Erasmus University Rotterdam

Erasmus School of Economics

Master Thesis Data Science & Marketing Analytics

Predicting insurance premiums with Machine Learning: the MS Amlin case

Name: Sven Groen

Student number: 478036

Supervisor: C. Cavicchia

Second Assessor: M.G. de Jong

Date: 25-8-2023

Version 5

Abstract

In this paper, the severity loss costs of the Ocean Hull product have been modeled for MS Amlin Marine NV. MS Amlin is an insurance company that sees opportunities to apply ML models in its decision-making process. The severity loss cost is a part of the price determination, where it predicts the expected claim cost for a product. Multiple ML methods are used to develop prediction models for predicting claim values. These ML models are compared to the MS Amlin gamma distributed GLM model they use for severity loss cost modeling. All models are compared based on their predictive ability (RMSE, MAE, and AIC). After that, the best-performing machine learning methods are interpreted. The results have shown that extreme gradient boosting and random forest performed better than the MS Amlin GLM model based on the performance metrics. The most important predictors are excess insurance, vessel value, vessel year, and weight. Overall, this research can be seen as a test for MS Amlin, and perhaps also other insurance companies, to see what new technologies, such as ML, can do for them. According to this research, ML has shown predictive performance that is interesting to invest in.

Table of Contents

1. Introduction

New technologies, such as artificial intelligence (AI) and machine learning (ML) are becoming more functional for companies. Many things in life are already influenced by AI and ML. For example, Netflix uses ML when suggesting a list of movies users might like to watch (Beam & Kohane, 2018). Here, Netflix shows just the tip of the iceberg of opportunities that AI and ML applications can bring to companies. The main advantages of AI and ML applications are cost-saving, revenue growth, product innovations, increased efficiency, etc. Therefore, the number of AI and ML applications in many sectors, such as the health- and real estate sectors, is rising. Take, for example, the current market value of a house. It was a common struggle for real estate owners to accurately estimate the value of their properties due to multiple factors that play a role in the prediction, such as location, size, number of bedrooms, and more. Human analysis of these complex factors can be hard to process. This is where ML can step in to assist. Advanced ML models can accurately predict real estate values by analyzing hundreds of factors. These models can learn from millions of past property sales and pricing data to make accurate price predictions for new properties.

So, by properly handling data, ML models can make accurate predictions, which are beneficial for making choices and creating strategies for a business. ML finds utility in any problem that contains time-series data and a goal to predict the future. Predictive ML methods, such as random forest, boosting, and support vector machines, are trained to recognize patterns and correlations within data to predict future values and occurrences. ML holds some advantages over traditional forecast methods, such as moving averages and linear regression. First, ML models possess the capacity to learn and adjust from new data volumes, while traditional forecasting methods mostly use predefined rules. ML can also adapt quite quickly to changes in the data set, whereas traditional forecasting methods can become less accurate over time. Further, compared to traditional forecasting methods, ML is not biased by human input, such as emotions or subjective opinions. To reduce the bias of an ML model cross-validation can be used to check the model's accuracy. Therefore, using ML for predictive purposes can probably generate more accurate predictions than traditional forecast models. These reasons make ML a powerful tool to use in decision-making processes within a business.

1.1 MS Amlin

The insurance industry is one of the oldest industries in the world, with marine insurance as the first form of insurance. As client data continues to grow, there are opportunities for improvement in methodologies used in the sector for policy enrollment and claims handling. Lately, the insurance industry has been moving toward new technologies (Stoeckli, Dremel & Uebernickel, 2018). These technological changes due to digitization give opportunities to tech-enabled products and business models (Rawat S., Rawat A., Kumar & Sabitha, 2021). Therefore, the popularity of ML is rising among insurance companies. One of the companies interested in these new technologies is MS Amlin. MS Amlin Marine NV is part of the Japanese MS&AD Insurance Group. MS Amlin is a service company operating in the global Marine & Specialty niche insurance market, selling several products on behalf of its insurance risk carriers (insurance companies) within the group and some other third-party risk carriers. MS Amlin's main activities are delivering excellent underwriting and claim-handling services to clients worldwide. They have seven main, niche products in Cargo, Fixed Premium P&I (Protection and Indemnity), Hull, Liability, Specie, War, and Yacht Insurance. MS Amlin is always open to new ideas and is interested in exploring new research fields. Hence, MS Amlin is curious to find out what ML models can offer them. MS Amlin is aware of other companies that are successfully implementing ML techniques in decision-making and strategy. As mentioned before, ML showed great predictive power that helped Netflix and the real estate industry. The real estate property pricing problem looks similar to the premium pricing problem that insurance companies need to deal with. So, ML could be applied in many areas of the business. However, MS Amlin is in the first place, interested in whether it can predict severity loss costs, which can be seen as the expected claim cost, to help set premium prices. MS Amlin sees the market is changing, and they feel that they are still technologically behind in some business areas. Implementing ML might be the next step for them to bring technological innovations to the company. Since most industries are getting more digital, MS Amlin is convinced, just as Braun & Schreiber (2017), that data has the potential to change the insurance business in a positive direction. In the insurance market, InsurTech companies are already trying to bring new technologies to the world of insurance (Rawat et al., 2021), both with and without success. MS Amlin sees these InsurTech companies as disruptors of the market that are trying to gain market share by implementing new technologies. Therefore, MS Amlin keeps a close eye on the movements these InsurTech companies make, such that they stay aware of their successes.

1.2 Idea of Machine Learning

ML was introduced to create models that describe the connection between a set of observable inputs and another set of related output variables. It can be defined as stimulating computers to make successful predictions using past experiences. It is achieved by analyzing the available data and optimizing a performance metric specific to the problem (Baştanlar & Özuysal, 2014). To help better understand what ML means, we can split the words and see how they influence each other. First, the definition of 'learning' is the means to obtain knowledge. Humans learn from experience because of their cognitive thinking. On the other hand, 'machines' learn from algorithms and, therefore, rely on data when they get trained. So, 'Machine Learning' can be seen as an algorithm that enables computers to think and learn (Gupta, 2020). In general, computers learn how to solve tasks that are difficult to solve and program by hand. An ML algorithm learns based on historical data, and this data gets split into training and test samples. An ML model gets trained on the training data such that it can later apply the knowledge it gained. Then, the model's accuracy is determined by testing it on the testing data before applying it for real. For ML, the process involves enabling computers to adjust their actions to improve accuracy. (Alzubi, Nayyar & Kumar, 2018). The intention is to learn from the data so analysts can recommend further actions (Mahesh, 2020).

1.3 Machine Learning models for premium pricing?

Insurance companies are facing challenges in many fields. However, one of their biggest challenges is finding the appropriate balance between covering costs while offering the lowest possible premium to attract and retain a large customer base. This problem is called the premium pricing problem. Insurance companies need to take the economic theories of decision-making under uncertainty and expected utility into account when they set a premium price (Laeven & Goovaerts, 2008). To help tackle the pricing challenge, insurers are exploring opportunities in new technologies. MS Amlin is one of these insurers who sees opportunities to apply ML models in their decision-making process. The focus of this research is on ML models and their use in predicting severity loss costs. This way they can better set the insurance premium prices for MS Amlin. Predicting claims is a critical challenge for insurers and has significant implications for their managerial, financial, and underwriting decisions (Bärtl & Krummaker, 2020). MS Amlin uses ML in the form of Generalized Linear Models (GLM) to help determine the expected loss costs. Linear Regression models force the prediction to be a linear combination of variables, which leads to interpretable models. Linear effects are easy to quantify and describe. Also, Linear Regression models are additive, so if you suspect variable interactions, you can add interaction terms or use regression splines since it is easy to separate the effect. So, there are many possibilities with these statistical models since they are intuitive, additive, and easy to interpret. However, Maulud & Abdulazeez (2020) state that Linear Regression models have some challenges and limitations. The limitations are that they are unable to capture nonlinear relationships, sensitive to outliers, and need a high amount of data to create accurate models. The ideal model has both a good predictive performance and is well interpretable. In practice, you will have to make a trade-off between the two of them. This thesis will test other ML models to predict the severity loss costs to see if there are stronger models to predict these expected costs.

1.4 Research Question

The research question that this thesis aims to address is:
Which models can best model severity costs to help set insurance premium prices?

Sub-questions:
i) What Machine Learning model gives the best predictions?
ii) What predictors are most important in predicting severity costs?
iii) What effects do the most important predictors have on the severity costs?
iv) How did the results of Machine Learning Models do compared to the results of the GLM model of MS Amlin?

This study aims to answer the research and sub-questions by applying models based on decision trees, random forest (RF), extreme gradient boosting (XGB), and support vector regression (SVR) to predict premium prices. The decision tree-based regression is the first method because it is relatively easy to interpret. In literature, decision trees often get outperformed by complexer methods. Therefore, the focus of this model is on interpreting rather than achieving the highest possible accuracy. This way, an intuitive display is shown in predicting the severity costs. After that, the other models get applied to see whether more complex ML models can predict the expected loss costs more accurately. The more complex ML methods (such as RF, XGB, and SVR) get tuned to achieve the highest possible predictive power. The predictive power is measured based on the test errors. The predictive power then gets compared, and the best-performing model will be used to analyze the effects of the most significant characteristics on the expected loss costs. Interpretation will be carried out using global interpretation methods. Variable importance plots will be used to show what predictors have the most effect on the prediction of the outcome variable. One example of doing this is by looking at how the model's error rises when a variable gets left out. Partial dependence plots show the marginal effect of a variable on the predicted outcome variable. A partial dependence plot mostly captures the relationship between the target variable and one or two predictors. In the end, the results of all ML models are compared to one other, and the results of the MS Amlin GLM model.

1.5 Managerial relevance

Since the insurance industry is getting more digital, data is starting to influence the business (Braun & Schreiber, 2017). Insurance companies have a lot of data available, so it is time to use it well. MS Amlin is curious to see what their data can tell through ML models. The thesis is managerial relevant for MS Amlin since they are interested to see whether ML models can predict the severity loss costs better than their current model. More accurate severity loss costs predictions can give insights into setting the premium price and business strategies. In terms of pricing strategies, it can positively impact their profitability. For MS Amlin, the underwriters use GLM models to help set premiums. Here, the model generates rating factors for all predictors that lead to the severity cost. Then underwriters use the predicted value of the expected loss costs and frequencies when setting the premium. The underwriter can alter the premium based on personal belief or motivation. The underwriter reduces or increases the premium by at most 5%. This study can help underwriters understand why premiums get predicted the way they are. ML and interpretation models can show what factors have a consequential influence on the severity cost predictions. This way, underwriters get an intuitive explanation for the loss cost value when they set the premium. If there is evidence that ML models predict the expected loss costs better, then MS Amlin might be interested in using ML algorithms in multiple business areas. This thesis will be a test for MS Amlin to see if it is worth it to start working with ML models. If the test is successful, there is a possibility they want to implement the use of ML further. Still, this thesis is also relevant to other insurance companies with lagging models. Implementing new technologies that could show more predictive power should be given a try.

1.6 Academical relevance

To determine prices of premiums, an insurer needs to consider the economic theories of choice under risk, financial economics, and actuarial pricing theory, and, in some cases, also ML models (Tsanakas & Desli, 2005). Classical actuarial pricing of insurance risks mainly relies on the economic theories of decision under uncertainty, in particular on expected utility theory (Von Neumann & Morgenstern, 1947) and subjective expected utility theory (Savage, 1954)). Also, the anticipated utility theory can be used to model individual risk preferences and calculate premiums that reflect those preferences (Luan, 2001). Some of the models go together with Linear models (Rosen, 1974), where premiums get derived from the expected loss costs. It is interesting to see how ML might help in the process of predicting premium prices. So, in this research, it is interesting to test whether ML models might be the next big step in predicting the expected loss costs that help set premiums since the insurance industry is transforming with the emergence of digital technologies such as big data analytics, mobile distribution models, etc. (Greineder, Riasanow, Böhm & Krcmar, 2020).

The insurance market is trying to implement new technologies. That means that, lately, more academic literature is becoming available at premium pricing with the help of new technologies. AI and ML can provide insurers with valuable insights to develop customized insurance plans by analyzing past data. For example, ML gets used in the insurance industry by creating personalized health insurance plans, churn models, fraud detection models, and customer profit value models (Kaushik, Bhardwaj, Dwivedi & Singh, 2022; Hanafy & Ming, 2021; Subudhi & Panigrahi, 2020; Fang, Jiang & Song, 2016). These are all examples of applications where ML models get used in the insurance market. The range of these applications is wide, but there is little academic research about setting premium prices with the help of ML. Additionally, most of this literature is about predicting the expected loss costs (Guelman, 2012; Bärtl & Krummaker, 2020) and risk exposure (Bertsimas & Orfanoudaki, 2021; Quan & Valdez, 2018) as part of pricing insurance premiums. Better predicting power for expected loss costs can help many insurers in the decision-making process. Therefore, this research can give insights into the use of ML for the prediction of expected loss costs for niche products. Insurance companies that have the same sort of nice product line as MS Amlin might benefit from this thesis since niche products can differ a lot from mainstream products.

2. Literature Review

In this chapter, economic literature is discussed in the field of insurance and ML technology. This chapter aims to explain insights into the current and past insurance industry. With this knowledge, it is interesting to see whether ML models can help the insurance industry, especially in the premium pricing area. By examining academic papers, books, and reports, the chapter will contain information from different angles to understand what insurance is and how they price their premiums. The chapter starts with an explanation of what insurance is and how it can be priced. After that, the problems that the insurance industry faces will be explained. The chapter ends by introducing ML as a new technology in the insurance industry.

