LUT
Lappeenranta
University of Technology

LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

School of Business and Management

Strategic Finance and Business Analytics

Tino Tuomi-Nikula

INSURANCE CLAIM RISK SCORING WITH MACHINE LEARNING ALGORITHMS: A CASE STUDY ON DEVELOPING A PREDICTIVE SYSTEM FOR ASSESSING CORPORATE CUSTOMERS' CLAIM RISK

Master's thesis

2018

1st Supervisor: Professor, D.Sc. (Econ. & BA), Mikael Collan
2nd Supervisor: Post-Doctoral Researcher, D.Sc. (Econ. & BA) Jyrki Savolainen

ABSTRACT

Author: Tino Tuomi-Nikula

Title: Insurance Claim Risk Scoring with Machine Learning Algorithms: A Case Study on Developing a Predictive System for Assessing Corporate Customers' Claim Risk

Faculty: LUT School of Business and Management

Major: Strategic Finance and Business Analytics

Master's thesis 73 pages, 12 figures, 7 tables

Examiners: Professor, D.Sc. (Econ. & BA), Mikael Collan,
Post-Doc. Researcher, D.Sc. (Econ. & BA), Jyrki Savolainen

Keywords: risk scoring, machine learning, insurance risk, classifier algorithms, predictive analytics

Predictive learning algorithms offer tools to automate and improve insurance risk management. The aim of this thesis is to study classification algorithms in risk scoring applications and to evaluate them in the creation of insurance claim risk scores from customer risk survey data. Another goal for the thesis is to quantify weights for risk survey questions using machine learning methods. This case study consists of a critical literature review and a quantitative analysis where the risk scoring systems were developed and their performance on test data evaluated. Logistic regression, Support Vector Machines, Extreme Gradient Boosted Trees (XGBT) and a Feed Forward Neural Network were used to predict claim risk from customer risk survey data. The results of the study show that with the test data, XGBT was the most accurate, predicting the risk class of a customer with 73% accuracy. Nonetheless, it is worth mentioning that all tested algorithms were suitable for the task. The results also suggest that machine learning risk prediction enables analysis of individual risk factors, and that feature importance metrics can be used to quantify weights for risk survey questions.

TIIVISTELMÄ

Tekijä:	Tino Tuomi-Nikula
Aihe:	Vakuutuskorvausriskiarvio koneoppimisalgoritmeilla: Case-tutkimus ennustavan järjestelmän kehittämisestä yritysasiakkaan korvausriskin arvioimiseksi
Tiedekunta:	LUT School of Business and Management
Pääaine:	Strategic Finance and Business Analytics
Pro Gradu –Tutkimus:	73 sivua, 12 kuviota, 7 taulukkoa
Tarkastajat	Professori, KTT, Mikael Collan Tutkijatohtori, KTT, Jyrki Savolainen
Hakusanat:	riskiarviointi, koneoppiminen, vakuutusriski, riskienhallinta, luokittelualgoritmit, ennustava analytiikka

Ennustavilla koneoppimisen algoritmeilla on suuri potentiaali vakuutusalan riskiarvioinnin automatisoinnissa ja kehityksessä. Tämän pro-gradu-tutkimuksen tavoitteena on tutkia luokittelualgoritmeja riskiarvioinnin työkaluna. Tarkoituksena on tuottaa yritysasiakkaan vahinkoriskiarvio riskikartoitusdatasta. Toisena tavoitteena on kartoituskysymysten painokertoimien määrittäminen koneoppimisen menetelmillä. Tutkimus koostuu kriittisestä kirjallisuuskatsauksesta sekä kvantitatiivisesta case-tutkimuksesta, jossa kehitettiin neljä eri algoritmeihin perustuvaa riskiarviojärjestelmää ja arvioitiin ne testidataa hyödyntäen. Case-tutkimuksessa ennustettiin korvausriskiä kartoitusdatasta käyttäen logistista regressiota, tukivektorikonetta, Extreme Gradient Boosted Trees (XGBT) -algoritmia ja eteenpäin syöttävää neuroverkkoa. XGBT riskiluokitteli testiasiakkaat parhaiten 73 prosentin osumatarkkuudella, mutta tulosten perusteella jokainen testattu algoritmi soveltuu tehtävään. Lisäksi tuloksista käy ilmi, että algoritmien laskemia muuttujien merkittävyyskertoimia voidaan käyttää kartoituskysymysten painokertoimien määrittämisessä.

ACKNOWLEDGEMENTS

Thank you,

Jyrki Savolainen and Nicolau Gonçalves for continuous feedback to steer my work into a meaningful complete master's thesis. Your comments on both scientific method and writing as in theoretical and technical accuracy were invaluable.

Mikael Collan and Eero Holmila for sparring and encouragement.

Rohea Ltd. for the chance to work on an interesting project with intelligent, friendly people.

Lappeenranta University of Technology staff and co-students for the two years' worth of new ideas and skills that made this thesis possible.

Table of Contents

List of abbreviations

AUC	Area Under the Curve
CM	Confusion Matrix
CRM	Customer Relations Management
CV	Cross-Validation
DT	Decision Tree
EM	Expectation-Maximation
FN	False Negative
FP	False Positive
GLM	General Linear Models
kNN	k-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
NB	Naïve Bayes
NN	(Artificial) Neural Network
RF	Random Forest
ROC	Receiver Operating Characteristic curve
SVM	Support Vector Machines
TN	True Negative
TP	True Positive

List of figures

List of tables

1. INTRODUCTION

Machine learning (ML) algorithms promise advancements in insurance risk management. Already, automatic credit risk scoring systems, based on ML algorithms, are behind most of the current credit decisions, and the topic is a widely discussed subject in academic literature, where numerous ML scoring systems have been developed and tested (Louzada, Ara & Fernandez, 2016). Insurance risk management has great potential to benefit from the same methods.

1.1 Motivation and background of the study

Currently, insurance risk management is carried out by calculating statistical probabilities based on very large samples (David, 2015b). Advances in computational power, and the increasing amount of available risk data and quality of the collections are promising improvements (Baecke & Bocca, 2017). These advances allow machine learning algorithms to learn patterns in data, indiscernible to human eyes, to predict future risks (Guelman, 2012). Systems based on these algorithms can automate expert analyst's work, make routine decisions independently and raise problematic cases for expert review. Valuable expert time can be allocated where it is most valuable while keeping the risk assessment up-to date and available for all the users.

Accurate risk assessment is a win-win situation, as in theory it should also lead to correct coverage on customer's risks. Insurers wants to quantify these risks to price the insurances correctly (Kafková, 2015), and to manage the total risk in its liabilities (David, 2015b). Currently, machine learning applications addressing this problem are scarce, at least in the academic literature, but automating risk assessment could potentially lead to better and more timely decisions in many different operations in insurance companies. One goal of this thesis is to find evidence about usability of ML risk scoring systems, in the insurance domain, for more efficient and accurate risk management.

1.2 Theoretical framework and the focus of the study

In the research universe this study is situated on three main areas: risk management, corporate insurance and machine learning. The main research problem and its motivation come from the corporate insurance domain and is specific to the risk management field, while the methodological tools come from the machine learning area. See the Venn diagram

of research area (Figure 1). The focus of the study is on the methodology and in finding evidence about machine learning algorithms' usability to this specific problem. In this thesis, I draw from the vast literature of machine learning based risk scoring systems to expand the applications to insurance specific risks.

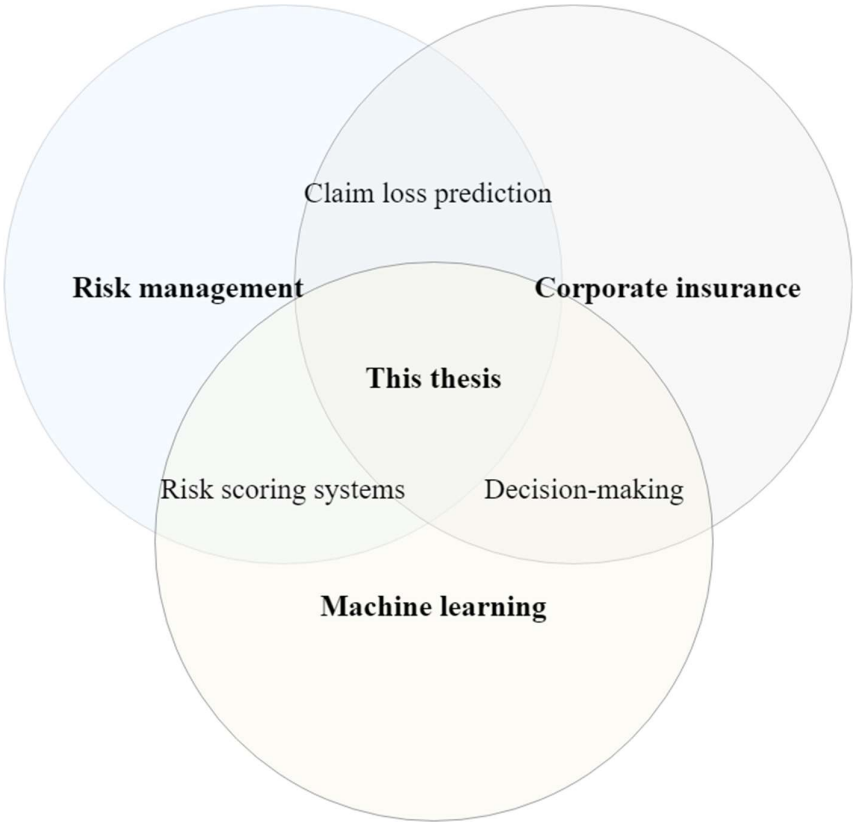

Figure 1. Venn diagram for the research area of the thesis

1.3 Research questions and objectives

The main goals for this thesis were to design an insurance risk scoring system utilizing machine learning algorithms and find ways to quantify relationships between risk factors and risks. Objectives to reach these goals were to find the best practices for ML algorithm-based risk scoring and to test and evaluate these methods with real customer and claim data. Based on these research objectives, I formulated two research questions with sub-questions:

1. *What does the literature tell us about risk prediction using ML?*
    a. What are the applications of ML algorithms in insurance industry?
    b. What does literature tell us about using ML algorithms for risk scoring?
    c. What algorithms should be used in insurance risk prediction and how can they be compared?
2. *How do different machine learning algorithms compare in predicting insurance claims based on customer data?*
    a. How do different methods compare in evaluation metrics with the given customer data set?
    b. Can ML algorithms be used to create insurance risk scores from the given customer data?
    c. Can ML analysis be used to quantify weights for the risk survey questions?

1.4 Research methodology

The approach to the research problem was three-folded. First, I conducted a literature review to find and map the recent literature on using machine learning in insurance industry and in risk scoring systems. The goals of the review were to fill my knowledge gaps about research areas touching the topic, to formulate proper research questions and to provide an overview of the research field for the reader. Second, after identifying the relevant literature, I focused on learning from the prior works and researching developments in the field to select the predictive algorithms to be used in the quantitative case study. Lastly, I tested the algorithms chosen from the literature, evaluated the results and sought evidence on their suitability in creating meaningful risk scores from the data.

The data for the case study was collected from a sales automation and risk management platform (PACE™). PACE™ is used by B2B insurance account managers and risk managers. Risk surveys and assignments to customers are platform's core features and data collected from it is used to train the risk scoring models. Unfortunately, the data cannot be published with the thesis.

1.5 Structure of the thesis

This thesis is structured as follows. In Chapter 2, I describe key methodological concepts necessary to understand the technicalities of the thesis. Chapter 3 provides the theoretical framework on which this research is built on. Firstly, I describe the structured literature

review I conducted to find the source literature, and to present an overview of the field of study. Next, I discuss the relevant literature to build the theoretical framework and premises for the experimental part. Lastly, I summarize the review and related work and answer first set of research questions. In chapter 4, I explain thoroughly the process of my case study on ML insurance risk prediction with its limitations and the considerations and decisions I made. Finally, I present the results in chapter 5 and answer the 2nd set of research questions. In the last chapter 6, I discuss the results and their limitations and arrive to conclusions and future research proposals. Illustration of the structure is presented in (Figure 2).

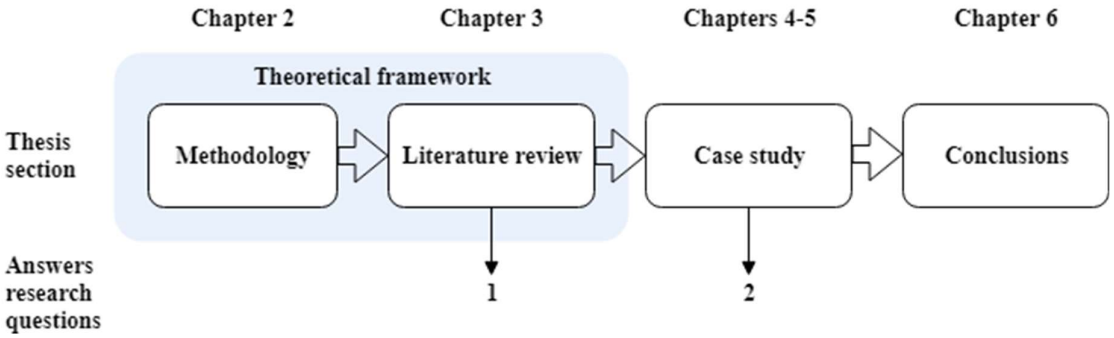

Figure 2. Structure of the thesis

2. MACHINE LEARNING CLASSIFICATION SYSTEMS

In this chapter, I describe the key machine learning concepts relevant to the study. I aim to summarize applied ML classification systems goals and methodologies in high-level. The goal is not to provide a definitive summary, but to introduce concepts through common tools used in the field.

Valiant (1984) defines machine learning as "algorithms that infer the function they compute from example data". In essence, ML are computer programs using defined rule bases to learn the structure of the data for different purposes. The most common tasks are *clustering* (grouping), *classification* (predicted value is categorical) and *regression* (predicted value is continuous). Machine learning algorithms can be divided into two main sub-classes based on their goal and data available: *supervised* and *unsupervised learning.* Supervised learning methods require labeled training data, labeled meaning the predicted class or value for the observations. Based on the pattern learned from the training data, supervised learning algorithms can make predict the label of new observations. Rather than predicting a value, unsupervised methods aim to learn the structure of the data to increase its interpretability often for further analysis. Two distinct goals for unsupervised methods are grouping and dimensionality reduction. Grouping the observations can allow labeling observations based on the groups and then using supervised methods or finding new relations between the features. With larger feature spaces dimensionality reduction methods help to summarize the data and filter or compress it into smaller number of features. Working with real world data and problems often requires combination of these tools. (James, Witten et al., 2013, 1-24)

2.1 Supervised classification methods

Supervised classification methods require class labels in training data to predict classes of unseen data. Classification means either binary (two classes, or in-class/not-in class) or multiclass labeling.

