Erasmus University Rotterdam

Erasmus School of Economics

Quantitative Finance

A Framework to Forecast Insurance Claims

Author:
Tim Pijl
371258

Supervisor:
M. van de Velden
Co-reader:
P. Groenen

A thesis submitted in fulfilment of the requirements for the degree of
Master of Econometrics and Management Science

August 2017

Abstract

For an insurance company, the forecasting of claims is central to a successful operation. This process can be divided into multiple subtasks. Data preparation, dimensionality reduction, classification, forecasting and evaluation. This research applies three dimensionality reduction techniques: variable elimination, reduction through a decision forest and multiple correspondence analysis. After dimensionality reduction, classification is used to determine the probability of issuing a claim for an observation to be predicted. Four classification techniques are used: a decision tree, a random forest, a binary logistic regression and a support vector machine. Once the probability of issuing a claim is estimated, it needs to be transformed into a predicted claim amount. As a benchmark, a naive model, called the ratio model, is used. This model uses ratios of risk groups with respect to the base premium to determine the final premium. For the evaluation of the models, classification measures, error measures and the normalized Gini coefficient are used. The results show that dimensionality reduction is not necessarily needed for this problem and that simple techniques, such as a decision tree or random forest, outperform the more statistically advanced techniques, such as a support vector machine, on out-of-sample results.

Acknowledgements

First I would like to thank my parents, Edwin and Angelique, as well as my girlfriend Sharida for their unconditional love and support and their never ending encouragement. They thought me to always do what you love and to never stop chasing your dreams, which helped me to grow as a person. I would also like to thank everybody at Finaps for making me feel at home during my internship. Andrew, Thomas, Simon and Youri in particular for helping me with the problems I have faced during this research, as well as Lonneke and Marlies for giving me this opportunity. Many thanks to my supervisor M. van de Velden which helped me to gain insights on the matter as well as being able to critically evaluate my own work. It was an experience I will carry with me through all my future endeavors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Helping clients in such a way that they are able to use their data to optimize their decision making process is a service Finaps[1] wants to provide. Having a reinsurer as a client, proposing a 'Framework to Forecast Insurance Claims' to them can help them to make better decisions. Giving the reinsurer insights in how their client data can be used to forecast upcoming claims, enables them to act accordingly. Furthermore, Finaps and SAS[2] have established a partnership and SAS wants to use this research as a guideline to a new product called 'Dynamic Pricing'. This product enables insurance companies to dynamically price premiums based on a risk profile.

1.1 Problem Definition

For an insurance company, the forecasting of claims is central to a successful operation. If the claims can be forecasted accurately, premiums can be adjusted accordingly, creating the opportunity to be one step ahead of the competitors. Charging a lower premium than the competitors, while maintaining a sufficient buffer to make profit to stay in business, will lead to more customers, which will in turn lead to more profit. However, charging a premium that is too low to cover the expected expenses can lead to bankruptcy in the long run.

Table 1.1 contains the premiums asked of four different insurance companies. Since the premiums are different, we can conclude that the different insurance companies have different pricing models.

[1] Finaps delivers innovative IT solutions using advanced technologies such as Mendix, SAS, Xamarin and Box. For more information see `https://www.finaps.nl`.

[2] SAS is specialized in statistical analytical software and solutions. For more information see `http://www.sas.com`.

	Achmea	Allsecur	ANWB	Ditzo
Premium[1](€)	702.36	659.16	874.08	861.96

[1] The premium mentioned is a yearly premium for a third party insurance (Wettelijke Aansprakelijkheid). The car to be insured is a Citroën C1 with license plate 58-NPF-3.

Table 1.1: Premium asked for different insurance companies

Assuming that people will always choose for the cheapest possibility and that the insurance companies did not underprice their premiums, AllSecur is one step ahead of its competitors. They charge a lower premium while maintaining a sufficient buffer to stay in business. It must be mentioned that in reality, people do not always choose the cheapest option. Customer service plays a big part in which insurance company someone chooses. However, customer service is not within the scope of this research and is impossible to measure given the data used.

To charge a competitive premium that is neither too high nor too low, it is important to forecast the upcoming claims accurately. To produce accurate forecasts and to give the insurance company insights in the methods that we use, we construct a framework which can be divided into the following five subtasks:

1. Data Preparation
2. Dimensionality Reduction
3. Classification
4. Forecasting
5. Model Evaluation

This research focuses on subtask two to five. The dataset presented in this research already accounts for subtask one, it is a complete dataset containing multiple variables that can directly be used. To understand the importance of subtasks two to five, simple examples are given.

Models that contain a large number of predictor variables relative to the number of observations have the tendency to overfit. That is, the model is excessively complex so that it fits the training data perfectly, but has a poor performance when applied to new data. As a solution, dimensionality reduction techniques may be used to extract only useful information of the dataset. For example, although the start date of each policy is stored into the database, it will most likely not influence the claim amount significantly. Therefore, it is possible to delete this variable to make the model less prone to overfitting and thereby obtain better out-of-sample results.

Using classification, it may be possible to assign a risk profile (e.g. probability to claim) to a client. For example, suppose customer 1 has a low probability to issue a claim next year and customer 2 has a high probability to issue a claim next year. It may not be optimal to charge both customers the same premium because you would overcharge customer 1 and undercharge customer 2. If other insurance companies do differentiate between the different risk profiles, clients similar to customer 1 are likely to leave since they are overpriced compared to the market. At the same time, clients similar to customer 2 are, with respect to their risk profile, underpriced compared to the market. This leads to an increase of clients similar to customer 2, which increases the risk of the portfolio. The premium that was initially charged is now too low, which can lead to bad results.

Once the risk profile of a (new) client is determined, the next step is to transform this risk profile into a predicted claim amount. The assigned risk profile is the predicted probability to issue a claim. To obtain predicted claim amounts, it is possible to give high predicted probabilities a high claim amount. Another possibility is to set a threshold such that only probabilities larger than this threshold issue a claim.

To determine the best performing model and whether this model is more informative than a benchmark model (e.g. random walk model), one can use evaluation metrics. An example is the absolute error. Let the total claim amount of the test data be €1.000.000. A model that forecasts the total claim amount to be €970.000 seems relatively accurate. However, how does it compare to other (more simple) models? Assume the benchmark model forecasts the total claim amount to be €940.000. In terms of absolute error of the total claim amount, the constructed model outperforms the benchmark model, which implies that information is gained from the modeling approach.

Accurately forecasting insurance claims helps insurance companies to improve their pricing model. Applying the correct methods can help insurance companies to be one step ahead of their competitors, which can result in more clients which in turn can lead to more profit.

1.2 Research Goal

This research combines multiple existing techniques to try to construct more accurate forecasts. For example, the research of Dal Pozzolo (2011) only uses classification techniques to make forecasts, whereas the research of Berridge (1998) tries to fit a distribution to historical data and simulate from it to generate forecasts. Both methods seem to work fairly well, but there might be room for improvement if these techniques are combined.

The goal of this research is to construct a step-by-step guide to forecast insurance claims. It is important to map and solve each step in the forecasting process, as SAS wants to use this research as a basis for their new Dynamic Pricing product. To be able to provide a framework and knowledge to Finaps and SAS, the following questions need to be answered:

- *Which steps are needed to prepare the data to be able to apply model building techniques?*
- *Do dimensionality reduction techniques on the dataset improve the forecasting performance?*
- *Which classification techniques can be used to determine a client's risk profile?*
- *How to transform the predicted client's risk profile into predicted claims?*
- *How to evaluate the constructed models?*

1.3 Structure

Chapter 2 discusses related work and how this research contributes to the already existing literature. In Chapter 3, the data and its summary statistics are presented. In Chapter 4 we explain the modeling process and techniques. This section tackles all five subtasks as given in Section 1.1. Chapter 5 discusses an alternative model which is commonly referred to as the ratio model. This method can be seen as a very naive method, in the sense that it is simplistic. However, this model is commonly used by insurance companies and therefore we use this model as the benchmark. Chapter 6 contains the expirimental results, where the different methods are evaluated. Finally, Chapter 7 presents the conclusions that can be drawn from this research as well as suggestions for future work and possible extensions in other areas of expertise.

Chapter 2

Related Work

Forecasting insurance claims is not a new area of expertise, actuarial sciences exist as long as the insurance business exists. All insurance companies have their internal models to forecast claims and determine insurance premiums. This chapter discusses some of the current approaches used by insurance companies as well as already existing forecasting methods. In Section 2.1 we discuss already existing literature about building an actuarial forecasting model. Section 2.2 presents literature in which individual techniques are used to predict insurance claims. Finally, in Section 2.3 we discuss relevant literature about the evaluation of actuarial models.

2.1 General Model-Building Information

Mata (2010) presents a step by step guide to design insurance rating models. The author works at *Matβlas*, which is an international insurance and actuarial consultancy company. They determine the final premium as a function of the pure premium and all costs included risks and profit loads. While the scope of our research does not include extra costs, a risk premium and a profit margin, the basic principles of constructing an insurance rating model can be used. Mata (2010) also provides a theoretical background on rating models and guidelines for choosing the base exposure. While the calculations and methods used in Mata (2010) are relatively simple, the theoretical background is of great importance.

The publication of Goldburd et al. (2016) is a comprehensive guide to creating an insurance rating plan using generalized linear models. It has an emphasis on application over theory and is written for actuaries practicing in the property and casualty insurance industry. The topic that is thoroughly covered is the model-building process: data preparation, selection of model form, model refinement and model validation. While our research does not use the generalized linear models that the publication of Goldburd et al. (2016) does, their publication adresses the technical aspects and gives a comprehensive guide of building an insurance claim forecasting model. It explains every step of the model-building process in depth, which gives useful insights for our research, since we also use these subtasks.

2.2 Model-Building Techniques

In Batty et al. (2010) it is stated that the use of advanced data mining techniques has taken root in property and casualty insurance. However, application of data mining techniques is still in a nascent stage. The authors describe how data mining and multivariate analytic techniques can be used to improve decision making. It is stated that data preparation provides a solid foundation for model development. They divide the data preparation into four steps: variable generation, exploratory data analysis, variable transformation and partitioning the datset to construct a model. These four steps provide a solid foundation for data preparation. After extracting information from the dataset using the four steps, Batty et al. (2010) use a final tool to extract information out of the data, namely a decision tree. A decision tree can be used to segment the population into different groups. While the authors discuss how to use the information revealed by data preperation, they leave the implementation for future discussion.

The research of Dal Pozzolo (2011) uses a combination of regression and classification techniques to forecast insurance claims. The data presented in the research are provided by the Allstate Claim Prediction Challenge and to construct forecasts, they use a combination of regression, dimensionality reduction and classification techniques. Dal Pozzolo (2011) classifies the observations to be predicted, removes the observations that are predicted as zero and applies regression on the rest. In the competition, the forecasting power is measured by the normalized Gini coefficient. Conclusions are based on the normalized Gini coefficient of the method that is used, a higher coefficient concludes a better forecasting method. The classification methods used in Dal Pozzolo (2011) are a decision tree, random forest, naïve Bayes, K-nearest neighbors, neural network, support vector machine and linear discriminant analysis. Between all the classifiers, they found that the three best performing methods are the decision tree, random forest and linear discriminant analysis. These methods outperform the results based on using a regression on all observations.

In Berridge (1998), the author tackled the problem of forecasting claims in motor vehicle insurance. Specificly focused on the dataset from State Insurance, the author tries to predict claims using five predefined clusters of risk categories. As an intermediate step, the author tries to fit a distribution to the claim amount. The author tries to fit well-known two parameter distributions to the claim data and finds that the log-normal and log-gamma distribution tend to fit relatively well, whereas the Pareto distribution can fit the heavy tail. Having fitted the distributions using the method of moments estimator, the author continues his research explaining the current approach of State Insurance, the rating approach. This approach determines the final premium as the base premium multiplied by individual rating factors. Berridge (1998) uses both methods to make predictions about future insurance claims. He first divides all clients into groups with similar risk characteristics, which are predefined and based on age and gender. Thereafter, the author proceeds to estimate the claim amount based on the fitted distributions of that particular cluster and forecasting the expected claim amount for each client.

2.3 Model Evaluation

In the research of Frees et al. (2014), the Lorenz curve and Gini index are combined and extended to a financial context by ordering insurance risks. They develop a Lorenz curve and Gini index that can cope with adverse selection and is able to measure potential profit. Frees et al. (2014) finds that the Gini index is a useful tool in predictive modeling, where the performance of the model is examined on an independent hold-out sample. They consider an 'ordered' Lorenz curve, which varies from the Lorenz curve in two ways. First, they look at the amount of insurance premium paid instead of wealth. Second, the premiums and claims are ordered by a third variable, called relativity. The reason to opt for such a modified Lorenz curve and Gini index rather than the mean squared error is due to the distribution of premiums and claims. Typically, the distribution of premiums tends to be relatively narrow and skewed to the right. In contrast, the losses have a much greater range and are predominantly zero. Therefore, they state that it is difficult to use the mean squared error to measure discrepancies between premiums and claims.

Chapter 3

Data

Insurance claim data are confidential. Insurers construct their internal models based on their data and try to outperform their competitors. To obtain policy and claim data, one must usually pay a large sum of money. However, by following the masterclass Actuarial Sciences[1] at the Erasmus University Rotterdam, we obtained a dataset that contains car insurance claim data. The summary statistics of this dataset can be found in table 3.1.

