised by having a majority of the observations belonging to one class and a smaller group of observations belonging to the other. In some cases the minority class can be comprised of less than one percent of the total number of observations. This can cause a problem when applying statistical learning methods as the models has relatively few observations to learn from to accurately classify the minority class.

This can be illustrated by a simple example. Imagine a data set with two classes, where one class accounts for 99 % of the observations, while the other only accounts for 1 %. In this case, a model may predict all observations as belonging to the majority class, leading to an accuracy score of 99 %. This is usually perceived as an excellent accuracy score. In this context however, this is a very poor result seeing as none of the samples from the minority class have been accurately classified.

Sun et al. (2007) note that in several real world applications imbalanced data cause problems for classifiers in regards to correctly classifying the minority class. This includes fraud detection, text classification and detecting rare medical conditions. They further state that classification algorithms and techniques such as neural networks, nearest neighbour, support vector machines and decision trees have been reported to be subpar when dealing with imbalanced data. It is therefore important to find and evaluate models that can handle this issue.

According to Sun et al. (2007) previous research has addressed the following aspects of the class imbalance problem:

1. In which domains class imbalances pose a problem for classifiers,
2. Potential solutions to this problem, and
3. Finding measures for evaluating classifiers' performance when dealing with imbalanced data.

This thesis will focus on evaluating solutions to the class imbalance problem using large real world data sets.

1.1 Research Problem

The aim of this thesis is to compare and evaluate methods for classification applied to large imbalanced real world data sets. The analysis is limited to binary classification problems and thus avoids multi-class classification applications.

The overarching research question is formulated as follows:

"Which is the best method to handle the class imbalance problem?"

The research question is narrowed down to answer the following:

"Which differences exist in performance of methods used for class imbalance for different data sets with varying degrees of class imbalance?"

"Which method used for class imbalance outperforms the others?"

How to evaluate which method performs "best" depends on the purpose of the analysis. Depending on whether it is important correctly classify as many as possible from the minority class or if it is of interest to find the model that creates a good balance between correctly classifying the minority and the majority class, different evaluation measures can be used. This aspect is discussed in more detail in the Methods chapter.

1.2 Aim and Scope

There are several aspects of the class imbalance problem that can be investigated. Which method to choose depends on the goal of the analysis. Creating synthetic data can be conducive for understanding which models perform best in different situations. However, creating data that mimics real world data is a difficult task. In this essay, the main goal is to evaluate how certain models behave in real world applications. Hopefully this can shed some light on which models to choose depending on the problem at hand.

Past research studies related to this topic have to a large extent focused on finding appropriate models to handle the class imbalance problem using either synthetic data, small data sets or a combination of the two. In this thesis, the aim is to evaluate existing models on large real world data sets in order to get an insight into how these models behave in these situations. Finding data sets that can be seen as typical for different applications is difficult. No attempts at stating which models perform best in given applications are made. Instead, the goal is to evaluate how models perform when applied to a number of real world data sets.

There are a variety of different approaches to the class imbalance problem. Only a handful are studied in this thesis. No attempt is made of determining which method is best of all available ones. Instead, a few are chosen and investigated. The chosen models and approaches to the imbalance problem evaluated are described in the Methods chapter.

1.3 Outline of the Thesis

The thesis is divided into the following parts:

1. **Introduction.**
2. **Theory.** Methods evaluated by other researchers are presented to show the context of the coming analysis.
3. **Data.** The data used in the analysis are introduced.
4. **Methods.** The scope of and the methods used in the analysis are presented.
5. **Empirical analysis.** The results from the analysis are displayed. Results are discussed.
6. **Conclusion.** Conclusions of the analysis are presented. Future research areas are proposed.

Theory

2.1 Previous Research

Several researchers have approached the problem of class imbalance. This chapter offers a brief overview of the class imbalance problem and some of the methods used to handle this particular challenge.

2.1.1 The Problem of Class Imbalance

Below follows an overview of types of class imbalance and some of the most common issues associated with classification tasks.

Types of class imbalance

Class imbalance can be categorised in different ways depending on the nature of the data and how the data have been collected. Depending on the type of class imbalance present in the data the difficulty of accurately distinguishing between classes can differ.

High class imbalance
Any data set with an unequal class distribution is imbalanced. The severity of such an imbalance can vary from minor to extreme. Some researchers have suggested that extreme or high imbalance can be of the order 100:1, 1000:1 and 10 000:1 (He and Garcia, 2009). However, this is not a strict definition. Others have suggested that high class imbalance is any imbalance that contributes to challenges regarding modelling and prediction of the minority class (Leevy et al., 2018).

Intrinsic class imbalance
Intrinsic class imbalance refers to imbalance that is a result of the nature of the data (He and Garcia, 2009). An example of this is rare diseases that only a small proportion of the population suffer from. This is thus a naturally occurring imbalance.

Extrinsic class imbalance
According to He and Garcia (2009) extrinsic class imbalance is not, as compared to intrinsic class imbalance, a result of the nature of the data. The original data may not be imbalanced, however, when the data is generated an imbalance can be created. According to the authors an example of this is data that is acquired from a stream of (balanced) data during some time interval, during which the transmission has random interruptions and some data fails to transmit. The data is thus

not originally imbalanced, however, when it is collected an imbalance can be created.

Relative class imbalance
Relative imbalance refers to data with a given proportion of the observations belong to the minority class (He and Garcia, 2009). An example of this is a data set with a class imbalance of 100:1, where there are 100 observations in the minority class if the size of the data set is 10 000 observations. If the number of observations is increased, the number of observations in the minority class is expected to rise proportionately. Thus in a data set of 200 000 observations, it would be expected that there are 2000 observations in the minority class. Now, the minority class might not be considered rare on its own, but rare relative to the majority class.

Class imbalanced due to rare instances
According to Weiss (2004) rare cases are comprised of "a meaningful but relatively small subset of the data" (p.7). Weiss (2004) makes a distinction between relative rarity (above described as relative class imbalance) and absolute rarity (here also noted as imbalance due to rare instances). For data sets with absolute rarity the number of observations from the minority class are very few, according to the author. The author notes that when dealing with these types of data sets the classifiers may have issues in distinguishing the two classes. A data set with relative class imbalance may be considered to have class imbalance due to rare instances if the data set includes only a few observations from the minority class. Thus in the example above, a data set with 10 000 observations out of which only 100 belong to the minority class, the class imbalance could still be regarded as imbalanced due to rare instances.

Challenges related to classification tasks

There are issues that are related not only to the class imbalance problem, but also to classification in and of itself. These are issues that are important to understand for all classification tasks including tasks with imbalanced data.

Data complexity and separability
Although class imbalance may hinder a classifier to accurately distinguish between the minority and majority class, this is not the only factor that causes such problems, according to He and Garcia (2009). The authors note that for some data sets with relative class imbalance, the classifier may still make accurate predictions of both classes. Instead they suggest that what they call data complexity is the primary reason why classifiers may fail at distinguishing the minority class. They further note the issues of data complexity present in a data set can be amplified when combined with relative class imbalance.

Data complexity entails a variety of issues such as small disjuncts and overlapping, according to He and Garcia (2009). The authors explain that classifiers try to create disjunct rules in order to accurately separate the classes. These rules are often ones that can cover large clusters of the data. When faced with small disjuncts there are several small clusters of the minority samples, making it difficult for the classifier to accurately distinguish the two.

Sun et al. (2007) note that order for a model to distinguish the two classes the classes need to be separable in some way. The more discriminative patterns an

algorithm can find, the easier it is for it to accurately predict to which class an observation belongs. A problem arises when the classes overlap and the algorithm subsequently fails at distinguishing the two classes.

Previous research by Japkowicz and Stephen (2002) has shown that if two classes are linearly separable, the class imbalance problem becomes much easier to solve. They note that as complexity increases the sensitivity to class imbalance increases as well.