2.1.1 Logistic regression

Logistic regression is a popular and versatile classification method. It is a good starting point explaining ML because of its closeness to linear regression, a statistical modeling technique which has been used across the scientific fields to model the relation between predictors X

and predicted variable *Y* across the by fitting a linear line to the data, minimizing the total error to actual observations. Pioneer work with logistic regression was done by David Cox (1958). In logistic model, a sigmoid function (output values between 0 and 1), is fit to the data. Despite of its name, it is used to classification problems rather than regression. It can be used to model the relation between predictor variables in nominal, relational or ordinal scales and predicted variable that is categorical. The algorithm computes probabilities for outputs which allows user to interpret the change in the predictors and the probabilities or "risk" of belonging to each category. Logistic regression can be represented as:

$$log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \ldots \beta_m x_m$$

Where π is probability of an event, β_i regression coefficients for each explanatory variable x_i. Solving π from the equation gives probability of belonging to the predicted class. (Bishop, 2006, 205-206)

2.1.2 Naïve Bayes

Naïve Bayes (NB) is based on Bayesian probability theory where a class of an object is determined by calculating its probability of belonging to each class and choosing the one with highest value. The "Naïve" in the name comes from the assumption that the different features of an object would be independent of each other. The relative advantages of NB include speed of computation, small amount of required training data. (Kubat, 2017, 19-40)

2.1.3 k-Nearest neighbor

k-Nearest neighbor (kNN) is a classifier formalized by Cover and Hart (1967). kNN is based on Euclidean distance between the observations. Each training data observation forms a feature vector in multidimensional feature space. Test object's feature vector is compared to the training data and the class of the closest (minimum distance between the vectors) training observation is given to the test object. *k* in the name refers to number of observations the test object is compared to. (Kubat, 2017, 43-49)

2.1.4 Support Vector Machines

Support Vector Machines (SVM) is a classification method based on distances between the observations. SVM was introduced by Cortes and Vapnik (1995). In a two-dimensional, two-group example the SVM algorithm finds a hyperplane (linear separator) that is equally

far from both of groups of observations. This is done by computing the distance from each observation to the hyperplane, finding minimum from of both of the groups and make sure they are equally far from the hyperplane. Two feature example of hyperplane presented in (Figure 3). New observations are labeled by which side of the hyperplane they fall to. (Clarke, Fokoué et al., 2009, 262-293)

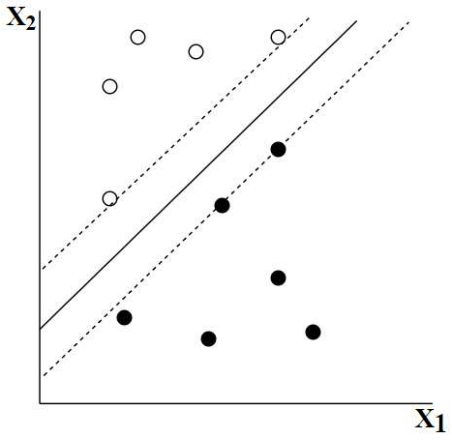

Figure 3. Hyperplane between two features

2.1.5 Tree based methods

Decision trees are non-parametric classification algorithms that predict the value of target variable based on simple *if-then-else* rules learned from the training data. A breakthrough on using decision trees for classification problems was made by Breiman, Friedman et al. (1984). The decision rules computed can be extracted which makes it an interpretable (*white box*) model, which can also be presented visually in an intuitive manner. Decision trees require relatively small amount of training data and process both numerical and categorical data. Their disadvantage is that they easily overfit to the training data which makes them perform badly on unseen data. Ensemble methods (such as Random Forests), where many independent trees are trained, and their results combined (e.g. by selecting the dominant class predicted by multiple trees) help with this problem. Decision trees work well with problems that can be expressed with *if-then-else* logic but struggle more complex concepts such as *else-or* relations. (Clarke et al., 2009, 249-262).

2.1.6 Neural networks

Artificial Neural Networks (NN) in ML are systems inspired by how neurons (called *nodes* in data science) function in human brain. NN is a network of nodes used to model complex interactions in the data. Neural networks used in ML often have a layered architecture, more layers meaning a deeper network (hence the name *deep learning*). NNs typically have an input layer, number of "hidden" layers in between and an output layer. These layers consist of nodes, which can be connected to each node in layers closest before and after it. Node takes inputs, makes computation based on its rules and sends the output forward. Each connection can be understood as weight. The data leaving the input layer is multiplied by the weight assigned to that connection and arrives as input to the next layer's node, which performs a computation on it and sends it forward through its connections to next layer. This goes on until output layer is reached and output received. (Bishop, 2006, 225-241)

There is no single inventor of neural networks as machine learning systems, their development has been a long series of developments in several fields of science. An illustration of a feed-forward network with one hidden layer is provided in Figure 4.

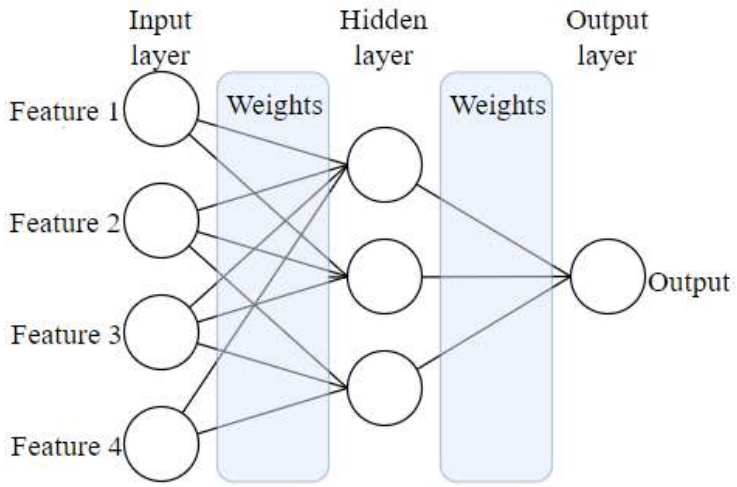

Figure 4. A single hidden layer neural network

Neural network is an umbrella term for algorithms that have the structure I described. They can differ for example in architecture (number of layers and nodes, how nodes are connected) or optimization methods which makes them completely different algorithms

under the hood. The key problem in ML neural networks is to find the set of weights for the inputs that maximizes the accuracy of the outputs. It is an optimization problem of very large scale as there can be hundreds of millions of weights in network. A popular optimization method for the problem is *gradient descent* where the parameter value is slowly tweaked until a minimum for *loss function* (error) is reached. (Bishop, 2006, 240-241)

2.2 Unsupervised classification methods – clustering

Clustering is an unsupervised method to explore the data, sometimes without a clear goal in mind. Clustering methods form groups of the observations based on the structure of the data and the statistics of the groups can reveal interesting relations in the data. Clustering is often used to preprocess data for supervised learning. (Clarke et al., 2009, 405-409)

2.2.1 k-Means clustering

k-means clustering is a distance-based grouping method. The algorithm was first formalized by Stuart Lloyd in 1957 but it was not published until 25 years later (1982). *k* is the number of groups to be extracted from the data, this might be known in advance or determined by different techniques depending on the goal. Solving which observation belongs to which group is done with expectation-maximization (EM) algorithm (Dempster, Laird et al., 1977). EM is an iterative method to find maximum when the distribution is unknown.

In the case of k-means clustering, the algorithm has three steps. First, the observations are represented as feature vectors and *k* number of centroids are randomly placed in the feature-space. Second, the distance from each observation to each centroid is calculated, and an observation is assigned to the centroid it is closest to. At the end of second step, initial groups are created. Third, the centroids are moved to the center of their clusters (the average position of the observations belonging to that cluster). Steps two and three are repeated until the change in the position of the centroids goes below certain limit or stops. (Clarke et al., 2009, 409-413)

2.2.2 Hierarchical clustering

Hierarchical clustering is a distance-based grouping method, where the process moves top-down or bottom-up. In bottom-up method, arguably more intuitive of the two, all observations start in the same cluster, are then divided into two clusters, which are again split apart. This process is continued until each observation is its own cluster, or other criteria

is met. The process is often presented as a tree-like dendrogram from which the distance in each split can be perceived i.e. how much different the clusters formed in the split are from the original cluster and each other. (Clarke et al., 2009, 413-422)

2.3 Feature selection and engineering

Feature selection is an important phase in all applied machine learning and especially predictive models. Collecting the right data to the problem often influences the results significantly more than model selection procedures. Scientific theories from the applied field can be helpful, but often availability of data might limit or determine the selection process. Missing important features leads to poor models, and including non-relevant features adds to complexity (longer computing times), reduces interpretability of the models and causes faulty. Transforming the features or generating new ones from the data to better answer the problem is called feature engineering. Feature engineering could be for example computing a new feature from two original features. (Correa Bahnsen, Aouada et al., 2016)

With predictive models *target/data leakage* is an additional consideration. It means that some predictive feature includes information from the predicted value. When leakage is present, the test results will be biased as actually some of the predictive power comes from information that is not available at the time of actual prediction. (Becker, 2017)

2.3.1 Dimensionality reduction

Dimensionality reduction aims to reduce the complexity of the data to either faster computations or to limit overfitting. Dimensionality reduction can also be useful feature selection method if there are too many features for manual selection or there is no theory or intuition about their relation to the problem. Many of the common methods exploit linear algebra to represent the data in simpler form while retaining the structure of the data. (Dougherty, 2013, 123-124)

Principle Component Analysis (PCA) is an unsupervised dimensionality reduction/feature selection analysis method. The data is represented as feature vectors, which are ranked by the variance they explain. A number of these uncorrelated feature vectors (principal components) are selected so that the explained variance stays high enough. Number of dimensions (features) can be understood as complexity of data and variance as data's

structure. In many cases, an analyst wants to reduce complexity while maintaining the structure. (Bishop, 2006, 561-570)

The goal of Singular Value Decomposition (SVD) is to reduce the size of the data. Machine learning methods are often computationally expensive, and each feature might exponentially increase the amount of computations in the system. The logic behind SVD is to represent the data matrix of feature vectors as a product of three simpler matrices. These matrices can be reduced to compress the data while maintaining a meaningful representation of the original matrix. When the amount of values in the matrices needed for representation of the underlying follows a logarithmic function, a larger reduction of dimensionality results in smaller loss of structure (e.g. predictive power). While the resulting features might be less interpretable for human analyst, they are often intended to be read by computers, other machine learning algorithms for example. (Clarke et al., 2009, 250-251)

Linear Discriminant Analysis (LDA) is a supervised method that is used both in in classification and dimensionality reduction. The LDA objective is to find a smaller subspace for the original feature space with axes that maximize the separation of the given classes. (Dougherty, 2013, 135-140)

2.4 Data preprocessing and model development

Machine learning algorithms require different data preprocessing and parameter tuning measures for optimal performance. I will next go through some of the common methods and considerations also discussed later parts of the thesis.

2.4.1 One-hot-encoding

Many algorithms only process numerical data and perceive numerical categories as values from continuous scale. One-hot encoding (known as dummy encoding in statistics) means making each category of a variable a feature of its own and allocating value 1 if the feature is true for the observation and value 0 if not. This procedure usually makes the dataset wider and shorter, e.g. more features, less observations. (McKinney, 2013, 202)

2.4.2 Handling missing data

Missing data, i.e. not all observations having values for all features is a common problem meaning there is no observation for all features in every sample. The common methods to

handle this problem are to either remove the observations with missing data or to impute values to them. Removing observations with missing data leads to more balanced datasets but reduces the amount of training data available and causes the information stored in the present values for features of the observation to be lost. Imputing data with some estimate, such as feature mean or median (or using predictive tools to estimate the value) is often the preferred solution. For the categorical variables one-hot encoding alleviates missing data issues and "data on feature *x* is missing" can be made its own variable, if considered important information. (Zhu, He et al., 2012)

2.4.3 Data normalization

Normalization, also scaling or standardization, is used to prevent bias due to features having different scales. There are many normalization techniques and the choice of method can have a large effect on the model. A simple example is *min-max* method where the minimum value of a feature is set to be minimum of the new scale, i.e. 0, and same for maximum, i.e. 1. Values in between are then transformed in relation to this new scale from zero to one. Standardization also normalizes the distribution of data, so that the different features have the same standard deviation. (Aksoy & Haralick, 2001)

2.4.4 Parameter tuning

Parameter tuning means adapting the parameters of an algorithm according to a specific criterion, e.g. improved accuracy. Rules of thumb can be used, but they require experience from applying the algorithms to datasets with different properties or trusting rules set by someone else. Manual search is also possible, but it too requires a starting point and could prove slow. One would need to change one parameter at a time, train the model and keep a list of results obtained from different parameter combinations. (Claesen & De Moor, 2015)

Grid-search is a method which allows testing for multiple parameters and their values at once by training multiple models and highlighting the best solution and its parameters. In essence, it means training models with different options for one parameter while maintaining the other parameters constant. The trained models are evaluated to find the parameter minimizing a chosen *loss-function* (for example error rate). Parameters found in grid search are then set as constants and same routine performed for next parameter until all wanted parameters are optimized. Large grid-searches are computationally ineffective, and usually it is not feasible to go through all the options. (Claesen & De Moor, 2015)

Recent studies suggest that another method, Random Search could be computationally more feasible solution to optimize for the parameters. In Random search a starting value for parameter is given, and an optimization method used to find the optimal value, rather than checking a set of predefined values. (Bergstra, Bardenet et al., 2011)

2.5 Evaluation of classification systems

Two important perspectives for classification systems evaluation are precision and speed. Precision meaning how accurately the system predicts the thing it is designed for. Speed meaning the time it takes to compute the solution. In this study, I will only discuss the evaluation of precision.

