Car Insurance Marketing Data LTV Analysis

Elena Andreou
Department of Mathematics and Statistics
University of Cyprus
Nicosia, Cyprus
andreou.a.elena@ucy.ac.cy

Grigorii Turchenko
Department of Business and Public Administration
University of Cyprus
Nicosia, Cyprus
turchenko.grigorii@ucy.ac.cy

Panikos Christou
Department of Computer Science
University of Cyprus
Nicosia, Cyprus
christou.panikos@ucy.ac.cy

1. INTRODUCTION

1.1 Problem statement

Businesses have always been concerned with the problem of customer segmentation. One approach to discriminate between various customer groups is calculation of customer lifetime value (CLV/LTV). It is a product of customer lifetime, i.e., longevity of customer's relationship with the company, and customer value, usually expressed in monetary terms. By computing CLV for each customer, a company can identify more valuable customer groups, and e.g., provide them with tailored marketing offers in order to build a long-term relationship and maximize their utility to the business.

However, an issue with LTV is a complexity of calculation. Despite the formula consists of just 2 variables, calculating those usually requires a lot of data, collected from various sources. This makes direct calculation time-consuming and rather expensive. Therefore, a statistical model could be of use here. On the one hand, it would be able to supply decision-makers with a quick estimation of CLV, and on the other provide factors influencing the lifetime value.

1.2 Goals

This study is concerned with statistical analysis of factors, influencing customer lifetime value of a car insurance company. In order to carry out the analysis, the following research questions are proposed:

- What factors influence the value of LTV?
- Is it possible to create a robust and explainable model, able of accurate LTV prediction?

2. EXPLORATORY DATA ANALYSIS

Our main objective at this stage is the construction of a thorough exploratory data analysis in order to understand the distributions of predictors, their relationship, as well as which kind of relationship could be captured between these parameters and the target. In order to check the relationship of the parameters and the target value (outcome) we will be using overlaying plots (histograms, boxplots, density estimation) by the value of the target variable, to deduce any significant difference between the different Customer Lifetime values.

The variables will be inspected for possible issues, as well as potential patterns will be explored.

2.1 Variables Description

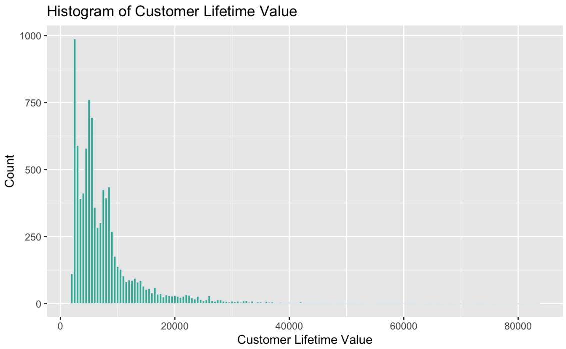
As a very first step let us introduce the variables in the dataset used in the study. The table below includes variable names, as well as their short meaning.

Variable Name	**Meaning**	**Range**
State	State of customer residence	Washington, Arizona, Nevada, California, Oregon
Customer Lifetime Value	Self-explanatory	1898.008 – 83325.38
Response	Response to a marketing offer	No, Yes
Coverage	Maximum amount paid out for claim	Basic, Extended, Premium
Education	Maximum education achieved	Bachelor, College, Master, HS or Below, Doctor
Effective To Date	Contract expiry date	56 unique dates
EmploymentStatus	Self-explanatory	Employed, Unemployed, Medical Leave, Disabled, Retired
Gender	Self-explanatory	F, M
Income	Annual income	0 – 99981
Location Code	Refer to "Range" column	Suburban, Rural, Urban
Marital Status	Self-explanatory	Married, Single, Divorced
Monthly Premium Auto	Monthly insurance cost	61 – 298
Months Since Last Claim	Self-explanatory	0 – 35
Months Since Policy Inception	Self-explanatory	0 – 99
Number of Open Complaints	Self-explanatory	0 – 5
Number of Policies	Self-explanatory	1 – 9
Policy Type	Refer to "Range" column	Corporate Auto, Personal Auto, Special Auto
Policy	Refer to "Range" column	Corporate L1–L3, Personal L1–L3, Special L1–L3
Renew Offer Type	Refer to "Range" column	Offer1, Offer2, Offer3, Offer4
Sales Channel	Self-explanatory	Agent, Call Center, Web, Branch
Total Claim Amount	The total pay-out a customer received	0.099007 – 2893.24
Vehicle Class	Refer to "Range" column	Two-Door Car, Four-Door Car, SUV, Luxury SUV, Sports Car, Luxury Car
Vehicle Size	Self-explanatory	Medsize, Small, Large