Sample size
Sun et al. (2007) note that the more data that is available for training the model, the easier it is for the model to find discriminatory patterns in the data. Thus a smaller sample size can hinder a model's ability to correctly classify the minority class. A small sample size may also lead to issues regarding imbalance of rare instances. Previous research has shown that, unsurprisingly, higher class imbalance and complexity together with small training set makes classifier more sensitive to the problem (Japkowicz and Stephen, 2002).

2.1.2 Possible Solutions to the Class Imbalance Problem

According to Sun et al. (2007), the solutions to the class imbalance problem can be divided into two main categories: Data level approaches and Algorithm level approaches. There are also possibilities to combine the two into what here will be referred to as Hybrid methods.

Data level approaches

When faced with a data set with an uneven class balance a common approach is to use methods that aim at balancing the class distribution. These methods solely focus on the data distribution and are thus referred to as data level approaches. There are two main ways of rebalancing the class distribution: *Undersampling* and *Oversampling*.

According to Sun et al. (2007), undersampling is performed by eliminating observations from the majority class in order to create a more even distribution of classes. Oversampling, on the other hand, aims at generating new examples from the minority class. There are several ways to perform these resampling techniques. Some examples of these techniques are described below.

According to He and Garcia (2009), random oversampling refers to randomly adding replications of existing samples from the minority class to the data set. The authors describe random undersampling as removing randomly selected observations from the majority class. Undersampling can also be performed based on distance criteria, according to Branco et al. (2016). However, the authors note, these methods may be extensively time consuming when faced with large data sets. The authors mention other undersampling techniques, such as Tomek links, that decide which observations to remove with regards to noisy observations or regions with overlapping classes. The authors also mention oversampling techniques that generate synthetic observations, such as the synthetic minority oversampling technique (SMOTE).

An advantage of data level approaches is that they are applicable to existing classifiers, as it forces the model to become biased to correctly predict the minority

class (Branco et al., 2016).

There are some drawbacks to using resampling techniques according to Sun et al. (2007). They note that to find the ideal class distribution in the training data can be difficult as this is most often unknown, it can cause loss of information when the larger class is undersampled and overfitting can occur if the small class is oversampled.

Algorithm level approaches

According to Sun et al. (2007), algorithm level approaches are essentially composed of modifications of existing learning algorithms to accommodate to the class imbalance problem. They state that to use this approach a thorough understanding of the algorithms chosen is needed in order to tweak the algorithms to fit the problem accurately for the task at hand. One of the most common ways to modify these algorithms is to implement costs and/or benefits to show the usefulness of predictions. This can be done by using cost-sensitive algorithms. Here, the goal is to minimise the overall cost, knowing that misclassification of the smaller class is associated with a higher cost.

Branco et al. (2016) claim that one advantage of the algorithm level approach is that the goal of correctly classifying the smaller class is incorporated into the chosen model and that this leads to that the models become more comprehensible for the user. Some disadvantages noted by the authors are that the analyst becomes restricted in terms of which models can be used and can be forced to develop specific algorithms for the given problem. It also requires a high degree of knowledge of the models implemented for the given task.

Sun et al. (2007) further state that the cost of misclassification is dependent on the nature of the problem the analyst aims to solve. As an example they mention fraud, where the amount of money involved will affect the importance of correctly classifying a fraudulent case. Incorrectly classifying a valuable customer as fraudulent could cost the bank or institution more than the opposite. The same goes for correctly classifying a medical diagnosis, which depends on the patient and the severity of the disease. There are thus many complicating factors to take into account given the nature of the problem the analyst is set to solve.

Hybrid methods

Hybrid methods simply creates a mix of the different approaches described above in order to obtain optimal results. This can for example mean that the analyst uses resampling techniques together with algorithms appropriate for the specific data at hand.

2.1.3 Evaluation Measures

Accuracy measures the proportion of correctly classified observations. In highly imbalanced data sets, classifiers may predict all observations as belonging to the majority class. This can lead to a high accuracy that in a balanced data setting would be considered good. However, this evaluation metric is not appropriate when dealing with imbalanced data. This is due to that the accuracy metric fails to show to what extent the classifier accurately predicts observations as belonging to the

minority class. He and Garcia (2009) note that accuracy is a poor performance metric due to that it is sensitive to the data distribution. If the class distribution changes, so does the accuracy. This leads to problems when the goal is to evaluate classifiers on different data sets, according to the authors. If the majority class constitutes 99 % of the data in one data set, and 80 % in another, it would seem as if the classifier performs better in the first data set. This, they claim, makes a relative analysis inherently flawed when using the accuracy metric. Due to this, several other evaluation methods have been proposed in previous research on the topic. In this section, some of these measures are shown.

In order to find such measures a confusion matrix can be used (see Figure 2.1). A confusion matrix offers a clear view of the possible outcomes of a classifier. The vertical cells correspond to the values predicted by the classifier. In this context the positive class refers to the minority class and the negative class refers to the majority class. The horizontal cells correspond to the true values of the observations. If a classifier performs a perfect classification there will only be values in the true positive and true negative cells.

		Predicted Value	
		Positive	Negative
Actual Value	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 2.1: Confusion Matrix

From the confusion matrix, appropriate measures for the class imbalance problem can be created (Sun et al., 2007). In this section four such measures are described, namely: recall, precision, F-measure and G-mean. Further, ROC-analysis is described.

Recall

If the researcher is solely interested in the positive (minority) class prediction performance recall can be used. Recall, or true positive rate, shows the proportion of the observations in the positive class that are labelled accurately (He and Garcia, 2009).

$$Recall = TP_{rate} = \frac{TP}{TP + FN}. \tag{2.1}$$

Precision

Precision, or positive predictive value, measures exactness according to He and Garcia (2009). It is thus a measure of the proportion of observations predicted as positive that are actually positive.

$$Precision = PP_{value} = \frac{TP}{TP + FP}. \tag{2.2}$$

F-measure

Recall and precision are useful in different situations according to He and Garcia (2009). The authors claim that neither are perfect as an evaluation measure for imbalanced data due to their respective flaws. Recall does not offer any information about how many observations are incorrectly predicted as being positive and precision does not show how many of the positive observations are misclassified. In an effort to combine the two measures the F-measure (see Equation 2.3) was created by Rijsbergen (1979). This measure aims at measuring the effectiveness of a classification by combining recall (R) and precision (P) into an average and represents the harmonic mean between the two (He and Garcia, 2009).

$$\frac{2RP}{R+P}. \tag{2.3}$$

G-mean

If the classification performance of both classes are of interest, the aim is that both the true positive rate and the true negative rate (TN_{rate}) are high (Sun et al., 2007). In this case the geometric mean (G-mean) (Kubat et al., 1998) can be used, which balances the performance of a model between the classes (see Equation 2.6). This measure has the property of being robust when the distribution changes with time or if the distribution differs between the training and test data sets (Kubat et al., 1998).

$$TN_{rate} = \frac{TN}{TN+FP}. \tag{2.4}$$

$$\sqrt{TP_{rate} \times TN_{rate}}. \tag{2.5}$$

ROC Analysis

According to He and Garcia (2009) both the F-measure and the G-mean are ineffective in terms of answering more generic questions in regards to classification evaluation. Because precision, like accuracy, is sensitive to changes in data distributions, so is the F-measure. Instead of these measures ROC (Receiver Operating Characteristic) analysis can be performed.

According to Sun et al. (2007), in the case where a classifier outputs a probability score for the prediction the class prediction may be altered by changing the threshold. In ROC analysis the TP_{rate} and FP_{rate} are plotted for different threshold values. The ROC curve (see Figure 2.2) thus shows the relative trade-offs between true positives (benefits) and false positives (costs) for different threshold values. A perfect classifier will create a ROC curve that hugs the top left corner, as it obtains a true positive rate of 1 and and a false positive rate of 0 (James et al., 2021, p.150-151). The blue curve in Figure 2.2 represents this ideal. A classifier that is as good as pure chance is a diagonal from the bottom left corner to the top right corner (James et al., 2021, p.150-151), as is shown by the black line in Figure 2.2.