2.5.1 Confusion matrix

Confusion matrix forms the basis point for evaluating the binary classification systems. It is a two-by-two matrix that presents the number of correctly and incorrectly classified observations versus the actual distribution to classes in the test set. The problem of binary classification can be presented: *does observation belong to a class?* This allows representation of observations to positives (belongs to the class) and negatives (does not belong to the class). The confusion matrix consists of following figures:

- True Positives (TP): predicted positive, actually positive
- False Positives (FP): predicted positive, actually negative
- False Negatives (FN): predicted negative, actually positive
- True Negatives (TN): predicted negative, actually negative

An example confusion matrix is represented in Figure 5. From the confusion matrix, different performance indicators can be computed for the classifier. *Accuracy* can be seen as an overall performance metric. It is the ratio of correctly classified observations versus all observations, but in many cases, it might not represent the wanted performance of the classifier. Especially when the distribution of classes in the test set is not balanced, accuracy might not be the best metric for overall performance. In an example case 90% of the test set belong to the class and 10% do not, a classifier only predicting positive would get accuracy of 0.9 but would be useless, because it completely misses the negative ones. Another example is a situation where other class is more important than the other, say it is critical to get the positives right but sometimes classifying negative as positive would be acceptable.

In this case *recall* or *sensitivity* (true positives of predicted positives) would be more suitable performance metric. Other common metrics are *precision* (true positives of total positives), *specificity* (True negatives of total negatives) and *F1-measure* which is a balanced mean between precision and recall. Selection of metrics should be done in accordance with the data and goal. (Fawcett, 2006)

n=120	Predicted Negative	Predicted Positive	
Actual Negative	TN=60	FP=30	90
Actual Positive	FN=20	TP=10	30
	80	40	

Figure 5. A confusion matrix

2.5.2 Receiver Operating Characteristic curve

Receiver Operating Characteristic curve (ROC) (see example Figure 6) is a visual representation of the relation between TP and FP rates of the classifier and can be used to analyze the trade-off between them. As a graphical tool, it can give better overall image of the performance. Shapes of the curves can also be used to compare different classifiers (Duda & Hart, 1973). Visual comparison of the curves might prove difficult and inaccurate, therefore a single figure measure, Area Under the Curve (AUC) was introduced to be used in formal single-measure evaluation of the classifiers. Formal and empirical proof has been shown that AUC is better single evaluation measure for machine learning classifiers than accuracy (Jin Huang & Ling, 2005).

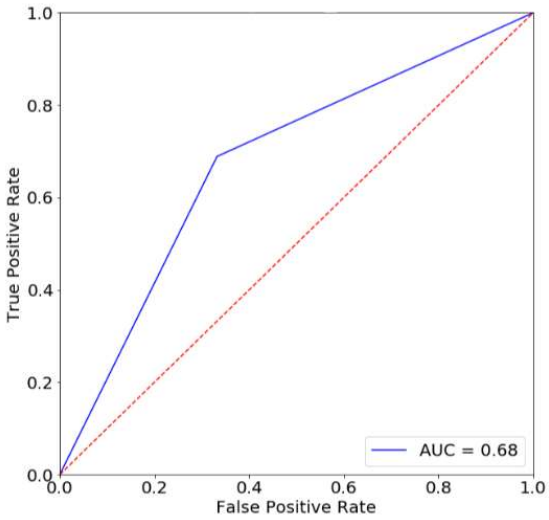

Figure 6. A ROC curve

2.5.3 Training and evaluation datasets

To detect *overfitting* and to avoid biased results, the ML model must be evaluated with data not used in its training. Overfitting means learning the specific structure of training data instead of a general pattern that would predict new, unseen observations. Holding out a large sample of the original data for testing means sacrificing the information in that sample but is essential to evaluate how the model works with out of sample data. There is no generally accepted theory on how much data to hold out, but the sample should acceptably represent the variance of the data. (Kohavi, 1995)

Cross-validation (cv) is an evaluation strategy where the train dataset is partitioned to *k-folds*, the same learning algorithm is taught *k* times and each time different part of data is left out in the training phase and used to evaluate the model. Cross-validation is often used in model development phase to evaluate changes. The final evaluation should still be conducted with holdout dataset. (Kohavi, 1995)

3. MACHINE LEARNING IN INSURANCE AND RISK ASSESSMENT – A LITERATURE REVIEW

In this chapter, I present the methodology, process and results of the literature review that I conducted to find relevant academic literature to build my research on. According to Webster and Watson (2002) the goals of a literature review are to motivate the research topic, describe key concepts, outline the past research and highlight the gaps in it. In addition, an important reason to conduct the review was to formulate, develop and focus the research questions and hypotheses.

3.1 Definitions

Claim reserving: Insurance business problem and a field of actuary studies on predicting the outstanding claims of insurance and optimizing the reserve to meet these future liabilities. Is linked to insurance premium and risk assessment. (Zhao, Zhou et al., 2009)

Claim risk/insurance risk: Uncertain event that can be insured against. (Zhao et al., 2009)

Customer: In the case study, customer refers to insured, the buyer of the insurance.

Insurance premium: The portion of the price of the insurance that insured has to pay based on risk assessment made by the insurer. Is linked to risk assessment, higher risk score can lead to higher premium. (David, 2015a)

Risk score/rating: A comparable numeric figure or a class for a certain risk, resulted from risk assessment process. (Kelly & Nielson, 2006)

(Stochastic) Risk assessment: A process that quantifies a certain risk into comparable risk score/rating based on the risk factors. (Kelly & Nielson, 2006)

3.2 Methodology

I used structured methodology to document the source literature collection process as suggested by Webster and Watson (2002). I adapted their approach for structured and documented collection of source literature. Their process has three steps:

1. Scan table of contents of leading journals, also look outside the main discipline. Look for major contributions.

2. Backward tracking: scan the citations of articles found in step 1 for more relevant sources.
3. Go forward: use web citation portals to find key literature citing the articles found in steps 1 and 2.

Webster and Watson (2002) also suggest using concept-centric approach to literature reviews. I used both author-centric and concept-centric approaches to arrange and classify the articles. Three separated streams of literature were presented in their own tables, of which one was author-centric, one concept matrix and third concept matrix with units of analysis.

3.3 Collection of source literature

Initial stock: I selected the databases to begin my research in. The selected portals were "EBSCO – Business Source Complete", "Elsevier ScienceDirect" and "Emerald Journals". These sources should provide comprehensive coverage of journals within my field of study. I limited the step 1 searches to find publications from January 1998 to December 2017. I formulated a broad search string, because I wanted it to capture interdisciplinary studies that are relevant to my research. Domain specific "insurance" and "claim" were supplemented with more generic "risk". The search string directed at abstracts, keywords and titles was: (insurance OR claim OR risk) AND ("machine learning" OR "data mining").

The search resulted in total of 1370 articles in these three portals. I scanned through all the titles of these 1370 articles and excluded the irrelevant ones. I scanned the abstracts of the articles that seemed relevant judging by the title and excluded further irrelevant articles. Many of the excluded texts were from the domain of medical sciences or geosciences, where applications include prediction of different medical conditions and wildfire, soil movement, and pollution modelling. I also excluded many mathematical studies that focused on development of ML algorithms as their specificity is outside the scope of this thesis. Algorithms for the experimental work were chosen from studies concerning relevant applications. I made an observation that there were very few articles about applications of ML in insurance industry, but two new rich discourses emerged: claim/loss reserving (actuary sciences) and credit risk assessment with ML (decision-making). I decided to include articles from these areas into my stock. After removing these articles from the list, from the initial 1370 articles, 42 were selected for reading.

Backward tracking: I read the related work and references sections of the 42 articles to find key contributions from the citations. I did the same procedure also to each of the article found. This was an important phase, as new areas of research were discovered in the first step. 22 articles were added to the stock in this step, mainly from credit risk assessment and claim loss reserving.

Forward looking: I used the same portals as in initial search to find citing articles to the ones in my stock and added relevant to the stock. I discovered only 6 new articles, as quite thorough work was done in the backward tracking step. Finally, the stock amounted to 70 articles. The whole process is presented in Figure 7.

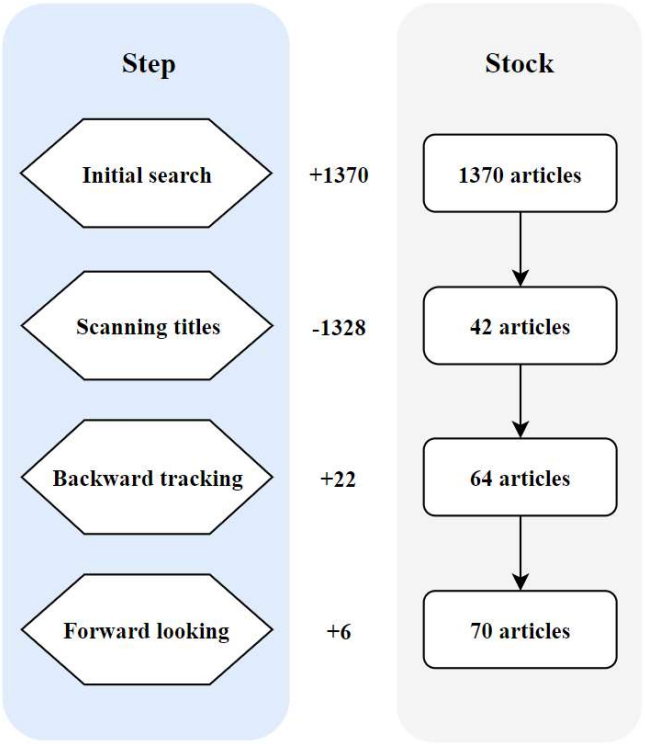

Figure 7. The source literature collection process

Through 2nd and 3rd step I was using *RefWorks* citation management software to process the articles. I classified the articles by discourse, domain, objective and methodology and coded

the classifications into the system. This allowed me to easily calculate the number of articles in each category.

3.4 Main literature streams in the stock

I discovered three streams of publications that were strongly connected to my research. The first stream I discuss is claim loss prediction, which seems to be the mainstream of risk prediction in insurance from the actuary point of view. This stream included insurance industry and risk prediction, but majority of methodology was outside ML. Second stream consisted of articles in which ML methodology was applied to insurance business problems. These articles were from different fields of studies but connected by insurance industry and ML methodology. The last stream discussed credit scoring stream that discussed ML credit scoring systems. It connected to risk prediction with ML methodology but excluded insurance industry.

3.5 Claim loss prediction stream

Claim loss stream consisted of 22 articles. Only one article used ML methodology, while the remaining 21 articles discussed the use of General Linear Models (GLM) in claim reserve prediction. These 18 articles concentrated on portfolio optimization and four on independent claims prediction. Main contributor was *Insurance: Mathematics and Economics* journal with 19 publications. Claim loss stream is presented in Table 1. I next go through the problems discussed within this field to introduce the other approach to claim risk prediction problem and to further explain the motivation behind the main research problem of this study.

Table 1. Articles in the claim loss stream

Year	Publication Title	Authors	Journal
2015	An Individual Loss Reserving Model with Independent Reporting and Settlement	Huang et al.	Insurance: Math. And Econ.
2015	Robust Loss Reserving in a Log-Linear Model	Pitselis et al.	Insurance: Math. And Econ.
2014	Conditional Least Squares and Copulae in Claims Reserving for a Single Line of Business	Pesta and Okhrin	Insurance: Math. And Econ.
2014	Individual loss reserving using paid–incurred data	Pigeon et al.	Insurance: Math. And Econ.
2014	Prediction in a Non-homogeneous Poisson Cluster Model	Matsui	Insurance: Math. And Econ.
2013	Bayesian Analysis of Loss Reserving Using Dynamic Models with Generalized Beta Distribution	Dong and Chan	Insurance: Math. And Econ.
2013	Predicting Multivariate Insurance Loss Payments Under the Bayesian Copula Framework	Zhang and Dukic	Journal of Risk and Insurance
2013	The Multi-year Non-life Insurance Risk in the Additive Loss Reserving Model	Diers and Linde	Insurance: Math. And Econ.
2013	Modeling Dependencies in Claims Reserving with GEE	Hudecova and Pesta	Insurance: Math. And Econ.
2012	Claims development result in the paid-incurred chain reserving method	Happ et al.	Insurance: Math. And Econ.
2012	Gradient boosting trees for auto insurance loss cost modeling and prediction	Guelman	Expert Systems With Applications
2011	Kernel Poisson regression machine for stochastic claims reserving	Shim and Hwang	Journal of the Korean Stat. Society
2010	Hybrid fuzzy least-squares regression analysis in claims reserving with geometric separation method	Apaydin et Baser	Insurance: Math. And Econ.
2010	Paid–incurred chain claims reserving method	Merz and Wüthrich	Insurance: Math. And Econ.
2009	Semiparametric model for prediction of individual claim loss reserving	Zhao et al.	Insurance: Math. And Econ.
2008	Estimation of loss reserves with lognormal development factors	Han and Gau	Insurance: Math. And Econ.
2008	Claims reserving: A correlated Bayesian model	de Alba and Niet-Barajas	Insurance: Math. And Econ.
2007	Claim Reserving with Fuzzy Regression and Taylor's Geometric Separation Method	de Andres-Sanchez	Insurance: Math. And Econ.
2006	Multivariate loss prediction in the multivariate additive model	Hess et al.	Insurance: Math. And Econ.
2006	Modelling negatives in stochastic reserving model	Kunkler	Insurance: Math. And Econ.
2000	An investigation into stochastic claims reserving models and the chain-ladder technique	Verrall	Insurance: Math. And Econ.
1998	Prediction of claim numbers based on hazard rates	Spreeuw and Goovaerts	Insurance: Math. And Econ.

3.5.1 Insurance pricing

Risk management is a core business problem in insurance industry. From the actuary point of view, it is statistical probabilistic process of estimating the risks in the insurance portfolio, pricing these risks accordingly and estimating and maintaining a loss reserve to meet the actualized liabilities. The definition of insurance from the insurer point of view can be reduced to risk pricing. David (2015a) has presented the modern history of the problem very well. Going back to Gaussian linear regression in the 19th century, developed by actuaries

Bailey and Simon (1960) with minimum bias procedure. Implementation by Nelder and Wedderburn (1972) to generalized linear models outside Gaussian distribution to exponential, Poisson, Binomial and Gamma distributions to name the ones most relevant to insurance pricing. This development made statistical testing of models possible and allowed their evaluation. According to David (2015a), most of recent literature is a continuation to and development of general linear models (GLM) and my literature review agrees to this notion. In her review, David (2015a) splits the pricing process to two methods, *a priori* and *a posteriori* pricing.