This dataset contains 10000 observations and 14 predictor variables, which is a relatively small number of observations for an insurance company. Datasets of over 13 million observations with over 30 predictor variables are available (see Dal Pozzolo (2011)) but to speed up the data preparation process which does not influence the modeling techniques, we use the (prepared) smaller dataset.

In this dataset, claim amounts range between €0 - 50000, except for three outliers. These three outliers have claim values between €4 - 4.5 million, over 1000 times the average claim value. We cap these outliers at €50000 to bring them more in line with the other claims and to get smoother results. For this transformation, we assume that these outliers are random and do not depend on specific predictor variables. Note that in the final premium calculation, the actual claim values need to be accounted for.

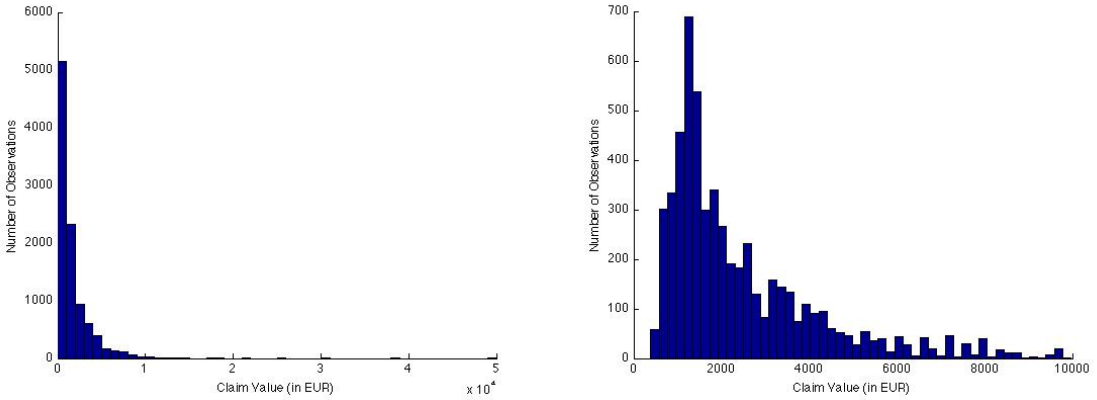
[1]The author has followed the masterclass Actuarial Sciences in 2015 at the Erasmus School of Economics. This masterclass was organized by D. Fok (`personal.eur.nl/dfok/`) in collaboration with Allianz. Note that this dataset is already prepared and ready for use, therefore a simplification of a real dataset.

			Values			
Variable	Abbreviation	Type	Min	Max	Mean	StDev
Age	A	Integers	18	88	37.8	11.6
ListedPrice	LP	Integers	2924	182129	26544.2	15786.7
CarAge	CA	Integers	0	37	11.2	5.9
Gender	G	Categorical				2 Levels
Province	P	Categorical				12 Levels
SocialClass	SC	Categorical				5 Levels
Urbanisation	U	Categorical				8 Levels
Education	E	Categorical				3 Levels
FinancialType	FT	Categorical				6 Levels
HousingType	HT	Categorical				6 Levels
Make	M	Categorical				5 Levels
Color	C	Categorical				6 Levels
Fuel	F	Categorical				4 Levels
Mileage	MA	Categorical				3 Levels
Claims > 0		Continuous	389	4562272	4036.6	72679.1
Capped Claims[1] > 0		Continuous	389	50000	2570.3	2383.3

[1] The capped value of a claim is 50000 ($CappedClaim = min\{Claim,\ 50000\}$).

Table 3.1: Summary statistics of the data used in this research

Figure 3.1A shows that most of the claims are equal to zero. In our dataset, 43.5% of the observations did not issue a claim. Figure 3.1B shows the distribution of claims that are greater than zero. For clarity, we have capped x-axis of the second figure at 10000, since only a relatively small number of claims ($< 1\%$) are outside this range and a more detailed histogram can be obtained.



(A) Capped Claim Values

(B) Claim Values > 0

Figure 3.1: Histogram of the distribution of claims in the dataset

Chapter 4

Modeling Process and Techniques

Finding a framework to forecast insurance claims is a multi-staged problem. The first step is to prepare the data. As a second step, it is possible to filter out irrelevant data to test if a model that uses all predictor variables overfits. That is, the model performs well on the training data, but has poor out-of-sample performance. The next step is to predict the probability to issue a claim by using classification techniques. To obtain forecasted claims, the predicted probability needs to be transformed into a predicted claim amount. The last step in the framework is to evaluate the constructed models.

4.1 Data Preparation

Once the data have been collected, data preperation provides a solid foundation for model development. Batty et al. (2010) provide four steps that summarize data preparation.

As a first step, it is important to create variables from raw data. To create variables, load the data into the system and assign a variable name and a data format to each data field. It is also possible to use the interactions between original variables as a new variable. An example for health insurance can be the new variable 'Body Mass Index', which is an interaction between the variables 'Length' and 'Weight', modeled by (Length^2 / Weight). This process is called feature generation.

Once the data have been loaded into the system, it is important for the analyst to become comfortable with the content of the modeling data. The data analyst should look at the descriptive statistics (min, max, mean, frequency, ...) to get a general grasp on the data. Furthermore, visualization of the data can be useful to spot any outliers or patterns. This step is referred to as the exploratory analysis.

If the exploratory analysis of the data reveals imperfections in the dataset, these issues must be solved. Imperfections can be excessive categorical values, missing values or outliers. Excessive categorical variables can lead to a model that overfits. If similar categories are grouped, the model may have better out-of-sample performance. Instances with missing values can be removed from the data set, but this might not be ideal due to removing useful information. A solution is to replace the missing value with an estimate, which can be the (conditional) mean of that variable. Outliers can skew the overall distribution to accomodate for a very small number of data points. If the tail of the distribution is not of any particular interest, these observations can be deleted.

The last step is to partition the data set into two parts, a 'train' and 'test' set. The test set is set aside until the end of the process. That set will only be used to assess the final out-of-sample results. Within the training set, cross-validation is used to tune the model parameters. Cross-validation evaluation consists of partitioning the data set into k subsets of equal size. Then, $k-1$ subsets are used as the training set and the remaining subset as the test set. This process is repeated k times, where in every round another subset is used as the test set. Model parameters are selected such that they give the best average cross-validated out-of-sample performance.

4.2 Dimensionality Reduction

Once the data have been prepared, it is possible to apply techniques that reduce the dimensionality of the data. Dimensionality reduction can improve the final classification accuracy of out-of-sample data. Removing uninformative data can help the algorithm find more general classification rules and thereby achieve better performance when the model is applied to new data (Silipo et al. (2014)).

Four different techniques to reduce the dimensionality are used. First, an a priori defined subset of variables is deleted. Secondly, a subset of the original variables are selected through forward selection. As a third technique, random forests are used to obtain the most informative variables, which are then used to built the model. The fourth technique is multiple correspondence analysis.

4.2.1 Variable Elimination

To reduce the number of variables, it is possible to use an a priori defined subset of the original variables or to stepwise include only significant predictor variables. The former technique can use expert knowledge as input variables. The latter technique can be compared to stepwise regression, where the choice of predictor variables in the model are carried out by an automatic procedure.

The two main approaches to include predictor variables are forward selection and backward elimination. Forward selection starts with a model without variable terms, but just a constant. The addition of each variable is tested using a chosen model fit criterion, adding the variable whose inclusion gives the most statistically significant improvement of the fit. This process is repeated until none of the variables significantly improve the model. Backward elimination uses the same procedure, but starts with all the variables and deletes insignificant variables (Lockhart (2008)).

4.2.2 Random Forest

A Random forest can be used for either dimensionality reduction or as a classification technique. It was first developed by Leo Breiman (Breiman (2001)). To apply dimensionality reduction using random forests, one generates a large set of decision trees and uses the usage statistics to find the most informative subset of variables (Silipo et al. (2014)). The usage statistic can be interpreted as how many times a variable is used as a split criterion in the decision trees. For more information about decision trees and how they are constructed, see section 4.3.1.

As a first step, construct a large number (N) of shallow trees. Each tree has two levels and randomly selects k out of n variables to perform splits on. To obtain the most informative variables, it is necessary to assign scores for each variable. The score is calculated by counting how many times it has been selected for a split and at which level. Levels one and two have one and two splits respectively. The score ($\dot{s}$) of a variable (v) in tree (t) is defined as:

$$\dot{s}_{v,t} = \wp_{v,t,1} + \frac{1}{2}\wp_{v,t,2}. \tag{4.2.1}$$

In (4.2.1) $\wp_{v,t,1}$ and $\wp_{v,t,2}$ are the number of splits that use variable v in tree t at level 1 and 2 respectively. The score for each variable, $\hat{s}_v$, is the sum of the scores in all trees. A higher score can be interpreted as a more informative variable. $\hat{s}_v$ is defined as:

$$\hat{s}_v = \sum_{t=1}^{N} \dot{s}_{v,t}. \tag{4.2.2}$$

At this point, one more step is needed to find out which variables score higher than you would expect by random chance. If it is assumed that each variable influences the target variable by the same amount, one would expect each variable to have the same score and to be in the same number of trees. The score is the expected number of times a variable is used in a split in the complete forest. Given the scoring scheme as in equation (4.2.1), the expected score is equal to:

$$\tilde{s}_v = \left(\frac{2}{k} \cdot N \cdot \frac{k}{n}\right) = \frac{2N}{n}. \tag{4.2.3}$$

If $\hat{s}_v > \tilde{s}_v$, the variable is included, otherwise the variable is eliminated.

4.2.3 Multiple Correspondence Analysis

Multiple Correspondence Analysis (MCA) allows one to analyze the pattern of relationships of several categorical dependent variables and is used to detect and represent underlying structures in a data set. MCA is the categorical counterpart of Principal Component Analysis (PCA), which tries to capture the maximum possible variance in a minimum number of linear independent variables (factors). MCA constructs a new dataset, where each observation has a factor score for each factor (Abdi and Valentin (2007)). The first few factors should capture most of the variance of the original dataset.

To get more into technical detail, first define an indicator matrix. To obtain an indicator matrix, code each level (category) of a categorical variable as a binary variable. For example *Gender* (Male or Female) is one categorical variable with two levels. The pattern for male will be [1 0] and for a female will be [0 1]. Table 4.1 illustrates how responses are coded as binary variables[1].

Obs.	**Gender**		**County**			**Mileage**			
	Male	Female	ZH	F	L	<10k	10k - 20k	20k - 35k	> 35k
1	1	0	0	1	0	0	0	1	0
2	1	0	1	0	0	1	0	0	0
3	0	1	0	0	1	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 4.1: Transformation categories to binary values for MCA

The matrix of 1's and 0's obtained by the coding scheme as illustrated in table 4.1 is further referred to as the super-indicator matrix (Greenacre (1984)). The super-indicator matrix has I observations and Q categories. Define the super-indicator matrix $\mathbf{Z} \equiv [\mathbf{Z_1}, \ldots, \mathbf{Z_Q}]$, where $\mathbf{Z_q}$ is an $I \times J_q$ indicator matrix corresponding to the qth categorical variable which has J_q levels.

To obtain the new factor scores, more notation is needed. First, define the Burt matrix $\mathbf{B} \equiv \mathbf{Z}^T\mathbf{Z}$ and transform this matrix into a probability matrix denoted $\mathbf{X} \equiv N^{-1}\mathbf{B}$. Here, N is the grand total of the table, that is, the sum over all rows and columns. Denote the vector $\mathbf{r} \equiv \mathbf{X}\iota$, the vector of row totals and $\mathbf{c} \equiv \mathbf{X}^T\iota$ the vector of column totals. Note that $\mathbf{r} = \mathbf{c}$ since $\mathbf{B} = \mathbf{B}^T$. Let $\mathbf{D}_c \equiv diag\{\mathbf{c}\}$ and $\mathbf{D}_r \equiv diag\{\mathbf{r}\}$, which are also equivalent. The factor scores are obtained from the following singular value decomposition (Abdi and Valentin (2007)):

[1]For example: Observation 1 is a Male that lives in F and drives between 20.000 and 35.000 km per year.

$$\mathbf{D_r}^{-\frac{1}{2}}(\mathbf{X} - \mathbf{rc^T})\mathbf{D_c}^{-\frac{1}{2}} = \mathbf{P\Delta P^T}. \tag{4.2.4}$$

In this decomposition, the matrix $\mathbf{\Delta}$ is the diagonal matrix of the singular values. Obtain the matrix of eigenvalues as $\mathbf{\Lambda} = \mathbf{\Delta}^2$.

The next step is to find a number of $M \leq Q$ factors, which can capture a large portion of the variance. In traditional PCA, the percentage of variance explained by factor j is $\frac{\lambda_j}{\sum_{j=1}^{J} \lambda_j}$. However, our technique created artificial additional dimensions, since one categorical variable is coded as several columns. As a consequence, the variance of the solution space is artificially inflated. Therefore, the percentage of variance explained by the first eigenvalue is severely underestimated and the traditional calculation no longer holds.