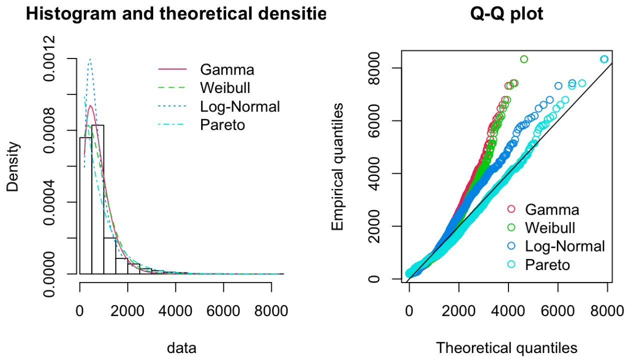
2.2 Dependent Variable

2.2.1 Customer Lifetime Value

Customer lifetime value in our case is measured on a continuous scale and is greater than zero.



The distribution is heavily right-skewed, and therefore heteroscedasticity is highly expected if the variable is used in the OLS model without any transformations. Therefore, in order to select appropriate statistical method, it was tested how well the variable fits several hypothesized distributions. The parameters of these distributions were estimated using maximum likelihood. The figure below shows the outcome of the testing.

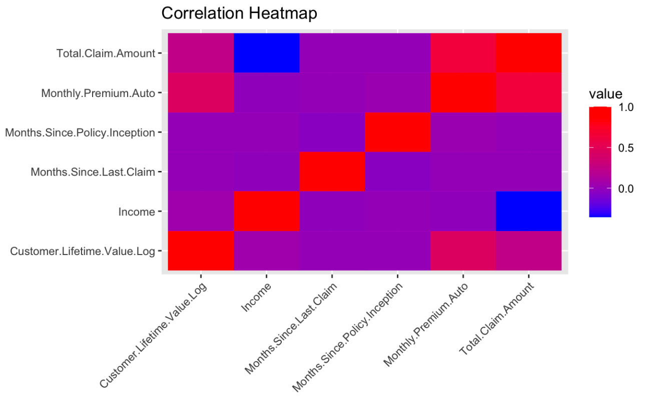


The best fit for CLV was the Pareto distribution. However, it is not as straightforward to model it, which is why it was decided to log-transform the target variable, so that potential heteroscedasticity is mitigated. Despite log-normal is a notably worse fit compared to Pareto, it is merely more practical to assume the variable comes from the former distribution. Applying a log transformation to the response variable can help to normalize the distribution and reduce the impact of outliers, making the data more suitable for linear regression analysis. Additionally, by taking the logarithm of the target variable, we can make the variance of the transformed variable more constant, which can help to reduce the problem of heteroscedasticity (which occurs when the variability of the response variable changes across different levels of the predictor variables.) that is expected due to the right-skewed distribution of the original variable.

2.3 Numeric Predictors

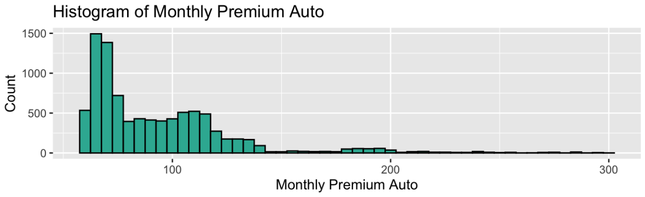
2.3.1 Numeric Predictors overview

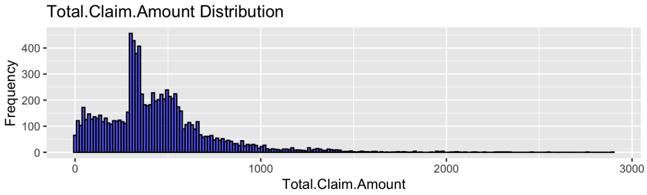
There are rather few numeric predictors in the dataset, which is why they will be briefly discussed in this section. To begin with, let us explore if there is some linear association among the variables. The heatmap below provides a laconic overview of pairwise correlation.