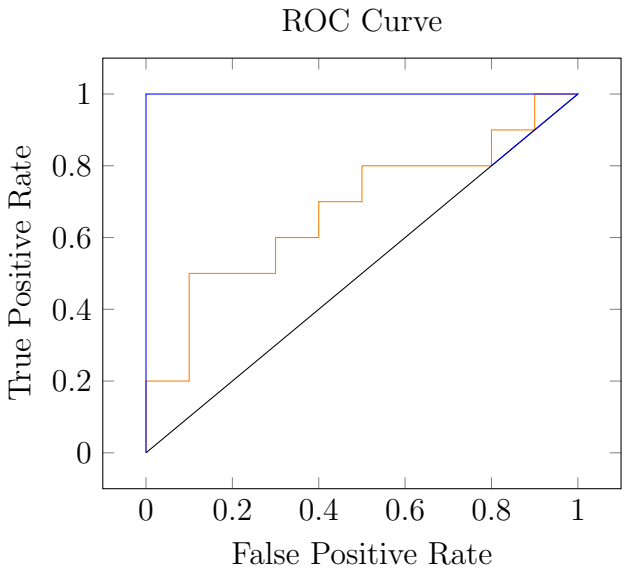

Figure 2.2: Example of an ROC Curve. The blue line represents the best possible classifier. The black line shows a classifier as good as pure chance. The orange curve shows how different threshold values may create different false positive and true positive rates.

Although the ROC curve can provide a good summary of a classification model, it can be difficult to use the ROC curves as a means of comparing different classifiers (Provost and Fawcett, 1997). Instead, the Area Under the ROC Curve (AUC) can be used according to Branco et al. (2016). According to the authors, the AUC provides a single metric to evaluate a classifier's performance on average.

$$AUC = \frac{TP_{rate} + TN_{rate}}{2}. \tag{2.6}$$

Data

Three data sets have been selected to test imbalanced learning techniques on real world data (see Table 3.1). The data sets are publicly available at Kaggle.com. All data sets are relatively large with varying degrees of imbalance. The number of observations in the minority class are also very different in the data sets. By comparing the performance of the models on these data sets it can hopefully shed some light on how class imbalance and number of observations affect the performance. This could potentially give indications on which model to use under certain circumstances.

From what is known about the data sets they all have intrinsic and relative class imbalance. There is nothing to suggest that the issues are extrinsic from the information available. The credit data could be viewed as imbalanced due to rare instances, given the very small number of observations in the minority class. How separable and complex the data sets are is not known. There are no missing values in any of the data sets.

Table 3.1: Data sets used for analysis

Data set name	Number of obs	Percentage of minority class	Number of obs in minority class	Number of variables
Credit Card Fraud Detection Data	284 807	0.178	492	30
Imbalanced Insurance Data	382 125	16.4	62 531	9
Personal Key Indicators of Heart Disease	319 795	8.6	27 373	17

Credit Card Fraud Detection Data

The credit card fraud detection data set (Machine Learning Group - ULB, 2015) was originally collected and used for analysis by the Machine Learning Group of Université Libre de Bruxelles and Worldline. The data is comprised of 284 807 transactions (observations) made by European cardholders' credit cards in September 2013, out of which 492 are fraudulent. This is thus deemed a highly imbalanced data set with only 0.178 % of the observations belonging to the minority group. Several other researchers have used this data (Dal Pozzolo et al.; Dal Pozzolo et al.; Dal Pozzolo et al.; Carcillo et al.; Lebichot et al.; Lebichot et al.; Carcillo et al.; Lebichot et al., 2015; 2014; 2017; 2017; 2019a; 2019b; 2019; 2021).

28 out of the original variables have been masked because of privacy reasons. The features are named V1, V2, ..., V28 and consists of the principal components. Two variables still have their original names and are not transformed via principal component analysis (PCA): Time and Amount. Time refers to the seconds elapsed between each transaction and the first transaction in the data set. Amount refers to the transaction amount. The response variable is named "Class" and has value 1 if the transaction is fraudulent and 0 otherwise.

Imbalanced Insurance Data

The data (Möbius, 2020) has been uploaded by a user on Kaggle.com. The source of the data is unknown. According to the contributor the data comes from an insurance company. With this data it is possible to predict if a customer is interested in purchasing Vehicle Insurance. The variables "id" and "region_code" have been removed in the analysis. The variables used in the analysis are shown in Table 3.2.

Table 3.2: Insurance Data Variables

Variable	Type	Description
Gender	Dummy (Male/Female)	Gender of customer
Age	Numerical	Age of customer
Driving License	Dummy (Yes/No)	Indicates if customer has a driver's license
Region Code	Numerical	Region code of customer
Previously Insured	Dummy (Yes/No)	Indicates if customer is previously insured
Vehicle Age	Dummy ($<$ 1 year, 1 - 2 years, $>$ 2 years)	Age of customer's vehicle
Vehicle Damage	Dummy (Yes/No)	Indicates if the customer's vehicle is damaged
Annual Premium	Numerical	Value of customer's premium
Vintage	Numerical	Days insured until now

Personal Key Indicators of Heart Disease

The data (Pytlak, 2022) is provided by the American governmental agency Centers for Disease Control and Prevention (CDC). The data was collected in 2020 from the Behavioral Risk Factor Surveillance System (BRFSS), which conducts annual telephone surveys regarding the health of U.S. citizens. The original data set contained 401 958 observations and 279 variables. In the data set on Kaggle.com the data has been reduced to 319 795 observations and 17 variables. The purpose of this data set is to predict whether respondents have had coronary heart disease (CHD)

or myocardial infarction (MI). In the available data 8.6 % of the respondents have reported having had heart disease. The variables used in analysis can be found in Table 3.3

Table 3.3: Heart Disease Data Variables

Variable	Type	Description
Sex	Dummy (Male/Female)	Gender of respondent
Age Category	Categorical (14 categories)	Age of respondent
BMI	Numerical	BMI of respondent
Smoking	Dummy (Yes/No)	Indicates if respondent has smoked 100 cigarettes in their life
Alcohol Drinking	Dummy (Yes/No)	Indicates if patient is a heavy drinker
Stroke	Dummy (Yes/No)	Indicates if a respondent has had a stroke
Physical Health	Numerical	How many days the past month the physical health was bad
Mental Health	Numerical	How many days the past month the mental health was bad
Difficulty Walking	Dummy (Yes/No)	Indicates if respondent has a difficulty walking
Race	Categorical	Indicates race of respondent
Diabetic	Dummy (Yes/No)	Indicates if respondent is/have been diabetic
Physical Activity	Dummy (Yes/No)	Indicates if respondent has been doing physical activity the past 30 days
General Health	Categorical	Indicates the respondent's general health
Sleep Time	Numerical	Indicates how many hours per day respondent sleeps
Asthma	Dummy (Yes/No)	Indicates if respondent has/have had asthma
Kidney Disease	Dummy (Yes/No)	Indicates if respondent has/have had kidney disease
Skin Cancer	Dummy (Yes/No)	Indicates if respondent has/have had skin cancer

Methods

In this chapter, the methods used for the experimental study are described. An overview of statistical learning is provided. Next, an outline of the empirical analysis is shown. The programming tools used for analysis are described. Data preprocessing and classification methods are accounted for.

4.1 Statistical Learning

The goal of statistical learning is to in one way or another understand data. Statistical learning techniques are divided into two groups: *Unsupervised* and *Supervised Learning* (James et al., 2021, p.1). In this thesis only supervised learning techniques are investigated.