A priori pricing means segmentation of the insurance contracts into groups by riskiness without accounting for their claim history. These groups form tariff classes which will pay uniform premium based on their risk class. This method weighs heavily on finding the relevant predictors for each risk and collecting that data a priori. (David, 2015b)

A posteriori accounts for the claim history which, in this context, is a measure of credibility of the policy holder. This means that the model tries to account for the fact that the same risks actualize for individuals in different rate, that is independent of the risk itself. (David, 2015b). Systems that try to allocate the claim loss fairly amongst policy holders, are called bonus-malus systems introduced in insurance business by Pesonen (1962), and are nowadays applied to for example car insurance bonus levels.

The optimization problem of insurance pricing persists, and recent literature suggests using both *a priori* and *a posteriori* methods. Kafková (2015) has used GLM for the segmentation and bonus-malus system to compute the relative premium for each customer. My thesis is an attempt to create a system that computes risk classes, based on both *a priori* and *a posteriori* methods. The scores produced by the system, could be used in the price determination process.

3.5.2 Claims loss cost and -reserving

Insurers need a reserve to guard them from financial distress arising from actualized risks. As I pointed out earlier, estimating the size of this reserve is one of the main problems discussed in the actuarial academic literature in the 21st century. It is mainly considered to be a regression problem for the whole portfolio (Spreeuw & Goovaerts, 1998; de Andres-Sanchez, 2007; Hudecova & Pesta, 2013), but research has also been done on predicting

individual claim loss (Zhao, Zhou & Jang., 2009; Pigeon, Antonio & Denuit., 2014). More recent studies also show applications of machine learning algorithms and redefining the problem into classification and predicting the severity of individual risk (Guelman, 2012).

3.6 Applications of ML in insurance stream

Applications of ML in insurance stream consisted only of 15 articles, of which eight discussed fraud detection problem, five risk prediction, one automated recommendation of insurance products and one automated claim handling. Full list of classified articles presented in Table 2. Methodologically, Neural networks (NNs) were used in five, logistic regression in three, decision tree or random forest in three, SVM in three studies. Self-Organizing Map (SOM), naïve Bayes and spectral ranking were each used in one article. Three articles were released before 2010 and 12 after 2009. The main contributors were *Decision support systems*, *Expert Systems with Applications* and The *Journal of Risk and Insurance*, each with two publications.

Table 2. Articles in applications of ML in insurance stream

Authors	Journal	Year	Objective	Algorithms	Data
Baecke et al.	Decision Support Systems	2017	Risk prediction	Decision tree, Logistic regression, Neural networks	automobile insurance dataset, vehicle telematics
Lau et al.	Fire Safety Journal	2015	Risk prediction	Support vector machines	fire safety records and fire occurance datasets
Paefgen et al.	Transportation Research: Part A: Policy and Practice	2014	Risk prediction	Logistic regression	automobile insurance dataset, vehicle telematics
Guelman	Expert Systems With Applications	2012	Risk prediction / Claim reserving	Decision tree	automobile insurance dataset
Cheng et al.	Expert Systems With Applications	2011	Risk prediction	Support vector machines	construction accident dataset
Wang and Xu	Decision support systems	2017	Fraud detection	Neural networks	automobile insurance dataset
Nian et al.	The Journal of Finance and Data Science	2016	Fraud detection	Spectral ranking	automobile insurance dataset
Sundarkumar and Ravi	Engineering Applications of Artificial Intelligence	2015	Fraud detection	k-Nearest neighors, Support vector machines	automobile insurance dataset
Xu et al.	2011 Fourth International Joint Conference on Computational Sciences and Optimization	2011	Fraud detection	Neural networks	automobile insurance dataset
Gudmunsson et al.	Biomedical Signal Processing and Control	2010	Fraud detection	Support vector machines	empirical
Viaene et al.	Expert Systems With Applications	2005	Fraud detection	Neural networks	automobile insurance dataset
Artís et al.	The Journal of Risk and Insurance	2002	Fraud detection	Logistic regression	automobile insurance dataset, vehicle telematics
Brockett	The journal of risk and insurance	1998	Fraud detection	Self-organizing map	automobile insurance dataset
Abbas et al.	Future Generation Computer Systems	2014	Recommendation	Decision tree	healthcare insurance plan dataset
Bertke et al.	Journal of safety research	2012	Claim handling	Naive bayes	work injury claim dataset

3.6.1 Machine learning in insurance risk prediction

Guelman (2012) compared traditional GLM method to a decision tree model in auto insurance claim loss prediction and found it to be a good alternative. An indication, that machine learning methods are surfacing in long GLM dominant loss cost prediction literature. Machine learning algorithms analyze data efficiently and have made it possible to

use car telematics data to assess individual drivers' characteristics for predicting auto accident risk. Baecke and Bocca (2017) found that using this data with neural network, random forest and logistic regression algorithms, improved AUC metric of claim prediction model to 0.618.

Lau et al. (2015) applied SVM credit risk scoring system methodology to predict building fire risk. They showed that SVM algorithm could classify buildings to reduced and increased risk classes with specificity of 0.755. They used building risk records as predictor variables, similarly to my case study, in which the majority of predictor variables are answers to risk survey questions.

3.6.2 Other use cases of machine learning with insurance data

Churn prediction as part of customer relations management (CRM) utilizing machine learning algorithms has been studied with insurance customer data by Risselada et al. (2010). Their findings suggest good performance of logistic regression and decision tree in this problem. Literature review which is part of their study also shows that there's a strong discourse in CRM literature studying use of machine learning algorithms for churn prediction purpose.

ML algorithms have been found to be very good at detecting fraudulent claims from auto insurance claim data. First ML proposal to the problem was Artis et al. (2002) with logistic regression. Xu et al. (2011) used ensemble neural networks to classify 88.7 % of the fraudulent claims in test data correctly. Sundarkumar and Ravi (2015) tested different algorithms, support vector machines and decision trees, to the same problem and reached 0.919 sensitivity on their test data. Recently, Wang and Xu (2017) improved their model by leveraging numerical and categorical claim data with features extracted from free text accident descriptions with natural language processing and still using neural network as classifier with sensitivity of 0.917.

My literature review suggests that the ML applications is not an active field of research in insurance studies. This is a somewhat surprising finding considering the accomplishments in credit risk scoring discussed in the next section. The few studies I discovered, suggest that ML techniques fit insurance problems and risk data can be used as training data in these systems.

3.7 Credit scoring stream

Credit scoring stream consists of 33 articles, of which two are reviews, eight algorithm comparisons and Table 3 presents the stream of credit scoring literature. SVM was most popular algorithm, discussed in 16/33 articles. Tree-based algorithms were used in 7/33, NN in 7/33 logistic regression in 4/33 and naïve Bayes in two articles. Five other algorithms occurred only in single article. Different methods were often compared and discussed in the same study. For algorithm performance evaluation, Confusion Matrix was used in 27 studies and ROC was interpreted in nine. All articles were released in after year 2003 and only eight before 2010. The biggest contributors were *Expert Systems with Applications* journal with 23, *Procedia Computer Science* with three *Journal of Banking and Finance* with three and *European Journal of Operational Research* with two publications.

Table 3. Articles in credit scoring stream

		Evaluation *		Algorithm used *					
Year	**Authors**	**CM**	**ROC/AUC**	**RF/DT**	**SVM**	**NN**	**NB**	**LR**	**Other**
2017	Barboza et al.								Review
2017	Bequé and Lessmann	x	x			x			
2017	Chen and Xian								Group Lasso
2017	Dahiya et al	x				x			
2016	Louzada et al.								Review
2016	Petropoulos et al.								Hidden Markow
2016	Butaru et al.	x		x				x	
2016	Punniyamoorthy and Sridevi	x			x	x			
2015	Florez-Lopez and Ramon-Jeronimo	x		x					
2015	Danenas and Garsva	x			x				
2015	Malekipirbazari and Aksakalli	x	x	x					
2014	Niklis et al.	x	x		x				
2014	Zhong et al.	x			x	x			
2013	Kruppa et al.	x	x					x	kNN
2012	Ribeiro et al.		x		x				
2012	Wang and Ma	x			x				
2012	Li et al.	x			x				
2011	Zhou et al.	x			x				
2011	Danenas et al.	x			x				
2011	Yu et al.	x			x				
2010	Khandani et al.	x							CART
2010	Khashman	x				x			
2010	Twala	x		x		x	x	x	
2010	Zhang and Härdle	x		x					
2010	Zhou et al.	x			x				
2010	Zhou et al.	x	x		x				
2010	Yu et al.	x	x		x				
2009	Xu et al.	x							
2009	Chen et al.	x			x				
2008	Sinha et al.	x	x	x	x	x	x	x	
2007	Yang	x	x		x				
2004	Cielen et al.								DEA
2004	Mues et al.	x		x					Decision Diagram
	Total	**27**	**9**	**7**	**16**	**7**	**2**	**4**	**8**
* Please refer to List of Abbreviations		**CM**	**ROC/AUC**	**RF/DT**	**SVM**	**NN**	**NB**	**LR**	**Other**

3.7.1 Development of machine learning based credit scoring systems

In the literature review chapter, I highlighted credit risk scoring as an active field of study in machine learning applications. Louzada et al. (2016) came to similar conclusions in their review where they analyzed 187 papers on using ML classification methods in credit scoring,

with a rising trend in number of publications from 1994 to 2015. Developing existing models and introducing new ones has shown continuous improvement in solutions to the problem (Louzada et al., 2016). Many of the ML algorithms have been invented decades ago, but their applications and improvements have truly emerged with the availableness of computational power and digital data. ML credit scoring has followed the similar path.

Wiginton (1980) was one of the firsts to publish on the topic. Wiginton used logistic regression algorithm to classify credit customers to "good" and "bad" accounts. The dataset consisted of 1908 observations in and the model reached and reached accuracy of 0.491 on the binary classification task, slightly worse than "choosing in random" would yield. Although Wiginton's experiment was not a success, his methodology and evaluation of the model was very close to what it is today. Wiginton had separated subsets of data for training and testing, discussed the balance of the dataset and the loss function relational to the objective. Further advance was made in related field of bankruptcy prediction. Tam and Kiang (1992) used neural networks with *back-propagation* to predict bank failures and achieved a misclassification rate of less than 0.10, an improvement, which was probably derived from the hidden layer in the neural network.

From mid-90s to year 2000 a series of different algorithms were proposed for the credit scoring problem. Davis et al. (1992) introduced decision tree model. Henley (1995) reintroduced logistic regression in his doctoral dissertation and kNN model with Hand (1997). Neural networks were applied to credit scoring classification by Desai et al. (1996) already with total accuracy of 0.80. These proposals set in motion a stream of literature that has been active ever since. Steady development of these algorithms continued, but major breakthrough was made by Lee (2007) as he proposed the use of SVM algorithm. SVM was found to be very suitable for the problem and it dominated the discourse until recent years. As Wiginton (1980) had suffered from shortage of predictive data, Lee (2007) already had 3007 observations on 279 features of business financial ratios at his disposal. Lee also used 5-fold cross validation to better validate the predictions, resulting in a mean error rates of 0.672 for five tests.

As SVM was included to the credit scoring methodology, the discourse shifted from proposing new algorithms to feature engineering and model development with the goal of improving out of sample accuracy (Sinha & Zhao, 2008). As Lee (2007) was using a wide

feature space data mining approach, Sinha and Zhao proposed use of expert domain knowledge to engineer more accurate predictors. Their approach was successful and use of experts yielded more accurate classification on two separate datasets, reaching AUC of 0.782 with SVM. They tested the same approach to other algorithms: logistic regression, naïve Bayes, NN, decision trees and kNN and had similar results across the board.

Ensemble methods started emerging in 2010 as another solution to out of sample performance problem. Ensemble methods means training multiple models (agents) and combining their decisions into one for example by voting or averaging. Yu et al.(2010) compared single agent SVM with different multiagent SVM. They incorporated diversity in the agents by applying different training datasets and kernel functions. In their experiment, multiagent SVMs were always more accurate than single-agent systems. Zhou et al. (2010) experimented with different ensemble diversifying also the algorithm's penalty parameters and concluded that it is difficult to find a single model that would work best with all datasets. In data science, this is called the *no free lunches theorem* which suggests that its often impossible to know in advance which algorithm works best for the data (Wolpert, 1996). Different ensemble methods such as random forest (Malekipirbazari & Aksakalli, 2015) and hybrid systems (G. Wang & Ma, 2012) were proposed and improved prediction accuracies reached. In recent years, some new algorithms such as Extreme learning machines (ELM, a feed forward NN) (Zhong et al., 2014), hidden Markov (Petropoulos et al., 2016) and Group Lasso (H. Chen & Xiang, 2017) were proposed.

As none of the algorithms or systems is unambiguously better than the others with every dataset, the merit of these studies is that practitioners now have a wide selection of tools to tackle the problem with the data they have available. The academia has provided strong evidence of usefulness of ML algorithms in credit scoring systems.

3.8 Summary of literature review

In the literature review I have found answers to first set of my research questions titled "*What does the literature tell us about risk prediction using ML?".* The first sub-question (1a) was posed as: *"What are the applications of ML algorithms in insurance industry?".* I found in the literature review that general linear models in claims reserve prediction seems to be the main stream in insurance risk prediction. Only few studies concentrated on prediction of individual risks or on other methodologies besides linear models. No distinct body of

research for applying ML algorithms to insurance problems seems to exist - the 15 articles I could label under "applying ML algorithms to insurance problems" discussed eight different problems. The most active problem discussed was automobile insurance fraud detection with 7 articles.

The second sub-question (1b) was posed as: *"What does literature tell us about using ML algorithms for risk scoring?".* Credit risk scoring was discovered to be an analogous problem to insurance risk scoring and it is an active field of research. I have presented the findings in credit risk scoring literature to show success in ML based risk scoring systems. Further, I have presented literature with evidence suggesting that same methodologies can be successfully implemented to insurance risks. Finally, empirical results from Lau et al. (2015) suggest that risk survey data can be used as predictors in these systems.