There is moderate linear relationship between logarithm of CLV and monthly premium auto, as well as total claim amount, whereas for the remaining variables the linear relationship does not seem to be the case. Another notable conclusion from the figure above is that total claim amount and monthly premium are highly positively correlated. This indicates a multicollinearity issue, which will be addressed with designated feature selection techniques further. All in all, at least at the stage of EDA it is unlikely that the majority of the continuous predictors will have significant effect on the target variable.

Another common issue with numeric predictors to be inspected is outliers, since they can cause high leverage points. Among the 5 numeric predictors, three have outliers, and the figure below depicts their distributions.

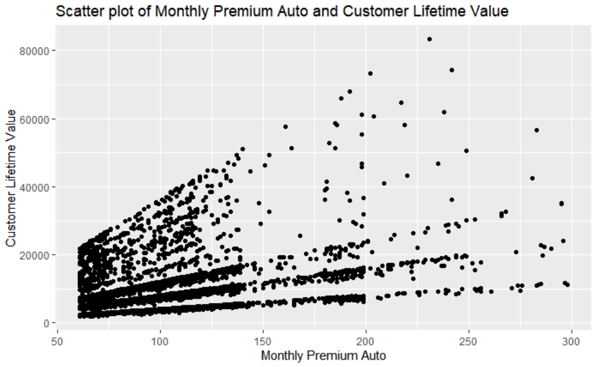




Not only these predictors are correlated, but also they have quite a large number of outliers, which cannot be simply dropped. At the modeling stage Cook's distance will be employed to inspect for high leverage points, and special attention will be paid to these variables in particular. In the next section the overview of categorical variables will be presented.

2.3.2 Monthly Premium Auto correlation

As we discussed above there seems to be some correlation between Monthly Premium Auto and a Customers lifetime value. When we plot a scatter plot of the variables, we observe this:



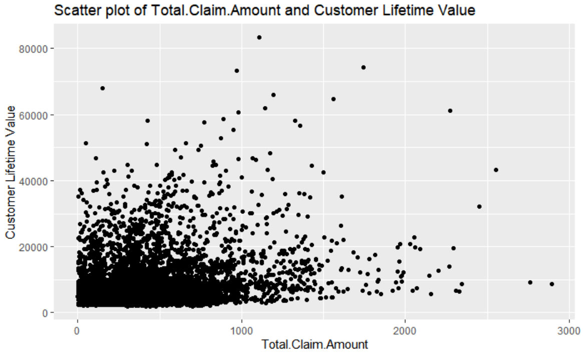
Here we clearly see a pattern emerging from these variables. Specifically, we see faintly 9 lines appearing. Each line has a different starting height and a different slope. Hopefully our model will be able to capture this relationship and also find a way to calculate it with the help of the other predictors.

Moreover, we can clearly see a positive funnel shape emerging which explains the positive correlation we saw before so our team expects Monthly Premium Auto to be a significant contributor in a customer's CLV.

2.3.3 *Total Claim Amount correlation*

Another relative correlation existence our team observed was in Total Claim Amount which was 0.23.

When we plot a scatter plot to see the relationship between them, we see this:

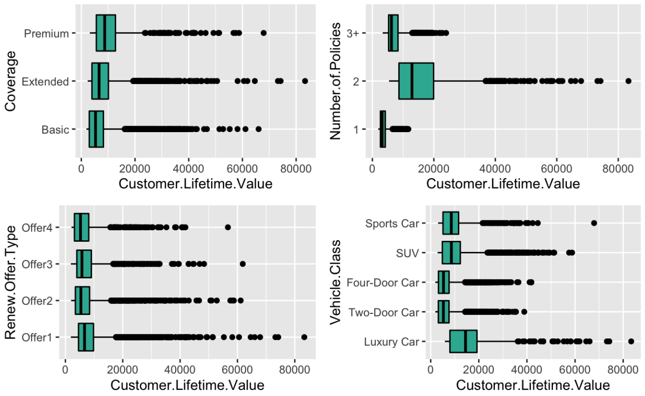


The shape looks rather chaotic with many values centered on the lower values for both variables. With that though we can clearly see that in general as the Total Claim Amount goes up, it relatively increases the CLV so we can faintly see a positive funnel emerging. Thus, we also expect this value to be significant, but not to the extent of Monthly Premium auto.