To account for this inflation of the variance and underestimation of the first dimension, the correction formula of Greenacre can be used (Greenacre (1996)). This correction takes into account that eigenvalues smaller than $\frac{1}{Q}$ are simply coding for the extra dimensions implied by MCA. The eigenvalues need to be corrected accordingly and the corrected eigenvalues, λ_j^c, are obtained as:

$$\lambda_j^c = \begin{cases} \left[\left(\frac{Q}{Q-1}\right)\left(\lambda_j - \frac{1}{Q}\right)\right]^2 & \text{if } \lambda_j > \frac{1}{Q} \\ 0 & \text{if } \lambda_j \leq \frac{1}{Q} \end{cases} \tag{4.2.5}$$

Given the corrected eigenvalues, use the traditional calculation $\left(\frac{\lambda_j^c}{\sum_{j=1}^{J} \lambda_j^c}\right)$ for the percentage of variance explained by each factor. Once the new percentages of explained variance are constructed, find a number of $M \leq Q$ factors which can capture a large portion of the variance. There is no 'rule of thumb' or criteria for keeping or rejecting dimensions for analysis based on the proportion of variance explained (Doey and Kurta (2011)). However, in traditional PCA, one can use an elbow plot to determine the number of factors, which can also be used for MCA.

4.3 Classification

Once the most important information is extracted from the dataset, either by stepwise eliminating variables or through multiple correspondence analysis, the next step in the modeling process is to classify each individual observation. Four classification techniques are used: a decision tree, a random forest, a binary logistic regression and a support vector machine algorithm. The objective of these techniques is to assign a probability of issuing a claim to each observation which needs to be predicted, since the dependent variable can be binary classified. Either there was a claim (1) or there was no claim (0).

4.3.1 Decision Tree

Decision trees provide a structure that divides a large data set into small subsets by applying decision rules (Quinlan (1986)). It is a flowchart like structure which tests for different attributes. It can be seen as a questionnaire which contains conditional questions. Figure 4.1 shows a decision tree with 2 levels graphically.

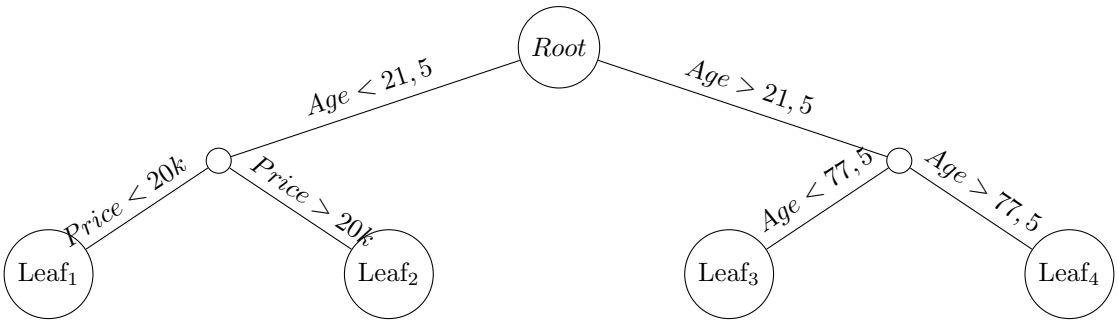


Figure 4.1: An example of a decision tree with two levels

To predict a new observation for a given tree, the classification starts at the 'Root' and goes down the tree testing for different attributes until an endpoint, a 'Leaf', is reached. Assume a decision tree equal to figure 4.1, the first question to predict a new observation is "Is the policyholder older than 21,5?". If the answer is "Yes", continue with "Is the policyholder older than 77,5?". If the answer is "No", the observation ends at Leaf_3.

To obtain a predicted probability to issue a claim, use the constructed decision tree. First, run the entire training set through the constructed tree and count the percentage of observations that issued a claim at each leaf. The percentage of observations in the training set that issued a claim which ended up at Leaf_i is p_i. To obtain the predicted probability of a new observation to issue a claim, run the new observation through the tree and if the new observations ends at Leaf_i, assign p_i. Define p_i as:

$$p_i = \frac{1}{O_i} \cdot \sum_{o \in \text{Leaf}_i} \mathbb{1}\{Y_o = 1\}, \tag{4.3.6}$$

where O_i is the number of observations that end up at Leaf_i in the training set and $\mathbb{1}\{Y_o = 1\}$ is the indicator function that returns 1 when observation o issued a claim and 0 otherwise.

One of the advantages of this approach is that making predictions is fast. No complicated calculations are involved, it is simply giving answers to the predefined questions in the tree. Therefore, it is easy to understand which variables are important in making the prediction. Another advantage is that predicting is still possible even when data for an observation is missing. To predict with missing variables, go through the tree until a question cannot be answered. Then, take the average probability of all the leaves that are still possible to reach (Shalizi (2009)).

When constructing a decision tree, common issues are which variable to perform the next split on and when to stop the splitting of variables. To obtain the variable to split, ingenious algorithms are constructed. These algorithms boil down to the following question: Which next split gives the best classification of the training set? That is, the next split is the split that minimizes the misclassification rate. The misclassification rate of the tree is the rate of which the tree predicts an observation as 1 when the actual value is 0 or vice versa. The constructed tree usually results in a large tree that provides a good fit to the training data. However, this large tree may have the tendency to overfit. A solution is to find a (smaller) tree that finds an optimal balance between the misclassification rate and the complexity of the tree. This balance can be found using the C4.5 algorithm developed in Quinlan (2014) or by using cost-complexity pruning (Bradford et al. (1998)).

For cost-complexity pruning, the optimal cost-complexity parameter CC needs to be found. This parameter controls the tradeoff between the complexity of the tree and the accuracy. The optimal value for CC is found by minimizing the cross-validatied misclassification rate. A higher value for CC results in a smaller tree.

4.3.2 Random Forest

A random forest is an ensemble of decision trees, which can correct for the overfitting of a single decision tree. Random forests average multiple decision trees, trained on different parts of the same training set. This can decrease the out-of-sample misclassification rate, at the expense of loss of interpretability. Generally, random forests outperform decision trees.

To train a random forest, repeatedly select a random sample (with replacement) of the training set and fit a tree to this sample. Random forests have one additional constraint, at each candidate split, only a random subset of variables can be split. The reason is to decrease the correlation of the trees. If one or a few variables are very strong predictors, these features will be selected in many trees, causing them to become strongly correlated (Breiman (2001)). The algorithm to construct the forest, given observations of predictors $\mathbf{X} = [\mathbf{x_1}, \dots \mathbf{x_n}]$ and responses $\mathbf{Y} = [y_1, \dots y_n]$, $y_i \in \{0, 1\}$ is:

$$\begin{aligned} &For\ b = 1, \dots, B \\ &\quad \text{Sample subset } X_b, Y_b \text{ from } X, Y; \\ &\quad \text{Train tree } t_b \text{ on } X_b, Y_b; \\ &end \end{aligned} \tag{4.3.7}$$

To obtain predicted probabilities in a random forest, either take the percentage of predicted 1's for an observation or take the average of the predicted probabilities for all individual trees. In SAS, the predicted probability obtained by the forest is obtained by averaging the predicted probabilities from all the individual trees. This research uses the SAS approach, that is:

$$\hat{f}(\mathbf{x_{n+1}}) = \frac{1}{B} \sum_{b=1}^{B} p_{t_b}(\mathbf{x_{n+1}}), \tag{4.3.8}$$

where $p_{t_b}(\mathbf{x_{n+1}})$ is the probability of issuing a claim of observation $n + 1$ with predictor variables $\mathbf{x_{n+1}}$ in tree b. This probability is obtained in the same way as in equation (4.3.6)

Recall the additional random forest constraint: "At each candidate split, only a random subset of variables can be split". Let the size of this subset be the variable *'VarsToTry'*, which can be optimized to minimize the cross validated misclassification rate. The optimal value for *VarsToTry* is found by comparing all possible values for *VarsToTry* and pick the value that minimizes the cross-validated misclassification rate.

4.3.3 Binary Logistic Regression

The binary logistic regression models the probability of issuing a claim using predictor variables. The objective is to compute the probability of success (Tranmer and Elliot (2008)), which is equivalent to the probability of issuing a claim. Each predictor variable influences this probability in a way which has yet to be determined. To determine the influence of the predictor variables, a logistic regression can be used.

Let π_i be the probability of observation i to issue a claim. It is convenient to have π_i be dependent on a vector of observed variables $\mathbf{x}_i$. The simplest idea is to let π_i be a linear function of $\mathbf{x}_i$, that is:

$$\pi_i = \mathbf{x}_i' \boldsymbol{\beta}. \tag{4.3.9}$$

One problem with this model is that π_i is a probability, which has to be between the values 0 and 1, while $\mathbf{x}_i'\boldsymbol{\beta}$, can take any real value. There is no guarantee that π_i will be in the correct range, unless complex restrictions are imposed on the coefficients.

A simple two-step solution is given in Rodriguez (2007). Transform the probability to remove the range restrictions and model the transformation as a linear function of the predictor variables. First, move from the probability π_i to the odds:

$$\text{odds}_i = \frac{\pi_i}{1-\pi_i}. \tag{4.3.10}$$

Second, take logarithms, calculating the logit, or log-odds:

$$logit(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right). \tag{4.3.11}$$

The logit is used to impose the restriction that π_i is between 0 and 1. In the logistic regression model it is assumed that the logit of the underlying probability ($logit(\pi_i)$) instead of the underlying probability (π_i) is a linear function of the predictors:

$$logit(\pi_i) = \mathbf{x}_i'\boldsymbol{\beta}. \tag{4.3.12}$$

Solving for π_i in equation (4.3.12), gives the following solution for π_i:

$$\pi_i = \frac{exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1+exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}. \tag{4.3.13}$$

By assuming that $logit(\pi_i)$ is a linear function of predictors, the probability of issuing a claim will be between 0 and 1. Irrespective of $\mathbf{x}_i'\boldsymbol{\beta}$, by definition $e^{\mathbf{x}_i'\boldsymbol{\beta}}$ is always greater than 0. Furthermore, it is known that $\frac{x}{1+x} < 1$. Both expressions combined indicate that $0 < \pi_i < 1$.

Equation 4.3.13 solves for π_i in terms of predictor variables. However, the structure of $\mathbf{x}_i'\boldsymbol{\beta}$ is still unknown. Assume two predictor variables, Mileage and Gender. Given this simple model, it is possible to compute five basic models of interest, ranging from the 'null model' to the 'saturated model'. Table 4.2 contains all five models, as well as the name of the model, a descriptive notation and the formula for the linear predictor.

Model	Notation	$\mathbf{x}_i'\boldsymbol{\beta}$
Null	ϕ	η
Mileage	M	$\eta + \alpha_i$
Gender	G	$\eta + \beta_j$
Additive	M + G	$\eta + \alpha_i + \beta_j$
Saturated	MG	$\eta + \alpha_i + \beta_j + (\alpha\beta)_{ij}$

Table 4.2: Structure of $\mathbf{x}_i'\boldsymbol{\beta}$ by *Mileage* and *Gender*

The saturated model uses all possible variable interactions as possible predictors. That is, if n variables are present, the saturated model uses all $1-, 2-, \ldots, n-$way interactions between them. Empirical evidence in Rodriguez (2007) shows that the saturated model fits the data the best for a small number of variables. However, when the number of variables exceeds 2, the number of category interactions increases rapidly. Then, parameter estimates and uncertainty must be taken into account. Another problem might occur: the data with many variables overfits. To find the best selection of variables is beyond the scope of this research and relevant research can be found in Tibshirani (1996). We use the saturated structure for $\mathbf{x}_i'\boldsymbol{\beta}$, which uses the interaction between all variables.

4.3.4 Support Vector Machine

A support vector machine (SVM) can be used to classify observations in binary problems. A SVM represents the training observations as points in a space. The two categories are divided by an optimal separation line (or hyperplane in multi-dimensional space). This optimal line is found by finding two parallel lines that seperate the categories, such that the distance between them (the margin) is as large as possible. The optimal separation line is the line that lies halfway between them. New observations are mapped into the same space and are predicted to belong to a category based on which side of the line they fall (Cortes and Vapnik (1995)).

Although the reader does not need to understand the underlying theory of SVM, we briefly introduce some SVM basics necessary for explaining the procedure. Given a training set of pairs of observations and predictors $(\mathbf{x_i}, y_i)$, $i = 1, \ldots, l$, where $\mathbf{x_i} \in \mathbb{R}^n$ and $\mathbf{y} \in \{-1, 1\}$, the SVM requires the following optimization problem (Boser et al. (1992), Cortes and Vapnik (1995)):

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\mathbf{w}'\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(\mathbf{w}'\phi(\mathbf{x_i}) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{4.3.14}$$

In this equation, the constants ξ_i are set such that the constraint in (4.3.14) holds for each training observation. The constant C is the penalty parameter which can freely be set by the researcher. Given the optimization problem, a higher C leads to an on average lower ξ_i.

A feature of a SVM is, is that it can be used when the original data are not linearly separable. Using a function $\phi(\mathbf{x_i})$ to map the training vectors $\mathbf{x_i}$ into a higher dimensional space, the SVM finds a linear separating line with the maximum margin in this higher dimensional space. However, finding the correct $\phi(\mathbf{x_i})$ for datasets with large dimensionality will quickly become intractable

(Kim (2013)). Luckily, Jordan and Thibaux (2004) show that during training, the optimization problem only uses the training examples to compute pair-wise dot products $\langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$. This is significant since there exist functions that implicitly compute the dot product of two vectors in a higher dimensional space, without explicitly transforming the vectors to this higher space (Kim (2013)). Such functions are called kernel functions, denoted by $K(\phi(\mathbf{x_i}), \phi(\mathbf{x_j}))$.