The last sub-question (1c) under title "*What does the literature tell us about risk prediction using ML?"* was posed as: "*What algorithms should be used in insurance risk prediction and how can they be compared?".* The question was posed in general level as the literature has not discussed my dataset. The no free lunches theorem suggested, that no ML method is superior in every problem and hence I am testing a selection of four methods based on the literature. I found the algorithms that are currently succeeding in solving problems analogous to insurance risk scoring and the best practices to evaluate them. Model selection and their evaluation is discussed in section 4.4.

Answers for the 1st set of research question summarize the theory part of this thesis. I will answer the 2nd set of questions in the end of experimental part of the thesis in chapter 5.

4. IMPLEMENTING CLASSIFICATION ALGORITHMS TO CUSTOMER DATA TO PREDICT CLAIM RISK – A CASE STUDY

In this chapter I present the experimental work of applying machine learning methodology from literature to predicting claim risk. I discuss the assumptions and choices made within the uncertainties and limitations in the methodology and data sections. Figure 8 outlines a simplification of applied machine learning prediction process, which I followed in the experimental study. The main titles of this chapter loosely follow the process figure.

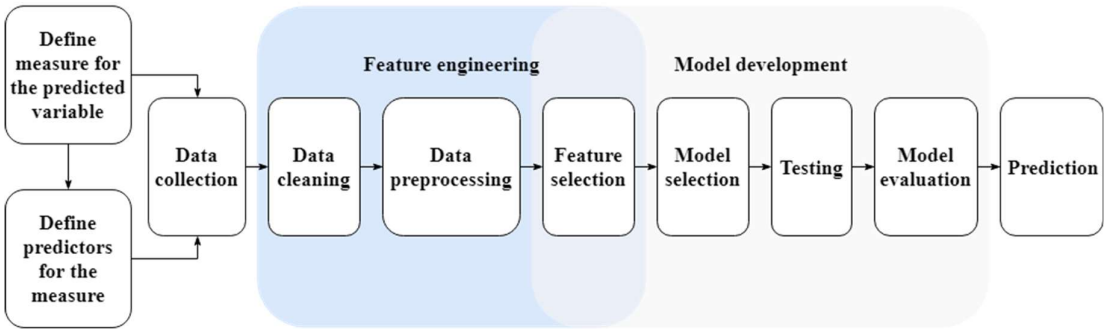

Figure 8. Implementing applied machine learning prediction systems

4.1 Descriptions of variables and data collection

The predicted variable, related to the total risk of an insurance customer, is the value of the yearly claim payments (EUR) paid to the insured. In other words, the goal was to predict the overall risk of an insurance customer, measured in total claim payments. The metric does not capture damages not covered by the insurance, therefore it defines the risk from the insurer perspective, rather than insured. Selecting total risk over specific individual risk as the predicted variable enables using a wider range of predictor variables and more data points to train and test the models. I collected historical data for the predicted variable from years 2013 to 2016.

The predictor variables (features) can be divided into three groups, according to their origin. The largest feature group was risk survey answers, obtained from questions selected by risk analysts for each customer individually. Having an answer for one question was a requirement for an insured to be present in the training data. The surveys map long-term

risks and answers up to the last day of the predicted year were considered as predictors. The second feature group was assignments made to the insured by the insurer, based on the survey answers. Basically, the insurer has detected a risk from the survey answers and has then suggested measures to reduce or remove that risk. The assignments made before 30th June of the predicted year were considered as to have possible effect on the year's claims. Answers and assignments were collected from year 2012 to 2016. The third and final group of features include revenues (EUR), salaries (EUR), number of employees and industry the insured company operates in.

These features together form the accumulated state of the customer at certain point of time. I acknowledge the problem with timing of the state. As the predicted metric is an accumulated figure of a whole year, I could not time the accidents and get the accurate state of the insured company at the time of the occurred damages. I generated the state based on the answers by assuming the past answers hold until the predicted year and that the reported state has been in force for whole predicted year. The goal of the study was to bring together ML methodology, insurance business and data rather than propose optimal solutions. The results should encourage future work to find the optimal data available for the specific problem at hand.

4.1.1 Feature selection

I did not conduct methodological feature selection because I wanted to maintain comparability between the learning algorithms, as they might unevenly benefit from those procedures. As the risk data used was survey answers and assignments to the customers, I could argue that feature selection in a sense was done by the risk manager who assigned them. This means the data should not include features that are insignificant to risks. Not using learning algorithm specific feature selection is a potential limitation to the performance of the classifiers, which I acknowledge. Overall performance of the classifiers should be analyzed with this restriction in mind.

4.1.2 Defining the risk classes

I had to define new risk classes because risk classes for this overall risk did not exist. The risk metric I had chosen was yearly total claims (EUR) and a logical way to define classes was to determine certain ranges of values for each class. Because no prior knowledge about the ranges or number of classes was available and the scope of the study is in the

methodology, I chose options that would limit methodological problems. Because of scarcity of training data, I used binary classes for *higher risk* and *lower risk.* I used statistical methods to split the observations into two balanced classes by calculating the median of yearly total claims in my sample. The two risk classes were therefore defined as:

higher risk	≥	median total claims
lower risk	<	median total claims

Adequate number of observations from all classes is needed to train and test the ML mode, as skewed distribution of class labels could amplify the problems caused by scarcity of data.

4.2 Data cleaning and preprocessing

I collected the data from the systems in comma separated values (CSV) files, each containing different features. The ML classification algorithms require the data to be in certain format and I needed the data in a format that represents the designed model. I designed a data matrix where each row represents a state of one insured in the predicted year. Each column is a specific feature in that state. For each possible answer/assignment/industry/etc. at given time, the insured would have **1** (one) indicating that feature is active, **0** (zero) for feature is not present, or a continuous value if the feature is continuous (i.e. revenue). To achieve this structure, and ML algorithm requirements, the datafiles needed to be merged, continuous features normalized, categorical features binarized and missing data handled. I conducted the data processing with Python, mostly using the Pandas library. For detailed description of Pandas, see McKinney (2013).

4.2.1 One-hot encoding of features

Answers, assignments, industry and number of employees were handled as categorical variables and one-hot encoded in the preprocessing phase. I also transformed the yearly claim payments to binary risk classes by encoding them to values **1** and **0** to represent the risk classes defined in 4.1.2. Higher risk values were encoded to **1** and lower risk to **0**.

4.2.2 Handling missing data

In my data the continuous features revenue and salaries had missing data. Categorical features were one-hot encoded which resolved the problem from their part. I did not want to lose any observations, so I chose to use imputing, instead of removing observations with

missing data. I conducted distribution analysis to find the appropriate imputing method for revenue and salaries. Figure 9 shows the long-tailed distribution shape on both features, the axis labels were not drawn, because the shape of the distribution is the dictating factor in this consideration. I decided to use feature median to impute both features as feature mean would lead to uncommon values.

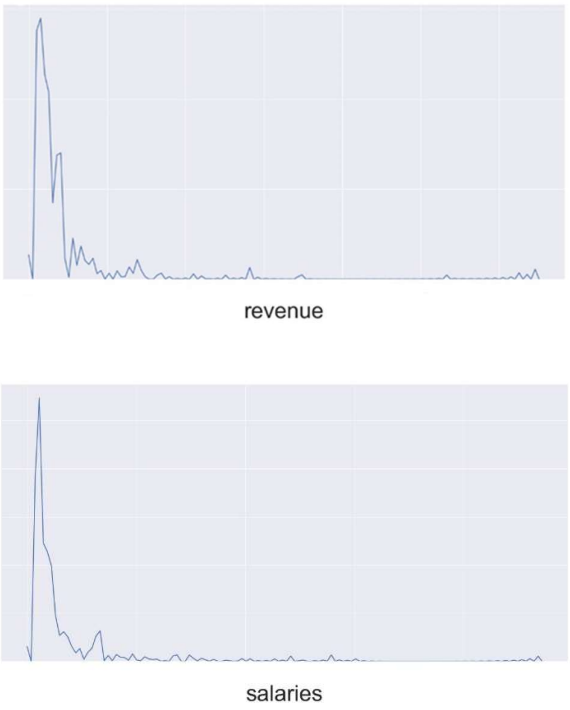

Figure 9. Observation distributions for features *revenue* and *salaries*

4.2.3 Normalization of features

The features have to be normalized because features in different scales can skew the models by emphasizing the effect of features on larger scales. One-hot encoded features were normalized to membership **1** and no membership **0**. In my data continuous features salaries and revenues needed to be normalized. There are many different normalization methods to consider and the choice could have a large effect on the performance of the classifier. Of my total 2143 features only two were continuous, so I assumed a limited effect from the optimization of the method. Optimizing is not a trivial task, nor in the scope of the study, therefore I decided to use a common L2 normalization method where each value in the

feature is divided by the square of the sum of squares of all the original feature values. The continuous variables are then scaled to have values between **0** and **1**. I then assured the normality of the matrix by computing its minimum and maximum values, which were **0** and **1** respectively. This means each value in the matrix is between 0 and 1, and the matrix is normalized to this scale.

4.3 Data structure and statistics

The dimensions of the cleaned dataset used in the analysis are 1945 rows (observations) for 2143 feature columns. The dataset is relatively short and wide. This might cause problems as there might not be enough samples to represent the complexity of the data. To revise, each row resembles a state of a customer in the predicted year, constructed of values in columns (features).

4.3.1 Splitting the data to train and test sets

I conducted 70/30 (train/test) split which can be considered an intuitive starting point. My dataset has 2143 features and pairwise comparison of each between train and test sets is not feasible or necessary, but I needed some evidence to validate the datasets.

I decided to explore the variances by finding a few important features and then comparing their variance in train and test datasets. I conducted the analysis by fitting an ensemble tree model (Extreme Gradient Boosted Trees, XGBT) into the whole dataset before splitting. The algorithm computes relative feature importance for each feature. The five most important features in the dataset for XGBT algorithm are presented in Figure 10. The most important features are *salaries* and *revenue* with relative importance of 0.123 and 0.071, respectively. The next features (prefix “ans” for risk survey question answer and prefix “ind” for industry) have much lower scores, so it was justified to use the two highest scoring features for variance analysis.

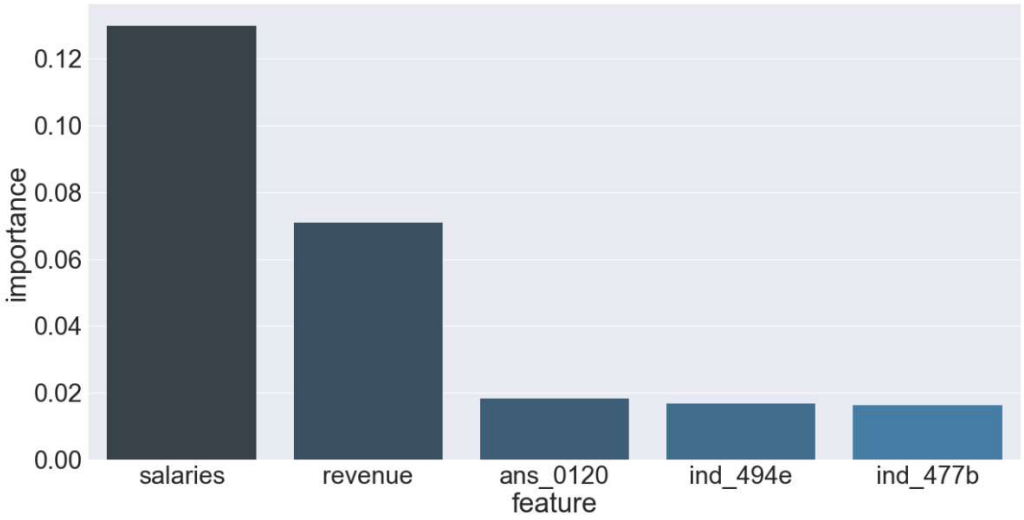

Figure 10. Feature importance for full dataset

Next, I shuffled the data and did a random 70/30 split to train and test sets. I computed mean and standard deviation (std.) for each dataset and compared them in the Table 4. For both features the standard deviations and means are close to each other in all datasets. Based on this analysis, I assume adequate similarity of variance between the datasets and that test set is a fair representation of the complete dataset. For the predicted variable, split function assures that the distribution of classes **1** and **0** are equal in both train and test datasets. Train set now has 1361 observations for 2143 features. Test set has 584 observations for 2143 features.

Table 4. Feature statistics for the datasets

	salaries				revenue			
dataset	mean	std.	min	max	mean	std.	min	max
full	0.00157	0.00529	0.00000	0.13001	0.00075	0.00334	0.00000	0.05610
train	0.00152	0.00538	0.00000	0.13001	0.00071	0.00307	0.00000	0.05610
test	0.00169	0.00506	0.00000	0.07294	0.00081	0.00356	0.00000	0.05610

4.4 Model selection

I did the model selection according to the discussion presented in chapters 2 and 3. I developed models based on four families of classifier algorithms:

1. Logistic regression
2. Support Vector Machines
3. Ensemble decision trees
4. Neural Network

I chose logistic regression for its simplicity and interpretability. Logistic regression is intuitive and common starting point for classification problems. It has been used to both insurance risk prediction and credit scoring problems. My hypothesis is that it will not have the best performance for this complex, high dimensional data. Logistic regression also serves the purpose of comparison to highlight the properties of other models. In addition to classification labels, logistic regression provides probabilities confidence values regarding its results.

I chose SVM because it was single the most used algorithm in both insurance and credit scoring streams and has undisputed success in latter. It was used in the study closest to my problem setting of fire risk prediction by Lau et al. (2015) with good results. It also makes sense to include at least one distance-based algorithm to the selection.

Ensemble decision trees emerged in the literature time to time and I want to test them for their interpretability and success also outside academic field. In recent years, they have been one of the most used algorithms in data science competition platform *Kaggle's* winning solutions (kaggle Inc., 2018). I chose Extreme Gradient Boosted Trees (XGBT) algorithm, which is an ensemble method which makes use of boosting (first trains weak learners, then focuses on the observations they misclassified). XGBT represents non-parametric tree models in my selection. XGBT computes relative feature importance that is used in dataset variance analysis and gives information for feature selection.

Neural networks have been extremely successful and popular in range of problems and emerged also in risk prediction literature. NN are so popular in the field of applied ML field, that it would be difficult to justify leaving them out of this study.