2.4 Categorical Predictors

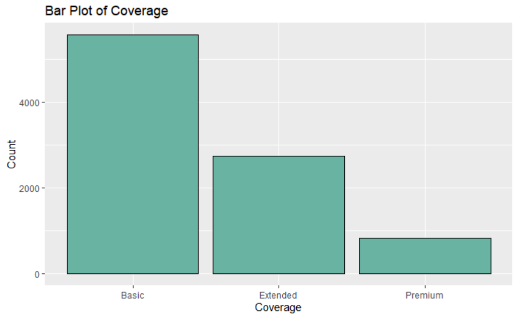
2.4.1 *Relationship with LTV*

As previously mentioned, categorical variables make up the majority of the predictors in the dataset. Nevertheless, most of them have little effect on the LTV. For each categorical feature, boxplots of customer lifetime value with respect to each group were plotted and median CLVs were compared. As a result, it was possible to point out only 4 predictors, which have more or less significant difference in the median value of the target variable. They are outlined in the figure below.

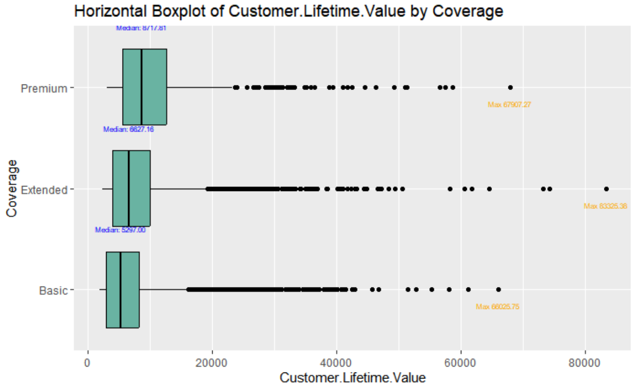


We see some categorical values have some impact on the CLV lets see these predictors in more detail.

2.4.2 *Coverage*



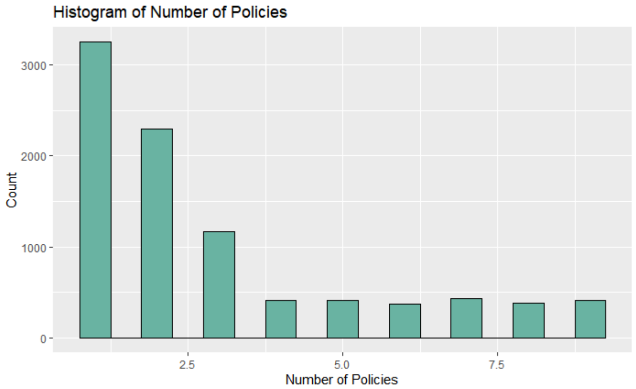
The Basic coverage is preferred by most customers, followed by the Extended coverage with half the frequency, and finally the Premium coverage is the least popular. When we plot boxplots of these values:



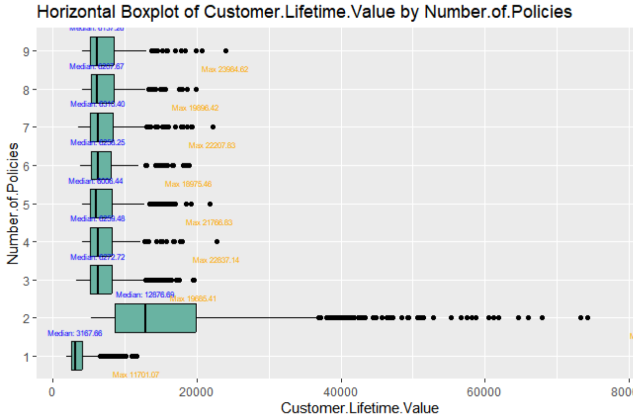
We clearly see that Premium has higher median and so does extended. This makes sense since a premium plan costs more than an extended plan which again is more expensive than a basic plan. So, we can expect to see that the coverage a customer has will affect the predicted CLV.

2.4.3 *Number of Policies*

The number of policies a customer can have is 1 – 9. And the majority has 1 then 2 then 3. Afterwards a small number of clients have 4+.