A kernel K effectively computes dot products in a higher dimensional space, while remaining in the original feature space, which makes it computationally easier than finding $\phi(\mathbf{x_i})$. Two types of kernels will be used, the linear kernel: $K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i}'\mathbf{x_j}$ and the radial basis function (RBF) kernel: $K(\mathbf{x_i}, \mathbf{x_j}) = exp(-\gamma||\mathbf{x_i} - \mathbf{x_j}||^2), \gamma > 0$. The linear kernel is a fast algorithm and works well when the number of variables is large. The RBF kernel generally has better performance when the model parameters are optimized to minimize the misclassification rate, but the algorithm is computationally harder (Hsu et al. (2003)).

Solving this optimization problem with a linear kernel has one parameter: C and with an RBF kernel it has two: C and γ, where C is the penalty parameter for misclassified training observations. It is not known which C and pair of (C, γ) are the best for a problem, thus we must search for the optimal parameters. Optimal parameters are found using a 'grid-search'. Basically, compare different values of the model parameters and the one with the best cross-validated accuracy is picked. Finding optimal parameters C and γ is important for out-of-sample forecasting. Using figure 4.2 as reference, we see that an outlier can decrease the margin by a large amount if one wants to fit the training data perfectly. If this outlier in the training data is allowed to misclassified, but given a penalty, it is possible to find a larger margin and thereby find a more general classification rule. This generally improves the out-of-sample performance.

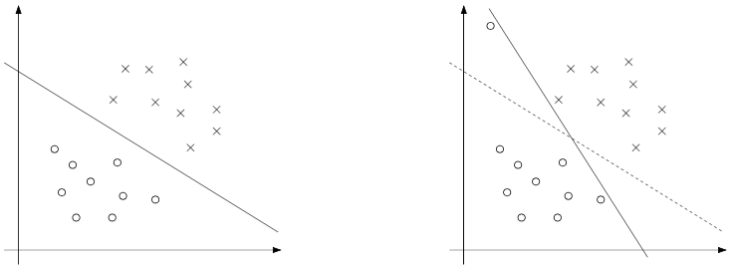


Figure 4.2: Change in SVM margin if an outlier is included

To obtain the predicted probability of an observation, the observations the furthest away from the separation line are given probabilities 0 and 1 respectively. The closer an observation to the separation line, the closer the probability will be to 0.5. The distance between an observation and the separation line depends on the difference in value of $\mathbf{w}' \cdot \phi(\mathbf{x_i})$ and b.

4.4 Forecasting

By applying dimensionality reduction and classification techniques, it is possible to assign a probability to issue a claim for every instance. However, a probability to issue a claim is not yet a forecasted claim amount. The objective is to apply a transformation on the predicted probability to obtain a predicted claim amount. Two questions need to be answered, the first being 'Does someone issue a claim?' and secondly 'If someone claims, how much do they claim?'.

4.4.1 From Probability to Binary Value

In practice, someone either issues a claim or does not issue a claim. To obtain accurate forecasts, it can be helpful to forecast claims as either 0 or as a value, which implies predicting 1's (does claim) or 0's (does not claim) first. The probability of a claim needs to be transformed into a binary value. A logical way to impose this transformation is to find a threshold τ for which the following holds:

$$\varphi_i = \begin{cases} 1 & \text{if } P(Y_i = 1) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{4.4.15}$$

Here, φ_i is the predicted value of issuing a claim for observation i, either 1 (issues a claim) or 0 (issues no claim) and $P(Y_i = 1)$ is the predicted probability of issuing a claim for observation i.

It is yet unknown which value of τ gives the best classification accuracy. For example, if $\tau = 0$, all observations will be predicted as 1, but this does not lead to accurate results. We opt for three different values of τ. First, let $\tau = 0.5$. This value performs well when the dataset is balanced but struggles when the number of 1's and 0's in the dataset are unbalanced. To account for this problem, it is possible to scale τ such that it corresponds with the percentage of 1's in the dataset. Since the objective is to maximize the classification accuracy, a more sophisticated possibility is to determine τ by maximizing the cross-validated *Accuracy* (see equation (4.5.24)). Let the three possibilities be defined by the following equations respectively:

$$\tau_h = 0.5, \tag{4.4.16}$$

$$\tau_x = \frac{\sum_i \mathbb{1}\{Y_{O_i} = 1\}}{\sum_i O_i}, \tag{4.4.17}$$

$$\tau_a = max_\tau\{Accuracy\}. \tag{4.4.18}$$

4.4.2 From Probability to Claim Amount

To obtain expected claim values, the probability needs to be transformed into a claim amount. We propose two solutions for this transformation. The first is to multiply the predicted probability of issuing a claim by the average claim amount of the training data:

$$_1\Xi_i = P\big(Y_i = 1\big) \cdot \overline{CappedClaims > 0}. \tag{4.4.19}$$

An advantage of this solution is that low predicted probabilities are assigned a low predicted claim amount and high predicted probabilites are assigned a high predicted claim amount. We have chosen to multiply by the average claim amount for the following two reasons. First, it prevents extreme expected claim amounts. If we let the expected claim amount be a proxy for the premium, we want to prevent extreme values, since people with extreme values will never take an insurance. Secondly, while it is possible to simulate a draw from the historical distribution, one can either be (un)lucky in their predicted claim amount due to the nature of simulating a draw from the historical distribution. A disadvantage of this approach is that it is impossible to obtain claim amounts that are greater than the average claim amount in the training data.

Another possibility is to first classify the observations and let each observation that is predicted to issue a claim have a predicted claim amount equal to the average claim amount of the training set. That is:

$$_2\Xi_i = \varphi_i \cdot \overline{CappedClaims > 0}. \tag{4.4.20}$$

If the model can classify accurately, this solution should be able to precisely predict the total claim amount. A disadvantage is that if the number of predicted false positives and false negatives are equal, even though there are many, this model also has a good performance. This is due to the fact that the false positives and false negatives cancel each other out in the total premium calculation.

4.5 Model Evaluation

The different techniques explained in the previous sections provide different probabilities of issuing a claim. However, it is yet unknown which combination of techniques is the most accurate. Earlier research (Myung (2000), Babyak (2004)) shows that complex models tend to overfit. That is, they return a minimal error when applied to the training data, but relatively poor results when applied to new test data. On the other hand, simple models tend to underfit. They cannot capture the dependency between the input variables and the output variable, the forecast. Hence, the model needs to be able to capture the important information, but should not overfit on the training data. The following sections explain different evaluation techniques that can be used to find the best model to the insurers liking.

4.5.1 Classification Measures

To evaluate the classification performance of the different techniques, a confusion matrix can be used. A confusion matrix is a 2×2 matrix with the two dimensions 'actual' and 'predicted'. The observations to be predicted have an actual and predicted value (true or false) and the performance of a model is determined by the number of correctly predicted observations (Visa et al. (2011)). Table 4.3 shows how a confusion table is defined.

		Predicted	
		True	False
Actual	True	TP	FN
	False	FP	TN

Table 4.3: Confusion table

In Table 4.3 TP is a 'True Positive'. That is, the actual value is 'True' and the predicted value is also 'True'. FP is a 'False Positive' (actual = false, predicted = true) and is regarded as a Type II Error. FN is a 'False Negative' (actual = true, predicted = false) and is regarded as a Type I Error and TN is a 'True Negative' (actual = false, predicted = false).

A model perfectly predicts if $(TP+TN) = n$, where n is the number of observations of which the classification needs to be predicted. However, in modeling human behaviour, almost no model can predict perfectly. Therefore, the following performance measures based on the confustion table can be used for evaluation. Equation (4.5.21), (4.5.22) and (4.5.23) are the True Positive Rate (TPR), True Negative Rate (TNR) and Positive Predictive Value (PPV) respectively. A higher value for each measure indicates a better performance. The performance measures can be defined as follows:

$$TPR = \frac{TP}{TP + FN}, \tag{4.5.21}$$

$$TNR = \frac{TN}{TN + FP}, \tag{4.5.22}$$

$$PPV = \frac{TP}{TP + FP}. \tag{4.5.23}$$

The objective is to minimize FP and FN simultaneously because on one hand, too many positives increases the premium, which can lead to customers switching to other companies (which classify better). On the other hand, too many negatives leads to a premium that is too low, which can lead to bankruptcy in the long run. To minimize the false positives and false negatives simultaneously, one should maximize the classification accuracy, which is defined as:

$$Accuracy = \frac{TP + TN}{n}. \tag{4.5.24}$$

4.5.2 Error Measures

While a confusion table can be used to determine the classification power of a model, it cannot determine the prediction error of the total claim amount. For an insurance company, it is important to know the total expected claim amount for the upcoming year, to set the premiums accordingly. To evaluate whether the upcoming total claim amount is forecasted correctly, use the difference between the predicted and actual total claim amount.

It can be useful to split the evaluation of a model into two parts. First, evaluate the classification and as a second step, evaluate whether the total claim value is forecasted correctly. If the model performs poorly in the first step, it can be regarded as a poor model. If the model can classify with great accuracy, but the predicted total claim amount does not match the actual total claim amount, the transformation of predicted probability to predicted claim amount is incorrect.

Assume a model that does not classify the observations correctly, but the predicted total claim amount is close to the actual total claim amount. Such a model might perform well on error terms, but is up for discussion whether this is desirable. For example, if FP = FN and the average claim amount in the training and test set are equal, the absolute error will be 0 if equation (4.4.20) is used. However, this might not be an optimal model for an insurance company.

4.5.3 Normalized Gini Coefficient

The Allstate Claim Prediction Challenge[2] uses the normalized Gini coefficient (NGC) to measure the results in the competition. In this metric, observations are sorted from 'largest prediction' to 'smallest prediction' and only the order determined by the predictions matters (Dal Pozzolo (2011)).

With the normalized Gini coefficient, it is important to correctly predict the relative size of the claim, rather than give a precise estimate. To get a high score, one needs to give high observed claims, high forecasted claims and policies that have a low claim amount or a claim amount equal to zero need to have a low predicted value. For example, assume three observed claims [4286, 1287, 0] and two predictive models that predict these claims as [3471, 5642, 928] and [12532, 3753, 760] respectively. With respect to the NGC, the second model outperforms the first due to predicting the order of claims correctly, despite having a larger absolute error.

[2]For more information about this challenge, see `https://www.kaggle.com/c/ClaimPredictionChallenge`.

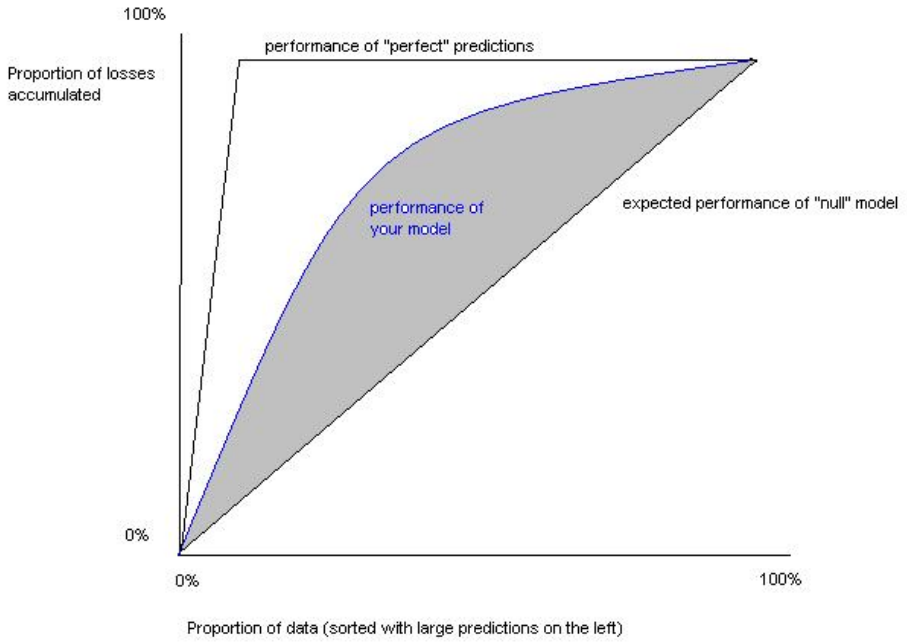


Figure 4.3: Normalized Gini coefficient: example figure

Figure 4.3 plots three lines, the performance of the null model (the benchmark), the performance of the constructed model and the performance of the perfect model. The x-axis represents the proportion of the data used, where the observations are sorted from largest claim to smallest claim. The y-axis represents the proportion of the cumulative losses for the leftmost x% of the data.

In Dal Pozzolo (2011) and on the website of the Allstate Claim Prediction Challenge, they use no model for a benchmark. They state that it is expected to accumulate 10% of the loss in 10% of the predictions when no model is used, so their benchmark model is a straight line. This benchmark line is needed to compute the normalized Gini coefficient.

The Gini coefficient corresponds to the area between the curve of the predictive model and the benchmark model, which is the shaded area in figure 4.3. The perfect model predicts the claims as the actual observed claims, obtaining the maximum Gini coefficient. The normalized Gini coefficient is obtained by deviding the Gini coefficient of the predictive model by the Gini coefficient of the perfect model.