2.1 Idea of insurance

Insurance can be seen as a technology of risk. In the insurance language, risk means an event or general event of an unfortunate kind. The risk goes together with chance, hazard, probability, and randomness (Ewold, 1991). In everyday life, people usually discuss risk in verbal qualitative terms that are not readily translatable into the probability language. The purpose of insurance is to protect individuals from suffering the full consequences of unpredictable risk (Spence & Zeckhauser, 1978). A contract between two parties, the insurer and the insured, defines this protection. The insurer is the company that offers insurance policies, and the insured is the person who purchases insurance to gain its benefits. The insurer pays a premium to the insurer, and then the insurer takes on the insurer's risk against future events or accidents (Rawat et al., 2021). For example, the insurer will cover the cost of repairing a car after an accident, medical treatment following an injury, or losing belongings abroad. These risks are unpredictable, but these costs can get high. For this reason, insurance is valuable to have.

2.2 The premium price and expected loss costs

The price or premium of insurance is the value for which the customer (insured) and insurer agree to exchange risk for certainty. Classical pricing of insurance risks mainly relies on the economic theories of decision-making under uncertainty and expected utility (Laeven & Goovaerts, 2008). Utility theory also provides insight into decision-making in the face of uncertainty. Further, Ang & Lai (1987) emphasize that the underwriters have to consider both the capital and insurance market when setting a premium. Equilibria for premium pricing in the insurance market can have particular characteristics. Classical premium principles depend on the predictors of the risk, which estimate expected losses. The expected loss costs are a prediction of the value of claims that the insurer has to pay (severity) and a prediction of the number of claims (frequency). The expected loss ratio gets estimated by dividing the value of claims by the total value of the premiums. To operate profitably, insurance companies need to have a healthy expected loss ratio (at least below 100% since a 100% rate is break-even). If the company expects a loss but does not have enough premium earnings or investments to compensate for that loss, they are at risk of going bankrupt. It is also realistic to consider setting a premium that considers general market conditions, such as aggregate risk, aggregate wealth, dependencies between the individual risk, and general market risk (Bühlmann, 1980). In all, to stay competitive, products must be priced suitably. Traditionally, insurance mostly relies on actuarial (regression) models. The frequency-severity model is a widely used approach in insurance claims analysis (Su & Bai, 2020), and MS Amlin also uses this pricing model. The model focuses on separately modeling two key aspects: claim frequency and claim severity. The claim frequency component of the model explores the number of claims occurring over a specified period. It investigates the patterns and factors influencing the frequency of claims, helping to understand the likelihood of claims occurring. The average claim severity component considers the average amount of money associated with each claim. It takes into account the monetary value of claims given that they have occurred, allowing for a deeper understanding of the financial impact of claims (Garrido, Genest & Schulz, 2016). Actuarial GLM methods have proven to be valuable tools in insurance analysis (Renshaw, 1994) due to their ability to express the means of the frequency and severity processes as linear combinations of rating variables, such as age, weight, and other relevant factors.

This research looks at the partial loss cost section of the Ocean Hull insurance of MS Amlin. Here, MS Amlin uses a GLM method with a Poisson distribution to model the partial loss frequency and a GLM method with a Gamma distribution to model the partial loss severity. However, since ML techniques are obtaining more popularity in the area of insurance analytics, it could be expected that the expected loss costs might also be modeled with the help of these ML models (Staudt & Wagner, 2021). Therefore, this thesis focuses on predicting the partial loss severity with ML methods to model the value of a claim as part of the expected loss cost. The ML models will be compared to the MS Amlin GLM severity model.

2.3 Risk aversion, adverse selection, asymmetric information and moral hazard

In a competitive insurance market, customers try to select contracts that maximize their expected utility (Utility theory). Since people tend to feel losses heavier than profits, we can tell that most customers for insurance are risk-averse (Holt & Laury, 2002). Risk aversion is experienced through game and pricing tasks (Harrison, 1989). A strong example of risk aversion for insurance is probabilistic insurance. Probabilistic insurance means that an insurer sometimes doesn't cover lower than 1% risk, this risk can be issues such as war. Wakker, Thaler & Tversky (1997) found that most clients demanded more than a 20% reduction in premium to offset a 1% default risk. Risk aversion is one of the core reasons for the existence of insurance since people who feel risk-averse want insurance against any form of risk. Insurers found that high-risk people, people that are more at risk, are more willing to pay higher premiums. Therefore, if insurance companies increase their premiums, the riskier people keep purchasing insurance, while lower-risk people might stop. This causes the equilibrium prices of these plans to rise. On the other hand, lower-risk client select less comprehensive coverage than they would otherwise prefer. This concept is called adverse selection (Akerlof, 1970).

In most cases, individuals are aware of their risks or accident probabilities, while companies are not. This concept is called asymmetric information (Chiappori & Salanié, 2013). Risk or accident probabilities can be seen as probabilities that events occur where a customer experiences loss. Since most customers are highly identical in their probability of having an accident, they should not be discriminated against by another. However, in practice, the risk can differ substantially per customer. Insurance companies have learned to use tools to model risks based on their characteristics. They use this to prompt the premium price to cover (potential) higher-risk customers. These higher-risk customers have higher expected loss costs for the insurer, so insurers need to increase the premium for these customers. Asymmetric information is one of the core problems for insurance companies (Rothschild & Stiglitz, 1978). It is the main reason adverse selection and moral hazards exist in insurance markets. Asymmetric information happens when the insured has better information about their risk (or expected loss costs), which the insurers lack. They then use this information in making insurance purchases (Cohen & Siegelman, 2010). Typically, private information is revealed in the equilibrium since high-risk and low-risk customers choose different contracts for different types of protection (Smart, 2000). Therefore, checking the willingness to pay is smart to do in the possible occurrence of asymmetric information.

The government finds that every individual has the right to be insured. However, the role of the government can negatively affect the efficiency of the insurance market. Subsidies and other forms of government support can lead to an imperfect market because of adverse selection and moral hazard problems. Moral hazard can be seen as the behavior of clients that changes after they purchase insurance. They take riskier actions since the insurance will financially cover the accidents if they occur. Insurance markets may be more efficient in providing insurance, as long as the insurability of the risk is sufficiently high (Chambers, 1989). Therefore, it is important to bear in mind moral hazards when modeling the expected loss costs. Cohen & Siegelman (2010) claim that the presence of adverse selection and moral hazard in insurance markets has important implications for policymakers. They argue that regulations designed to reduce adverse selection can result in increased premiums and decreased coverage levels. However, they recommend that policymakers consider alternative approaches, such as risk adjustment and reinsurance, to reduce the effects of adverse selection. This while still offering affordable coverage options for consumers. The trade-off between risk aversion and moral hazard in insurance economics is an ongoing problem. Higher insurance coverage implies less risk-bearing by the insured but induces more moral hazard (Newhouse, 1996).

2.4 New technologies

Insurers are beginning to see the possibilities of adapting to new technologies. Over 50% of insurers' IT budget gets spent on running costs rather than research and development (Cortis, Debattista, Debono & Farrell, 2019). However, it seems likely that all insurance companies will be using ML and AI as new technologies behind their underwriting decisions over time. For example, the buffet-style approach, where you pay the same amount irrespective of use, would not apply to motor insurance if the price gets determined with the help of AI telematics devices (Azzopardi and Cortis 2013). Here, AI helps to accurately tell how much an individual is driving so that everyone pays a fair premium based on how much they drive. Also, ML models can be used in fraud detection, risk selection, marketing, and pricing strategies. Many upcoming changes due to digitization will generate new challenges that rely on tech-enabled products and business models (the start of InsureTech startups). InsurTech is simply bringing technology into the world of insurance (Rawat et al., 2021). Industry observers believe that some of these startups may have the potential to disrupt the insurance market in the long run. As a reaction to the expected pressure from InsureTech companies, many insurance companies are currently screening the InsurTech landscape for technology that may help them adapt to digital change. For example, Insurtech company Lemonade strives to address claim process issues with the help of AI by implementing a self-service, user-friendly approach. Lemonade's technology is designed to enable customers to quickly and effortlessly save up on their insurance expenses within 90 seconds.

2.5 Machine learning models for insurance

Predictive models may give new insights into the problem that the expected loss is highly dependent on the characteristics of an individual policy (Yang, Qian, & Zou, 2018). Machine learning can process vast volumes of data in real time and give insightful information based on past data. This way, insurers can identify and develop insurance plans for all unique clients. Kaushik et al. (2022) found that based on an artificial neural network, insurers can provide a personalized health insurance plan rather than a health insurance package where clients pay for services they may not use. Therefore, ML in healthcare insurance boosts efficiency and treatment effectiveness. The study by Hanafy & Ming (2021) provided an accurate model for insurance companies to predict whether the customer relationship with the insurer gets renewed. They state that applying ML models in insurance can be the same as in other industries where it is already in use. The goal is to optimize marketing strategies, improve the business, enhance income, and reduce costs. A random forest (RF) was the most efficient model to classify whether an insurer would renew his contract for a new period. Subudhi & Panigrahi (2020) reported that a support vector machine model achieved high accuracy in detecting fraudulent claims, outperforming existing fraud detection models in the literature. In a study by Fang et al. (2016) an RF customer profitability model can predict the customer profit value by using only a small subset of variables. Guelman (2012) described an application of gradient boosting (GB) to the analysis of loss cost for auto insurance. GB can handle complex and missing data better than linear models, and that is why accuracy is shown to be higher for GB relative to a generalized linear model approach. Most of the ML models above are tree-based models. Decision tree-based models are popular since they are considered to be nonparametric. Additionally, they generate algorithms that can handle missing data. Further, it can provide a variable selection procedure by assessing the relative importance of the explanatory variables

(Quan & Valdez, 2018). These points counter some of the challenges that linear regression models have.

3. Data

This chapter provides information on the data set that will be used for modeling in this research. The chapter starts with the data description. Here, the distribution of the target variable gets explained, and there is an emphasis on the choices for the predictor variables. Also, where needed, complex predictor variables will have extra explanations. This way the target variable and the predictors used for modeling are clear. In the second part of this chapter, the data transformations will be discussed. Here, information on one-hot encoding and standardizing of the data will be given.

3.1 Data description

The MS Amlin pricing team has provided the data for this research. Since the focus is on predicting severity loss cost for the MS Amlin 'Ocean Hull' pricing division, all available information about 'Ocean Hull' claims is needed. The dataset that gets used is the same as the dataset that MS Amlin used to create their GLM prediction severity model in 2020. Originally, the plan was for me to gather recenter data within MS Amlin, such that I would make a prediction model based on the last years. However, eventually, this task was too time-consuming, so we went for the above-mentioned dataset. The data contains information on 4421 different claims from 2006-2020. This information is on both vessel and policy levels. Vessel-level information is specific to each unique vessel, and policy-level information is data that is more or less the same for all vessels within the same policy. The dependent variable is the value of the claims incurred measured in US dollars. The highest claim incurred is around 35 million US dollars, while the lowest is just a few US dollars. Therefore, the range of the claim value is enormous. The dependent variable, the claim value incurred, is skewed to the right. The dependent variable is skewed because around 81% of the claims are lower than the average (380,000 US dollars). Also, almost 1000 claims (more than 20%) are lower than 5000 US dollars. In Figure 1, the distribution of the dependent variable is displayed. In this Figure, it has been chosen to exclude all claim values over 1 million US dollars because only 8% of all claims are above 1 million US dollars, and these claims values are too spread out (1-35 million US dollars range). Therefore, to best show the distribution, it has been chosen to exclude them from Figure 1. The dependent variable will not be transformed. This way the results of all models can be compared based on the same values.

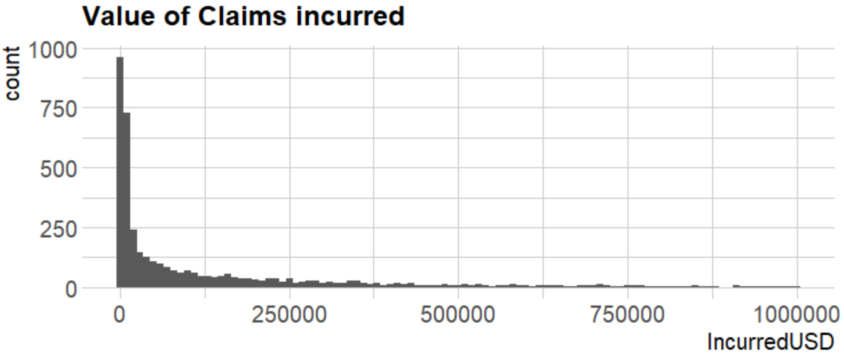


Figure 1. Distribution of the value of claims incurred (range 0 - 1 million US dollars).

The data contained 35 columns. To get high-end performance ML prediction models, only variables that affect the dependent variable need to be selected. Most policy-level variables are not interesting in predicting the claim value of a specific vessel. All uninteresting columns that have no predictive power, such as policy ID and vessel name, get excluded. Therefore, out of the 35 original columns, 12 variables are used in training the ML models. The vessel type, age, vessel value, and weight variables tell about vessel characteristics. For weight, there are three weight variables in the data. Gross tonnage describes how much a vessel weighs by itself in tons. Net tonnage is the weight of how much a vessel can carry in tons. Deadweight tonnage is the weight of everything. It is the sum of the weights of cargo, fuel, fresh water, ballast water, provisions, and crew. Therefore, it does not mean that gross and net tonnage added is deadweight! ExcessUSD and DeductibleUSD tell about the worth of the policy for a specific vessel. The ExcessUSD and DeductibleUSD variables are alike, but there is a difference. A deductible is the portion of a claim the insured must pay before the insurance company covers the remaining amount until the agreed limit is reached. On the other hand, excess insurance is an extra insurance policy purchased to provide coverage for losses that exceed the limits of the primary insurance. The P&I club variable seems interesting as well. In the next paragraph, more information on this variable since it has to be transformed before it is interesting. The vessel years variable can not be excluded either since it tells about the length of insurance. Most insurance policies are one year exactly, but for some, it is a few months more or less. The vessel years variable is a variable that captures the length of the vessel exposure. On the recommendation of the pricing team, the last variable in the data is the deductible per deadweight tonnage variable, which gives the ratio of how much the insured thinks his weight is worth. Unfortunately, some variables that seem interesting, such as Flag_id and

Classification Society, are excluded from the research. These variables are categorical variables with too many levels. Therefore, they contain less predictive power than initially hoped. Further, there were no other interesting variables. In Table A1 and A2 in Appendix A, you can see the summary statistics for the used numeric and categorical variables respectively.