In supervised learning the goal is to predict some kind of outcome or output based on a number of features or inputs (Hastie et al., 2009, p.1-2). The outcome can be quantitative or in the case of classification, categorical. The data used to create the prediction model (learner) is the training data. The trained learner can then be used to predict outcomes on previously unseen data, referred to as test data. Before commencing the analysis available data is divided into training and test data. The models are trained on the training data to find distinguishing patterns. Next, this trained model is tested on the remaining data to evaluate the model's performance.

In this thesis the focus lies on supervised learning with categorical outcomes. There are a number of classification methods included in the statistical learning framework such as Linear Discriminant Analysis (LDA); Logistic Regression; Support Vector Machines; Tree Based Methods; and Deep Learning. Only a handful of the available methods will be investigated in the experimental study.

The learning tasks can be described as using an input vector X to make predictions of the output Y, denoted $\hat{Y}$ (Hastie et al., 2009, p.10-11). When prediction concerns binary classification, $\hat{Y}$ will lie in $\{0, 1\}$, i.e. either take on a value of 0 (majority class) or 1 (minority class).

4.2 Outline of empirical analysis

Figure 4.1 displays a flowchart of the empirical analysis. The different aspects of the analysis is further presented in this section.

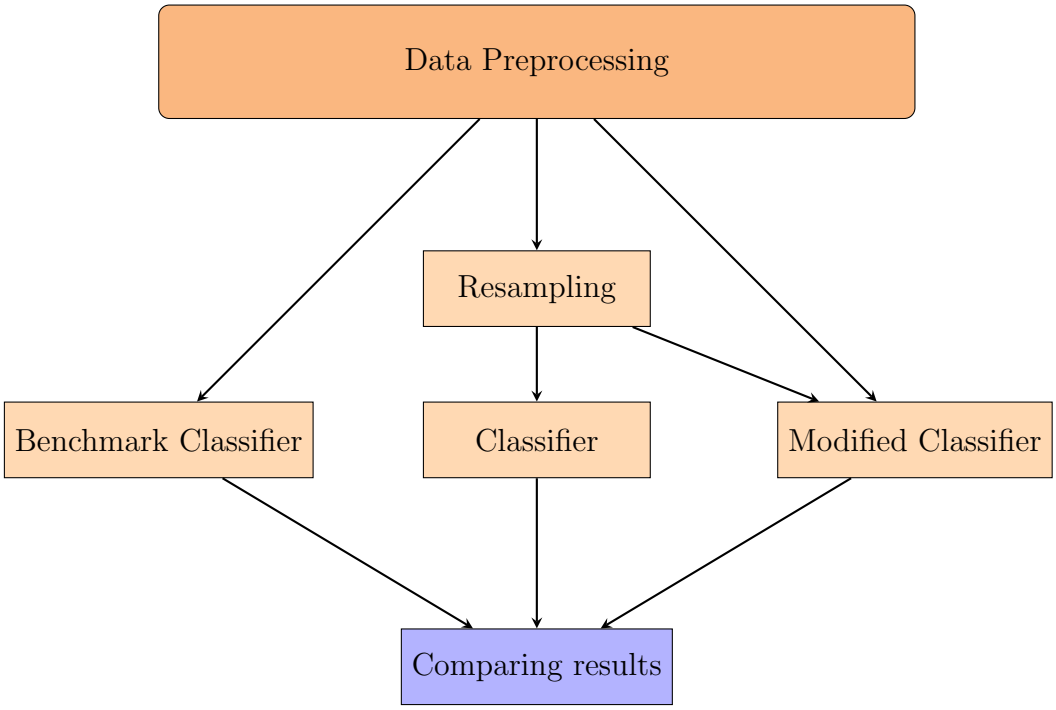

Figure 4.1: Method for empirical analysis

The analysis is broken down into the following steps:

1. The three data sets are preprocessed in order to function with the chosen models.
2. Three benchmark classifiers are used to evaluate how the classifiers behave with the imbalanced data sets.
3. Two resampling methods are used on the training data. The resampled training data is then used to train the classifiers.
4. Modified classifiers that take into account the class imbalanced is used on the original data.
5. A combination of resampling and modified classifiers are used.
6. Finally, the outcome of these techniques are compared and discussed.

4.2.1 Overview of the methods used

The methods evaluated in the empirical analysis are:

- Benchmark classifiers: linear SVM (SVC), logistic regression and classification trees
- Resampling methods: random undersampling and SMOTE in combination with benchmark classifiers
- Cost-sensitive versions of the classifiers (SVC, logistic regression and classification trees)
- Resampling methods in combination with cost-sensitive classifiers

The methods have been chosen in order to see how a few of the existing methods used for imbalanced learning behave. There a numerous ways to tweak and change the chosen algorithms in order to create more powerful classifiers. Such improvements will not be attempted here. Instead, the "out-of-the-box" classifiers will be used to get an overview of relatively simple classifiers. This is done in order to limit the scope of the thesis and also to see how the standard versions of these classifiers behave. In a real world application, these classifiers would need to be tweaked in order to create the best possible outcomes for the specific data at hand.

The aim of the empirical analysis is twofold:

1. First, the aim is to compare and contrast how resampling and cost-sensitive learning behave.
2. Second, the aim is to evaluate how these methods behave on different data set types with varying class imbalance.

Limitations of the method

As noted in the Theory chapter, there are numerous classification methods for imbalanced learning. The scope of the analysis needs to be narrowed down in order to be able to make some sort of evaluation of such methods. Here, methods that have similar approaches are compared. Both cost-sensitive versions of classifiers and resampling techniques aim at taking the imbalance found in the data into account. Of course, it would be interesting if other types of methods were investigated. In order to limit the scope of the thesis, this is not attempted here, which can be viewed as a limitation. However, hopefully, this analysis may still provide some interesting insights for the community.

4.2.2 Programming

The analysis is performed using the open software program Python. The library mainly used is scikit-learn (Pedregosa et al., 2011). For decision trees the Decision Tree Classifier in scikit-learn is used. For the resampling tasks, scikit imbalanced-learn (Lemaître et al., 2017) is used.

To implement SVM in scikit-learn (Pedregosa et al., 2011) there are two libraries that can be used: LIBLINEAR (Fan et al., 2008) and LIBSVM (Chang and Lin, 2011). The chosen library is LIBLINEAR, seeing as it is efficient in when dealing with large-scale data. The library supports linear SVM (SVC) and logistic regression, which is what is used in this research.

4.2.3 Data Preprocessing

In order to use the chosen classifiers some data preprocessing need to be performed. All categorical variables are coded into dummy variables. Further, all continuous variables are scaled. This is done in order to speed up the computations as the large data sets require a substantial amount of computing power in the case of logistic regression. Support vector classification requires scaled variables.

4.2.4 Algorithm level methods

In this section the classifiers used are briefly described. The modifications of the classifiers are also described.

Logistic Regression

Logistic regression uses a set of training observations $(x_1, y_z)...(x_n, y_n)$ to build a classifier. The model determines a probability of an observation belonging to a particular category (James et al., 2021, p.134). This probability can be written as $p(X) = Pr(Y = 1|X)$, i.e. the probability that Y belongs to class 1 given X. The probability ranges between 0 and 1. In the most common case, a probability larger than 0.5, i.e. $p(X) > 0.5$, means that the observation will be classified as belonging to class 1. Other threshold values can be used with different results as shown by the ROC curve in Figure 2.2.

In order to achieve outputs between 0 and 1 the logistic function (James et al., 2021, p.134) is used:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \tag{4.1}$$

The maximum likelihood method is used to fit the model 4.1. The goal is to find the parameters β that maximizes the likelihood (James et al., 2021, p.135). This is equivalent to minimizing the negative log likelihood, which is the cost function of logistic regression according to li Zhang et al. (2021). The authors further note that the negative log likelihood can be divided into two parts: the costs for misclassifying majority observations and costs for misclassifying minority observations. The model assumes that both costs are equal, leading to the model maximizing the overall accuracy.