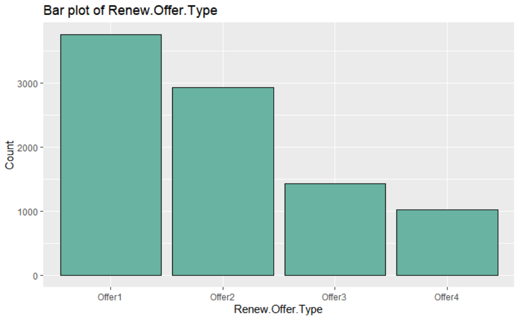
When we plot boxplots on how each number of policies effects CLV. We clearly see that customers who have 1 policy have on average the least CLV. Weirdly enough, customers with 2 have much higher CLV's than the rest and then from 3+ we see somewhat the same distribution.



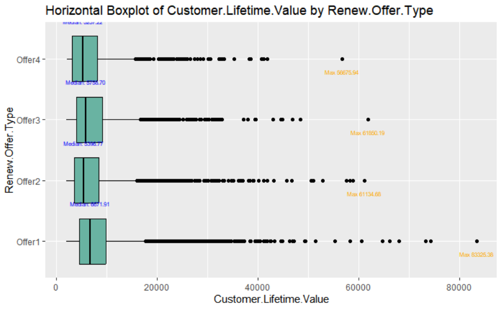
With this in mind, our team decided to group these into 3 factors, 1, 2 and 3+ as you see in the previous page. We believe this will help the model since it reduces dimensionality and captures the behaviors better.

2.4.4 *Renew.Offer.Type*

When it comes to renewal offers types. We see we have 4 options ranging from 1-4. Most Customers choose the first option then the second and so forth.



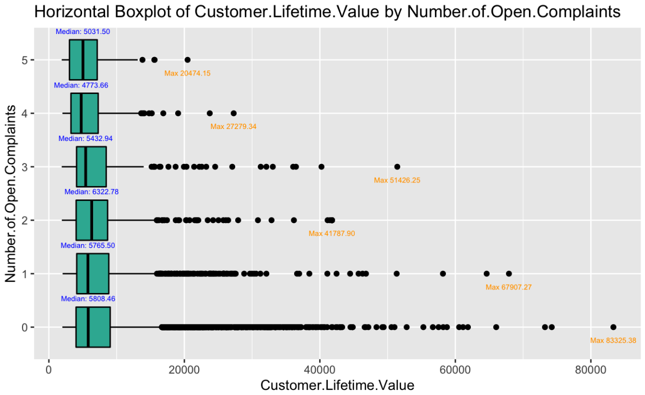
When we plot the boxplots however, we see that offer 1 has a restively higher median than rest, as well as offer 4 having slightly less. Thus, we expect these 2 values to affect our models in the future.



2.4.5 *Transformations and Cleaning*

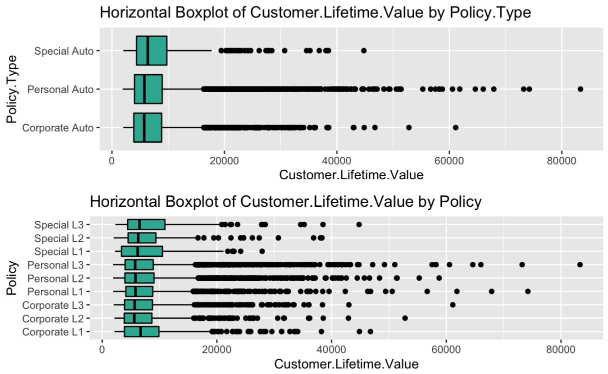
Due to the fact that there are about 20 predictors in total, and most of them are categorical, the resulting design matrix of the model will have quite significant dimensionality. In order to reduce it, the variables were transformed by merging some categories into a single one. The main criterion for this was distributions similarity. If for several categories LTV has approximately the same median, then it makes sense to merge them. This way we do not lose a lot of information contained in each level.

However, for some variables a more common-sense approach was applied. Consider the variable indicating the number of open complaints.



Even though customers who have 2 complaints and less have relatively higher LTV, it was decided to convert the predictor to the dummy one, which takes the value 1 if a customer had ever complained, and 0 otherwise. It makes sense, because clients who filed a complaint at least once had already expressed their dissatisfaction with the services, and are already in a way different from those, who had not. Moreover, there are way more non-complainers than complainers, and uniting the 5 classes resulted in a balanced dummy variable.

Lastly, some variables had a very similar semantic meaning, which is why keeping both did not make sense. In particular, these were policy and policy type.



In general, information contained in policy is also present in policy type, which is explicit collinearity. Therefore, it was concluded to drop the "Policy" predictor prior to feature selection, since in this case there is a commonsense justification for this.