3.2 Data transformations

3.2.1 Variable transformation

The biggest variable transformation happened to the 'PIClub' variable. An P&I Club (Protection and Indemnity Club) is a non-governmental, not-for-profit insurance association providing marine insurance. The 'PIClub' variable returns the name of a P&I club that represents the third-party insurance for the specific vessel. For example, if an oil leak happens, a P&I club would pay out all third parties affected by this accident. Originally, this variable contained over 20 different clubs, and therefore, the variable would be a big categorical variable. However, based on the MS Amlin-derived rating factor, these P&I clubs can be classified as good, average, or bad. Generally, good P&I clubs are known for having high standards in the vessels they insure, while average or bad P&I clubs also take on the 'riskier' vessels. Therefore, a good P&I club can be seen as a characteristic of a vessel's quality.

3.2.2 One-hot encoding

The dataset contains two categorical variables. For the tree-based methods, this does not matter, but for the support vector regression (SVR) it does matter. This method can not handle categorical variables. The categorical variables are therefore encoded using one-hot encoding to make them numerical. One-hot encoding means that a separate column gets created for each unique value of a factor. Therefore, a separate dataset is constructed for the SVR method. Here, the total number of variables after one-hot encoding equals 26. It is important to note that one of the new columns of the formerly categorical variables has to be the baseline. Therefore, one of the formerly categorical variables has to be excluded from the SVR dataset.

3.2.3 Standardize

For many of the variables, the mean and variance differ a lot from one another. Again, decision-tree-based methods are not that much affected by this. However, the optimal hyperplane of an SVR gets influenced by the scale of the input variables. If the variables are on different scales, the hyperplane will be heavily affected by the variables with a larger standard deviation, potentially leading to suboptimal results. So, it is recommended to alter the one-hot encoded SVR dataset once again and normalize all variables. By standardizing, all variables will have a mean of 0 and a variance of 1. Variables get normalized using z-score normalization. The formula for calculating z-scores is as follows:

$$z = \frac{X - \mu}{\sigma}. \tag{1}$$

Here, X is a value of the variable, μ is the mean of the variable, and σ is the standard deviation of the variables

4. Methodology

In this chapter, the methods used in this research will be explained. First, it gets addressed that ML can be split into unsupervised and supervised ML. In these sections, the differences between them will be explained, and for both cases, examples of methods will be given. After that, the ML methods, the gamma distributed GLM method, and the interpretation methods that will be used in this research will be elaborated. In the end, information on the performance metrics will be provided.

4.1 Machine Learning

The use of machine learning enables computers to perform tasks by constantly learning and gaining experience to understand the complexity of a problem (Alzubi, Nayyar & Kumar, 2018). In general, computers learn how to solve tasks that are extremely difficult to program by hand. The idea is that when applying ML, the model gets trained. The data used for training purposes is called training data. Therefore, the original data gets divided into two parts, the training and test data. The trained model can provide new insights into how input variables are mapped to the output, and it can make predictions for input values that were not part of the training data (Baştanlar & Özuysal, 2014). To check how accurate the trained model is, you can run it on test data. After the model makes its predictions, you can compare the prediction values with the actual values of the test data. Machine Learning can be split into two main categories depending on whether the output values are present in the data. The groups are Unsupervised and Supervised Machine Learning.

4.1.1 Unsupervised ML

Unsupervised learning techniques do not require output values in the data. Because of that, the data for Unsupervised ML can be unlabeled. There is often a structure to the input space such that certain patterns occur more often than others, and it is important to identify what generally happens and what does not (Alpaydin, 2020). Unsupervised learning algorithms need to discover these patterns in data. Mainly Unsupervised ML models get used for clustering (grouping) and dimension reduction. Examples of clustering models are k-means and hierarchical clustering. It is not possible to directly measure the performance of clustering because there are no correct output labels. Clustering gets used for target marketing and customer segmentation. On the other hand, an example of dimension reduction is the principal component analysis (PCA). Dimension reduction methods get used mainly for reducing the predictors in a dataset (Ringnér, 2008). On the other hand, PCA has also found applications in face recognition, image compression, and neuroscience (Paul., Suman & Sultan, 2013).

4.1.2 Supervised ML

Supervised learning techniques do require output values in the data (labeled data). All algorithms learn patterns from the training dataset and apply them to the test dataset for prediction or classification (Choudhary & Gianey, 2017). When the output variables of a predictive model are continuously valued, it is referred to as a regression problem. For instance, predicting the severity costs for MS Amlin involves a regression function. Examples of regression problems are (generalized) linear regressions, decision tree-based regressions, and support vector regressions. On the other hand, if the output variables can only take on a discrete set of values, the predictive model is known as a classifier. A familiar example of a binary classification problem is a medical diagnosis. Here, the characteristics of a patient can lead to classification predictions. Then it can be classified as having a disease or not (Baştanlar & Özuysal, 2014). Examples of classification problems are logistic regression, decision tree-based classification, and support vector machine.

4.2 Methods

In this part, the following models will be explained. A regression tree, random forest regression, extreme gradient boosting regression, and support vector regression will be explained since these are the prediction models. A permuted importance plot and partial independence plot will be explained since these models help interpreting the predictions models.

4.2.1 Decision tree-based regression

A decision tree is named that way because it visually looks like a tree. A decision tree starts with the whole dataset in the root note, which then branches in smaller parts of the data. Each decision node represents a question or condition regarding the data, and the branches that lead to leaves represent potential answers. Trees aim to group observations based on similar behavior, splitting up the sample per branch. When a decision tree predicts a numeric value, it is called a regression tree. It is useful in areas where the relationship between the variables is non-linear. Further, there is little data preparation necessary, so for example, there is no need to standardize predictors as decision trees are not sensitive to the variance in the data. To create a decision tree, the data gets split into training and testing data. The decision tree is then trained on the training data, and the predictive power is evaluated by predicting the outcomes of the test data.

The process of splitting in regression trees involves looking at the sum of squared residual sum of squares (RSS) for all potential splits per predictor. This can be done by dividing the predictor space boxes. The goal is to find boxes R1, . . ., RJ that minimize the RSS, given the following formula:

$$RSS = \sum_{j=1}^{J} \sum_{i \in Rj} (y_j - \hat{y}_{Rj})^2. \tag{2}$$

Here, $\hat{y}_{Rj}$ is the mean response for the training observations within the jth box. The greedy approach gets used because it splits recursive binary. The method is referred to as top-down since it starts at the top of the tree, which contains the whole dataset, and then proceeds to split the predictor space step by step. It is considered greedy since it makes the best split at each particular step of the tree-building process (James, Witten, Hastie & Tibshirani, 2013).

When a big dataset gets used, there is a chance of overfitting, which can lead to an interpretation problem of the tree. Overfitting means that the tree will predict almost perfectly for the training data but poorly for the test or other out-of-sample data. Additionally, too many split conditions can make the tree too complex and hard to interpret. To prevent the regression tree from getting too big the tree will be pruned by finding the best complexity parameter (cp). The cp gets used to specify the minimum improvement required before proceeding, thereby allowing control over the size of the decision tree. The strategy is to grow a large tree with cp = 0, and after that, select the cp where the cross-validated cp error is the lowest, and finally reset the cp to that value to shrink the big tree again (Myles, Feudale, Liu, Woody & Brown, 2004). This tree should give the best predictions since it will not overfit. In this research, the interpretability of the regression tree is more important than the prediction power because other models can achieve higher predictive power easier. Therefore, it is better to focus on interpretability, which might mean that the tree shrinks more. This hurts the predictive power but helps make the tree more interpretable.

4.2.2 Random forest (RF)

Random forest (RF) is an ensemble machine learning model, which means that the model combines several weak learners to produce one optimal predictive model (Breiman, 2001). RF uses multiple decision trees to make a better model. Decision trees may have relatively lower accuracy, but this is fine since it is easy to understand and interpret. On the other hand, RF will generate higher predictive power but is harder to interpret. To create an RF, first, the data gets split into train and test data. Then the train data gets bootstrapped for each tree in the RF. The number of trees to create is determined beforehand. Each bootstrap sample is constructed by randomly selecting observations from the training data, with replacement. This process allows for different data samples for each tree. Then, M predictors get selected randomly for all available predictors for splitting. These randomly chosen predictors destroy the correlation between the trees in the forest. The decision for the splits of the M predictors is based on the RSS for all trees in the RF. Using a larger M increases the likelihood of selecting important predictors related to the outcome variable for most of the splits, which decreases the randomness of the trees. The number of trees depends on the strength of the individual trees and the correlation between them. By using a large number of trees, it follows from the Law of large Numbers that adding more trees does not lead to overfitting (Breiman, 2001).

For the predictions when using regression trees, the average of all trees will determine the output value (James et al., 2013). All regression trees produce an estimate, and the final prediction is an equally-weighted prediction of each estimate. This can be captured in the formula below:

$$\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B} f_b(x_i). \tag{3}$$

Here, B is the number of trees. RF is an algorithm that provides good results in the default settings (Fernández-Delgado, Cernadas, Barro & Amorim, 2014). However, a small performance gain is possible if parameters get tuned. By tuning the randomly selected M predictors (or called mtry), the model has the highest chance to gain on performance. The default for mtry is $\sqrt{p}$ for regression problems. Other parameters that can be tuned are sample size, node size, the number of trees, and the splitting criterion (Probst, Wright & Boulesteix, 2019). These parameters can be tuned on the hand of the Out of Bag error (OOB). The OOB-error is the error of the data that is not selected by the bootstrap procedure (OOB sample), which is thus not used to train any decision tree. The OOB-sample provides an efficient and reasonable approximation of the test error. In Figure 1, you can see the steps of the algorithms that construct an RF.

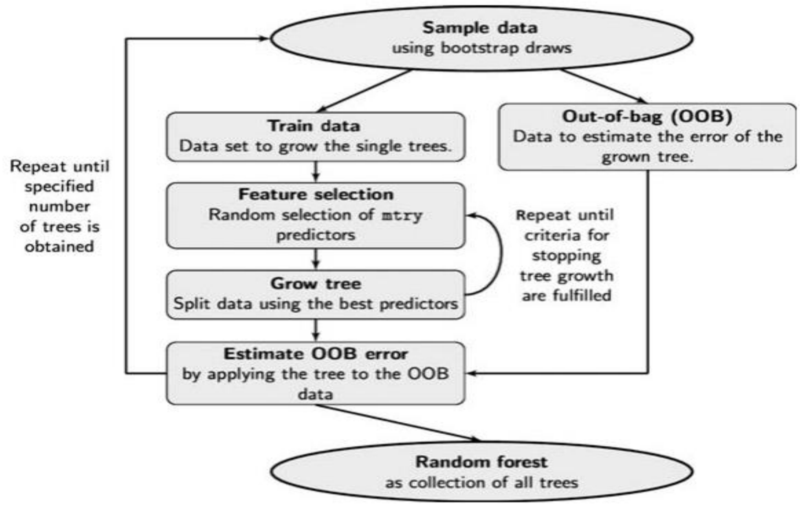

Sample data
using bootstrap draws
Train data
Data set to grow the single trees.
Out-of-bag (OOB)
Data to estimate the error of the grown tree.
Repeat until specified number of trees is obtained
Feature selection
Random selection of mtry predictors
Repeat until criteria for stopping tree growth are fulfilled
Grow tree
Split data using the best predictors
Estimate OOB error
by applying the tree to the OOB data
Random forest
as collection of all trees

Figure 1. Algorithms of a RF (from Boulesteix, Janitza., Kruppa & König, 2012)

4.2.3 Extreme gradient boosting (XGB)

Boosting is, just like RF, an ensemble method that combines several weak learners to produce one optimal predictive model (Friedman, 2002). However, boosting builds a model by adding new weak learners sequentially, rather than independently (which is the case with RF). Gradient boosting (GB) trains on the residual errors of the previous predictor. For example, by fitting each new weak tree concerning the residuals, the model will continuously improve its fit. GB is a type of boosting that uses gradient descent to train new trees. Gradient descent is an optimization algorithm that gets used to find the minimum of a function. In regression problems, it is to minimize the RSS. The algorithm works by measuring the local gradient of the cost function for the model's predictors and taking steps towards the decreasing gradient (Friedman, 2002). Extreme gradient boosting (XGB) is an optimized version of the GB algorithm because it is quicker and allows for regularization terms, which can reduce overfitting. This results in a good balance between bias and variance. Chen & Guestrin (2016) show in their paper how they created the algorithm. In this part, the steps of the XGB algorithm will be explained. The goal of the XGB algorithm is to minimize the following objective function:

$$L(\varphi) = \sum_{i} l(y_i , \hat{y}_i) + \sum_{k} \Omega(f_k), \tag{4}$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2} \lambda ||w||^2. \tag{5}$$

Here, l is a differentiable loss function, which measures the differences between the actual values y_i and the predicted values $\hat{y}_i$. Ω is the regularization term that penalizes the complexity of the model. Here, T is the number of terminal nodes (or leaves), and γ is a regularization parameter called the tree complexity parameter. The values of this interaction can be set to penalize the model for having more leaves. λ is also a regularization parameter, it can be set to get more smooth prediction to reduce overfitting. w is the square root of the sum of the squared of the weights. It can be seen as an output value, and we need to minimize $L(\varphi)$ (Chen & Guestrin, 2016). The model will be trained in an additive manner. So, if we let $\hat{y}_i^{(t)}$ be the prediction of the i-th instance at the t-th iteration, f_t needs to be added to minimize the following objective:

$$L(\varphi) = \sum_{i}^{n=1} l\left(y_i , \hat{y}_i^{(t-1)} + f_t(X_i)\right) + \Omega(f_t). \tag{6}$$