Cost-sensitive logistic regression
Logistic regression in its original form is not optimal for class imbalance problems, seeing at it is biased towards the majority class. In order to combat this problem, penalty weights can be added to the log likelihood function, ensuring that misclassification of the minority class has a higher cost than the reverse (li Zhang et al., 2021).

LIBLINEAR offers a setting to choose the penalty weights (called class weights) in logistic regression. In the analysis, the weights are set to be balanced, i.e. take into account the class imbalance in the training data.

Support Vector Machines

The SVM algorithm was originally developed for binary classification. The classes are called the positive and negative class with class labels 1 and -1 respectively (Lin et al., 2002).

In order to understand support vector classifiers, the hyperplane must first be understood. The hyperplane in p-dimensional space refers to a flat affine subspace of dimension $1 - p$ (James et al., 2021, p.368). For $p = 2$, i.e. two dimensions, the hyperplane can be described by the following equation:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_x = 0. \tag{4.2}$$

for parameters β_0, β_1 and β_2. Equation 4.2 defines the hyperplane such that if a point $X = (X_1, X_2)^T$ satisfies 4.2 then X is on the hyperplane (James et al., 2021, p.368). However, if X does not satisfy 4.2 and instead is larger than or smaller than zero, X does not lie on the hyperplane.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_x > 0. \tag{4.3}$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_x < 0. \tag{4.4}$$

The hyperplane creates a dividing border between two halves of the p-dimensional space. By calculating the sign of 4.2 it is thus possible to decide at which side of the hyperplane a point lies (James et al., 2021, p.368). Depending on which side of the hyperplane an test observation lies, it is classified as belonging to one of the two classes.

If the classes in the data can be perfectly separated by a hyperplane there will exist an infinite number of such hyperplanes. The best choice of such hyperplanes is commonly referred to as the *maximal margin hyperplane* (James et al., 2021, p.371), see Figure 4.2. This is the separating hyperplane that lies farthest from the training observations. The margin is the distance between the training observations on either side of the hyperplane and the hyperplane. These training observations are shown in red in Figure 4.2 and the hyperplane is shown by the straight black line. The maximal margin hyperplane is the one that creates the largest margin of all possible hyperplanes. The training observations that are on the border of the margin (the dashed lines) are referred to as *support vectors* (James et al., 2021, p.368).

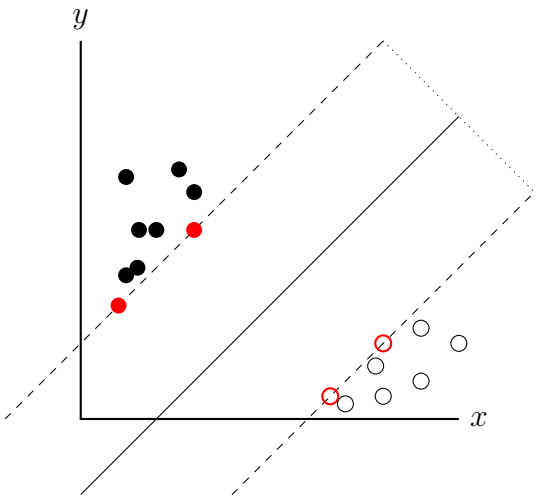

Figure 4.2: Maximal margin hyperplane. The black diagonal line shows the hyperplane. The filled points refer to points in one class and the circles refer to points in the other class. The red circles and filled points lie on the dotted lines (the margins) and refers to support vectors.

In most situations, the maximal margin hyperplane cannot fit a line that separates all observations of the two classes. There is thus needed some acceptance of misclassification. The support vector classifier (or soft margin classifier) allows for some observations to be placed on the wrong side of the margin and hyperplane (James et al., 2021, p.373-375).

Support vector machines is a classifier that allows for non-linear decision boundaries (James et al., 2021, p.379). In this thesis, the linear support vector classification (SVC) is used by implementing the LIBLINEAR library in scikit-learn.

Cost sensitive SVC
An assumption of SVM is that the cost of misclassification is the same for both classes (Lin et al., 2002). However, as have been pointed out, that may not be the case. Much like in cost-sensitive logistic regression, the cost of misclassification can be altered by taking into account the class imbalance in the data. LIBLINEAR allows for changing the class weight in order to increase the accuracy of prediction of the minority class.

Classification Trees

Tree based methods can be used for both classification and regression tasks. Here, classification trees are described.

The aim of the classification tree is to find a systematic approach to predicting classes given some set of measurements. The decision tree algorithm applied in scikit-learn is the CART-algorithm (Classification and Regression Trees) (Breiman et al., 2017). The CART decision splits the data into smaller parts by asking "yes" or "no" questions. In the example in Figure 4.3 the first question is: "Is the patient younger than 85 years old?" Depending on the answer to this question, the tree can either end up in a terminal node, where a classification is made, or continue to another question or node. The algorithm searches the available variables to find the

optimal split in the data. The optimal split is one that divides the data into two parts with the highest possible homogeneity (Razi and Athappilly, 2005).

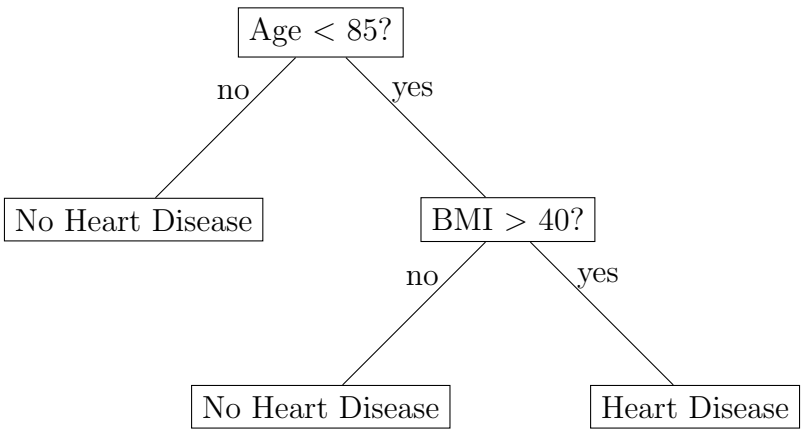

Figure 4.3: Example of a classification tree

Cost sensitive decision trees
In the decision tree classifier in sci-kit learn, the cost of misclassification can be changed, much like in the LIBLINEAR library described earlier.

4.2.5 Data level methods

Random Undersampling

There is a long list of methods used for undersampling the majority class. In the coming analysis random undersampling is used. Random undersampling simply randomly removes observations belonging to the majority class in order to create a more balanced distribution (He and Garcia, 2009). In the analysis scikit imbalanced-learn's RandomUnderSampler is used.

SMOTE

Oversampling by replication (random oversampling) can lead to overfitting. In order to improve generalization SMOTE was created. SMOTE (Synthetic Minority Oversampling TEchnique) (Chawla et al., 2002) is an oversampling method where the minority class is oversampled by generating synthetic samples. The oversampling is done by taking the minority class sample and creating synthetic examples by means of interpolation of neighbouring minority class observations (Fernández et al., 2018).

There are numerous extensions and modifications of the SMOTE algorithm. In the empirical analysis the regular SMOTE algorithm is used.

Analysis

In the this chapter the results of the analysis are presented. In all of the analyses the data are randomly divided into training and test data sets with 75 % of the data belonging to the training data and 25 % to the test data. The classifiers are applied to the training data in order to learn a classification rule that is then applied to assign class labels to new unseen observations in the test data. The classifications are evaluated using the metrics accuracy, precision, recall, F-measure, G-mean and AUC. The results of these metrics are presented in tables below. The presentation of the results are followed by a discussion of the findings.