2.5 Interaction Terms

Based on the information above, it is clear the data does not provide a lot of promising predictors. Therefore, it was decided to employ interaction terms as well. Four interaction terms were created and will be briefly discussed in this section.

2.5.1 Monthly Premium and Coverage

Monthly premium represents the amount paid by a customer to the insurance provider, and coverage implies the extent to which client's expenses are covered should an insurance case happen. In the case of our dataset, coverage is divided into "Basic", "Extended", and "Premium". It makes sense these variables could have a synergetic effect, because, for instance, "Premium" coverage explicitly implies higher price, which results in a higher LTV of the customer holding this type of policy.

2.5.2 Vehicle Class and Vehicle Size

The meaning of these predictors is quite straightforward. An interaction term between these variables allows to account for specific cases, which are not present when the variables are employed separately. For example, among vehicle classes there are luxury cars. These cars can be quite different, both small-sized (retro), medium-sized (sport cars), as well as big-sized. Accounting for this may potentially unveil additional insights about the effect on CLV.

2.5.3 Coverage and Renew Offer Type

Renew offer type implies a promotional message, which the client had received shortly before the policy expiry. Offers are just encoded as "Offer 1", "Offer 2", etc., which is why their semantic meaning is unclear. However, there could be possible synergy of the variable with coverage. Obviously, each coverage plan has its peculiarities, and the offers have to be tailored to the specifics of the plan. Accounting for this helps to reduce unobserved heterogeneity and gain additional insights about the data.

2.5.4 Sales Channel and Policy Type

To remind, the sales channels of the insurance company are "Agent", "Call Center", "Web", and "Branch", while the policy types are "Personal", "Corporate", and "Special". It makes sense to assume that for different policy types, different sales are employed or at least work better. For example, personal insurances are likely to be sold on a high scale, which is why "Web" and "Call Center" channels can provide sufficiently high sales. On the other hand, corporate and special insurance policies should be more custom and complex, which is why an agent participation, or a branch visit should be a more effective channel. Since sales directly influence LTV, accounting for such a synergetic effect may yield interesting results.

While it was possible to create some other interaction terms, it was decided to limit their number to these four, since the dimensionality data was already high enough.

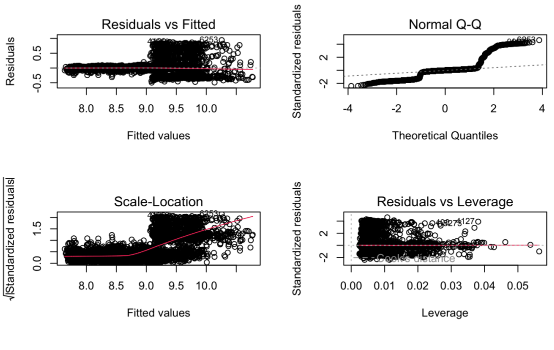
3. REGRESSION ANALYSIS

In this section our team decided to train some more regression models to see if any of them have the capacity to capture the relationship between the target variable and the predictors. The methods used included Multiple Linear Regression, Ridge and Lasso Regression, Best Subset Selection algorithms, as well as Decision Trees, Random forests, Gradient boosting Mahcines and K Nearest Neighbour Regression.

3.1 Multiple Linear Regression

3.1.1 OLS Regression

The first model utilized was multiple linear regression using the OLS method to estimate the model parameters. It provides a high degree of interpretability and serves as a good starting point for the analysis because it is easy to implement and provides a baseline model to compare against more complex models. However, from the EDA there were concerns that the method may not be applicable due to unobserved linear relationship between predictors and the target variable. Therefore, after fitting the linear model, we immediately proceeded with inspection of the assumptions, in particular, of homoscedasticity or constant variance assumption. In order to check it, the plot below was employed.

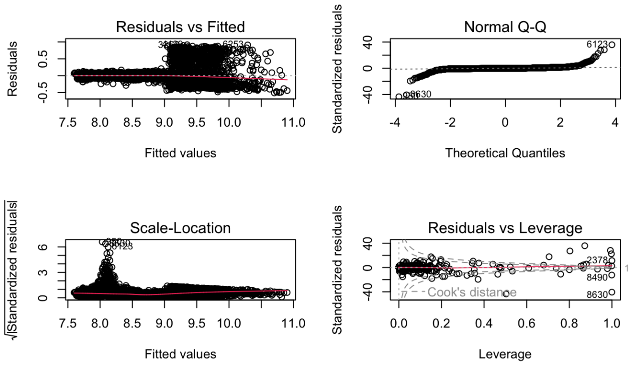


Based on the figure, heteroscedasticity is evident. Firstly, on the "Residuals vs Fitted" chart it can be seen that variance before point 9.0 on the x-axis and after it has significantly different magnitude. Secondly, the normal qq-plot shows the residuals are not distributed normally. The distribution resembles Cauchy or any other heavy-tailed distribution rather than standard normal. Therefore, a first problem with OLS model was identified.