It greedily adds the f_t that most improves the model, such that the model fits better over time. Further, the model can use the second-order approximation to simplify the loss function:

$$l\left(y_i, \hat{y}_i^{(t-1)} + f_t(X_i)\right) \approx \sum_{i}^{n=1} l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(X_i) + \frac{1}{2} h_i f_t^2(X_i). \tag{7}$$

In the second-order approximation it is stated that the new loss function is approximately the old loss function, plus the first derivative of the loss function called g, plus the second derivative of the loss function called h. The new loss function (7) can be substituted in the second-order approximation (8). In the next step, $Ij = \{i|q(xi) = j\}$ gets defined as the instance set of leaf j. This way $f_t(X_i)$ and w can be rewritten as w_j. This looks like:

$$L(\varphi)^{(t)} \approx \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j \ + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T. \tag{8}$$

The derivative of this function (7) with respect to the weight can be taken and set to 0. This way the optimal weight w_j of leaf j can be computed by w_j^* (9). By substituting the optimal w_j in the $L(\varphi)^{(t)}$, the scores can be estimated for all leaves (10). The formulas are shown below:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \tag{9}$$

$$L(\varphi)^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \tag{10}$$

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. *In Proceedings* *of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

This formula can be used as a scoring function to measure the quality of a tree structure q. To give a more intuitive insight, if you put in the derivative of g_i you get negative residual, which simply gives the RSS in the numerator. If you also put in the second derivative of h_i we see that all $h_i = 1$. So, the denominator is just the number of residuals and λ. This gives:

$$L(\varphi)^{(t)}(q) = \frac{RSS}{number\ of\ Residuals\ +\ \lambda} + \gamma T. \tag{11}$$

The highest score becomes the split. However, it is hard to calculate the score for all tree structures q. This strengthened the use of a greedy algorithm that starts from a single leaf and keeps adding branches. Next, assume that I_L and I_R are the sets of left and right nodes after the split. Letting $I = I_L \cup I_R$, the scores for the left and right nodes can be calculated. The loss reduction after the split, without any constants, is given by:

$$L_{split} = \frac{(\sum_{i \in IL} g_i)^2}{\sum_{i \in IL} h_i + \lambda} + \frac{(\sum_{i \in IR} g_i)^2}{\sum_{i \in IR} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}. \tag{12}$$

The L_{split} gets calculated by adding up the individual scores of the left and right node minus the initial split node. The highest L_{split} is the highest score for this split, and therefore, becomes the split. In the end, to prune the tree, regularization parameter γ can be set to make sure that the tree does not make bad splits. If $L_{split} - \gamma$ is positive, the split remains in the tree and there is no pruning. Otherwise, the split gets pruned and the next branch will be checked by this rule. After all the trees have been made, the model starts with an initial prediction, and then adds all outputs of the trees times a learning rate.

The XGB model has some hyperparameters that can be tuned such that the model does not overfit the train data. Just as with GB, XGB adds an interaction with a learning rate to every tree. For XGB, the default for the learning rate ε is 0.3. λ and γ are regularization terms that help against overfitting. The default values for λ and γ are both 0. Lastly, the number of terminal nodes T helps counter uninformative splits. The default number for terminal nodes is 6. Above is the algorithm of XGB explained in a mathematical manner. As mentioned before, in Appendix A, a more intuitive reasoning behind the boosting algorithm can be found.

4.2.4 Support vector regression (SVR)

Support vector machine (SVM) is popular for classification problems, but support vectors can also be used for regression problems. This part starts with a small explanation of SVM before diving in SVR. SVM creates nonlinear boundaries, or hyperplanes, by constructing a linear boundary in a transformed, higher-dimensional space where points can be classified into their respective classes. The optimal position of the hyperplane is determined by maximizing the margins between the hyperplane and the support vectors, where the support vectors are the closest data points of both groups to the hyperplane. SVM is complex but can achieve great predictive power. The use of support vectors to create a hyperplane has as main benefit the kernel trick, which can capture non-linearities through various kernel functions (Drucker, Burges, Kaufman, Smola & Vapnik., 1996). The hyperplane is a separating line between two data classes in SVM. But in SVR, the hyperplane is the line that will predict the continuous output. In contrast with decision trees, models that use support vectors are based on distance, it is necessary to standardize the predictors as part of the data preparation. If there is no optimal function that separates all observations perfectly, then SVR allows for misclassification for better results in the long term. The allowing of misclassifications is called the soft margin (Cortes & Vapnik, 1995).

SVR is a powerful tool for regression tasks where there are complex relationships between the predictors and the outcome variable since the model has the ability to capture non-linearities through kernel functions. SVR is an algorithm that finds a hyperplane that best fits data points in a continuous space while minimizing the prediction error. SVR can perform nonlinear regressions by transforming the input variables into a higher-dimensional space, and then perform linear regressions in that space. So, the first step is to start with a linear regression function that looks like this:

$$f(x) = w^T(x) + b, \tag{13}$$

where b represents the bias or intercept term of the equation. x represents the regressor values of a data point. The vector w can be seen as the regression coefficient vector, which the SVR model wants to minimize. The primary objective of SVR is to find the optimal hyperplane that remains as close as possible to all data points, while also minimizing the error of the hyperplane. The orientation and slope of the hyperplane get determined by the vector w. Cortes & Vapnik (1995) came up with a ε-insensitive loss function. They try to find $f(x)$ that has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time is as flat as possible. In other words, errors less than ε are fine (Smola & Schölkopf, 2004). To allow for misclassification introduce slack variables ξ_i, ξ^* was introduced to SVM to cope with infeasible constraints of the optimization problem. In Figure 2 you can see how the soft margin looks. Data points that lay in the 'tube' are not considered in the loss function, where ξ denotes the soft margin. The data points outside the gray area get penalized.

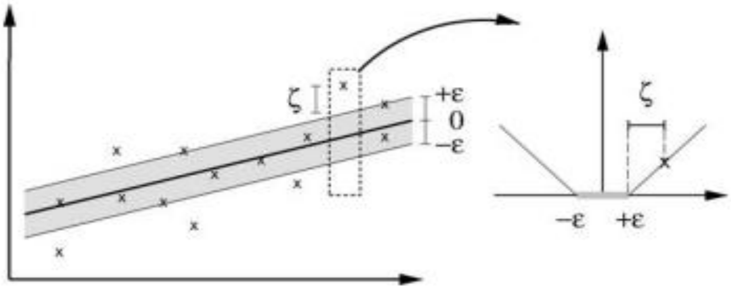


Figure 2. The soft margin loss setting for a linear SVM (from Schölkopf & Smola, 2002)

Slack variables are used as upper and lower constraints to get to the next SVR formulation:

$$\text{Minimize} \qquad \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi^*)$$

$$\text{With subject to} \qquad \begin{cases} y_i - w_i x_i - b| \leq \varepsilon + |\xi_i| \\ \xi_i, \xi^* \geq 0 \end{cases}. \tag{14}$$

The constant $C > 0$ determines the trade-off between the flatness and the amount up to which deviations larger than ε are tolerated. This corresponds to dealing with a so-called ε-insensitive loss function. The higher C the more the function gets penalized. However, the higher C more misclassification is allowed which prevents overfitting. The hyperparameter C needs to be tuned. To obtain the most optimal hyperplane the kernel trick will be used since computations in a high-dimensional space are difficult. The kernel trick works with a dot product, which gives the high-dimensional coordinates of the data. The key idea is to construct a Lagrange function from the objective function and the corresponding constraints, by introducing dual formulation. This way the kernel trick can be applied. The optimization looks as follows:

$$\text{Maximize} \qquad \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(a_i - a_i^*)(a_j - a_j^*)K(x_i - x_j^*) \\ -\varepsilon\sum_{i=1}^{l}(a_i - a_i^*) + \sum_{i=1}^{l} y_i(a_i - a_i^*) \end{cases}$$

$$\text{With subject to} \qquad \begin{cases} \sum_{i=1}^{l}(a_i - a_i^*) = 0 \\ a_i, a_i^* \in [0, C] \end{cases}. \tag{15}$$

Here, a represents the lagrange multipliers (Schölkopf & Smola, 2002). $K(x_i - x_j^*)$ is the kernel function. There are many kernel functions, the most popular ones are the linear, polynomial and radial kernels (Gunn, 1998). The choice for a kernel will be tuned. In the end, the SVR function that makes the prediction is:

$$f(x) \; = \sum_{i=1}^{l} (a_i - a_i^*) K(x_i - x) + \; b. \tag{16}$$

4.2.5 Generalized linear regression model with gamma distribution

Just like a simple OLS regression, a Generalized Linear Model (GLM) is also an ML method. However, a GLM extends the concept of linear regression by introducing a link function that connects the linear model to the response variable. An OLS regression states the error distribution of response variables as a normal distribution, while this might not always be the best fit. A GLM generalizes this assumption by allowing the linear model to have another distribution with the help of linking functions. Therefore, GLM allows the variance to be a function of its predicted value, providing flexibility in modeling heteroscedasticity. According to Nelder & Wedderburn (1972), a GLM is described by three core characteristics. In the first place, a GLM has a random component, which defines the probability distribution of the response variable, such as the Normal, Poisson, or Gamma distribution. Second, a GLM has a systematic component, which describes a linear combination of the explanatory variables in the model. Lastly, The GLM has a linking function. The linking function combines the random component with the systematic component. Among other things, a simple OLS regression is a special case GLM.

In predictive modeling, the gamma distribution is frequently employed in the context of GLM models when the response variable follows a skewed distribution with positive support. According to Laudagé, Desmettre, & Wenzel (2019), a continuous random variable X is said to have a gamma distribution with parameters a > 0 and B > 0, shown as $X \sim$Gamma(a, B), if $x \geq 0$ and its density function is as follows:

$$\hat{f}_S(x_S) = \frac{B^a x^{a-1} e^{-Bx}}{\Gamma(a)}. \tag{17}$$

If $X \sim$Gamma(a, B) then the mean and variance are as follows:

$$\mathrm{E}(X) = \frac{a}{\mathrm{B}}, \tag{18}$$

$$Var(X) = \frac{a}{\mathrm{B}^2}. \tag{19}$$

4.2.6 Permuted importance plot

Interpreting ML models can be difficult. In general, linear models and decision trees are well interpretable. On the other hand, ensemble methods and SVR are so-called black box models that mainly focus on achieving high accuracy rather than interpretability. To help interpret these black-box models, a Permuted importance plot by Breiman (2001) gets used. Permutation-based variable importance involves measuring the increase in a model's prediction error after the feature has been permuted. The association between the predictor and the target variable gets disrupted if the predictor values get shuffled. More accurate results can be obtained if the shuffling gets repeated a few times. Then the average of the errors gets used. A larger increase in the prediction error indicates a more important predictor. This is because the model relies on the feature for the prediction. Predictors that seem unimportant can be removed from the model to see if the accuracy increases.

4.2.7 Partial dependence plot

Partial dependence plots are also a useful tool to help interpret black box models. The concept of partial dependence allows you to understand the impact of a variable on the model's prediction output, considering the variable's contribution while keeping other variables constant (Friedman, 2001). A partial dependence plot helps identify whether the relationship between the variable and the output is linear, monotonic, or more complex. The function $\hat{f}_S$ is estimated by calculating averages in the training data, and looks as follows:

$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}\left(x_S, x_C^{(i)}\right). \tag{20}$$

Here, x_S are the predictors that the partial dependence plots will plot. $x_C^{(i)}$ are the remaining predictors used in the ML models, where n is the number of predictors (bradleyboek). A line will be plotted through all the averaged predictions and the corresponding unique values x_S. Partial Dependence plots mostly capture the relationship between one or two predictors with the outcome variable. This makes the plots 2-dimensional, and therefore, easy to interpret.

4.3 Model evaluation

The predictive power of ML models can be evaluated by accuracy or errors. Since in this research regressions get used, it is impossible to calculate the accuracy of the models. Therefore, the performance of a regression model gets evaluated on the error of predictions. Errors are interesting because they show how close the predictions were to the expected values. This paper will use the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Adjusted R Squared.

4.3.1 Root Mean Squared Error

The mean squared error (MSE) calculates the average of the squared errors across all samples. This means that the MSE calculates the difference between the predicted and actual values, squares the results to lose negative values, and then takes the average. The root mean squared error (RMSE) adds a step and takes the square root of the MSE. By adding this step, the RMSE provides an error measure in the same unit as the output variable, which makes it more interpretable. The RSME gets calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}{N}}. \tag{21}$$

Here, in the numerator is the RSS, which is explained in 4.2.1., and in the numerator is N, which is the total number of observations. A lower RMSE indicates a better fit, and therefore a better accuracy than a higher RMSE.

4.3.2 Mean absolute error

The nean absolute error (MAE) calculates the average of the absolute errors. In contrast with the RMSE, the MAE loses negative values by taking the absolute value of the difference between the predicted and actual values. The MAE also provides an error measure in the same unit as the output variable by not squaring the errors. The MAE gets calculated as follows:

$$MAE = \frac{\sum_{j=1}^{N}\left|y_j - \hat{y}_j\right|}{N}. \tag{22}$$

Here, in the numerator, you see the absolute value of the sum of the residuals, and in the numerator is N. Just as with the RMSE, a lower MAE indicates a better fit.

4.3.3 Akaike information criterion

Akaike information criterion (AIC) serves as a model selection criterion, enabling the assessment of various models' relative quality while considering the trade-off between model fit and complexity. Here, the number of features penalizes the AIC-score. The AIC formula used in this research is as follows (Panchal, Ganatra, Kosta & Panchal, 2010):

$$AIC \; = \; n * log\left(\frac{RSS}{n}\right) + \; 2 * k. \tag{23}$$

In the formula, n represents the sample size, RSS represents the residual sum of squared errors from the model, and k represents the number of predictors in the model. Originally, the AIC gets calculated with the help of the log of the likelihood of the model, but in the case of tree-based methods, this can be hard. Therefore, the above function is used to criticize the models. The AIC would require a bias-adjustment if the ratio of $n/k > 40$. The model with the lowest AIC is the best.