The different evaluation metrics are shown in order to compare and contrast these measures. Depending on the goal of the analysis, different measures may be appropriate. The aim of visualising several metrics at once is also to show how these measures are related to one another.

In Table 5.1 the number of observations and the proportion of the majority class in the test data sets are presented. This information can aid in understanding how the models behave.

Table 5.1: Proportion of majority class in the test data used for analysis

Data set name	No of observations	Proportion of majority class
Credit Card Fraud Detection Data	71 202	0.9984
Imbalanced Insurance Data	95 004	0.8344
Personal Key Indicators of Heart Disease	79 949	0.9131

5.1 Results

5.1.1 Benchmark classifiers

In order to get a view of how the classifiers behave without any changes on data- or algorithm level, three benchmark classifiers are applied to the training data sets. The results are shown in Table 5.2.

Table 5.2: Results: Benchmark Classifiers

Decision Trees (Unmodified)						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9993	0.8316	0.6991	0.7596	0.8360	0.8494
Insurance Data	0.8352	0	0	0	0	0.5000
Heart Data	0.9136	0.5466	0.0312	0.0591	0.1765	0.51438
Logistic Regression (Unmodified)						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9992	0.8500	0.6018	0.7047	0.7757	0.8008
Insurance Data	0.8347	0.4927	0.1007	0.1673	0.3141	0.5401
Heart Data	0.9142	0.5346	0.1000	0.1686	0.3150	0.5458
SVC (Unmodified)						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9993	0.8404	0.6991	0.7633	0.8360	0.8495
Insurance Data	0.8339	0.4700	0.0601	0.1066	0.2436	0.5234
Heart Data	0.9143	0.5996	0.0403	0.0756	0.2005	0.5189

The results of the benchmark classifiers for the credit data shows that the accuracy is higher than the proportion of the majority class in the test data. It can also be seen that the true positive rate, recall, is between ca 0.6 and 0.7. This shows that the models have succeeded in correctly predicting a relatively high proportion of the observations in the minority class. It can also be noted that precision, F-measure and G-mean are relatively high, indicating that there is a decent balance between correctly predicting both classes. It can also be noted that the G-mean is consistently higher than the F-measure and that the AUC scores are quite high.

The same high values are not obtained for the remaining two data sets. Here, the accuracy rate is very close to the proportion of the majority class. This indicates that the model has simply predicted most of the observations as belonging to the majority class. This can further be shown if looking at recall, which have values close to or equal to zero. This shows that a low proportion of the observations in the minority class are correctly labelled. The decision tree algorithm for the insurance data is shows the extreme case where all observations are predicted to belonging to the majority class, leading to that precision, recall, G-mean and F-measure all have values of 0.

The classification tree has the worst performance for the heart and insurance data sets. It seems as this classifier is very sensitive to imbalanced data. In contrast, logistic regression seem to perform slightly better.

5.1.2 Data level methods

The results of the data level methods is shown in Table 5.3 and 5.4. Again, there is an obvious distinction in how these methods behave depending on the data set used.

Table 5.3: Results: Random Undersampling

Classifier: Decision Tree						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9451	0.0255	0.9027	0.0496	0.9237	0.9236
Insurance Data	0.7586	0.3995	0.9240	0.5578	0.8190	0.8250
Heart Data	0.7524	0.1991	0.6122	0.3005	0.6847	0.6890
Classifier: Logistic Regression						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9706	0.0480	0.9292	0.0912	0.9497	0.9499
Insurance Data	0.7040	0.3540	0.9649	0.5180	0.7935	0.8087
Heart Data	0.7501	0.2266	0.7774	0.3509	0.7623	0.7625
Classifier: SVC						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9718	0.0500	0.9292	0.0948	0.9503	0.9506
Insurance Data	0.7021	0.3538	0.9776	0.5196	0.7957	0.8127
Heart Data	0.7479	0.2252	0.7794	0.3494	0.7620	0.7622

Table 5.4: Results: SMOTE

Classifier: Decision Tree						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9157	0.0174	0.9381	0.0341	0.9268	0.9269
Insurance Data	0.7457	0.3865	0.9249	0.5452	0.8105	0.8176
Heart Data	0.7031	0.1705	0.6255	0.2680	0.6667	0.6680
Classifier: Logistic Regression						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9736	0.0531	0.9292	0.1005	0.9511	0.9514
Insurance Data	0.7040	0.3537	0.9624	0.5173	0.7927	0.8077
Heart Data	0.7252	0.2000	0.7214	0.3132	0.7235	0.7235
Classifier: SVC						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9787	0.0645	0.9204	0.1206	0.9491	0.9496
Insurance Data	0.7020	0.3537	0.9773	0.5195	0.7956	0.8125
Heart Data	0.7204	0.1978	0.7258	0.3109	0.7229	0.7229

SMOTE and random undersampling performs similarly for all three data sets and models. For the credit data, the accuracy is lowered slightly compared to the benchmark classifiers. The precision has gotten close to zero and the recall rate has increased to above 90 % for all models. This indicates that the models predict many more observations as belonging to the minority class. However, this is done at a great cost. The false positive rate has increased tremendously, meaning that the model predicts a large number of observations as being fraudulent, that are in fact not. As an example see Figure 5.1. The figure depicts the confusion matrix for classification trees with random undersampling for the credit data. Here, the model accurately predicts 102 of the fraudulent cases as fraud. However, the cost of increasing the true positive rate is that now 3 939 observations are incorrectly classified as fraud.

This kind of misclassification can cause a large cost to the bank or institution if these many non-fraudulent cases are deemed fraud.

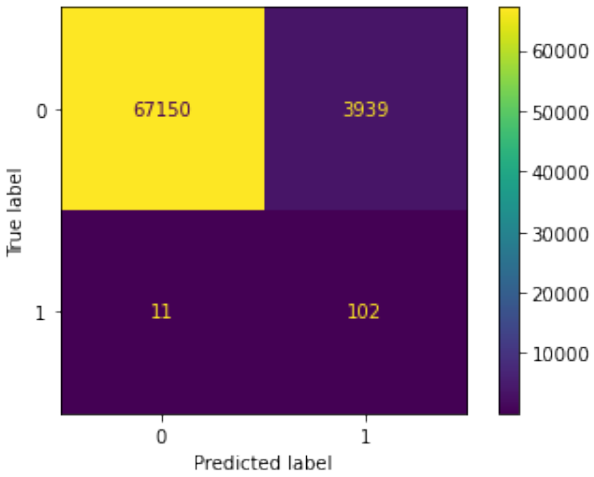

Figure 5.1: Confusion Matrix: Credit data, decision trees with Random Undersampling

An interesting observation to make here is the relationship between the F-measure and the G-mean. The values for the F-measure in the credit data is close to zero for the resampled credit data. The G-mean on the other hand has increased from around 0.8 up to over 0.92. The F-measure reflect the low precision values, whereas the G-mean does not. This may be due to that the false positive rate does not influence G-mean to the same extent as it influences the F-measure. Thus, if the false positive rate is of great interest to the analyst, the F-measure may be appropriate to use for evaluation.

For the remaining data sets all measures except for accuracy and precision has increased. Again, the G-mean is higher than the F-measure.

Undersampled decision trees seem to have the best overall performance for the insurance data. Logistic regression and SVC perform relatively similar for the insurance data. For the heart data, the best overall performance is obtained by logistic regression and SVC.

5.1.3 Algorithm level methods

In this section, modifications of the algorithms are made in order to account for the class imbalance. The modified algorithms are then applied on resampled training data to see how this further affect the results. The modifications made to the algorithms are that they are cost-sensitive, i.e. that they take into account the class imbalance present in the data when creating classification rules.