Chapter 5

Baseline Method: Ratio Model

A whole spectrum of econometric methods exist that should be able to predict future insurance claims. However, it is found that insurance companies often use a more simplistic approach. Mata (2010) explains that many insurance companies use a so called 'Ratio Model'. The final premium is calculated by multiplying a base premium with the cluster risk ratios. Section 5.1 explains the basics of this model and section 5.2 explains how to construct clusters of categories that can produce credible results (for example, a cluster with only 5 observations is not credible).

5.1 Introduction

The ratio model is an approach that measures the risk of a category relative to the risk of the complete dataset. This model is easily explained by using a simple (car insurance) example. Let the objective be to calculate the expected claim of a record in the database. Assume a simple database with four variables with their categories and the relative risk ratio of a category respectively as given in Table 5.1.

Gender		Age		County		Mileage	
Cat.	Ratio	Cat.	Ratio	Cat.	Ratio	Cat.	Ratio
Male	1.01	< 30	1.21	Zuid-Holland	0.98	< 10000	0.89
Female	0.99	30 - 50	0.94	Friesland	1.04	10000 - 20000	0.95
		50 - 70	0.92	Limburg	0.97	20000 - 35000	1.02
		> 70	1.07			> 35000	1.12

Table 5.1: Example of categories and their ratios for the ratio model

Calculating the expected claim amount for an observation is easy, but to mathematically define the model some notation is needed. Let $\overline{C}$ be the average of all the claims in the training set (including claims equal to 0), referred to as the 'base' or 'base risk'. Furthermore, denote the variables as v ($v = 1, \ldots, V$) and categories as c. Let $\overline{C_{v,c}}$ be the average claim of the observations in the training set that are of category c for variable v and compute the ratio, $\Upsilon_{v,c}$, as:

$$\Upsilon_{v,c} = \frac{\overline{C_{v,c}}}{\overline{C}}. \tag{5.1.1}$$

Once the base risk and the ratios are known, it is possible to calculate the expected claim amount for an observation. Assume the dataset contains V unique variables. The expected claim amount for observation i with response categories $\mathbf{c}_i$ is[1]:

$$\Psi_i = \overline{C} \cdot \prod_{v=1}^{V} \Upsilon_{v,\mathbf{c}_i}. \tag{5.1.2}$$

An extension to the ratio model is to include additional optional coverages, denoted by Θ. Common practice for policy holders is to limit the amount of coverage provided for certain types of losses. Those with greater exposure to specific types of losses are encouraged to buy additional coverage. The risk premium of these optinal coverages is additive rather than multiplicative, which implies the following extension of the model:

$$\Psi_i = \overline{C} \cdot \prod_{v=1}^{V} \Upsilon_{v,\mathbf{c}_i} + \sum_{o=1}^{O} \Theta_{o,\mathbf{c}_i}. \tag{5.1.3}$$

5.2 Merging Categories

The example in section 5.1 is very straightforward. However, databases used at insurance companies contain more variables and categories. A problem might occur when there are too many possible combinations of categories relative to the number of observations, or when a category does not have enough observations to make credible statistical statements. In table 5.1, the categories of the variable *Age* are already partitioned into four clusters. However, in practice it is more common to know the exact age of the driver, which can lead to a lot of *Age* categories.

Two problems occur when there are too many categories for a variable, explained by an example with the variable *Age*. First, it is likely that not a lot of policies have a policy holder aged 18, making it hard to produce credible statements about the ratio of 18 year olds. Second, the ratio of people aged 22 and 23 will most likely be statistically the same, is it best practice to rate them differently, or is it more common to combine them into one cluster? The following two steps can be used to overcome these problems:

[1] For example, the risk premium of a 25 year old male who lives in Friesland and drives between 10.000 and 20.000 km per year is equal to: $\overline{C} \cdot 1,01 \cdot 1,21 \cdot 1,04 \cdot 0,95$.

- Merge categories that have too few observations

  If a category has only a few observations, it is difficult to make statistically credible statements. A solution is to merge categories that have too few observations. Categories are merged until the number of observations in a cluster exceeds an arbitrary sufficiently large number N. Merging is done from the top down and from the bottom up simultaneously.

Age	Observations	
18	12	} Merge
19	20	} Merge
20	78	
21	175	
⋮	⋮	
75	142	
76	89	
77	43	} Merge
78	5	} Merge

Table 5.2: Merging categories with few observations

- Merge clusters that have statistically the same ratio

  Even when the categories are combined into clusters with observations that exceed N, the number of clusters can still be large. A solution is to combine clusters that have similar risk ratios. An advantage is to produce even more credible statements. Furthermore, if clusters are not merged, the model might calculate the ratios on idiosyncratic features of the training data, which might lead to poor out-of-sample performance. However, a disadvantage of merging clusters is that there is a slight loss of information.

Age(s)	Observations	Ratio	
18, 19, 20	110	2.18	
21	175	1.16	} Merge
22	225	1.14	} Merge
23	255	1.15	
⋮	⋮	⋮	

Table 5.3: Merging clusters with similar ratios

To determine whether two categories are statistically different, a t-test is used. Clusters are significantly different if the calculated test statistic exceeds the threshold for statistical significance (1.64 for a 95% confidence interval). The test statistic can be calculated as:

$$T = \frac{\bar{X}_i - \bar{X}_j}{\sqrt{\frac{S_i^2}{N_i} + \frac{S_j^2}{N_j}}} \sim t(p, N_i + N_j - 2). \tag{5.2.4}$$

For numerical clusters, only consecutive elements will be merged. That is, even though people with ages 22 and 52 might have the same ratio, they will not be merged if any other cluster in between has a significantly different ratio. For categorical variables, we might merge non-consecutive unique elements, since the euclidian distance between two categorical variables is unknown.

Chapter 6

Experimental Results

This chapter presents the experimental results based on the dataset presented in chapter 3. Section 6.1 starts with the model preparation. We explain which variables are selected through the different dimensionality reduction techniques, how the parameters can be tuned and which threshold to use to obtain optimal classification accuracy based on the training data. In section 6.2 we discuss the out-of-sample results based on the classification accuracy, the normalized Gini coefficient and the total error.

6.1 Model Preparation

Given the structure of the framework, three model preparation steps are needed. First, the (reduced) datasets need to be constructed by using the dimensionality reduction techniques. To obtain an optimal classification accuracy, tune the parameters for the different classification techniques. As a final step, it is possible to find the optimal classification threshold, which can further improve the classification accuracy. To assess the cross-validation results, 8000 of the 10000 observations and 10-fold validation are used, which are further referred to as the training data.

6.1.1 Variable Selection

The first 'dimensionality reduction' technique is to apply no reduction. All predictor variables in the original dataset are used in the classification techniques. Another simple approach is to use expert knowledge to determine the predictor variables (Section 4.2.1). Our expert used his prior knowledge and common sense to predict that the following 5 predictor variables are of significant influence on the probability of issuing a claim: *Age, Province, Make, ListedPrice* and *Mileage*.

Another possibility is to stepwise include significant predictor variables through forward selection (Section 4.2.1). This procedure starts with a model without variables but just a constant and stepwise includes variables until the model selection criterion no longer improves. In SAS, the model selection criterion is the Schwarz Bayesian Criterion (SBC) and the summary of the selection procedure can be found in table 6.1. The optimal SBC is found after the variable 'Gender' is included in the model. This selection procedure includes the following 6 variables: *Urbanisation, Province, Mileage, Color, Make* and *Gender*.

Step	Effect Entered	SBC
0	Intercept	-14.022,07
1	Urbanisation	-17.845,83
2	Province	-21.337,57
3	Mileage	-22.508,80
4	Color	-23.183,79
5	Make	-23.606,79
6	Gender	-23.639,94*
7	Age	-23.636,42
8	CarAge	-23.628,53

* Optimal Value of SBC Criterion

Table 6.1: Forward selectionpProdecure details

As a fourth possibility, a random forest can be used to obtain the most informative variables (Section 4.2.2). Variables that are present in more than the expected number of splits are informative and included in the model. The forest to determine the most informative variables used 2000 trees with a depth of 2 and randomly selected 3 of the 14 variables in the dataset to perform a split on. This leads to an expected number of splits of 429 (Equation (4.2.3)). Variables that are used in more than 429 splits are included in the model. Table 6.2 presents the number of splits in the forest of each variable. This selection procedure includes the following 6 variables: *Urbanisation, Province, Mileage, Age, Color* and *Make*.

Variable	# Splits	Variable	# Splits
Urbanisation	1139*	Gender	217
Province	1090*	FinancialType	173
Mileage	906*	Fuel	131
Age	699*	CarAge	101
Color	593*	HousingType	70
Make	531*	SocialClass	32
Listed Price	291	Education	27

* Predictor variable that outperforms the expected number of splits.

Table 6.2: Random forest variable scores

As a final technique, multiple correspondence analysis is used (Section 4.2.3). This technique does not select a subset of the original variables, but imposes a transformation such that the first few dimensions in the new dataset try to capture the most of the variation as possible. Once the new dataset is constructed, the next step is to determine the number of dimensions to include.

As a first solution, we decided to use the first two dimensions as the new dataset. Figure 6.1 shows that after two dimensions, there is a large drop in the proportion of variance explained by the following dimensions. This can be compared to to the elbow plot which is commonly used to determine the number of dimensions for principal component analysis. As a second possibility, 10 dimensions are used. Multiple correspondence analysis can only use categorical variables as input variables and the dataset presented in this research contains 10 categorical variables. We decided to use 10 dimensions to see whether the transformation implied by multiple correspondence analysis can help with the classification accuracy.

Principal Inertia	Adjusted Inertia	Percent	Cumulative Percent
0.18497	0.00891	14.71	14.71
0.18425	0.00876	14.46	29.17
0.15601	0.00387	6.39	35.56
0.15022	0.00311	5.14	40.70
0.14705	0.00273	4.51	45.21
0.14080	0.00206	3.39	48.60
0.13615	0.00161	2.66	51.26
0.13188	0.00125	2.07	53.33
0.13016	0.00112	1.85	55.19
0.12393	0.00071	1.17	56.35
0.12191	0.00059	0.98	57.33
0.11996	0.00049	0.81	58.14
0.11427	0.00025	0.41	58.56
0.11118	0.00015	0.25	58.81
0.10944	0.00011	0.18	58.99
0.10911	0.00010	0.17	59.16
0.10588	0.00004	0.07	59.23
0.10401	0.00002	0.03	59.27
0.10324	0.00001	0.02	59.29
0.10163	0.00000	0.01	59.29
0.10136	0.00000	0.00	59.30
0.10073	0.00000	0.00	59.30
0.10072	0.00000	0.00	59.30
0.10014	0.00000	0.00	59.30
	0.03593	59.30	

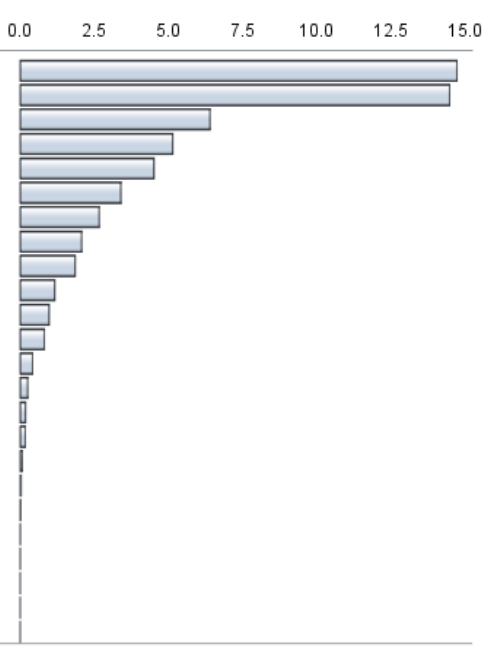


Figure 6.1: Inertia explained by the MCA components

6.1.2 Parameter Tuning

Once the selected variables are known, the parameters for the classification techniques can be tuned in such a way that the cross-validated misclassification rate is minimized. First, we find the optimal cost-complexity parameter CC for the decision tree. For the random forest, the optimal number of $VarsToTry$ need to be found and finally we obtain the optimal penalty parameter C and model parameter γ for the support vector machine. Recall that the binary logistic regression does not have any parameters to optimize.

In SAS, the procedure that constructs the decision tree selects the cost-complexity parameter value that minimizes the average misclassification rate, obtained by 10-fold cross-validation. However, it is possible that the average misclassification rates for several other values of the cost-complexity parameter are nearly the same, while resulting in a smaller tree. Breiman's 1-SE rule chooses the parameter that corresponds to the smallest subtree for which the misclassification rate is less than one standard error above the minimum misclassification rate (Breiman et al. (1984)).

Table 6.3 shows the selected cost-complexity parameter and the corresponding average cross-validated misclassification rate for every dimensionality reduction technique. This table also presents the number of leaves in the final decision tree. A smaller number of leaves corresponds with a less complex tree. Corresponding figures that show how the misclassification rate varies with the cost-complexity parameter can be found in Appendix A.