5. Results

In this Chapter, the results of all severity models will be shown. First, the MS Amlin GLM model gets explained to see what accuracy to strive for. Secondly, the decision tree gets interpreted to help understand how ML models learn. After that, the results of the RF, XGB, and SVR will be displayed to show that a more powerful model should improve the prediction power of the easily interpretable decision tree. Later, with the help of importance plots and partial dependence plots, some of the models with low errors get interpreted to give insights into their predictions. In the end, when all the results are displayed they get compared to the GLM model of MS Amlin. For all models, the data got split 70-30 for training and testing respectively. For SVR models, the data is transformed, as mentioned in Chapter 3, and then it gets split. For all models, the values of the hyperparameters will be mentioned in the text, but there will also be a table in Appendix B.

5.1 MS Amlin GLM model

Although the MS Amlin pricing team had a lot of variables available (see Chapter 3), the pricing team decided to fit three variables for the MS Amlin severity model. The following variables: vessel type, vessel value, and accounted year, play a big role in the GLM model. By one-hot encoding of the vessel type variable, the total of predictors eventually became 17. The accounted year variable has been fitted as a one-way factor in the GLM model to rebase the GLM fit for any time effects, such as inflation. Further, the pricing team chose not to add extra variables, due to potential overfitting problems. The pricing team expected that some variables such as gross, deadweight, net tonnage, P&I club, and Classification society could contribute to the GLM model. However, some of these variables missed data points, or there was too little confidence, while the fear of overfitting was extant. Overall, the GLM model with a gamma distribution has an RMSE of 1.482.068 and an MAE of 487.798. The pricing team believes that deductibles and excess could be major factors in the model despite not fitting them in the model. The GLM model mostly shows that if the vessel value rises, the expected claim value rises, too. Further, among the vessel types, the LNG (liquefied natural gas carrier), passenger vessel, and tanker, the claim values rise the most, while for general cargo, LPG (Liquefied petroleumgas vessel), reefer, Ro-Ro, and Vehicle vessels the expected claim value would be lower.

5.2 Decision tree

A decision tree is one of the easier ML models to understand. For that reason, this research starts by explaining a relatively simpler model. So, the main objective of this method is not to strive for the highest prediction power but to display an interpretable model. The trick with creating good decision trees is to follow two steps. In the first step, the complexity parameter (cp) gets set to 0, which makes the tree as big as possible. This tree contains over 250 splits, making it difficult to interpret. Despite the overfitting, the tree achieves an RMSE of 1.587.982 and an MAE of 528.258, which is worse than the GLM. However, the tree requires pruning to reduce its size and improve accuracy and interpretability. In step two, the pruning is achieved through increasing the cp value. 10-fold cross validation is used to find the best cp. Normally, to get the best accuracy, it is suggested to pick the cp value with the lowest cross validated error. However, since the decision tree is not a strong predictor compared to the models that come later, a cp gets chosen that is easy to interpret. Therefore, a cp value of 0.0068416 has been chosen since it got 9 splits, whereas if the cp would be higher the tree would get 3 or less splits. In Figure 4, you can see the tree where the cp is set to 0068416, which is nicely interpretable.

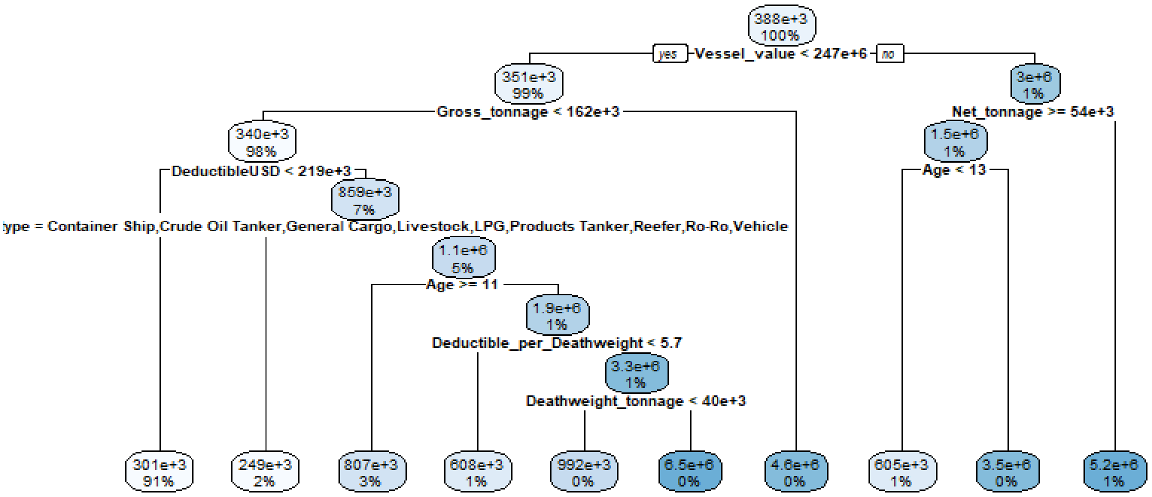


Figure 4. Decision tree with 9 splits.

In Figure 4, you can see that every node contains a prediction per node. Below the predictions are the proportions of the data in that node. It starts at 100% and then drops when the data gets split more often. The leaf nodes show the prediction if the data follows that specific split path. Normally, it is preferred that a decision tree tries to split the data in half by each split, but since the distribution of the dependent variable is skewed, the splits are also skewed. Around 2.5% of the data gets predicted as a claim value of over 1 million US dollars, which is not bad since only 4% of all claims are above 1 million. The RMSE of this tree is 1.546.868, and the MAE is 497.981, which is a big improvement over the unpruned tree. Since a decision tree splits based on the variable that gives the lowest RSS, the first splits can be seen as the most important variable. Therefore, the first node says that ‘vessel value’ is the most important variable. It states that, if the vessel value is more than 247 million US dollars the data follows the right path, which leads to higher predictions. Hence, a higher vessel value leads to a higher predicted claim value. ‘Gross’ and ‘net tonnage’ are also important. For these variables, it states that if the net or gross tonnage gets higher, the predicted claim gets higher. The sixth split is also quite interesting. From the GLM model it was seen that most of the vessel types in split 6 lower the expected claim costs, which exactly happens in the decision tree.

5.3 Random Forest, XG Boosting, and Support Vector Regression

5.3.1 Random Forest

Generally, an RF can reach higher predictive performance than a decision tree since it uses multiple different decision trees. Initially, a default random forest is created to assess its predictive performance. However, this default random forest consists of 500 trees, an mtry of 3, and a max_nodes of 10. The numbers of trees and mtry are the default values, but the number of max_nodes is increased from 5 to 10 since decision trees tend to create either 3 or 10 terminal nodes. So, a default value of 5 for max_node would only create small trees. The default random forest yields an RMSE of 1.492.695 and an MAE of 481.676, which is already an improvement compared to the individual decision tree. However, it is possible to enhance the models' accuracy by tuning the random forest. First, the number of trees gets tuned, and then the mtry. The tuning is visualized in Figures B1 and B2 in Appendix B. Out of these Figures, you can see that 495 is the optimal number of trees since the error is lowest there. After that, when using 495 trees, it is best to pick mtry is 2. Now, a new random forest model with tuned hyperparameters gets trained and tested. Here, the RMSE decreased to 1.482.810, and the MAE increased to 482.956. All in all, tuning the random forest improves the model a little bit. But that is not weird since random forests have a strong default setting.

5.3.2 Extreme gradient boosting

For XGB, one tree gets created, and then this tree gets boosted over and over until there is no improvement. It keeps adjusting the predictions based on its errors, and this way it learns from its former tree. Through a learning rate, the improvement per new tree gets regulated. The plan to train the best XGB model follows three steps. In the first step, a default XGB gets trained. In the second step, eta (the learning rate) gets tuned with the help of 10-fold cross validation. When the eta is tuned, that eta value gets entered in the new XGB model. In the last step, this model will tune max depths and max leaves with the help of 10-fold cross validation. The default XGB has an RMSE of 1.510.645 and an MAE of 450.043. Figure B3 in Appendix B shows that eta is 0,01, giving the lowest cross validated test error. Although the standard deviation is also the highest, it still has been chosen to pick eta as 0,01. This new XGB with the tuned eta has an RMSE of 1.505.469 and an MAE of 427.004, which is a big improvement compared to the default XGB. Figure B4 in Appendix B shows that the combination of max depth is 3 and max leaves is 15, giving the lowest cross validated test error. Here, both the max depth and max leaves were tuned on values between 3 and 21 with steps of 3. The newly tuned XGB has an RMSE of 1.469.762 and an MAE of 465.373. Now, the RMSE is way lower, but the MAE is higher than the default. Still, based on both errors, the tuned XGB outperforms the RF.

5.3.3 Support vector regression

SVR can be a strong ML model. Unlike the models above, SVR is not a tree-based method. However, SVR possesses other skills, such as the kernel trick, which enables the SVR to operate in higher dimensions. This way, SVR can capture nonlinear relations as well as linear relationships. In this research the following kernels will be used: the linear kernel, the radial kernel, and the polynomial kernel. The linear kernel is strongest if the data has linear relationships, the radial kernel is strongest when the data is more complex and nonlinear, and the polynomial kernel is strongest when the data contains nonlinear relationships. First, the linear kernel gets trained. For the linear kernel, only one hyperparameter needs to be tuned, which is the cost parameter. The cost can contain values between 0,01 and 100. The higher the cost, the more the function gets penalized. So, if the cost is high, the penalty for misclassification is high, and the margin becomes smaller, which could lead to overfitting. A cost value that is too small might underfit. The default SVR for a linear kernel with a cost of 1 gives the best model since tuning does not improve the model further. The SVR with linear kernel has an RMSE of 1.515.800 and an MAE of 385.939. The SVR with radial kernel gets trained next. For this SVR, besides the cost, the gamma parameter can also be tuned. The gamma parameter defines how far the influence of a single training example reaches. A higher gamma will consider only points close to the hyperplane, while a lower gamma will consider wider data points. Gamma is one divided by the dimension of the data by default, which is 0,04. The tuned SVR with a radial kernel has a cost of 5 and a gamma of 0,25. The RMSE and MAE of this SVR are 1.502.523 and 400.002. An SVR with a polynomial kernel is the last SVR that gets trained. For the SVR with a polynomial kernel, the cost and degree hyperparameters can be tuned. The degree parameter represents the degree of the polynomial, which by default is 3. After tuning the SVR with a polynomial kernel, the cost is 0.1, and the degree is 2. Here, the RMSE is 1.502.530, and the MAE is 391.762. Overall, the results of all SVRs with different kernels are quite alike. What stands out is that for all SVR models, the MAE is very low compared to earlier models. On the other hand, the RMSE gets not below 1,5 million.

5.4 Compare the models

All models get compared based on their RMSE, MAE, and AIC. For all these performance measurement methods the lower the value the better the performance. For each of the prediction models the best (tuned) models get compared. In table 1, you can see the performance of all the models on the test data.

Table 1. Performance of all ML methods

	RMSE	**MAE**	**AIC**
Decision tree	1.546.860	497.982	125.779
RF	1.482.810	482.956	125.662
XGB	**1.469.762**	465.373	**125.584**
SVR lin	1.515.802	**385.961**	125.885
SVR rad	1.502.523	400.002	125.807
SVR pol	1.502.530	391.762	125.807
GLM	1.482.068	487.799	125.680

Overall, all models are quite similar in most performance metrics. The AIC for all models is between 125.584 and 125.885, from the XGB and SVR with linear kernel respectively. The AIC of the GLM and the other SVR models are also higher than the tree-based methods, except for the decision tree. That seems logical since the AIC penalizes for using more variables, and the GLM and SVR models have to one-hot encode their categorical variables, which increases the number of variables. The RMSE and MAE were also mentioned in the above paragraphs, and the GLM errors are the errors to strive for. In that case, the XGB model is the only model that has both a lower RMSE and MAE than the GLM. Therefore, based on the performance metrics in Table 1, it can be stated that the XGB is the most accurate prediction model. The RF is extremely similar to the GLM model since their metrics are close, but the RF has a better MAE and AIC. All SVR models have low MAEs. In the case of the SVR with linear kernel, the MAE is even more than 100,000 US dollars lower than the GLMs. However, if you look at Figure 5, it is clear that, although the MAE is low for the SVR with a linear kernel, it predicts the bigger claim values poorly. The MAE of the SVR with a linear kernel is low since most of its predictions are quite low compared to the XGB and actual claim values. So, on average, these predictions are fine, but if a larger weight gets added to outlier predictions, the error becomes bigger. This is what happens within the RMSE. Because every residual gets squared, it penalizes bigger differences between actual and predicted values, increasing the RMSE. Therefore, having one low-performance metric does not make the model directly a good predictor. It is good to see that both the XGB and RF perform around the same or better than the GLM model. Maybe with more time, these models could take more economic factors into account to improve their predictions even more.