Developing and testing models based on this selection of methods should give adequate evidence of viability of the application. These methods should also bring up a good coverage of considerations to present applied ML prediction process and my knowledge about it. All these algorithms have different properties, limitations and strengths. They treat the data in different ways and stress different properties of data to solve the problem. My hypothesis is, that even developed and tuned, they will give differing results.

I did all analysis using Python programming language and its analysis libraries. Python (Python Software Foundation, 2018) is a versatile open source programming language, popular in the data science community (Pedregosa, Varoquaux et al., 2011). The base algorithms 1-3 I listed, are available through the *scikit-learn* library (Buitinck, Louppe et al., 2013) and the neural network was built using *keras* (keras.io, 2018). As all the libraries are open source, the source code and specifications can be read from their documentation found through references. The libraries offer a framework in which the algorithm can be tuned and optimized for specific purposes.

4.5 Model evaluation

There are three different evaluation phases for the models. Firstly, models are cross-validated in the model development phase. This validation is done to evaluate the model development activities – to observe if changing the model improved it. Secondly, in-sample performance is evaluated for each optimized model – this tells how well the model fits the training data. Lastly the final models are evaluated with out-of-sample test data for final performance scores.

The literature suggests using the confusion matrix and ROC AUC metrics for classifier performance evaluation. I used both in the evaluation of the final models. The confusion matrix metrics depend on the classifier use-case and if there is increased incentive or risk predicting positives or negatives correctly. In claim risk prediction I chose the "risk-averse" strategy, where I considered misclassifying higher risk as lower risk larger error than misclassifying lower risk as higher risk. In confusion matrix terms, I favor Recall over Precision, which I accomplished by using F_β score as evaluation metric. F_β is computed according to the formula:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

where β is the weight for recall. I will use β = 1.5 to favor recall slightly in the score. For cross-validation I used ROC AUC metric. Accuracy is used as baseline metric for the model fitness, as an indicator of model-data fit. F_β is used as the final determinant to rank the models

4.5.1 Evaluating the interpretability of the models

Model interpretability is a vague term but an important consideration in real-world applications such as insurance risk prediction. As I am unaware of a good framework to compare the interpretability of the models, I will report my findings from studying the different algorithms and using them for classification problem in the experimental study. I will assess two aspects of interpretability: *prediction confidence* and *feature importance*. In claim risk scoring applications, user would often be interested in prediction confidence i.e. "how sure" model is of individual predictions. Prediction confidence should be analyzed with prediction accuracy in mind. If the classifier has high out-of-sample accuracy, one might set looser threshold for ambiguous cases. With lower accuracy the classifier is making more mistakes and its user should be more aware of the confidence metrics. Prediction confidence is available for LR in form of conditional probability score of the positive class. This score can be interpreted as "this customer belongs to the higher risk class with a probability of ...". As XGBT uses a logistic loss function, its predictions are also conditional probabilities. For SVM there is a similar metric. SVM is based in distances and for the final classifier each observations distance from the decision boundary can be measured. Platt's scaling (Platt, 1999) can be used to transform the distance to a probabilistic figure. For neural networks, deriving probabilities is less straight-forward and I am not familiar with any plausible method to obtain them.

Another look into interpretability is feature importance. It answers the question of what features were the most important determinants for the prediction, in other words, why was certain label assigned to the case. The different feature importance metrics can be used in feature selection phase of model development to improve the performance of the classifier. They also give important information on the decision process for further risk analysis. LR solves coefficients which can be interpreted as probabilistic weights, which can be used to

analyze the magnitude of each feature to the prediction. For SVM the coefficients resemble the feature vectors' absolute distances from the separator hyperplane. Larger size of a coefficient corresponds to larger feature importance as these features are situated farther from the hyperplane. Interpretation of these coefficients is less straight-forward to those of LR, but Guyon, Weston et al. (2002) have proposed using their coefficient squares as ranking for feature importance. XGBT computes metric called *relational importance* is based on how many times each feature was used to determine a split in the trees and how much each split improved the model predictive power. From single-hidden layer NN the weights of the first inputs are maybe the most intuitive place to measure for relative feature importance. Relative, as the rest of the network has a large influence on the result. In fully connected network each input node is connected to each node in hidden layer. Each input's connections should be therefore summed, and the sum normalized with the corresponding figure from other inputs. See Table 5 for summary of interpretability properties.

Table 5. Prediction confidence and feature importance properties

	LR	**SVM**	**GB**	**NN**
Prediction confidence	Conditional Probabilities	Conditional Probabilities	Conditional Probabilities	Not available
Feature importance	Probabilistic weights	Relational importances	Relational importances	Partial relational importances

As there is no general metric for interpretability and the metrics or statistics provided by different models are not comparable in quantitative methods, I simply describe the models' properties in Results chapter 5 and analyze their implications in the Conclusion chapter 6.

4.6 Model development

Model development in this study includes model specific data preprocessing and algorithm parameter optimization. I chose to use grid-search method with cross-validation for parameter optimization. My model is static and not continuously updated with new data, so the longer training times are not a major problem, but noteworthy especially in dynamic

applications. I used 15-fold cross-validation in parameter optimization to evaluate the changes in the parameters by comparing the AUC means of the cross-validated models.

I used same model development process for each of the four models. First, I trained and cross-validated a baseline model with minimal consideration into parameters. This evaluation would be used as comparison to make sure the change in parameters improves the performance of the model. I then proceeded to grid-search parameter optimization and defined the "optimal" model. I performed grid-search for one parameter at time, by setting a range of values to cross-validate. If the best performing value was extreme value in the range, I shifted the range up and ran grid-search again. I evaluated that model with test dataset and iterated back to baseline model if there was a fixable problem with the results. I will next present the development processes for each model.

4.6.1 Logistic regression model specification

For logistic regression, there are a few parameters to control. I used grid-search to optimize the most important ones: *solver algorithm (solver), penalty* and *regularization. Solver* is the algorithm used to optimize the logistic regression problem. *Regularization* is used to penalize for complexity to achieve better out of sample performance. It can also be thought as limitation to how unusual patterns in the data are incorporated to the model. *Penalty* parameter is also used to control overfitting.

***Baseline model LR0*:** I trained the baseline model with the scikit library's preset parameters (*solver = liblinear, penalty = l2, regularization/C =1.0).* The metrics for baseline model LR0 were:

Accuracy (in-sample):	0.8832
AUC Score (in-sample):	0.9543
F-score (in-sample):	0.8763
CV AUC Score:	Mean - 0.7368 | Std - 0.0599 | Min - 0.6364 | Max - 0.8305

CV AUC of 0.737 is a good first result and std. of 0.060 tells that there was significant variance of performance between the data batches partitioned in cross-validation. Higher in-sample AUC of 0.954 suggests overfitting to the data.

Model LR1: The best performing parameters I found with grid-search were *SAGA solver* (Defazio, Bach et al., 2014)*, l1 norm* for *penalty* and *regularization term* of 1.0. The cross-metrics for model LR1 were:

Accuracy (in-sample):	0.8259
AUC Score (in-sample):	0.8948
F-score (in-sample):	0.8117
CV AUC Score:	Mean - 0.7523 | Std - 0.0625 | Min - 0.6485 | Max - 0.8375

Parameter optimization increased the cross-validated mean AUC by 0.016 to 0.7523. The change in penalty and regularization terms has also decreased the gap between in-sample and CV AUC scores which implies a less overfit model. Change in the solver algorithm is likely to have caused the improvement in precision and *l1* penalty term is the probable cause for reduced overfitting.

4.6.2 Support Vector Machines specification

SVM parameters include *kernel, gamma* and *C.* The kernel determines the form of the function used to draw the hyper-planes that separate the observations. Linear kernel was presented in Figure 4, but it can be for example sigmoid or polynomial function. Gamma is a coefficient for the kernel determines how exact fit model attempts to achieve. *C* penalizes the error term and controls the "smoothness" of the decision boundary. In general terms, smoother classifier generates "clearer" groups, and could generalize better. Higher *C* means less restricted model and higher accuracy.

Baseline model SVM0 was trained with linear *rbf* (Radial Basis Function) kernel, *C*=1 and gamma of $\frac{1}{2143}$ ($= \frac{1}{number\ of\ features}$). Results for SVM0 were:

Accuracy (in-sample):	0.5121
AUC Score (in-sample):	0.7532
F-score (in-sample):	0.7733
CV AUC Score:	Mean - 0.6982 | Std - 0.0500 | Min - 0.6045 | Max - 0.7771

SVM0 had CV AUC of 0.698, but accuracy of only 0.512 which tells that this classifier is only barely better than random guess in classifying the samples. The fit to data is not good and I would not trust the CV results.

Model SVM1*:* Best classifier performance was obtained with linear kernel, penalty term *C* of 0.1 and gamma of 0.1:

Accuracy (in-sample):	0.8303
AUC Score (in-sample):	0.8954
F-score (in-sample):	0.8161
CV AUC Score:	Mean - 0.7384 | Std - 0.0452 | Min - 0.6623 | Max - 0.8066

The accuracy of the model went up from 0.512 to 0.830, a significant improvement in model fitness. CV AUC improved by 0.04 to 0.738 and is much more credible, with improved accuracy score. The most significant change in parameters was change to linear kernel which seems to work better with this high dimensional data. The initial gamma might have been too small for the complexity of the data and caused the model to underfit.

4.6.3 Extreme Gradient Boosted Trees specification

For Extreme Gradient Boosted Trees (XGBT) classifier there are more parameters to optimize. The parameters can be divided into three groups: tree-specific, boosting, and other. Tree-specific parameters affect each individual tree in the model. They are constraints to the set of rules, which the algorithm generates from the data. I optimized the model for following parameters.

- *Minimum samples per split* defines how many samples are required in a node for it to be considered for splitting. Lower values allow formation of rules based on smaller number of samples, higher values limit generation of new rules and therefore form more general models.
- *Minimum samples per leaf* sets limit to how many samples are needed to for a terminal node. The logic is analogous to the previous.
- *Maximum tree depth* represents the desired overall complexity of the model e.g. how many consecutive splits are allowed for a "branch" of a tree.
- *Max features considered for a split* is used to control how many features each split takes into account. Limiting this prevents the model for using all features for all the splits and just learning the structure of the training data. All these parameters control for overfitting by limiting the complexity of the model.

The boosting parameters are higher level considerations:

- *Learning rate* determines how much each tree contributes to the final result. Boosting is an iterative process where first one tree is modeled, next tree complements the previous estimates and so forth. Learning rate controls how much each iteration can change the estimates.
- *Number of trees* is the number of individual trees that are modeled and iterated over.

- *Subsample per tree* determines how much of the whole dataset is used for each tree and increases variance between the trees. Complexity versus generalization is the main question in these parameters too but number of trees also has a significant effect on the number of computations needed to solve the function, which is an important consideration with limited computational power.

Baseline model GB0. I am not aware of a theoretic method to determine the starting parameters. I used the library preset parameters again for the baseline:

- Learning rate = 0.1
- Number of trees = 100
- Subsample per tree 1.0
- Minimum samples per split = 2
- Minimum samples per leaf = 1
- Maximum tree depth=3
- Maximum features considered for a split=2143.

The results for GB0 were:

Accuracy (in-sample):	0.8119
AUC Score (in-sample):	0.9053
F-score (in-sample):	0.7928
CV AUC Score:	Mean - 0.7838 | Std - 0.0636 | Min - 0.6381 | Max - 0.8792

GB0 had good baseline results with CV AUC of 0.784. Higher in-sample AUC 0.905 suggests overfitting. Std of 0.064 again implies of variance between the models which could suggest variance in the CV data batches or variance in their sparsity.

Model GB1: The best performing parameters found with grid-search were:

- Learning rate = 0.1
- Number of trees = 140
- Subsample per tree = 0.8
- Minimum samples per split = 14
- Minimum samples per leaf = 1
- Maximum tree depth = 7
- Maximum features considered for a split = 536).

The results for GB1 were:

Accuracy (in-sample):	0.9177
AUC Score (in-sample):	0.9767
F-score (in-sample):	0.9127
CV AUC Score:	Mean - 0.7914 | Std - 0.0680 | Min - 0.6354 | Max - 0.8757

Parameter tuning improved the CV AUC by 0.008 to 0.791. Very high in-sample AUC of 0.977 suggests overfitting, but as out-of-sample performance improved, GB1 can be accepted as the better model.

4.6.4 Neural network specification

Neural networks offer a huge range of options and considerations for the architecture and specifications. I decided to use an architecture similar to the one Baecke and Bocca (2017) used in auto insurance claim risk prediction. The network has one input layer, one hidden layer and one output layer. This architecture (provided with enough nodes and training data and a non-linear activation function) should theoretically be enough to approximate any function (Hornik, 1991). The network is linear and dense, all the nodes in a layer are connected to all nodes in the next layer. The number of nodes in the first layer corresponds to number of features (2143) and output layer has a single node. Choosing a shallow (less layers) network might limit the networks capability to learn but maintains a level of interpretability. As no accepted theory exists for number of nodes in hidden layer, I used a rule of thumb: mean of nodes in input and output layers, 1072. I wanted to optimize for:

- Algorithm to be used to optimize the weights
- Initial weights for a starting point to begin the optimizing from
- Number of epochs (iterations)
- Size of data batch fed through the network in one iteration

Baseline model NN0: For the baseline model I used *RMSprop* optimizer (Hinton, 2012), 30 epochs, batch size of 10 and *glorot uniform* method (Glorot & Bengio, 2010) for weight initializer. The results for NN0 were:

Accuracy (in-sample):	0.9471
AUC Score (in-sample):	0.9932
F-score (in-sample):	0.9530
CV AUC Score:	Mean - 0.7387 | Std - 0.0360 | Min - 0.6663 | Max - 0.8116

Baseline model CV AUC was 0.739, with std of 0.036. Significantly higher in-sample accuracy and AUC scores suggest overfitting to the data.

Model NN1: The parameters found in grid-search were *ADAM* optimizer (Kingma & Ba, 2014), glorot uniform initializer, 200 epochs and batch size of 130. The results for NN1 were:

Accuracy (in-sample):	0.9456
AUC Score (in-sample):	0.9938
F-score (in-sample):	0.9586
CV AUC Score:	Mean - 0.7487 | Std - 0.0391 | Min - 0.6748 | Max - 0.8328

Parameter tuning improved CV AUC by 0.010 to 0.749.