Modified Decision Trees

The results of the cost-sensitive decision trees can be found in Table 5.5. For the credit data, it can be noted that the results vary depending on if the modified decision tree is used on resampled training data or not. Without resampling, the modified decision trees lead to a good overall performance. Precision and recall are both relatively high, which is shown by the high F-measure value. The G-mean is still higher than the F-value and the AUC value is also high. However, when combined

with random undersampling something changes. The precision is lowered, leading to a low F-measure value. The recall is suddenly much higher leading to both a high G-mean and AUC value. When combined with SMOTE the precision is also lower, but not as low as with random undersampling. It seems, for modified decision trees, the best option is to avoid resampling for the credit data in order to get the best overall performance.

The same does not hold for the other two data sets. Here the F-measure is actually lower for the modified decision tree without any resampling. The G-mean and AUC values are also lower. For these data sets the best overall performance is given by the cost-sensitive decision trees together with random undersampling.

Table 5.5: Results: Cost-sensitive Decision Tree

Without Resampling						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9990	0.6860	0.7345	0.7094	0.8568	0.8670
Insurance Data	0.8051	0.4089	0.4086	0.4087	0.6008	0.6460
Heart Data	0.8638	0.2228	0.2281	0.2254	0.4616	0.5762
With Random Undersampling						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.8846	0.0120	0.8850	0.0238	0.8848	0.8848
Insurance Data	0.7563	0.3772	0.7352	0.4986	0.7477	0.7478
Heart Data	0.6749	0.1632	0.6641	0.2619	0.6700	0.6700
With SMOTE						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9977	0.3909	0.7611	0.5165	0.8716	0.8796
Insurance Data	0.7998	0.4016	0.4382	0.4191	0.6178	0.6547
Heart Data	0.8236	0.1894	0.3139	0.2362	0.5232	0.5930

Modified Logistic Regression

The results of applying cost-sensitive logistic regression to the data sets can be found in Table 5.6. In regards to the credit data, cost-sensitive logistic regression behave similarly with and without resampling. For all of these methods the precision is very low and recall very high, leading to a low F-measure. Interestingly, both the G-mean and the AUC are much higher than for the cost-sensitive decision tree, at values around 0.95.

The results of cost-sensitive logistic regression combined with SMOTE and random undersampling are almost exactly the same as regular logistic regression with these resampling techniques for all data sets. It thus seems as if cost-sensitive logistic regression behave much in the same way as regular logistic regression combined with resampling. Thus it seems unnecessary to combine these techniques.

However, there is a difference between regular logistic regression and cost-sensitive logistic regression. For the credit data, regular logistic regression outperforms all cost-sensitive logistic regression and logistic regression combined with resampling techniques. For the heart and insurance data, there the cost-sensitive logistic regression outperforms the regular logistic regression (without resampling).

Table 5.6: Results: Cost-sensitive Logistic Regression

Without Resampling						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9758	0.0571	0.9204	0.1075	0.9477	0.9481
Insurance Data	0.7035	0.3536	0.9653	0.5176	0.7932	0.8086
Heart Data	0.7490	0.2257	0.7775	0.3499	0.7617	0.7619
With Random Undersampling						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9706	0.0480	0.9292	0.0912	0.9497	0.9499
Insurance Data	0.7040	0.3540	0.9649	0.5180	0.7935	0.8087
Heart Data	0.7501	0.2266	0.7774	0.3509	0.7623	0.7625
With SMOTE						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9736	0.0531	0.9292	0.1005	0.9512	0.9514
Insurance Data	0.7040	0.3537	0.9624	0.5173	0.7927	0.8077
Heart Data	0.7252	0.2000	0.7214	0.3132	0.7235	0.7235

Modified SVC

Just as with cost-sensitive logistic regression, cost-sensitive SVC result in very similar measures when combined with SMOTE and random undersampling, respectively.

For the credit data, cost-sensitive SVC result in lower precision and higher recall than regular SVC. G-mean and AUC are also higher for cost-sensitive SVC, however, F-measure is slightly lower. Precision is lower when SVC (both cost-sensitive and regular) is combined with either random undersampling and SMOTE. G-mean and AUC are both close to 0.95.

For the other two data sets the cost-sensitive SVC and SVC combined with random undersampling and SMOTE yield similar results. It is hard to state a clear winner among these. The only thing that is clear is that cost-sensitive SVC or SVC used on resampled data outperform the regular SVC classifier.

Table 5.7: Results: Cost-sensitive SVC

Without Resampling						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9992	0.7177	0.7876	0.7511	0.8873	0.8936
Insurance Data	0.7021	0.3540	0.9791	0.5200	0.7962	0.8133
Heart Data	0.7471	0.2246	0.7793	0.3486	0.7614	0.7616
With Random Undersampling						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9718	0.0499	0.9292	0.0947	0.9503	0.9505
Insurance Data	0.7021	0.3538	0.9776	0.5196	0.7957	0.8127
Heart Data	0.7479	0.2252	0.7794	0.3495	0.7620	0.7622
With SMOTE						
Data set name	Accuracy	Precision	Recall	F-measure	G-mean	AUC
Credit Data	0.9789	0.0651	0.9204	0.1216	0.9492	0.9497
Insurance Data	0.7020	0.3537	0.9773	0.5194	0.7956	0.8125
Heart Data	0.7204	0.1978	0.7258	0.3109	0.7229	0.7229

5.2 Discussion

In this section some of the results found in the empirical analysis are discussed. Below, the highest measurements of recall, precision, F-measure, G-mean and AUC for the different data sets are presented.

Recall
Recall shows the proportion of observations in the minority class that are correctly labelled. It takes into account the true positives and the false negatives. The more false negatives present, the lower the recall will be. A high recall score thus show that there are relatively few observations in the minority class that are labelled as belonging to the majority class.

The best method for the credit data in terms of recall is decision trees combined with SMOTE with a value of 0.94. For the insurance data the methods that gave the highest recall score of around 0.98 was cost-sensitive SVC and SVC combined with SMOTE and random undersampling. For the heart data, the best recall measure was about 0.78 with similar results found using cost-sensitive SVC, cost-sensitive logistic regression, and undersampled SVC and logistic regression. This shows that using the methods chosen in the analysis the heart data obtained relatively more false negatives than the other data sets.

Precision
The precision score shows the proportion of observations predicted as belonging to the minority class that actually belong to the minority class. The fewer false positives present, the higher the precision score will be.

The best precision score of 0.85 for the credit data was obtained when using unmodified logistic regression. The other unmodified classifiers gave similar results of above 0.83. For the insurance data the best precision score was 0.49 which was obtained using unmodified logistic regression. The worst however, was the unmodified decision tree, leading to a precision score of 0. The best precision for the heart data was almost 0.6. This was obtained from the unmodified SVC.

This shows that the highest precision rates were all given by an unmodified version of the classifiers. However, which classifier that performed best related to this measure differs. These scores further show that both the heart data and the insurance data had relatively higher false positive values than the credit data.

F-measure
Seeing as the F-measure combines recall and precision into one score, it accounts for both the issue of false negatives and false positives. Both of these values are important. An example of this is fraud detection, where it is important to avoid false negatives (i.e. that classifying a transaction that is fraudulent as non-fraudulent) and false positives (i.e. classifying non-fraudulent transactions as fraudulent). Both of these of types misclassification can cause harm to individuals and institutions.

The best F-measures for the credit data is the unmodified SVC and classification tree at a value of about 0.76. This is also the highest F-measure value for all data sets. The highest value for the F-measure for the insurance data was around 0.55 which was obtained for resampled versions of classification trees. The best F-measure for the heart data was around 0.35 obtained with undersampled versions

of logistic regression and SVC. When looking at these results, it seem that none of the models used for the heart and insurance data sets were very successful.