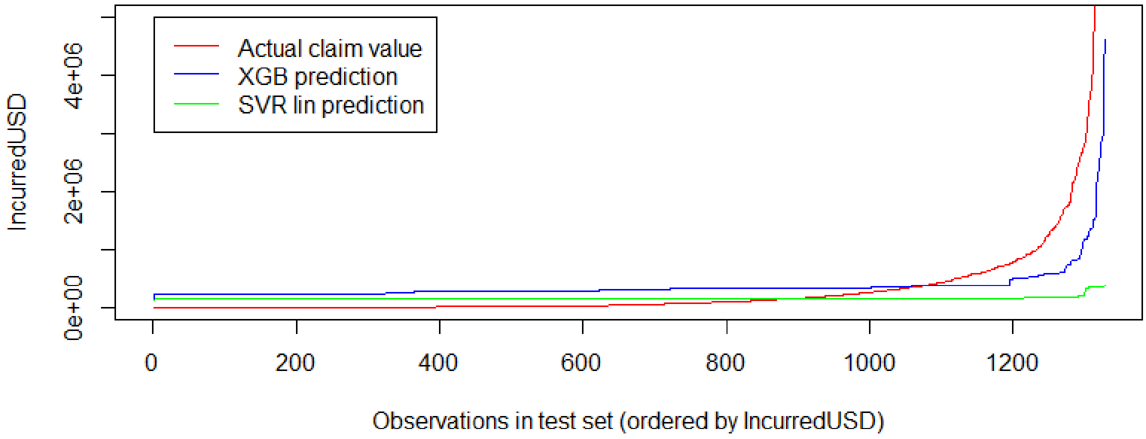


Figure 5. Comparison between actual claim values and the predicted values of the XGB and SVR with a linear kernel. Note that the observations of the test set are ordered such that all lines become clearer to follow.

5.5 Interpretation

This part is intended to help interpret the results of the XGB and RF models. Based on the performance metrics from above, these models should predict the severity costs the best. Importance plots help compute what variables are important to the model's predictions. It gives either their importance based on gain or loss in RMSE. The importance variables of both models get checked and compared. After that, The partial dependence plots get displayed for the most important variables. Thanks to the partial dependence plots the relationships between the dependent and important independent variables become clear.

5.5.1 Importance plots

In Figure 6, you can see the variable importance plot for the XGB model. This important plot is based on the gain of the boosted trees in the XGB model. Gain is the relative contribution of the corresponding variables to the model calculated by taking each variables contribution for each tree in the model. A higher score suggests the variable is more important in the boosted tree's prediction. According to Figure 6, ExcessUSD is the most important variable. Excess insurance is an extra insurance policy purchased to provide coverage for losses that exceed the limits of the primary insurance. It could be expected that if Excess insurance increases, the insured would be more afraid of a higher claim value. Therefore, it makes sense that this variable is important. Vessel value is also important according to the XGB importance plot, which seems logical. For higher value vessels, the claims can get higher. The vessel years variable is surprisingly quite important for the XGB model. It can be said that the length of the insurance protection has a big effect on claim value.

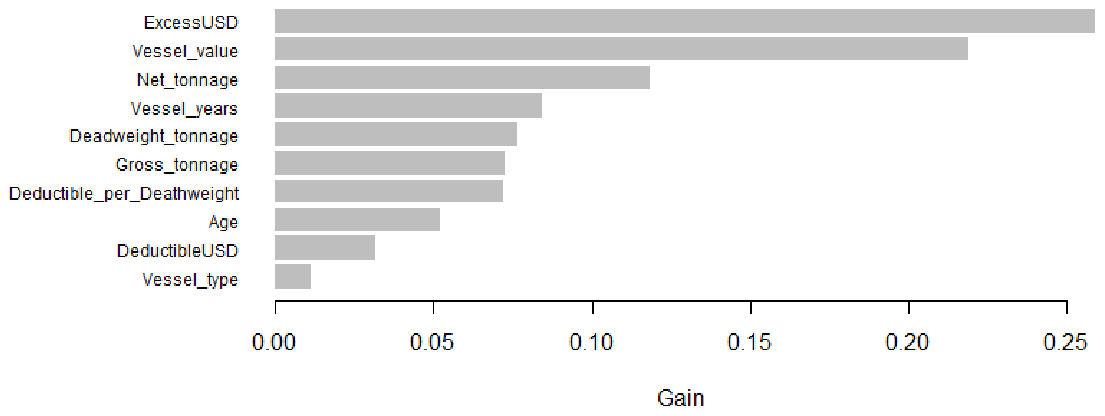


Figure 6. XGB variables importance plot (based on Gain).

In Figure 7, you can see the RF permuted variable importance plot. Here, the variable importance involves measuring the increase in the RFs RMSE after the variable has been permuted. A larger increase in the prediction error indicates a more important predictor because the model relies on the variable for the prediction. Therefore, without either the ExcessUSD or gross tonnage variables, the RMSE rises with almost 50.000. For ExcessUSD, it is the second time it is the most important variable. Gross tonnage is also in the XGB importance plot, the fourth important variable. For all weight variables, it seems logical that the more the vessel weighs, the more the claim could be. Again, the vessel value appears important for the RF, as well as for the XGB model. Something else that is quite remarkable, is that vessel type and age are not as important to the model's predictions compared to other variables. Age and vessel type seem like variables that could influence the claim value. At some time, when a vessel gets older, it would make sense that the costs of making a vessel ready to sail would increase. Further, it seems feasible to believe that some different vessel types have higher risks, and therefore, higher average claims costs. However, they are just not that important to the prediction models. Vessel type might be a bit biased since tree-based methods can handle categorical data but when a categorical variable has too many categories, it can lead to overfitting and high complexity. The vessel type variable has 15 levels, which could be too big a choice for a decision tree to split on. Lastly, P&I club is the least important variable in the predicting models of the XGB and RF. Excluding this variable might increase the predictive performance of the ML models.

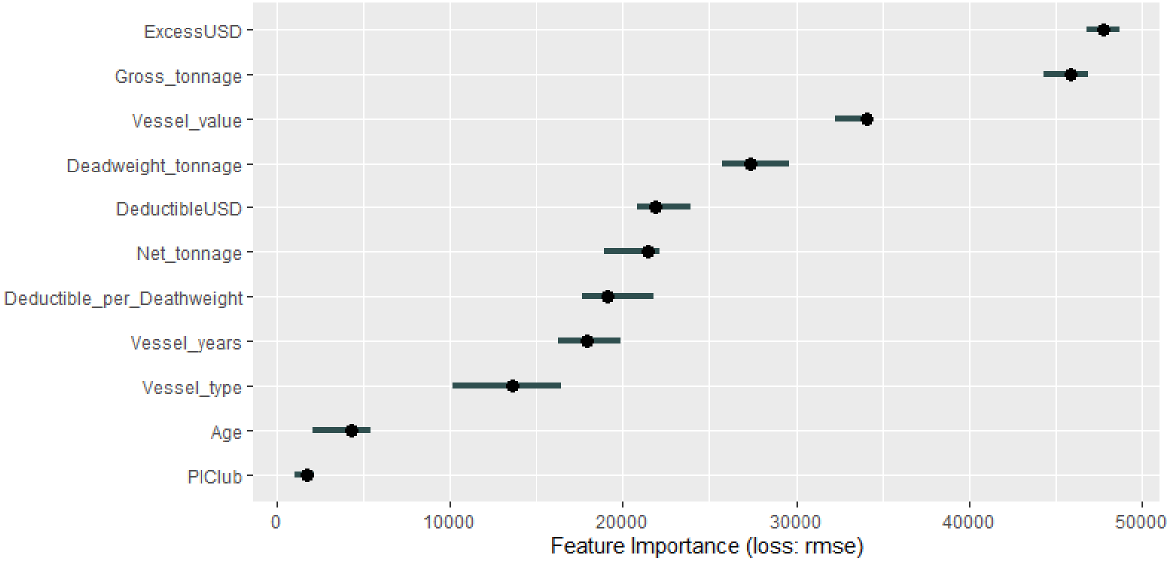


Figure 7. RF permuted variable importance plot (based on loss in RMSE).

5.5.2 Partial dependence plots

Partial dependence plots illustrate the individual impact of a variable on the model's output. It has been chosen to only create two-dimensional partial dependence plots since these plots are easier to interpret. In Figure 8, you can see the relationships between the six most important variables and the target variable, which is the IncurredUSD variable. At the top left, ExcessUSD is displayed on the x-axis. When ExcessUSD rises, the expected claim value rises softly until ExcessUSD becomes larger than around 18 million US dollars, which makes it a strong positive relationship, resulting in the predicted claim value rising stronger. At the top right, you can see that the vessel value and IncurredUSD follow a positive nonlinear relationship. Overall, if the vessel value increases, the predicted claim value increases, just as expected. For the gross tonnage variable, the relationship is a lot like the ExcessUSD variable. Generally, there is a small positive relationship until gross tonnage exceeds 150.000 tons, and then the relation with the predicted claim value is way stronger. For the deadweight tonnage variable, there is a positive nonlinear relationship in the form of a stair. This relationship gets stronger in two phases. In the first phase, there is a quite strong positive relationship until the deadweight is around 70.000 tonnage. Then there is a weak positive relationship until deadweight is around 170.000 tonnage, which changes in a strong effect again. Not long after 170.000 tonnages, there is no effect anymore. At the bottom left, you can see that the net tonnage variable has an overall strong positive relationship with the claim value. A net tonnage higher than 60.000 does not have an effect anymore. In the bottom right, the vessel year variable is displayed. Most insurance policies are one year, but for some, it is a few months more or less. All values lower than 1 represent policies that are smaller than a year, and values bigger than 1 represent longer policies. For vessel year values that are at least 1, there is no clear effect between the vessel year and predicted claim value. IncurredUSD rises slightly, the more it takes a value of two, but this effect is minimal. For vessel year values smaller than 1, there is a strong effect on the predicted claim value. There is a nonlinear strong positive effect between vessel year value 0-0,3, while after that, there is mostly a strong negative effect until the vessel year value of 1 again.

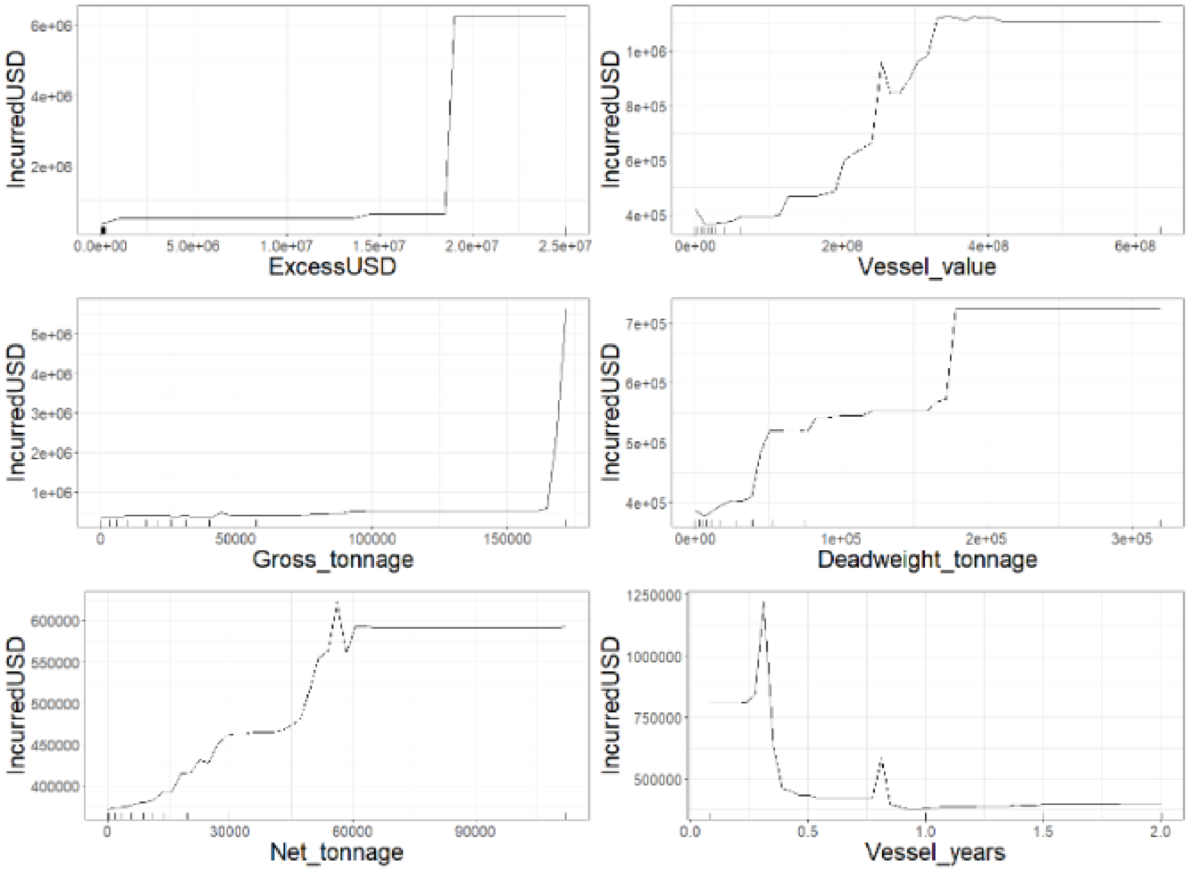


Figure 8. Partial dependence plots of the most important variables according to the XGB and RF prediction models.

6. Conclusion and discussion

In this chapter, the conclusion to the findings will be defined. Here, the main research questions and sub-questions will be answered based on both the literature and the results of this research. After answering all the research questions, the content will be given on implications and recommendations on managerial and academic bases. In this part, the main focus is on how ML can benefit MS Amlin. In the end, the limitations of the research will be given.

6.1 Conclusion

The previous chapters aimed to determine the most effective method for predicting severity loss costs for the MS Amlin 'Ocean Hull' pricing division. Multiple ML methods were used to test whether they could outperform the gamma distributed GLM model from MS Amlin. Furthermore, importance and partial dependence plots were used to gain understanding regarding the most important factors influencing the height of the claim, and their respective relationships. This all to answer the following research question:
Which models can best model severity loss costs to help set insurance premium prices?