4.6.5 Summary of model development

Each of the base models improved with parameter tuning procedures. The cross-validated AUC scores for different models were in a surprisingly small range (min 0.738, max 0.791). I assumed that there would be larger variation in performance between the models. Signs of overfitting were present with all models, as the in-sample scores were significantly higher than the cross-validated scores. If this assumption about overfitting is correct, the evaluation scores with test data should be lower than the cross-validated scores. All in all, the model development phase results are promising as the scores imply that each algorithm performs better than random guessing on data collected with minimal feature selection.

5. RESULTS

In this chapter, I report the test results of the final models LR1, SVM1, GB1 and NN1 and answer the 2nd set of research questions introduced in section 1.1. I first present and discuss the confusion matrices, proceed to analyze the Receiver Operating Characteristic curves and finally summarize the main evaluation metrics. Evaluation of interpretability of the models is documented in a separate section at the end of this chapter.

5.1 Confusion matrices

Confusion matrices for each model are presented in Figure 11. All the evaluation metrics used (refer to chapters 2.5.1 and 4.5) are computed from confusion matrix figures. I begin the analysis of the results from overall prediction accuracy presented in Table 7. GB1 reached the highest accuracy of 0.726 followed by SVM1 with 0.721 and LR1 with 0.702. NN1 had the lowest accuracy score of 0.678. In general terms, the best (GB1) model predicted 72.6% of the unseen risk classes correctly.

Confusion matrix for LR1

n = 584	Predicted Lower risk	Predicted Higher risk	
Actual Lower risk	TN = 225	FP = 84	309
Actual Higher risk	FN = 90	TP = 185	275
	315	269	

Confusion matrix for SVM1

n = 584	Predicted Lower risk	Predicted Higher risk	
Actual Lower risk	TN = 234	FP = 75	309
Actual Higher risk	FN = 88	TP = 187	275
	322	262	

Confusion matrix for GB1

n = 584	Predicted Lower risk	Predicted Higher risk	
Actual Lower risk	TN = 227	FP = 82	309
Actual Higher risk	FN = 78	TP = 197	275
	305	279	

Confusion matrix for NN1

n = 584	Predicted Lower risk	Predicted Higher risk	
Actual Lower risk	TN = 199	FP = 99	298
Actual Higher risk	FN = 89	TP = 197	286
	288	296	

Figure 11. Confusion matrices

To evaluate the confidence in these figures, I reviewed how the observations distribute between classes in the test data and analyzed the confusion matrix. Earlier, in chapter 4.1.1, I defined the risk classes so that 50% of observations would be higher risk and 50% lower risk. The split to train and test datasets was also conducted so that the same distribution was maintained, in test set 47.1% of observations were higher risk and 52.9% lower risk. From this distribution I can already conclude that 0.726 accuracy was not achieved just by predicting same class for each sample. This means each model has learned some kind of general structure from the data.

Next, I wanted to analyze how the performance compares between positive and negative test subjects. As I discussed in section 4.5., in this specific problem it is more important that high risk is not predicted as low risk than vice versa. To assess this, I computed *False Negative Rates* ($\text{FNR} = \frac{FN}{FN + TN}$) for each model (Table 6). The larger the rate, the worse the classifier in the aforementioned criteria, so GB1 with FNR of 0.172 was the most accurate in this sense. For comparison, let us look at the *False Positive Rates* ($\text{FPR} = \frac{FP}{FP + TP}$). As desired, each model had significantly lower FNR than FPR. SVM1 had the best FPR of 0.286.

Table 6. False Negative and False Positive rates

	LR1	SVM1	GB1	NN1
FNR	0.200	0.188	0.172	0.224
FPR	0.315	0.286	0.294	0.334
rank by FNR	3	2	1	4

5.2 Receiver Operating Charasteristic curves

For each model, I drew a ROC curve (Figure 12) to present its TPR and FPR in different classifying thresholds (different limits for probability to determine if observation belongs to class or not). There is no large difference in the shapes of curves and there is no significant variance in their behavior between different thresholds. The red diagonal line in each graph represents the “random guess classifier” benchmark. As each ROC stays above the diagonal, they seem to work better than random chance in all thresholds. The ROC metric was computed for each model to represent their overall predictive power. GB1 had the best out-

of-sample AUC of 0.781 and NN1 was the worst performing classifier with AUC of 0.729. LR1 scored 0.751 and SVM1 0.752 respectively.

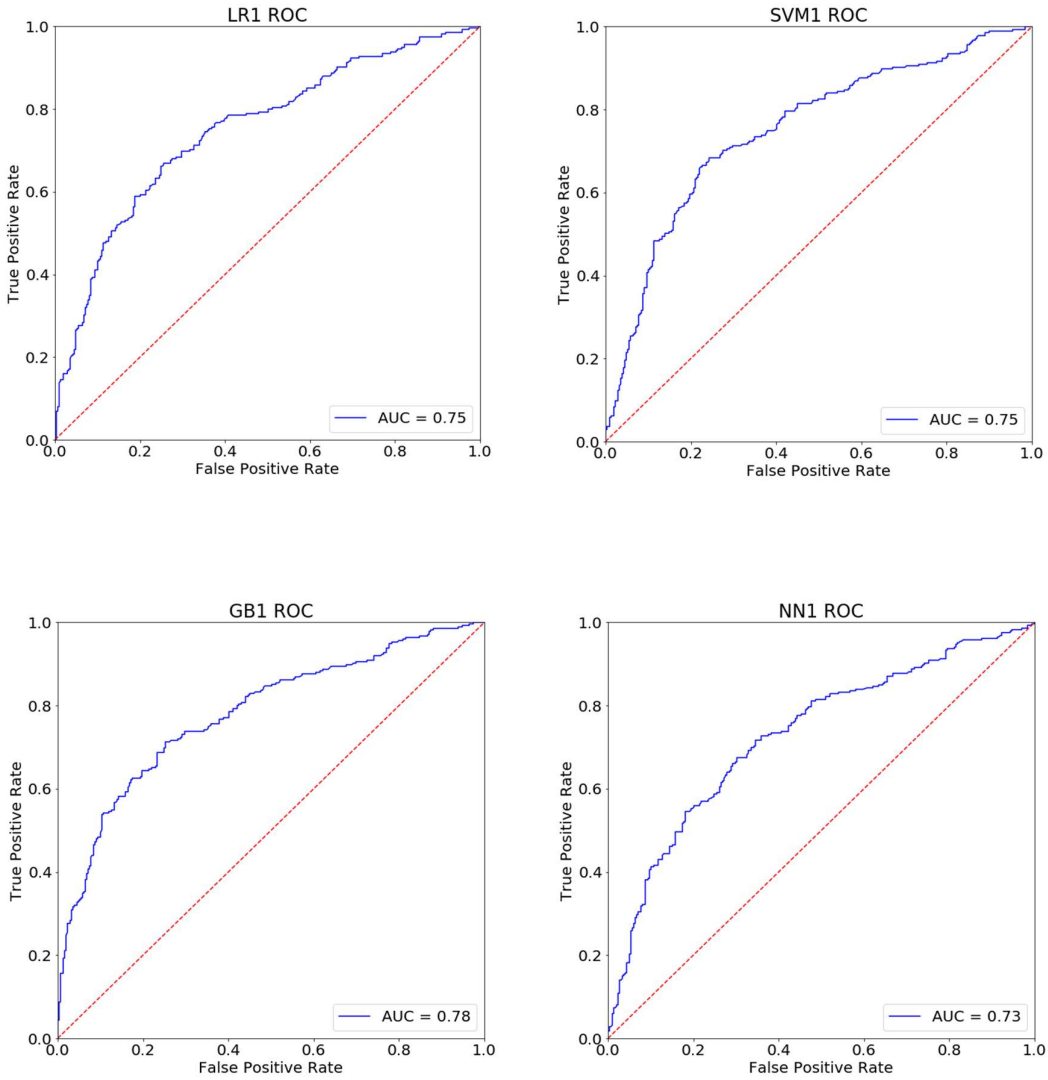

Figure 12. Receiver Operating Characteristic curves

5.3 Summary of evaluation metrics

Based on the results of the experimental study I can answer the 2nd set of research questions titled: *"How do different machine learning algorithms compare in predicting insurance claims based on customer data?"*. The first sub-question (2a) was posed as: *"How do*

different methods compare in evaluation metrics with the given customer data set?". Table 7 presents the summary of the key evaluation metrics. Accuracy was discussed earlier section 5.1. The overall rank is dictated by higher F_β − score (refer to chapter 4.5). GB1 outperformed other classifiers with F_β of 0.713, followed by SVM1 (0.690), NN1 (0.681) and LR1 (0.677). GB1, model based Extreme Gradient Boosting Trees algorithm was the best performing classifier in all selected metrics with my test data. I will further discuss the limitations of the experiment in Conclusions chapter 6 but note that the results resemble the data used and model development choices made.

Table 7. Summary of evaluation metrics

	LR1	SVM1	GB1	NN1
Accuracy	0.702	0.721	**0.726**	0.678
F-score	0.677	0.690	**0.713**	0.681
ROC AUC	0.751	0.752	**0.781**	0.729
Overall rank	3	2	**1**	4

The second sub-question (2b) was posed as: *"Can ML algorithms be used to create insurance risk scores from the given customer data set?"*. The best classifier labeled 73% of the test customers to correct risk classes, which can be considered a very promising performance for a real-world application. More research invested in any phase of the ML prediction process introduced in chapter 4 could potentially improve the classifier performance. In this experiment, I designed an ML risk prediction system that created risk scores from customer risk data.

5.4 Prediction confidence and feature importance metrics

In the experimental study I created risk prediction models using risk survey questions as predictors. In this last section, I present my findings on interpretability properties of these models to answer the last sub-question posed as: "*Can ML analysis be used to quantify weights for the risk survey questions?"*. Normalized feature importance metrics of a model could be used as weights representing the weight for risk questions. For RL, SVM and GB these metrics were equally available. NN did not have a prediction confidence metric and the feature importance metric was derivable by feasible methods only partially.

6. CONCLUSIONS

In this chapter I will analyze the implications of the results, their limitations and further research directions. On the basis of results attained in this thesis, machine learning methods can be used to create insurance claim risk scores from customer risk data. The results attained in the case study present in this thesis imply that Extreme Gradient Boosted Trees method would be most suitable for this task. My results align with Guelman (2012), who found XGBT to fit well to vehicle insurance claim data for loss cost prediction. However, the applicability of different methods is very much dependent on the data structure and definition of the predicted classes, therefore the ranking of the algorithms cannot be generalized to other datasets. The results imply that all the tested methods were suitable for the classification task. Baecke et al. (2017) had similar results with vehicle data using logistic regression, neural network and random forest classifiers, in terms of AUC they all achieved similar precision on a scale, and the ranking of algorithms changed with each different feature selection methods used.

6.1 Analysis of results

The results of my experimental study were interesting from two perspectives. Firstly, the evidence suggests machine learning predictive algorithms can viably be used in insurance risk prediction for risk scoring. There is much room for improvement, but the fact that all four very different classifiers were performing at this level (AUC > 0.73) suggests that the ML methodology works, and further research could should lead to improved performance. An ML risk scoring system allows computing the risk scores to whole customer portfolio simultaneously and automatically updating them when new risk data is available. Automated system frees up expert risk analyst time for ambiguous cases, risk data gathering and risk reduction/mitigation procedures. The ambiguous cases could be determined for example by setting a threshold for a prediction confidence metric to flag the customers the scoring system is unsure of. The business processes, such as sales or CRM benefit from fast, up-to-date risk assessments. In this thesis, I defined the insurance risk as overall claim risk, but creating similar systems for individual risks would allow things such as automated risk-based product recommendations and risk mitigation assignments.

Secondly, the analysis suggests that the methods could be used to measure the importance (weights) for risk survey questions. As risk assessment through surveying the customer is

no trivial task, it is important to know the right questions to ask. Weighted risk survey questions can be used to find optimal set of questions for each customer. Assessing the individual risk factors (features), and targeting risk reduction or mitigation measures to them, could allow insurance companies to reduce the claim payments and therefore increase overheads without increasing prices and losing competitive advantage.

I have shown evidence for usability of machine learning classification algorithms in claim risk prediction. I also have highlighted several points where the models I experimented with could be further developed. Further, I found that machine learning risk prediction enables analysis of individual risk factors and the feature importance metrics can be used to quantify weights for risk survey questions.

6.2 Limitations and ideas for future research

The limitations of this thesis should be considered with its scope and goals. The scope was in relatively high level and the goal was to create an overall picture of ML claim risk prediction systems and their problems and possibilities. I chose this approach, because an overall picture of the research area was not previously available. To maintain the scope, I had to overlook many interesting areas. I hope the readers of this thesis recognize these parts as inspiration for future research. I do not propose new methodology for claim loss reserving or try to challenge the currently used models for this problem. Further, I do not propose optimal system for insurance claim scoring problem nor did I attempt to find them. Learning classifiers are also dependent on the data they were trained with. I want to highlight out a few ideas of future research that sparked my own interest.

Many of the model development steps require model specific data pre-processing and feature selection. For the purpose of comparability of the models, I excluded such methods, and used exactly same dataset to train all the models. Unarguably, feature selection is a very important phase of model development and could have improved the classifier performances significantly. For example, some algorithms could have benefited from dimensionality reduction and others not. Interesting topic for further research would be to choose one of the algorithms and go through the same process by optimizing each step for that algorithm. I am optimistic, that significant improvements to classifier performance (more accurate risk scores) could be achieved.

As I mentioned in the previous section, moving the scope to individual risks (for example fire or automobile damages) is another interesting direction for future research. Instead of basing the risk score on predicted overall claims, the individual risks could be scored, and overall risk assessment would be a combination of them. Individual risks can be measured from claims, which have accident date, sum of damages and other data which allow more accurate timing of the accident and a timestamp from which to collect records for customer state.

My study concentrated on corporate insurance, but one future research direction is to apply the methods for private household customers. Arguably, the risks of households are more homogenous than of those of businesses, especially when the households are segmented to relevant groups. Patterns in data with larger number of more homogenous observations would intuitively be easier for learning algorithms to pick out. In household insurance where number of contracts is larger, but their values smaller, automation of risk assessment process is a business goal with clear benefits.

REFERENCES

Aksoy, S., & Haralick, R. M. (2001). Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters,* 22(5), 563-582.