3.1.2 WLS Regression

As a remedy, weighted least squares (WLS) regression was used. The estimated regression coefficients obtained from WLS are more efficient than the ordinary least squares

(OLS) estimates when the variance of the errors is heteroscedastic. It is different from the OLS in a way that each observation is assigned a weight, calculated as the inverse of the variance of each observation, and therefore observations with lower variance are assigned a higher weight. However, this technique did not help to solve the heteroscedasticity issue. Consider the figure below.



The variance of the residuals had not improved almost at all, in a sense that it is as non-constant as before. As for the distribution of the residuals, the tails had become slightly shorter, but it is a minor improvement.

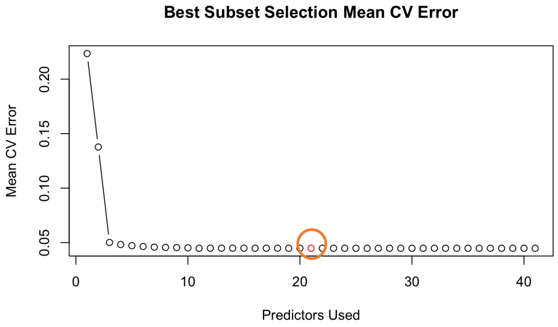
All in all, it was concluded that least squares regression cannot be implemented for the dataset at hand, since homoscedasticity assumption of the model is violated. The main causes of these are relatively high dispersion of the target variable, as well as absence of linear relationship with the predictors.

Given that, there were two options on how to proceed. On the one hand, we could recall from the exploratory analysis that lifetime value most likely follows Pareto distribution and consequently find a method, suitable for modelling of such a dependent variable. On the other hand, we could employ any method which does not required estimation of sigma. The examples are subset selection algorithms, as well as shrinkage methods. Our team decided to opt for the second approach due to it being more straightforward and easier to implement.

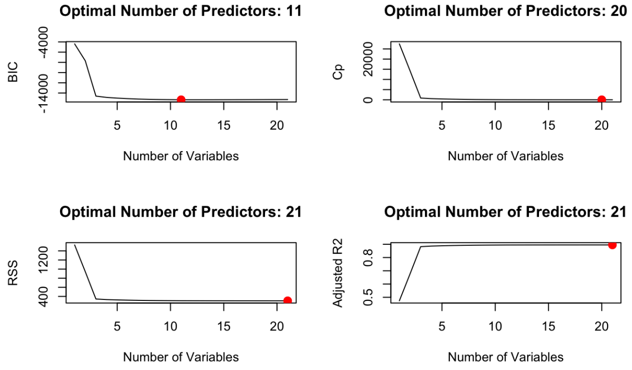
3.2 Cross Validation with Feature Selection Methods

3.2.1 Best Subset Selection

Heteroscedasticity can affect the reliability of metrics such as adjusted R-squared, BIC, etc., as these metrics assume normality of residuals and homoscedasticity (equal variances of residuals). Therefore, relying solely on these metrics for feature selection may not be appropriate in such cases. Instead, performing feature selection during the cross-validation phase can be a better approach. To begin with, an exhaustive search for the best subset was carried out. In this case, interaction terms were not included due to a very high resulting computational cost. The selection was carried out with the use of 5-fold cross-validation (CV). The data was split into training and test sets for the purpose of robustness check. The result of the process is represented in the figure below.