First, the focus will be on the sub questions that help answering the research questions:

i) What Machine Learning model gives the best predictions?
To find out what ML model gives the best predictions, three performance metrics have been used to compare all models. First, the MAE describes the average prediction error of the ML model. It does this by taking the absolute value of the residuals. The next metric is the RMSE, the RMSE describes more about the standard deviation of the ML model. By taking the square of the residuals, it penalizes bigger differences between actual and predicted values. Lastly, AIC tells how well the model fits the data by penalizing the number of predictors. For all of these metrics, the lower their value, the better. By checking these metrics, the XGB model showed the lowest RMSE and AIC, followed by the RF. The SVR models give the lowest MAE, but their RMSEs are relatively high. After checking the predictions of the SVR models, it showed that these models only predict well for lower claim values. However, a severity model should also account for claims that can get high claim value. Therefore, the SVR models are not the best predictors. The XGB model has the best mix of low error values. Therefore, the XGB is the best prediction model. After the XGB, the RF also shows good performance metrics, which makes it a good second.

ii) What predictors are most important in predicting severity loss costs?

Importance plots can help interpret the ML models by showing the value a predictor has for the model. Since the XGB and RF were the best-performing models, two important plots were constructed. In the first plot, an importance plot based on the gain for the XGB model was created, where the gain is the relative contribution. The higher the gain, the more important the predictor. From this plot, ExcessUSD, vessel value, net tonnage, vessel years, and deadweight tonnage were the five most important variables for the XGB. P&I club and vessel type were the least important variables. In the second plot, a permuted importance plot based on an increase in RMSE was created for the RF. So, if the predictor is not in the data, then the RFs RMSE would increase. Here, the five most important variables were ExcessUSD, gross tonnage, vessel value, deadweight tonnage, and DeductibleUSD. If ExcessUSD were excluded from the data, the RMSE of the RF would increase by almost 50.000. Again, the P&I club was not important to the predictions. Based on this finding, it might improve the predictive performance of the ML models if the model got trained without the P&I club variable.

iii) What effects do the most important predictors have on the severity loss costs?

Partial dependence plots were used to illustrate the individual impact of a variable on the model's output. They showed the relationships between the six most important variables and the predicted value of the claims. For the ExcessUSD, vessel value, weight, and vessel year variables were partial dependence plots created. In general, all variables, except the vessel year variable, had positive and nonlinear relationships with the predicted claim value. Overall, if these variables increase, the predicted claim value increases as well. For example, for the ExcessUSD variable, the effects on the predicted claim value were slightly positive, until ExcessUSD became larger than around 18 million. Then, there was a strong positive effect on the claim value. Quick after that strong positive effect, there was no effect any longer. The vessel year variable is different. For vessel year values that are at least 1, there is no clear effect between the vessel year and predicted claim value. IncurredUSD slightly increases, the more it takes a value of two, but this effect is minimal. For vessel year values smaller than 1, there is a strong effect on the predicted claim value. There is a nonlinear strong positive effect between vessel year value 0-0,3, while after that, there is mostly a strong negative effect until the vessel year value of 1 again.

iv) How did the results of Machine Learning Models do compared to the results of the GLM model of MS Amlin?

Initially, the GLM model of MS Amlin for severity modeling fitted only three variables. However, after one-hot encoding of the vessel type predictor, the number of predictors became 17. One of the variables they used in their model was the accounted year variable. This variable was used to rebase the GLM fit for time effects, such as inflation. Due to overfitting, the vessel value was the last fitted predictor, while other variables might also have been expected to influence the claim value. The GLM model with a gamma distribution has an RMSE of 1.482.068 and an MAE of 487.798. If this is compared to the ML performance metrics, it is seen that XGB shows both lower RMSE and MAE. Further, the RF is extremely similar to the GLM model since their metrics are close, but the RF has a better MAE and AIC. Lastly, the SVR models show good MAE, but as discussed before, the RMSE is way too high, which makes them worse prediction models. Overall, the performance metrics of all models in this research are quite close.

To conclude, *which models can best model severity loss costs to help set insurance premium prices?* If the performance metrics are the guidelines, then the XGB model predicts the claim value the best. Also, if the SVR models get excluded from the performance metric table because of their inaccurate predictions, the XGB model has the lowest value for every metric. The RF and GLM are alike, and both show good predictive power. XGB and RF, both tree-based methods, provided accurate predictions, and both models provided an ability to capture nonlinear effects. However, the interpreting of these ML models can get difficult since interpretation methods have to come into play. The importance and partial dependence plot help with that, but in general, simple decision trees and linear regressions are easier to interpret. So, on that note, the GLM method is better at interpreting the results. The predictive power and interpreting will always be a pay-off. Another strength of the MS Amlin GLM model is that it fits the accounted year variable to rebase the GLM fit for time effects, such as inflation. Something that is not taken into account in the ML models. So, the GLM model is adjusted to a more real-life scenario. Nevertheless, I think, the XGB model is the best predictor since its performance metrics are the best, and while it is still being explainable due to interpretation methods. And, if more economic factors, such as inflation, could be added to the model, it might achieve higher predictive power. I think, if the XGB model would account for these factors, it could potentially be the method for a new severity model.

6.2 Managerial implications and recommendations

This thesis can be seen as a test for MS Amlin to see whether it is worth starting working with ML models. Overall, I think both MS Amlin and I are happy with the results. The MS Amlin GLM severity model shows good predictive performance, while some of the ML models do as well. As it is now, It remains questionable whether the XGB model is better than the GLM model. Based on the findings of this research, the XGB model still proves it can achieve high predictive power while being interpretable. For the MS Amlin pricing team, I would not suggest putting all their money and resources into creating a new prediction model with the help of ML. However, it is probably wise to invest some time in ML prediction models. In the long term, these ML prediction models are likely to outperform the GLM models. Especially, when more economic factors get incremented into the ML models to make them more flexible, and thus more accurate for real-life cases. Therefore, I suggest that the MS Amlin pricing team examine the power of ML further, such that they can maybe benefit from these new technologies in the future.

Overall, this research shows a positive application of what is possible with ML models. This research has passed the test to see if it is worth working with ML models. Now, MS Amlin is interested in the range of what is possible with ML models. Predictive modeling is valuable in many divisions. In the first place, ML is advantageous to the pricing division, as discussed before. In the long term ML models could be used in the whole pricing flow. This research focussed on severity modeling (expected claim cost), but ML can be of value for frequency (expected amount of claims) modeling of the pricing prediction plan as well. Further, the pricing division could also implement ML prediction modeling for other products than Ocean Hull.

Financial crime screening is the second division that might benefit from an ML prediction model. The financial crime screening screens all new potential clients of MS Amlin to find out what kind of risk they bring. In the first stage, they check multiple factors to classify potential clients as either low, average, or high risks. For example, some countries are riskier in transporting products. The screening happens in the second stage, where the department has to screen high-risk clients intensely, while low-risk clients require just a small security check. Financial crime screening indicates that they would be interested in an ML model to take over the first stage. The first stage is a classification problem with three classes, which now gets done manually. An ML prediction model could save time and money in this division.

Third, the head of the finance and data department listed many topics for which he believes ML could help the team. The analytics area contains a lot of predictive opportunities. For example, fraud detection or prevention, credit risk detection, and market analysis models that predict trends. In credit risk detection, prediction models could lead to better financial risk management, and for trend predictions, they could lead to better-assessing growth opportunities. ML can quickly help find patterns that would take a lot of time if done manually. Better resource allocation then makes the business more efficient.

The above-mentioned recommendations could also apply for other insurance companies. ML shows good predictive power, so it should be given a try.

6.3 Limitations

Before jumping into any limitations, I want to address that MS Amlin models price premiums by modeling the severity loss cost and the frequency of claims. Together, they become the price flow. So, in this research, only the severity loss cost is modeled. To predict prices, the claims frequency needs to be modeled as well. Therefore, this research modeled just a part of the pricing flow. For future research, one could model both the severity and claim frequency, and compare on premium model instead of their respective model.

6.3.1 Data

The data used for modeling the severity loss cost is data from 2006-2020. In the first place, the intention was for me to gather my own data, such that I could take the last couple of years into account as well, and see if there were changes in the COVID-19 period. However, after months of data preparation there was still too little progress so the plans changed, and I would use the old dataset on which the MS Amlin GLM model was trained. By not altering the data any further the new ML models could be compared to the GLM model. However, some observations missed datapoints, and therefore some predictors were excluded from the model. In this research, the P&I club variable got transformed, where named P&I clubs were transformed to good, average, and bad. This variable did not appear to be important to both the

XGB and RF models. In future research, unimportant variables, such as P&I club, could be removed from the data to achieve higher accuracy.

6.3.2 Economic factors

As discussed before, the XBG model achieves better accuracy than the MS Amlin GLM model. However, the GLM model should be better suited for real, since the MS Amlin pricing team accounts for economic factors in their model, such as inflation. Therefore, a limitation of this research is that most of these economic factors are not implemented in the ML models. The ML models that are displayed in this research lack the economic background factors of real-life cases. These background factors could be from broader external market data. For this reason, I want to be cautious with my recommendations on ML models. In future research, taking more economic attributes in the model should make it fit the real-life better since the ML only trained on client and vessel related data.

6.3.3 Interpretation

A big advantage of the GLM model over more complex ML models is that it is relatively easy interpretable. For every variable it gives a factor that shows the strength and relationship to the dependent variable. In this research, the interpretation of the ML was only done with the help of importance and partial dependence plots. The methods help explain the ML models on a global level. However, interpretation can also be done on a local level, where you can see how a single prediction is formed by a machine learning model. For further research, it might be helpful to look into and compare different methods of local interpretation.

7. References

Akerlof, G. A. (1970). The market for “lemons”: Quality uncertainty and the market mechanism. *The quarterly journal of economics, 84(*3), 488-500.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. *In Journal of physics: conference series* (Vol. 1142, p. 012012). IOP Publishing.

Ang, J. S., & Lai, T. Y. (1987). Insurance premium pricing and ratemaking in competitive insurance and capital asset markets. *Journal of Risk and Insurance*, 767-779.

Azzopardi, M., & Cortis, D. (2013). Implementing automotive telematics for insurance covers of fleets. *Journal of technology management & innovation, 8*(4), 59-67.

Baştanlar, Y., & Özuysal, M. (2014). Introduction to machine learning. *miRNomics: MicroRNA biology and computational analysis*, 105-128.

Beam, A. L., & Kohane, I. S. (2018). Big data and machine learning in health care. *Jama, 319*(13), 1317-1318.

Bertsimas, D., & Orfanoudaki, A. (2021). Pricing algorithmic insurance. *arXiv preprint arXiv:2106.00839*.

Boulesteix, A. L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2*(6), 493-507.

bradleyboekmhke.github (8-5-2023) Chapter 16 Interpretable Machine Learning. bradleyboekmhke.github: https://bradleyboehmke.github.io/HOML/iml.html

Braun, A., & Schreiber, F. (2017). T*he current InsurTech landscape: Business models and disruptive potential* (No. 62). I. VW HSG Schriftenreihe.

Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.

Bühlmann, H. (1980). An economic premium principle. *ASTIN Bulletin: The Journal of the IAA, 11*(1), 52-60.

Chambers, R. G. (1989). Insurability and moral hazard in agricultural insurance markets. *American Journal of Agricultural Economics*, 71(3), 604-616.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Chiappori, P. A., & Salanié, B. (2013). Asymmetric information in insurance markets: Predictions and tests. *Handbook of insurance*, 397-422.

Cohen, A., & Siegelman, P. (2010). Testing for adverse selection in insurance markets. *Journal of Risk and insurance, 77*(1), 39-84.

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

Cortis, D., Debattista, J., Debono, J., & Farrell, M. (2019). InsurTech. *Disrupting finance: FinTech and strategy in the 21st century*, 71-84.

Choudhary, R., & Gianey, H. K. (2017, December). Comprehensive review on supervised machine learning algorithms. In 2017 *International Conference on Machine Learning and Data Science (MLDS)* (pp. 37-43). IEEE.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems, 9.*

Ewold, F. (1991). Insurance and risk. *The Foucault effect: Studies in governmentality*, 197210, 201-202.

Fang, K., Jiang, Y., & Song, M. (2016). Customer profitability forecasting using Big Data analytics: A case study of the insurance industry. *Computers & Industrial Engineering, 101*, 554-564.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems?. *The journal of machine learning research, 15*(1), 3133-3181.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 1189-1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, *38*(4), 367-378.

Garrido, J., Genest, C., & Schulz, J. (2016). Generalized linear models for dependent frequency and severity of insurance claims. Insurance: Mathematics and Economics, 70, 205-215.

Greineder, M., Riasanow, T., Böhm, M., & Krcmar, H. (2020). The generic InsurTech ecosystem and its strategic implications for the digital transformation of the insurance industry. *40 Years EMISA 2019*.

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

Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report, 14*(1), 5-16.

Gupta, M. FUTURISTIC ROLE OF MACHINE LEARNING: EXPLORING DOMAINS.

Hanafy, M., & Ming, R. (2021). Machine learning approaches for auto insurance big data. *Risks, 9*(2), 42.

Harrison, G. W. (1989). Theory and misbehavior of first-price auctions. *The American Economic Review*, 749-762.

Holt, C. A., & Laury, S. K. (2002). Risk aversion and incentive effects. *American economic review*, 92(5), 1644-1655.

Kaushik, K., Bhardwaj, A., Dwivedi, A. D., & Singh, R. (2022). Machine learning-based regression framework to predict health insurance premiums. *International Journal of Environmental Research and Public Health, 19*(13), 7898.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

Laeven, R. J., & Goovaerts, M. J. (2008). Premium calculation and insurance pricing. *Encyclopedia of quantitative risk analysis and assessment*, 3, 1302-1314.

Luan, C. (2001). Insurance premium calculations with anticipated utility theory. *ASTIN Bulletin: The Journal of the IAA, 31*(1), 23-35.

Mahesh, B. (2020). Machine learning algorithms-a review. International *Journal of Science and Research (IJSR).[Internet],* 9, 381-386.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics: *A Journal of the Chemometrics Society, 18*(6), 275-285.

Newhouse, J. P. (1996). Reimbursing health plans and health providers: efficiency in production versus selection. *Journal of economic literature, 34*(3), 1236-1263.

Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2010). Searching most efficient neural network architecture using Akaike's information criterion (AIC). *International Journal of Computer Applications, 1*(5), 41-44.