G-mean

The G-mean takes into account both the true negative rate and the true positive rate. The true negative rate measures the proportion of observations in the minority class that is correctly predicted (see Equation 2.4). Like precision, the true negative rate includes false positive values. Thus, both the G-mean and the F-measure accounts for false positives to some extent. The F-measure scores are low when precision values are low. However, when precision values are low, the G-mean scores are still high compared to the F-measure. Seeing as precision is low when there are many false positives present, it seems as if the G-mean is more sensitive to false positives than the F-measure. Thus, if it is important to avoid false positives it may be better to use the F-measure.

The highest G-mean scores for the credit data were obtained from resampled versions of SVC and logistic regression and cost-sensitive logistic regression at around 0.95. The best G-mean value for the insurance data was obtained from resampled versions of decision trees with values around 0.81. The best heart data results for the G-mean came from undersampled SVC and logistic regression and cost-sensitive logistic regression and SVM, with values at around 0.76.

AUC

Seeing as both the AUC and the G-mean include the true negative rate and the true positive rate, it is not surprising that they give similar results. Both the AUC and the G-mean have higher scores than the F-measure for all data sets and methods.

The highest AUC values for the credit data were obtained from resampled versions of SVC and logistic regression and cost-sensitive logistic regression at around 0.95. The best AUC value for the insurance data was obtained for undersampled decision trees at a value of 0.83. The best values of AUC for the heart data come from cost-sensitive SVC and logistic regression and undersampled SVC and logistic regression at values around 0.76.

Conclusion

The aim of this thesis was to compare and evaluate methods for classification applied to large imbalanced real world data sets. The overarching research question was chosen to be:

"Which is the best method to handle the class imbalance problem?"

The question was further narrowed down to the following questions:

"Which differences exist in performance of methods used for class imbalance for different data sets with varying degrees of class imbalance?"

"Which method outperforms the others?"

The answers to these questions are:

- Out of the methods evaluated in this thesis, none proved to outperform all others.
- There are several differences in performance between methods and data sets used for analysis.

The results of the analysis show that the different data sets have obtained varying results depending on which methods have been used. The F-measure stands out, as neither of the heart nor the insurance data obtain any high values for this measure. However, all data sets obtained relatively high values for recall, G-mean and AUC when applying some models. Regarding precision, all data sets but the heart data obtained high scores. There is no model or resampling technique that clearly outperformed the rest.

The analysis further shows how the evaluation measures differ. The G-mean and the AUC fails to show the cost of false positives. Seeing as false positives are important to measure in many imbalanced data situations these measures may not be appropriate to use. The F-measure, on the other hand, shows the cost of both false positives and false negatives simultaneously. For those data sets with low F-measure scores the analyst may need to choose which cost is worse: false positives or false negatives. The analysis shows that when one increases the other decreases, leading to low overall F-measure scores across the board.

It seems as if the models' performances are dependent on the data set used. The data set that performed best was the credit data. Here several models were able to distinguish the two classes. It is unclear why the performance differ to this

extent between the data sets. The number of observations in the minority class or degree of class imbalance does not seem to be the reason for the difference, as the credit data had the best performance. One possible reason for the difference could be differing degrees of data complexity, such as issues regarding overlapping or small disjuncts. However, this is something that for now is unknown, and thus no conclusion regarding this hypothesis can be drawn.

The analysis shows that the issue of class imbalance is a complicated one. Depending on the data set used for analysis with its structure, number of observations, class separability, variables etc. different approaches to the problem yield varying results. There is not one size fits all when it comes to solutions to this problem. Which evaluation measures to use is further a complex task and depends to a large extent on the purpose of the analysis as the cost of misclassification can differ widely between different applications.

6.1 Future Research

This thesis have been able to show some results on three specific data sets. However, the results cannot be generalized to certain areas of study, such as fraud detection or health in general. Further research is needed in order to understand why the models behave differently in separate data sets. What would be interesting to understand is how data complexity and separability changes the performance of methods used for imbalanced learning. Perhaps there could be some rule of which method to use depending on the complexity of the given data.

References

P. Branco, L. Torgo, and R. P. Ribeiro. A survey of predictive modeling on imbalanced domains. 49(2), 2016. ISSN 0360-0300. doi: 10.1145/2907070. URL https://doi.org/10.1145/2907070.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification And Regression Trees*. 10 2017. ISBN 9781315139470. doi: 10.1201/9781315139470.

F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi. Scarff : a scalable framework for streaming credit card fraud detection with spark. *Information Fusion*, 41, 09 2017. doi: 10.1016/j.inffus.2017.09.005.

F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi. Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 05 2019. doi: 10.1016/j.ins.2019.05.042.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, jun 2002. ISSN 1076-9757.

A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41:4915–4928, 08 2014. doi: 10.1016/j.eswa.2014.02.026.

A. Dal Pozzolo, O. Caelen, R. Johnson, and G. Bontempi. Calibrating probability with undersampling for unbalanced classification. 12 2015. doi: 10.1109/SSCI.2015.33.

A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–14, 09 2017. doi: 10.1109/TNNLS.2017.2736643.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, jun 2008. ISSN 1532-4435.

A. Fernández, S. Garcia, F. Herrera, and N. Chawla. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 04 2018. doi: 10.1613/jair.1.11192.

T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. doi: 10.1109/TKDE.2008.239.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*. Springer, 2021.

N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pages 429–449, 2002.

M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Mach. Learn.*, 30(2–3):195–215, feb 1998. ISSN 0885-6125. doi: 10.1023/A:1007452223027. URL https://doi.org/10.1023/A:1007452223027.

B. Lebichot, Y.-A. Le Borgne, L. He, F. Oblé, and G. Bontempi. *Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection*, pages 78–88. 01 2019a. ISBN 978-3-030-16840-7. doi: 10.1007/978-3-030-16841-4_8.

B. Lebichot, Y.-A. Le Borgne, L. He, F. Oblé, and G. Bontempi. *Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection*, pages 78–88. 01 2019b. ISBN 978-3-030-16840-7. doi: 10.1007/978-3-030-16841-4_8.

B. Lebichot, G. M. Paldino, W. Siblini, L. He, F. Oblé, and G. Bontempi. Incremental learning strategies for credit cards fraud detection. *International Journal of Data Science and Analytics*, 12, 08 2021. doi: 10.1007/s41060-021-00258-0.

J. Leevy, T. Khoshgoftaar, R. Bauder, and N. Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5, 11 2018. doi: 10.1186/s40537-018-0151-6.

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.

L. li Zhang, T. Geisler, H. Ray, and Y. Xie. Improving logistic regression on the imbalanced data by a novel penalized log-likelihood function. *Journal of Applied Statistics*, 2021.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Mach. Learn.*, 46(1-3):191–202, 2002. URL https://doi.org/10.1023/A:1012406528296.

Machine Learning Group - ULB. Credit card fraud detection, 2015. Data set retrieved from Kaggle.com https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

Möbius. Learning from imbalanced insurance data, 2020. The data has been downloaded from Kaggle.com https://www.kaggle.com/datasets/arashnic/imbalanced-data-practice.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

F. Provost and T. Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD'97, page 43–48. AAAI Press, 1997.

K. Pytlak. Personal key indicators of heart disease, 2022. The underlying uncleaned data has been obtained from the CDC's BRFSS 2020. The data has been downloaded from Kaggle.com https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease.

M. Razi and K. Athappilly. A comparative predictive analysis of neural networks (nns), nonlinear regression and classification and regression tree (cart) models. *Expert Systems with Applications*, 29:65–74, 07 2005. doi: 10.1016/j.eswa.2005.01.006.

C. J. V. Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979. URL citeseer.ist.psu.edu/vanrijsbergen79information.html.

Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang. Cost-sensitive boosting for classification of imbalanced data. 40(12), 2007. ISSN 0031-3203.

G. M. Weiss. Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl.*, 6 (1):7–19, jun 2004. ISSN 1931-0145. doi: 10.1145/1007730.1007734. URL https://doi.org/10.1145/1007730.1007734.