	CC	Misc. Rate	*nleaves*
No Reduction	0.0004	0.0420	83
Expert Opinion	0.0031	0.1530	6
Forward Selection	0.0003	0.0509	67
Random Forest	0.0004	0.0556	51
MCA 2 Dimensions	0.0018	0.2425	1
MCA 10 Dimensions	0.0010	0.1531	26

Table 6.3: Optimal cost-complexity parameter for decision trees

The results in Table 6.3 show that the best performing model applies no dimensionality reduction. When forward selection or a random forest are used to reduce the number of variables, there is only a slight loss in accuracy. The decision trees for these techniques have a low cost-complexity parameter, indicating that the selected variables can keep on finding splitting criteria which also lead to a good out-of-sample performance. Furthermore, this table shows that both the expert and multiple correspondence analysis are not able to accurately classify out-of-sample data using a decision tree. Having relatively small trees, the variables used cannot capture the important factors that determine whether or not someone issues a claim.

To construct a random forest in SAS, the procedure needs the input variable *VarsToTry*. That is, the size of the random subset of variables that can be used for each split. The SAS procedure does not find the value of *VarsToTry* that minimizes the average misclassification rate, thus we have to compare all possibilities and find the optimal value ourselves. Table 6.4 presents the optimal value of *VarsToTry* for all dimensionality reduction techniques. The figures that show how the average misclassification rate behaves for different values of *VarsToTry* as the number of trees of the forest increases can be found in Appendix A.

	$VarsToTry$	Misc. Rate
No Reduction	12	0.0304
Expert Opinion	1	0.2030
Forward Selection	5	0.0594
Random Forest	3	0.0596
MCA 2 Dimensions	2	0.4143
MCA 10 Dimensions	4	0.1631

Table 6.4: Optimal number of *VarsToTry* for random forest

To construct the support vector machine, we considered both a linear or an RBF kernel. To decide which kernel to use, there are two main factors to consider, computation time and accuracy. Solving the optimisation problem with a linear kernel is faster. The CPU time for an RBF kernel can be one hour, whereas the CPU time for a linear kernel is usually less than one second in SAS. However, typically, the predictive performance is better for an RBF kernel (Hsu et al. (2003)).

Given the scope of this research as well as computational limits, we have chosen only the following parameter values for the support vector machine with an RBF kernel[1]: $C = \{0.1,\ 1,\ 10\}$ and $\gamma = \{0.01,\ 0.1,\ 1\}$. For a linear kernel, it is possible to compare more values of $C = \{0.05,\ 0.1, \ldots,\ 10\}$. The parameter values that minimize the average cross-validated misclassification rate, as well as the misclassification rate itself, are given in Table 6.5.

	Linear Kernel		RBF Kernel		
	C	Misc. Rate	C	γ	Misc. Rate
No Reduction	0.75	0.0711	10	1.00	0.0607
Expert Opinion	1.00	0.2315	10	1.00	0.2300
Forward Selection	0.10	0.0720	1	1.00	0.0405
Random Forest	0.05	0.0720	1	1.00	0.0715
MCA 2 Dimensions	1.00	0.4354	1	0.10	0.4210
MCA 10 Dimensions	1.70	0.2244	1	0.10	0.1885

Table 6.5: Optimal parameters and misclassification rates for SVM

[1]We have chosen these values of C and γ by doing our own a priori grid-search. During the training of the models, we compared multiple pairs of (C, γ) and found that these values work best. For example, we compared $C = 0.001$ and $C = 10$ and found that $C = 10$ gives better results for different values of γ.

Given Table 6.5, we see that the RBF kernel outperforms the linear kernel for all dimensionality reduction techniques. The best performing model uses forward selection to determine the predictor variables. A support vector machine is the only technique where eliminating variables has a positive effect on the classification accuracy. When less informative variables are eliminated, the support vector machine is able to find a more general separation line through the data, resulting in a better out-of-sample performance. This can be explained due to the fact that when less informative variables are taken into account, different observations may seem to be more alike then they should be, resulting in a less general separation line.

6.1.3 Optimization of Classification Threshold

By tuning the classification threshold τ, it may be possible to further improve the cross-validated classification accuracy. The optimal value for τ is found by computing the *Accuracy* for different values of τ and choose the τ that obatins the maximum *Accuracy*. For each combination of dimensionality reduction and classification technique, the optimal *Accuracy* (*Acc.*) and corresponding τ are given in Table 6.6. The figures that show how the *Accuracy* varies with the value of τ can be found in Appendix A.

	DT		RF_{class}		BLR		SVM	
	Acc.	τ	*Acc.*	τ	*Acc.*	τ	*Acc.*	τ
None	0.9679	0.34	0.9971	0.56	0.9328	0.50	0.9393	0.50
Expert	0.7938	0.39	0.8329	0.48	0.7851	0.55	0.7700	0.50
Forward	0.9453	0.35	0.9540	0.48	0.9320	0.56	0.9595	0.50
$Forest_{dim}$	0.9406	0.34	0.9595	0.60	0.9328	0.51	0.9285	0.50
MCA 2D	0.5839	0.47	0.9703	0.54	0.5708	0.54	0.5780	0.50
MCA 14D	0.8114	0.38	0.9959	0.46	0.7765	0.55	0.8115	0.50

Table 6.6: Cross-Validatied *Accuracy* for optimal threshold τ

Comparing Table 6.6 with Table 6.3 - 6.5, we see that by optimizing the classification threshold τ, it is possible to slightly improve the cross-validated classification accuracy, except for the support vector machines. The results in Table 6.6 show that for this particular dataset, the combination of using no dimensionality reduction and a random forest as classification technique obtains the best cross-validated *Accuracy*. Note that the optimal value for τ is relatively close to 0.5. This is due to the fact that our dataset is relatively balanced, with 56% of the observations issuing a claim.

6.2 Out-of-Sample Results

To assess the final out-of-sample results, the remaining 2000 of 10000 observations that were not selected for the training set are used. The models will be evaluated on three different metrics. First, we evaluate the classification accuracy. The second metric to assess out-of-sample model performance is the normalized Gini coefficient and finally, model performance is evaluated on the error of the total claim amount.

The main focus of the out-of-sample results is on the classification accuracy. Given the constructed framework, it is possible to extend this research into other areas of expertise. If we can classify accurately, any data-driven decision that can be answered with 'Yes' or 'No' can be solved. This, in turn, is useful for Finaps as they can provide this service to their clients.

6.2.1 Classification Accuracy

Confusion tables are used to evaluate the classification performance of the models. A confusion table is a 2×2 matrix with the two dimensions 'actual' and 'predicted'. The observations to be predicted have an actual and predicted value (true or false) and the performance of a model is determined by the number of correctly predicted observations. Table 6.7 presents the out-of-sample confusion tables. The results in this table are obtained using the optimal model parameters and τ as specified in Section 6.1. To easily present how accurately we can classify the out-of-sample data, the *Accuracy* of each model is given in Table 6.8.

		DT		RF_{class}		BLR		SVM	
		1'	0'	1'	0'	1'	0'	1'	0'
None	1	1046	47	1056	37	1014	79	1022	71
	0	54	853	26	881	89	818	63	844
Expert	1	864	229	906	187	809	284	825	268
	0	234	673	244	663	162	745	192	715
Forward	1	1033	60	1029	64	1001	92	1044	49
	0	60	847	39	868	65	842	32	875
$Forest_{dim}$	1	1039	54	1025	68	1014	79	1016	77
	0	115	792	76	831	85	822	87	820
MCA 2D	1	994	99	698	395	1085	8	1032	61
	0	762	145	423	484	892	15	821	86
MCA 10D	1	884	209	951	142	857	236	944	149
	0	244	663	219	688	259	648	228	679

Table 6.7: Out-of-Sample confusion tables for all constructed models

	DT	RF_{class}	BLR	SVM
No Reduction	94.95	96.85	91.60	93.30
Expert Opinion	76.85	78.45	77.70	77.00
Forward Selection	94.00	94.85	92.15	95.95
Random Forest	91.55	92.80	91.80	92.85
MCA 2 Dimensions	56.95	58.65	55.00	57.80
MCA 14 Dimensions	77.35	81.95	75.25	81.15

Table 6.8: Out-of-Sample *Accuracy* for all constructed models (in %)

Given Table 6.7 and 6.8, we see that the combination of applying no dimensionality reduction and using a random forest to classify has the best out-of-sample classification performance, misclassifying only 63 of the 2000 observations, obtaining an accuracy of 96.85%. A close second is the combination of forward selection and a support vector machine with an accuracy of 95.95%. A downside of both models is that the interpretation for someone without statistical knowledge is difficult. It is not immediately clear how each variable influences the probability to claim. Therefore, we must give a honorable mention to the most simple model, the combination of no reduction and a decision tree. In a decision tree, interpreting the influence of variables is easy.

At the same time, we see that the variables picked by the expert are not able to classify as accurately, but we are not surprised by this result. Statistical methods of finding the most informative variables should outperform 'randomly' selected variables. Furthermore, the transformation using multiple correspondence analysis does not improve classification accuracy.

Given these results, we can conclude that for this problem, a model that uses all the variables available in the dataset outperforms. This result can be expected. An insurance company has to invest time and money into obtaining predictor variables. It would be a waste of resources to obtain predictor variables that have no effect on the classification performance. A useful result for further research or other applications is that it is possible to reduce the number of variables significantly, with only a slight loss of accuracy. This is especially useful when the number of variables is large, when dimensionality reduction shows its benefits.

6.2.2 Normalized Gini Coefficient

For an insurance company, being able to predict the (relative) size of the claims is useful. We use the normalized Gini coefficient to evaluate how accurately we can predict the relative size of the claims. In this metric, observations are sorted from 'largest prediction' to 'smallest prediction' and only the order determined by the predictions matters. To obtain a high score, it is important to correctly predict the relative size of the claim, rather than the actual value (Dal Pozzolo (2011)).

We expect the ratio model to perform well on the normalized Gini coefficient, especially when the size of the claims are relatively large. The ratio model includes the size of the claims, which helps identifying large expected claims. The constructed models only use the binary claim value, which might not be able to accurately identify large claims. However, since the constructed models are able to accuractely classify, we expect them to outperform the ratio model when the size of the claims goes to zero. The ratio model is unable to accurately predict claims of size zero, due to the nature of the model. Table 6.9 presents the normalized Gini coefficient of the constructed models. The normalized Gini coefficient of the ratio model is 0.664 and is outperformed by 11 of the 24 constructed models. Figures 6.2A - 6.2F show the performance of the models. These figures are in line with our expectations. The ratio model outperforms the constructed models on the relatively large claims, but as the size of the claim goes to zero, the constructed models tend to outperform.

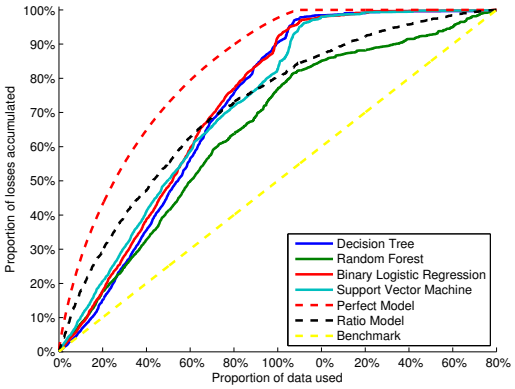
	DT	RF_{class}	BLR	SVM
No Reduction	0.668	0.454	0.698	0.676
Expert Opinion	0.087	0.521	0.735	0.545
Forward Selection	0.710	0.706	0.206	0.641
Random Forest	0.702	0.490	0.703	0.694
MCA 2 Dimensions	0.143	0.617	0.703	0.497
MCA 10 Dimensions	0.675	0.657	0.232	0.655

Table 6.9: Out-of-Sample normalized Gini coefficient

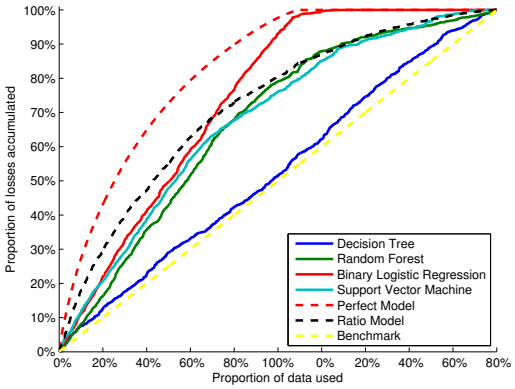
The best performing model on the normalized Gini coefficient is obtained by the expert when a binary logistic regression is used to obtain the probabilities to issue a claim. This is a rather remarkable result. For all other classification techniques, the expert has a poor performance. Furthermore, Table 6.7 shows that this combination does not classify accurately. Given these results, we regard the outperformance of this model as luck and do not consider this result credible.

The next best performing model uses the combination of forward selection and a decision tree. Table 6.9 shows that decision trees obtain a high score, except when expert opinion or multiple correspondence analysis with 2 dimensions is used. This is due to the fact that these techniques obtain very small trees with only 6 and 7 leaves respectively. Therefore a lot of similar probabilities will be found and exactly similar probabilities need to be randomly sorted, which leads to a low score.

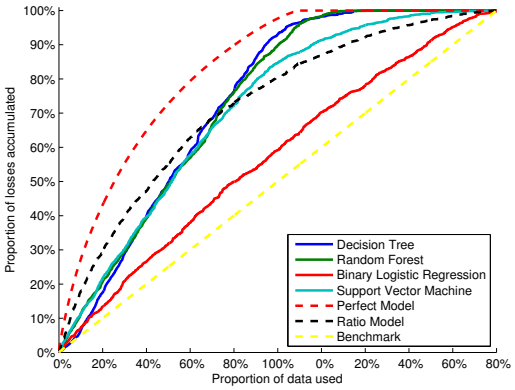
For an insurance company, we recommend to use a combination of the ratio model and the constructed models to predict the relative size of the claims. Use the ratio model to identify large expected claims. However, as the size of the claims goes to 0, use the models that have a high *Accuracy*. If a model classifies an observation as 0, give it 0 expected claim. Using a combination would most likely generate a better performance than using the individual models.