Paul, L. C., Suman, A. A., & Sultan, N. (2013). Methodological analysis of principal component analysis (PCA) method. *International Journal of Computational Engineering & Management, 16*(2), 32-38.

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery, 9*(3), e1301.

Quan, Z., & Valdez, E. A. (2018). Predictive analytics of insurance claims using multivariate decision trees. *Dependence Modeling, 6*(1), 377-407.

Rawat, S., Rawat, A., Kumar, D., & Sabitha, A. S. (2021). Application of machine learning and data visualization techniques for decision support in the insurance sector. *International Journal of Information Management Data Insights, 1(*2), 100012.

Renshaw, A. E. (1994). Modelling the claims process in the presence of covariates. *ASTIN Bulletin: the Journal of the IAA, 24*(2), 265-285.

Ringnér, M. (2008). What is principal component analysis?. *Nature biotechnology, 26*(3), 303-304.

Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy, 82*(1), 34-55.

Rothschild, M., & Stiglitz, J. (1978). Equilibrium in competitive insurance markets: An essay on the economics of imperfect information. *Uncertainty in economics* (pp. 257-280). Academic Press.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Savage, L. J. (1972). *The foundations of statistics*. Courier Corporation.

Smart, M. (2000). Competitive insurance markets with two unobservables. *International Economic Review, 41*(1), 153-170.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14, 199-222.

Spence, M., & Zeckhauser, R. (1978). Insurance, information, and individual action. I*n Uncertainty in Economics* (pp. 333-343). Academic Press.

Staudt, Y., & Wagner, J. (2021). Assessing the performance of random forests for modeling claim severity in collision car insurance. *Risks, 9*(3), 53.

Subudhi, S., & Panigrahi, S. (2020). Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection. *Journal of King Saud University-Computer and Information Sciences, 32*(5), 568-575.

Su, X., & Bai, M. (2020). Stochastic gradient boosting frequency-severity model of insurance claims. *PloS one, 15*(8), e0238000.

Tsanakas, A., & Desli, E. (2005). Measurement and pricing of risk in insurance markets. *Risk Analysis: An International Journal, 2*5(6), 1653-1668.

Von Neumann, J., & Morgenstern, O. (2007). Theory of games and economic behavior. In *Theory of games and economic behavior.* Princeton university press.

Wakker, P., Thaler, R., & Tversky, A. (1997). Probabilistic insurance. *Journal of Risk and Uncertainty*, 15, 7-28.

Yang, Y., Qian, W., & Zou, H. (2018). Insurance premium prediction via gradient tree-boosted Tweedie compound Poisson models. *Journal of Business & Economic Statistics, 36*(3), 456-470.

Appendix A

Table A1. Summary statistics for all numeric variables.

	Minimun	Median	Mean	Maximum
ExcessUSD	0	125.000	265.382	25.000.000
Gross_Tonnage	0	21.211	26.929	201.384
Deadweight_Tonnage	0	17.157	33.198	400.000
Net_Tonnage	0	8.826	12.740	135.084
Age	0	11	16,33	86
Vessel_value	0	17.789.533	31.140.824	633.333.334
DeductibleUSD	0	100.000	129.058	1.000.000
Vessel_years	0	1	1,06	2
Deductible_per_Deadweight	0	5,30	20,51	3118,24
IncurredUSD	0	45.091	383.329	35.000.000

Table A2. Levels of the catagorical variables.

	Numder of levels	Levels
Vessel_type	15	Bulker
		Chemiacal tanker
		Container ship
		Crude oil tanker
		General Cargo
		Lifestock
		LNG
		LPG
		Ore bulker
		Passenger
		Products tanker
		Reefer
		Ro-Ro
		Tanker
		Vehicle
PIClub	3	Good
		Average
		Bad

Appendix B

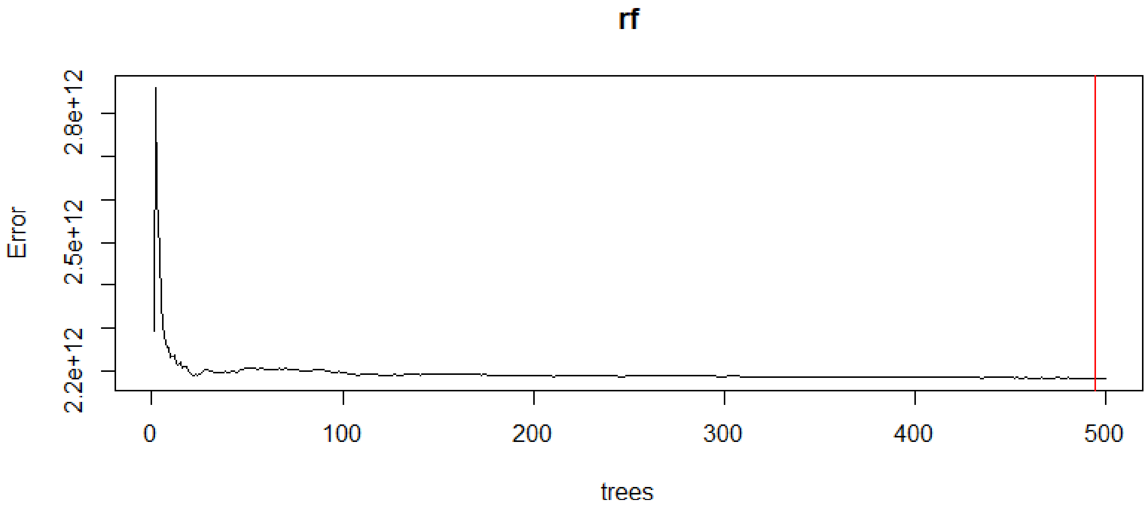


Figure B1. Tuning number of trees for RF. The red line represents the lowest Error. This happens when the number of trees is 495.

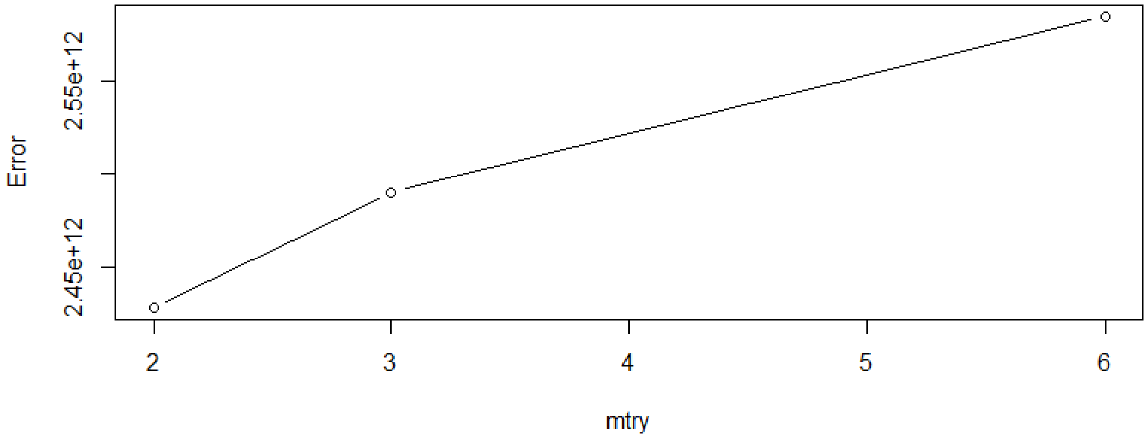
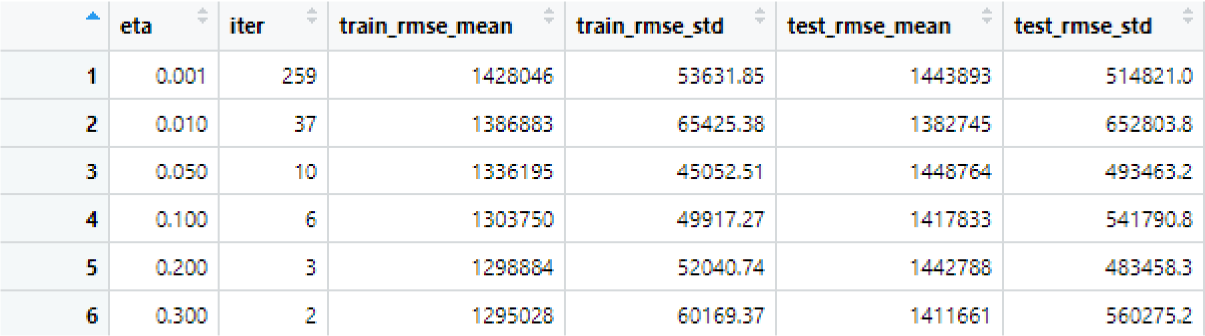
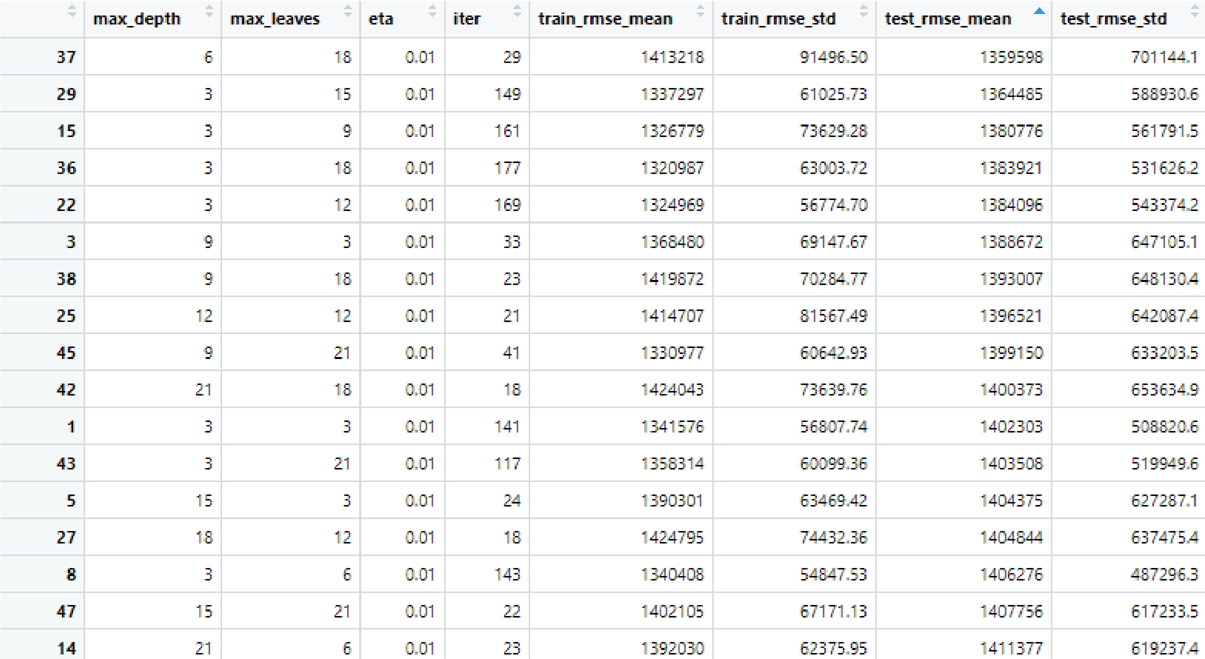


Figure B2. Tuning mtry for RF (number of trees set to 495). Mtry is 2 gives the lowest error.



	eta	iter	train_rmse_mean	train_rmse_std	test_rmse_mean	test_rmse_std
1	0.001	259	1428046	53631.85	1443893	514821.0
2	0.010	37	1386883	65425.38	1382745	652803.8
3	0.050	10	1336195	45052.51	1448764	493463.2
4	0.100	6	1303750	49917.27	1417833	541790.8
5	0.200	3	1298884	52040.74	1442788	483458.3
6	0.300	2	1295028	60169.37	1411661	560275.2

Figure B3. Tuning eta for XGB. Eta is 0,01 gives the lowest error.



	max_depth	max_leaves	eta	iter	train_rmse_mean	train_rmse_std	test_rmse_mean	test_rmse_std
37	6	18	0.01	29	1413218	91496.50	1359598	701144.1
29	3	15	0.01	149	1337297	61025.73	1364485	588930.6
15	3	9	0.01	161	1326779	73629.28	1380776	561791.5
36	3	18	0.01	177	1320987	63003.72	1383921	531626.2
22	3	12	0.01	169	1324969	56774.70	1384096	543374.2
3	9	3	0.01	33	1368480	69147.67	1388672	647105.1
38	9	18	0.01	23	1419872	70284.77	1393007	648130.4
25	12	12	0.01	21	1414707	81567.49	1396521	642087.4
45	9	21	0.01	41	1330977	60642.93	1399150	633203.5
42	21	18	0.01	18	1424043	73639.76	1400373	653634.9
1	3	3	0.01	141	1341576	56807.74	1402303	508820.6
43	3	21	0.01	117	1358314	60099.36	1403508	519949.6
5	15	3	0.01	24	1390301	63469.42	1404375	627287.1
27	18	12	0.01	18	1424795	74432.36	1404844	637475.4
8	3	6	0.01	143	1340408	54847.53	1406276	487296.3
47	15	21	0.01	22	1402105	67171.13	1407756	617233.5
14	21	6	0.01	23	1392030	62375.95	1411377	619237.4

Figure B4. Tuning max depth and max leaves for XGB (keeping eta 0.01). Max depth is 3 and max leaves is 15 has been chosen since the RMSE between the two best options differs not that much but the standard deviation does.

Table B1. Tuned hyperparamters of the ML models.

	Parameter 1	**Parameter 2**	**Parameter 3**
Decision tree	cp = 0068416	-	-
RF	number of trees = 495	mtry = 2	-
XGB	eta = 0,01	max_depth = 3	max_leaves = 15
SVR lin	cost = 0,1	-	-
SVR rad	cost = 5	gamma = 0,25	-
SVR pol	cost = 0,1	degree = 2	-