Artís, M., Ayuso, M., & Guillén, M. (2002). Detection of automobile insurance fraud with discrete choice models and misclassified claims. *The Journal of Risk and Insurance,* 69(3), 325-340.

Baecke, P., & Bocca, L. (2017). The value of vehicle telematics data in insurance risk selection processes. *Decision Support Systems,* 98, 69.

Bailey, R. A., & Simon, L. J. (1960). Two studies in automobile insurance ratemaking. *ASTIN Bulletin,* 1(4), 192-217.

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications,* 83, 405-417.

Becker, D. (2017). Data leakage. Retrieved from https://www.kaggle.com/dansbecker/data-leakage

Bequé, A., & Lessmann, S. (2017). Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications,* 86, 42-53.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems,* 24

Bertke, S. J., Meyers, A. R., Wurzelbacher, S. J., Bell, J., Lampl, M. L., & Robins, D. (2012). Development and evaluation of a naïve bayesian model for coding causation of workers' compensation claims. *Journal of Safety Research,* 43(5-6), 327-332.

Bishop, C. M. (2006). Pattern recognition and machine learning. New York, NY: Springer.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees (1st ed.). Wadsworth, Belmont, CA: Taylor & Francis.

Brockett, P. L. (1998). Using kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *The Journal of Risk and Insurance,* 65(2), 245-274.

Buitinck, L., Louppe, G., & Blondel, M. (2013). API design for machine learning software: Experiences from the scikit-learn project. *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases,*

Butaru, F., Chen, Q., Clark, B., Das, S., Lo, A. W., & Siddique, A. (2016). Risk and risk management in the credit card industry. *Journal of Banking and Finance,* 72, 218-239.

Chen, H., & Xiang, Y. (2017). The study of credit scoring model based on group lasso. *Procedia Computer Science,* 122, 677-684.

Chen, W., Ma, L., & Ma, C. (2009). Mining the customer credit using hybrid support vector machine technique. *Expert Systems with Applications,* 36(4), 7611-7616.

Cheng, M., Peng, H., Wu, Y., & Liao, Y. (2011). Decision making for contractor insurance deductible using the evolutionary support vector machines inference model. *Expert Systems with Applications,* 38(6), 6547-6555.

Cielen, A., Peeters, L., & Vanhoof, K. (2004). Bankruptcy prediction using a data envelopment analysis. *European Journal of Operational Research,* 154(2), 526-532.

Claesen, M., & De Moor, B. (2015). Hyperparameter search in machine learning. *MIC 2015: The XI Metaheuristics International Conference in Agadir,*

Clarke, B., Fokoué, E., & Zhang, H. H. (2009). Principles and theory for data mining and machine learning. Dordrecht: Springer.

Correa Bahnsen, A., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications,* 51, 134-142.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning,* 20(3), 273-297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory,* 13(1), 21-27.

Cox, D. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological),* 20(2), 215-242.

Dahiya, S., Handa, S. S., & Singh, N. P. (2017). A feature selection enabled hybrid-bagging algorithm for credit risk evaluation. *Expert Systems,* 34(6), n/a.

Danenas, P., & Garsva, G. (2015). Selection of support vector machines based classifiers for credit risk domain. *Expert Systems with Applications,* 42(6), 3194-3204.

Danenas, P., Garsva, G., & Gudas, S. (2011). Credit risk evaluation model development using support vector based classifiers. *Procedia Computer Science,* 4, 1699-1707.

David, M. (2015a). Auto insurance premium calculation using generalized linear models. *Procedia Economics and Finance,* 20, 147-156.

David, M. (2015b). A review of theoretical concepts and empirical literature of non-life insurance pricing. *Procedia Economics and Finance,* 20, 157-162.

Davis, R. H., Edelman, D. B., & Gammerman, A. J. (1992). Machine-learning algorithms for credit-card applications. *IMA Journal of Management Mathematics,* 4(1), 43-51.

de Alba, E., & Nieto-Barajas, L. E. (2008). Claims reserving: A correlated bayesian model. *Insurance Mathematics and Economics,* 43(3), 368-376.

de Andres-Sanchez, J. (2007). Claim reserving with fuzzy regression and taylor's geometric separation method. *Insurance: Mathematics and Economics,* 40(1), 145-163.

Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems,*

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the Em algorithm. *Journal of the Royal Statistical Society, 39*(1), 1-38.

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research,* 95(1), 24-37.

Diers, D., & Linde, M. (2013). The multi-year non-life insurance risk in the additive loss reserving model. *Insurance: Mathematics and Economics,* 52(3), 590-598.

Dong, A. X. D., & Chan, J. S. (2013). Bayesian analysis of loss reserving using dynamic models with generalized beta distribution. *Insurance: Mathematics and Economics,* 53(2), 355-365.

Dougherty, G. (2013). Pattern recognition and classification. New York, NY: Springer.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis. New York, NY: Wiley.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters,* 27(8), 861-874.

Florez-Lopez, R., & Ramon-Jeronimo, J. M. (2015). Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal. *Expert Systems with Applications,* 42(13), 5737-5753.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics,* 9(1), 249-256.

Gudmundsson, S., Oddsdottir, G. L., Runarsson, T. P., Sigurdsson, S., & Kristjansson, E. (2010). Detecting fraudulent whiplash claims by support vector machines. *Biomedical Signal Processing and Control,* 5(4), 311-317.

Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Systems with Applications,* 39(3), 3659-3667.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning,* 46(1), 389-422.

Han, Z., & Gau, W. (2008). Estimation of loss reserves with lognormal development factors. *Insurance Mathematics and Economics,* 42(1), 389-395.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society),* 160(3), 522-541.

Happ, S., Merz, M., & Wüthrich, M. V. (2012). Claims development result in the paid-incurred chain reserving method. *Insurance Mathematics and Economics,* 51(1), 66-72.

Henley, W. E. (1995). Statistical aspects of credit scoring Open University

Hess, K., Schmidt, K., & Zocher, M. (2006). Multivariate loss prediction in the multivariate additive model. *Insurance Mathematics and Economics,* 39(2), 185-191.

Hinton, G. (2012). Neural networks for machine learning, lecture 6a. Retrieved from https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks,* 4(2), 251-257.

Huang, J., Qiu, C., Wu, X., & Zhou, X. (2015). An individual loss reserving model with independent reporting and settlement. *Insurance: Mathematics and Economics,* 64, 232-245.

Hudecova, S., & Pesta, M. (2013). Modeling dependencies in claims reserving with GEE. *Insurance: Mathematics and Economics,* 53(3), 786-794.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. New York: Springer.

Jin Huang, & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering,* 17(3), 299-310.

Kafková, S. (2015). Bonus-malus systems in vehicle insurance. *Procedia Economics and Finance,* 23, 216-222.

kaggle Inc. (2018). Kaggle inc. Retrieved from https://www.kaggle.com/).

Kelly, M., & Nielson, N. (2006). Age as a variable in insurance pricing and risk classification. *The Geneva Papers on Risk and Insurance. Issues and Practice,* 31(2), 212-232.

keras.io. (2018). Keras. Retrieved from https://keras.io/

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking and Finance,* 34(11), 2767-2787.

Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Systems with Applications,* 37(9), 6233-6239.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations,*

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence,*

Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications,* 40(13), 5125.

Kubat, M. (2017). An introduction to machine learning (2nd ed.). Cham: Springer International Publishing.

Kunkler, M. (2006). Modelling negatives in stochastic reserving models. *Insurance Mathematics and Economics,* 38(3), 540-555.

Lau, C. K., Lai, K. K., Lee, Y. P., & Du, J. (2015). Fire risk assessment with scoring system, using the support vector machine approach. *Fire Safety Journal,* 78, 188-195.

Lee, Y. (2007). Application of support vector machines to corporate credit rating prediction. *Expert Systems with Applications,* 33(1), 67-74.

Li, S., Tsang, I. W., & Chaudhari, N. S. (2012). Relevance vector machine based infinite decision agent ensemble learning for credit risk analysis. *Expert Systems with Applications,* 39(5), 4947-4953.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory,* 28(2), 129-137.

Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science,* 21(2), 117-134.

Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. *Expert Systems with Applications,* 42(10), 4621-4631.

Matsui, M. (2014). Prediction in a non-homogeneous poisson cluster model. *Insurance: Mathematics and Economics,* 55, 10-17.

McKinney, W. (2013). Python for data analysis (1st ed.). Sebastopol: O'Reilly Media.

Merz, M., & Wüthrich, M. V. (2010). Paid–incurred chain claims reserving method. *Insurance Mathematics and Economics,* 46(3), 568-579.

Mues, C., Baesens, B., Files, C. M., & Vanthienen, J. (2004). Decision diagrams in machine learning: An empirical study on real-life credit-risk data. *Expert Systems with Applications,* 27(2), 257-264.

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized linear models (3rd ed.). London: Chapman & Hall.

Nian, K., Zhang, H., Tayal, A., Coleman, T., & Li, Y. (2016). Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science,* 2(1), 58-75.

Niklis, D., Doumpos, M., & Zopounidis, C. (2014). Combining market and accounting-based models for credit scoring using a classification scheme based on support vector machines. *Applied Mathematics and Computation,* 234, 69-81.

Paefgen, J., Staake, T., & Fleisch, E. (2014). Multivariate exposure modeling of accident risk: Insights from pay-as-you-drive insurance data. *Transportation Research: Part A: Policy and Practice,* 61, 27-40.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research,* 12, 2826-2860.

Pesonen, E. (1962). A numerical method of finding a suitable bonus scale. *ASTIN Bulletin,* 2(1), 102-108.

Pesta, M., & Okhrin, O. (2014). Conditional least squares and copulae in claims reserving for a single line of business. *Insurance: Mathematics and Economics,* 56, 28-37.

Petropoulos, A., Chatzis, S. P., & Xanthopoulos, S. (2016). A novel corporate credit rating system based on student's-t hidden markov models. *Expert Systems with Applications,* 53, 87-105.

Pitselis, G., Grigoriadou, V., & Badounas, I. (2015). Robust loss reserving in a log-linear model. *Insurance: Mathematics and Economics,* 64, 14-27.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.

Punniyamoorthy, M., & Sridevi, P. (2016). Identification of a standard AI based technique for credit risk analysis. *Benchmarking: An International Journal,* 23(5), 1381-1390.

Python Software Foundation. (2018). Python. Retrieved from https://www.python.org/

Ribeiro, B., Silva, C., Chen, N., Vieira, A., & Carvalho das Neves, J. (2012). Enhanced default risk models with SVM. *Expert Systems with Applications,* 39(11), 10140-10152.

Risselada, H., Verhoef, P. C., & Bijmolt, T. H. A. (2010). Staying power of churn prediction models. *Journal of Interactive Marketing,* 24(3), 198-208.

Shim, J., & Hwang, C. (2011). Kernel poisson regression machine for stochastic claims reserving. *Journal of the Korean Statistical Society,* 40(1), 1-9.

Sinha, A. P., & Zhao, H. (2008). Incorporating domain knowledge into data mining classifiers: An application in indirect lending. *Decision Support Systems,* 46(1), 287-299.

Spreeuw, J., & Goovaerts, M. (1998). Prediction of claim numbers based on hazard rates. *Insurance Mathematics and Economics,* 23(1), 59-69.

Sundarkumar, G. G., & Ravi, V. (2015). A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence,* 37, 368-377.

Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of neural networks: The case of bank failure predictions. *Management Science,* 38(7), 926-947.

Twala, B. (2010). Multiple classifier application to credit risk assessment. *Expert Systems with Applications,* 37(4), 3326-3336.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM,* 27(11), 1134-1142.

Verrall, R. J. (2000). An investigation into stochastic claims reserving models and the chain-ladder technique. *Insurance Mathematics and Economics,* 26(1), 91-99.

Viaene, S., Dedene, G., & Derrig, R. A. (2005). Auto claim fraud detection using bayesian learning neural networks. *Expert Systems with Applications,* 29(3), 653-666.

Wang, G., & Ma, J. (2012). A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine. *Expert Systems with Applications,* 39(5), 5325-5331.

Wang, Y., & Xu, W. (2017). Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decision Support Systems,* 105, 87-95.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly,* 26(2), 13-23.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *Journal of Financial and Quantitative Analysis,* 15(3), 757-770.

Wolpert, D. (1996). The lack of A priori distinctions between learning algorithms. *Neural Computation,* 8(7), 1341-1390.

Xu, W., Wang, S., Zhang, D., & Yang, B. (2011). Random rough subspace based neural network ensemble for insurance fraud detection. *Fourth International Joint Conference on Computational Sciences and Optimization,* , 1276-1280.

Xu, X., Zhou, C., & Wang, Z. (2009). Credit scoring algorithm based on link analysis ranking with support vector machine. *Expert Systems with Applications,* 36(2), 2625-2632.

Yang, Y. (2007). Adaptive credit scoring with kernel learning methods. *European Journal of Operational Research,* 183(3), 1521-1536.

Yu, L., Yao, X., Wang, S., & Lai, K. K. (2011). Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection. *Expert Systems with Applications,* 38(12), 15392-15399.

Yu, L., Yue, W., Wang, S., & Lai, K. K. (2010). Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Systems with Applications,* 37(2), 1351-1360.

Zhang, Y., & Dukic, V. (2013). Predicting multivariate insurance loss payments under the bayesian copula framework. *Journal of Risk and Insurance,* 80(4), 891-919.

Zhao, X. B., Zhou, X., & Wang, J. L. (2009). Semiparametric model for prediction of individual claim loss reserving. *Insurance Mathematics and Economics,* 45(1), 1-8.

Zhong, H., Miao, C., Shen, Z., & Feng, Y. (2014). Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings. *Neurocomputing,* 128, 285.

Zhou, L., Lai, K. K., & Yu, L. (2010). Least squares support vector machines ensemble models for credit scoring. *Expert Systems with Applications,* 37(1), 127-133.

Zhou, X., Jiang, W., & Shi, Y. (2010). Credit risk evaluation by using nearest subspace method. *Procedia Computer Science,* 1(1), 2449-2455.

Zhou, X., Jiang, W., Shi, Y., & Tian, Y. (2011). Credit risk evaluation with kernel-based affine subspace nearest points learning method. *Expert Systems with Applications,* 38(4), 4272-4279.

Zhu, B., He, C., & Liatsis, P. (2012). A robust missing value imputation method for noisy data. *Applied Intelligence,* 36(1), 61-74.