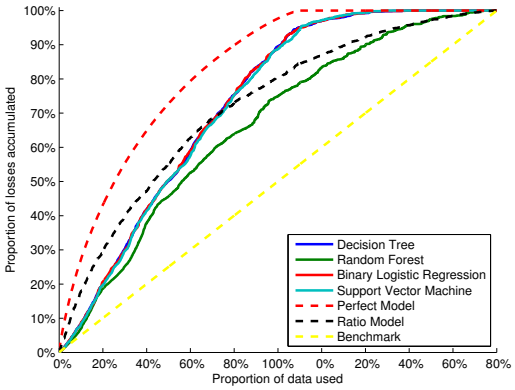
(A) No Reduction



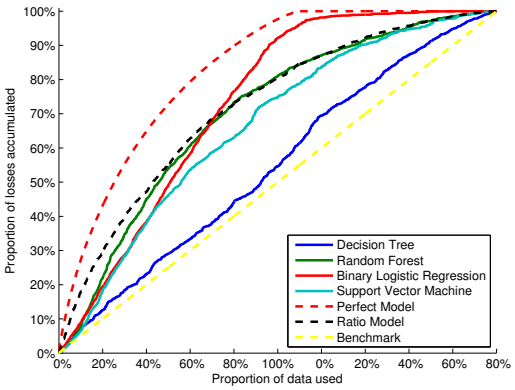
(B) Expert Opinion



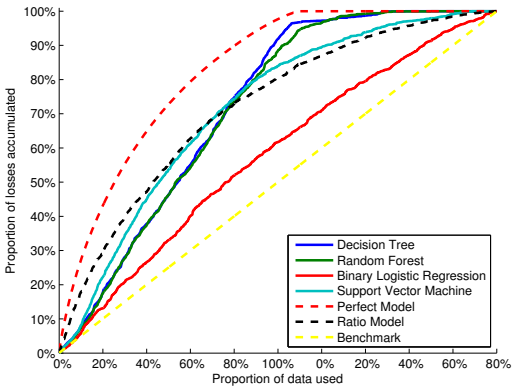
(C) Forward Selection



(D) Random Forest



(E) MCA 2 Dimensions



(F) MCA 10 Dimensions

Figure 6.2: Results of the out-of-sample normalized Gini coefficient

6.2.3 Error

Being able to classify correctly or to determine the size of the claims correctly is useful, but for an insurance company it may be more important to correctly predict the total claim mount. Eventually, this total claim amount needs to be at least covered by the premiums.

Table 6.10 presents the out-of-sample error for the total claim amount for all models. ${}_1\Xi$ and ${}_2\Xi$ are defined in Equation (4.4.19) and Equation (4.4.20) respectively. The error is measured as the sum of the predicted claims in the test data minus the sum of the actual claims in the test data. The total amount claimed in the test data is equal to €2.766.230.

	DT		RF_{class}		BLR		SVM	
	${}_1\Xi$	${}_2\Xi$	${}_1\Xi$	${}_2\Xi$	${}_1\Xi$	${}_2\Xi$	${}_1\Xi$	${}_2\Xi$
None	72.0	70.9	81.1	24.4	89.5	78.6	-114.1	32.2
Expert	147.0	65.7	154.4	199.8	110.5	-261.8	-192.4	-120.0
Forward	102.9	52.8	93.8	-11.8	85.9	-16.8	-117.0	11.5
$Forest_{dim}$	121.4	210.1	111.7	73.4	90.8	68.3	-46.7	107.0
MCA 2D	163.6	1762.8	141.1	125.0	169.7	2332.8	348.5	581.6
MCA 10D	137.4	143.1	151.5	251.4	150.5	112.1	-31.5	251.4

Table 6.10: Out-of-Sample error (in €1.000) on total claim amount

The best performing model on ${}_1\Xi$ is the combination of using multiple correspondence analysis with 10 dimensions and a support vector machine. This model has an error of €31.541, which is an error of 1.14% on the total claim amount. While this model misclassifies many observations, it still can predict the total claim amount accurately. The best performing model on ${}_2\Xi$ applies forward selection and uses a support vector machine. This model has an error of €11.532, which is an error of 0.42% on the total claim amount. Looking back at table 6.7, this model also is able to classify relatively accurate, which is a very useful result for further applications.

A simple benchmark model for this metric is the number of observations in the test set, multiplied by the average claim amount of the training set. This leads to a total claim amount of €2.925.891, which is an error of €159.661. The ratio model performs bad on this metric. Assume the expected claim amount is equal to the premium as defined in equation (5.1.2). The sum of the expected claims in the test set is equal to €4.203.288, which is an error of €1.437.057. Our best performing models outperform both benchmarks, indicating that we have gained information from the modeling approach.

Chapter 7

Conclusion and Future Work

For an insurance company, the forecasting of claims is central to a successful operation. If the claims can be forecasted accurately, premiums can be adjusted accordingly. To forecast expected claim amounts, models are constructed that use a combination of a dimensionality reduction and classification technique. The constructed models are evaluated on their classification accuracy, normalized Gini coefficient and error of the total claim amount. An overview of the findings is presented in Section 7.1. Section 7.2 discusses suggestions for future work, as well as possible extensions into other areas of expertise.

7.1 Conclusion

This research constructed a framework to forecast insurance claims. While the forecasting of insurance claims is not a new area of expertise, we decided that given the clients of Finaps and their direction as a company, we can apply machine learning techniques to obtain accurate forecasts that can be extended to other areas of expertise.

In this research, three dimensionality reduction techniques and four classification techniques were introduced to tackle the problem. First, we explained how we can extract useful information using either forward selection, a random forest or multiple correspondence analysis. Once the most useful information is extracted, classification techniques can predict whether observations are likely to issue a claim. To classify, we either used a decision tree, a random forest, a binary logistic regression or a support vector machine.

We further improved the performance of the constructed models by optimizing the model parameters. For a decision tree, optimize the cost-complexity parameter CC such that the tree can find more general rules while maintaining a minimal misclassification rate. When constructing a random forest, the size of the subset of variables that can be used for each split, *VarsToTry*, can be optimized to minimize the average cross-validated misclassification rate. For the support vector machine, we optimized the model parameters C and γ. As a final step for all techniques, it is possible to optimize the classification threshold τ, such that the average cross-validated misclassification rate is minimized.

As a benchmark, we used a commonly used model, the ratio model. This model estimates the expected claim amount as the base risk of the historical dataset, multiplied by a relative risk ratio. This ratio is calculated by the features of a client and the relative risk of these features compared to the base risk of the complete dataset. The base risk is the average claim amount of the historical dataset and the risk ratio is the conditional average claim amount on a feature.

We evaluated the models on three metrics, the classification accuracy, normalized Gini coefficient and error on total claim amount. We find that the two models that are able to classify most accurately use the combination of 'No Reduction' and 'Random Forest' or the combination 'Forward Selection' and 'Support Vector Machine'. These models have an classification accuracy of 96.85% and 95.95% respectively. To obtain a minimal error on total claim amount, a support vector machine can be used. The ratio model should be used to identify large expected claims, but has a poor performance when used to identify small expected claims or to determine the total expected claim amount.

To model data-driven decisions that can be answered with either 'Yes' or 'No', we recommend to use either a random forest or support vector machine to classify when interpretation is not necessarily needed. If interpretation of variable influence is a must, a decision tree can be used without too much loss of accuracy. For an insurance company, to predict the relative size of the claims, we recommend a combination of the ratio model and the best performing model on classification accuracy. The ratio model is used to identify large expected claims, whereas the model that classifies accurately can identify claims equal to zero.

7.2 Future Work

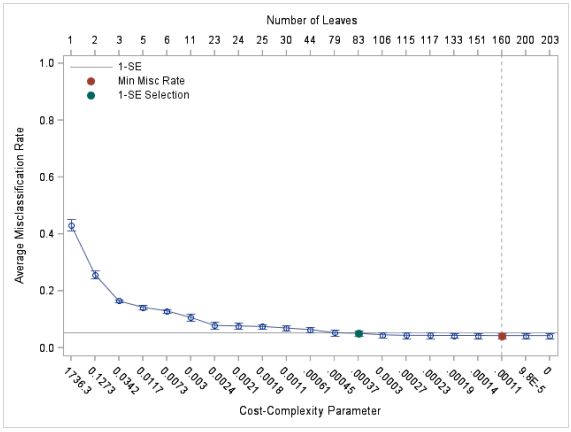
The methods presented in this research are an attempt to construct a framework that can forecast insurance claims as accurately as possible. There are still areas where the methods can be improved. We decided to reduce the forecasting of claims to a binary classification problem, but it is possible to formulate the problem as a multiclass classification problem. Instead of forecasting whether someone issues a claim, it is possible to divide the claim value into bins and predict the probability of the observation being in each bin. This will most likely improve individual accuracy, but may have a negative impact on the error of total claim amount.

By solving a multiclass classification problem, individual claims may be more accurately estimated. However, this may have a negative impact on the error of total claim amount, since this problem focuses more on individuals than on the grand scheme of things. We may use a combination of both, where we try to forecast the total expected claim amount through a binary classification problem and specialize on individuals by using a multiclass classification problem.

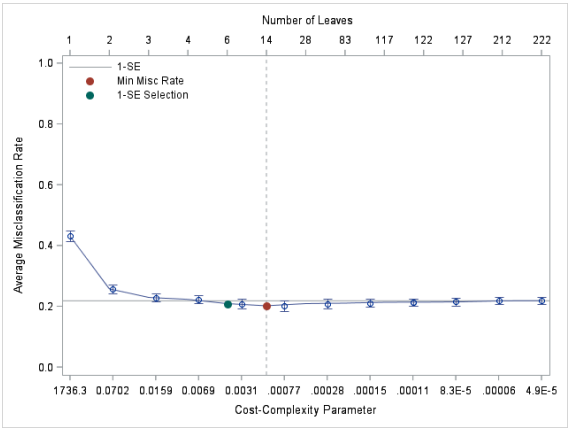
Improvement on the transformation from claim probability to claim value is possible when more data is available. For instance, we now multiply by the average claim amount of the complete training set, but we can also multiply by a conditional average claim amount. That is, the average claim amount of similar records in the historical database. However, a much bigger dataset is needed since our dataset is too sparse and this would lead to very volatile results.

By essentially solving a binary classification problem with many predictor variables, this framework can be extended into other areas of expertise. Almost every data driven decision that is needs to be answered with either 'Yes' or 'No' can be solved with this framework. A direct business case for Finaps is CV-OK. CV-OK is specialized in employment screening and offers the possibility to screen candidates and employees through an online portal. CV-OK screens whether somone has lied on their resume or not, which can also be reduced to a binary classification problem. Using our constructed framework and the characteristics of a person, we can built a model that can predict the probability that someone has lied on their resume or not.

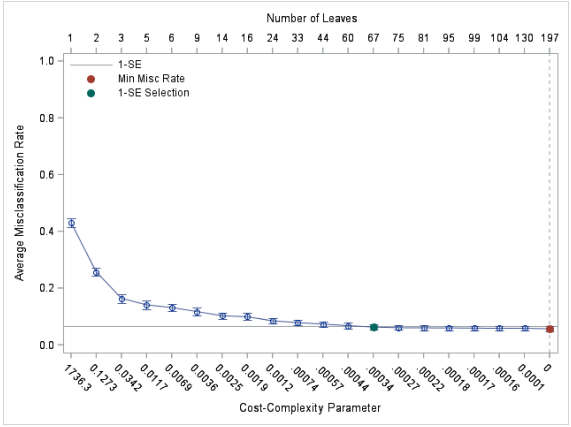
Appendix A



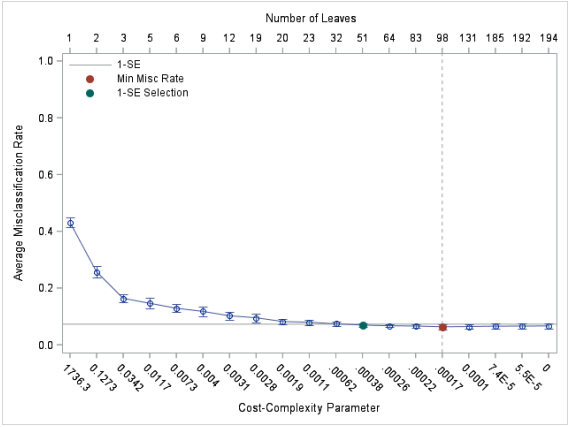
(A) No Reduction



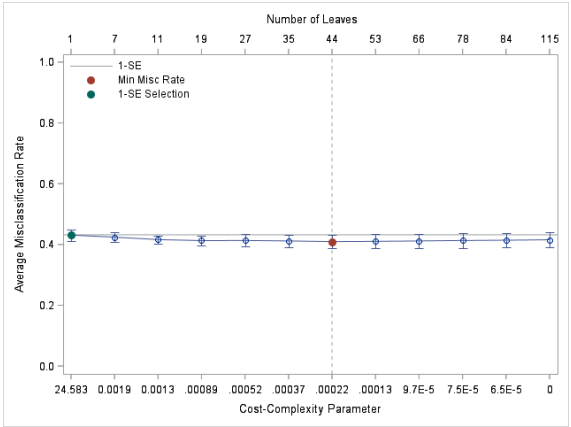
(B) Expert Opinion



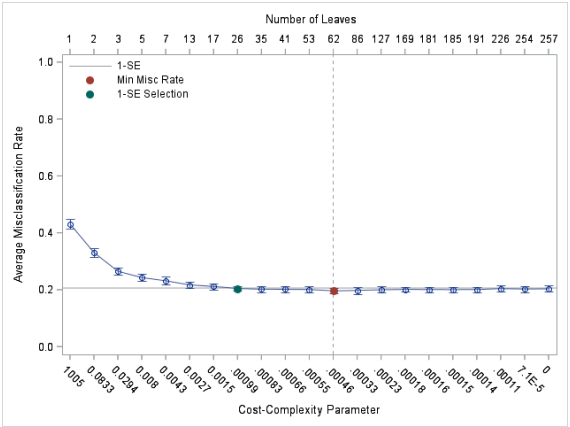
(C) Forward Selection



(D) Random Forest

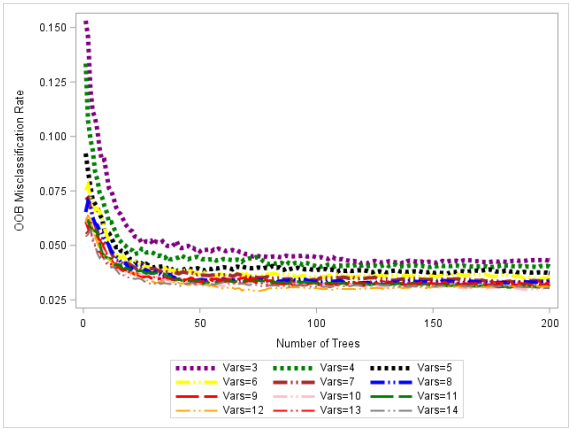


(E) MCA 2 Dimensions

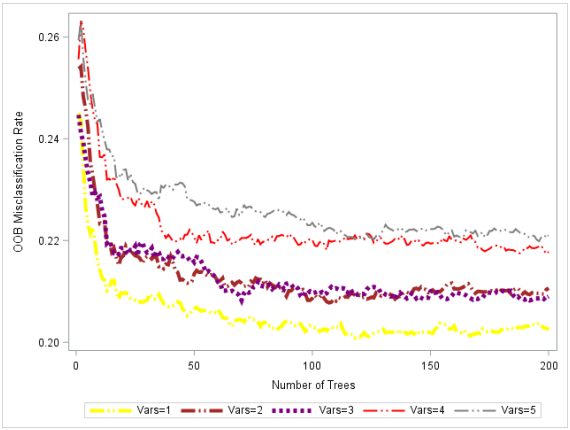


(F) MCA 14 Dimensions

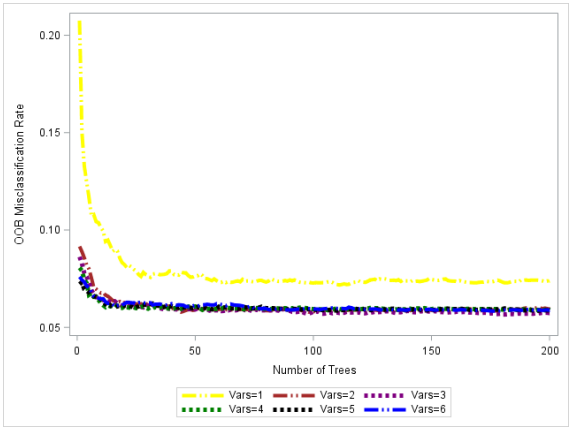
Figure .1: Selected and minimum cost-complexity parameter for decision trees



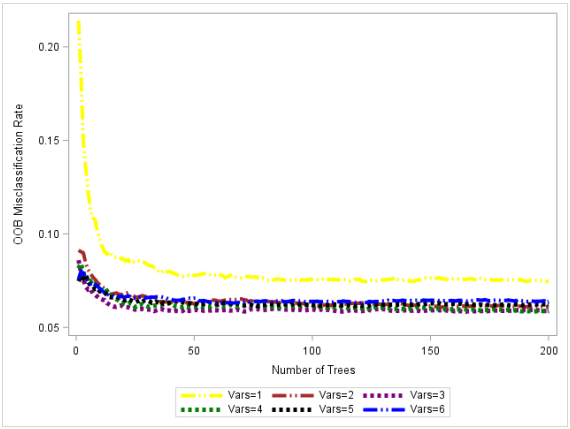
(A) No Reduction



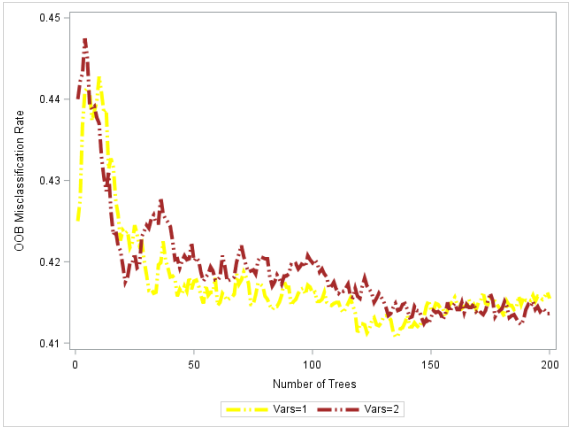
(B) Expert Opinion



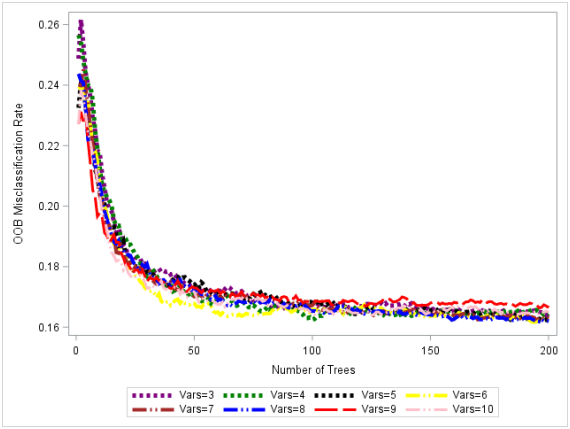
(C) Forward Selection



(D) Random Forest

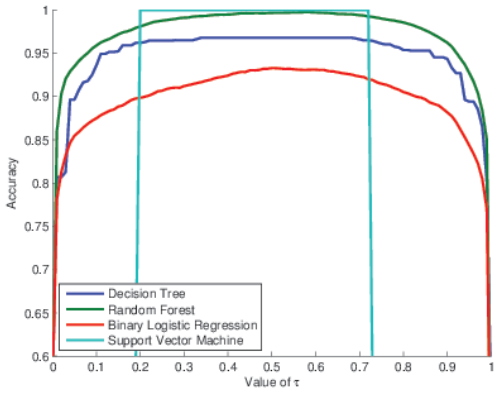


(E) MCA 2 Dimensions

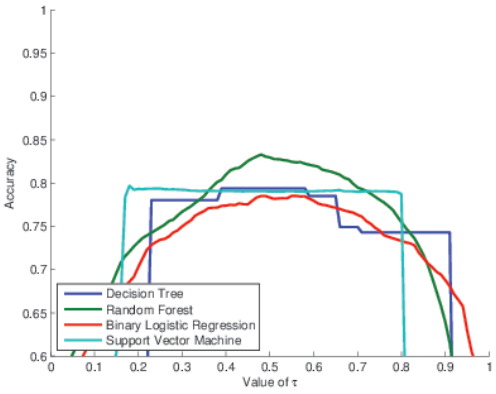


(F) MCA 10 Dimensions

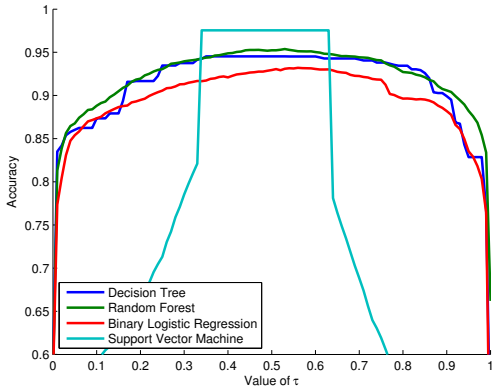
Figure .2: Misclassification rate for different values of *VarsToTry*



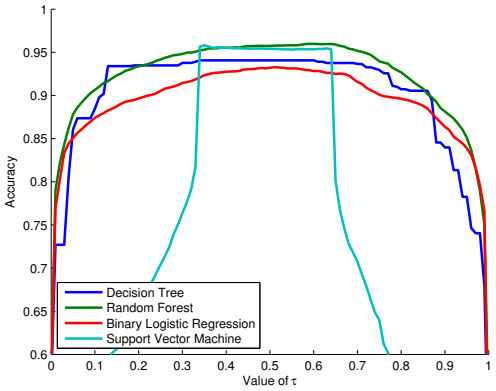
(A) No Reduction



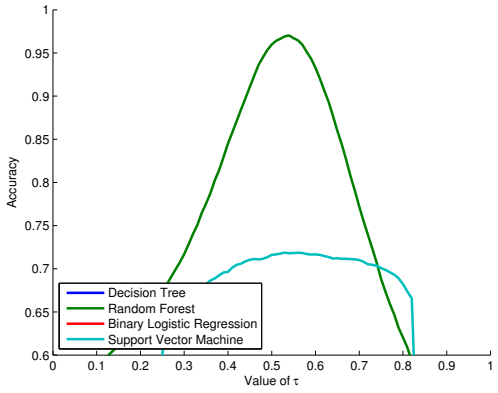
(B) Expert Opinion



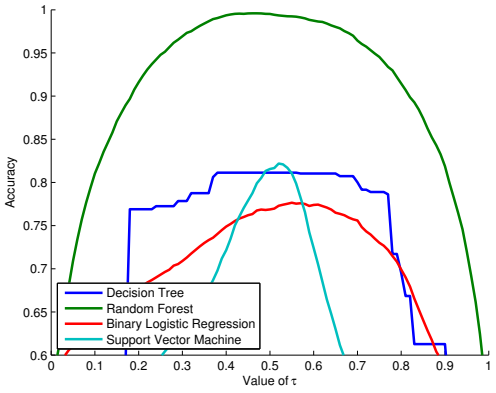
(C) Forward Selection



(D) Random Forest



(E) MCA 2 Dimensions



(F) MCA 10 Dimensions

Figure .3: Classification accuracy for different values of τ

Bibliography

H. Abdi and D. Valentin. Multiple correspondence analysis. *Encyclopedia of measurement and statistics*, pages 651–657, 2007.

M. A. Babyak. What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models. *Psychosomatic medicine*, 66(3):411–421, 2004.

M. Batty, A. Tripathi, A. Kroll, C. Wu, D. Moore, C. Stehno, L. Lau, J. Guszcza, and M. Katcher. Predictive modeling for life insurance. *Ways Life Insurers Can Participate in the Business Analytics Revolution (Deloitte Development LLP, New York)*, 2010.

S. Berridge. Forecasting claims in motor vehicle insurance. 1998.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. E. Brodley. Pruning decision trees with misclassification costs. In *European Conference on Machine Learning*, pages 131–136. Springer, 1998.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. CRC press, 1984.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

A. Dal Pozzolo. *Comparison of Data Mining Techniques for Insurance Claim Prediction*. PhD thesis, University of Bologna, 2011.

L. Doey and J. Kurta. Correspondence analysis applied to psychological research. *Tutorials in quantitative methods for psychology*, 7(1):5–14, 2011.

E. W. J. Frees, G. Meyers, and A. D. Cummings. Insurance ratemaking and a gini index. *Journal of Risk and Insurance*, 81(2):335–366, 2014.

M. Goldburd, A. Khare, and C. D. Tevet. Generalized linear models for insurance rating. 2016.

M. J. Greenacre. *Theory and applications of correspondence analysis*. 1984.

M. J. Greenacre. Correspondence analysis in practice. *Psychometrika*, 61(1):187–190, 1996.

C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. 2003.

M. I. Jordan and R. Thibaux. The kernel trick. *Lecture Notes*, 2004.

E. Kim. Everything you wanted to know about the kernel trick. 2013. URL `http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html#[1]`.

R. Lockhart. Stat 350: Variable selection. 2008.

A. J. Mata. *A Step by Step Guide to Designing Insurance Rating Models*. 2010.

I. J. Myung. The importance of complexity in model selection. *Journal of mathematical psychology*, 44(1):190–204, 2000.

J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

G. Rodriguez. Lecture notes on generalized linear models. 3, 2007. URL `http://data.princeton.edu/wws509/notes/`.

C. Shalizi. Classification and regression trees. *Statistics*, pages 36–350, 2009.

R. Silipo, I. Adae, A. Hart, and M. Berthold. Seven techniques for dimensionality reduction. *KNIME*, 2014.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

M. Tranmer and M. Elliot. Binary logistic regression. *Cathie Marsh for Census and Survey Research, Paper*, 20, 2008.

S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap. Confusion matrix-based feature selection. In *MAICS*, pages 120–127, 2011.