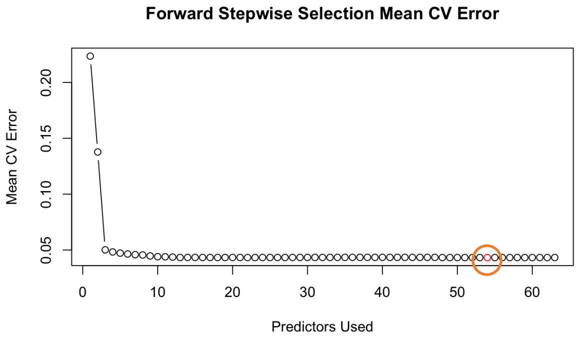
The red dot highlights the number of predictors with the lowest mean cross-validation error. According to the procedure, 21 predictor is optimal. The best subset selection regression model was then refitted with the above mentioned number of variables, and was analyzed within the metrics, such as Bayesian information criterion (BIC), Mallow's Cp, residual sum of squares (RSS), and Adjusted R-squared. When performing feature selection using cross-validation, the issue of heteroscedasticity can still affect the reliability of these metrics, however, we can evaluate the performance of different models on independent test sets, which can provide a more realistic estimate of the model's predictive performance on new data. The figure below shows the optimal number of features with respect to each of the metrics.



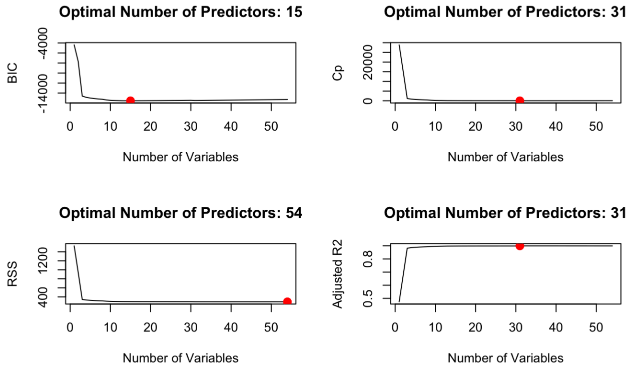
While R-squared and RSS suggested to keep the optimal predictors on the level of 21, Cp and BIC offered 20 and 11 variables respectively. Due to the fact mentioned before that the design matrix of the model has a relatively high dimension, we decided to be more selective in terms of the resulting number of predictors, and therefore made a decision to proceed with 11 features.

3.2.2 Forward Stepwise Selection

Next the forward selection algorithm was implemented. Since it does not perform an exhaustive search of the best subset and usually "stops" early, interaction terms were included as well. The optimal number of predictors from cross-validation is depicted in the figure below.



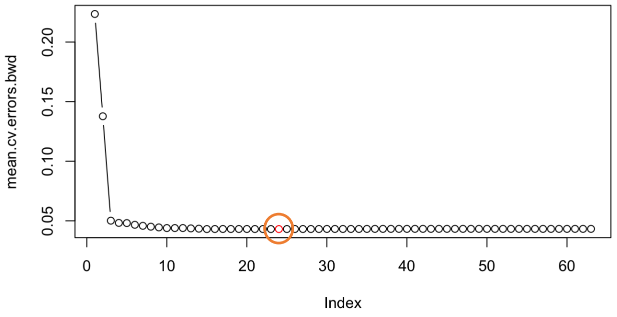
Interestingly enough forward selection actually resulted in a quite complex model. It is usually not the case, since the algorithm starts with an empty model, and then iteratively adds a predictor. Since 54 is quite a large quantity of features, let us consider a refitted forward stepwise selection regression model with 54 maximum features.



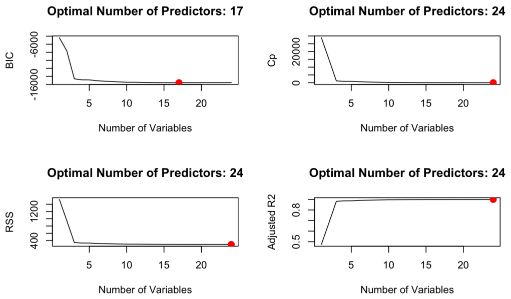
While other metrics suggest still keeping quite high number of variables, BIC, being in general highly discriminative criterion, indicates 15 predictors to be optimal. The conclusion looked quite reasonable, which is why we proceeded with this number.

3.2.3 Backward Stepwise Selection

The last subset selection algorithm attempted was backward selection. It was implemented out with 5-fold cross-validation as well and included interaction terms. The optimal predictors number based on the mean cross-validation error is shown in the chart below.



Once again, the results are not as expected, since backward selection starts with a full model and iteratively removes predictors, usually resulting in a more complex model. However, 24 variables is also not a low number, which is why consider refitted backward selection model with 24 maximum features.



As in case of best subset selection and forward stepwise selection, BIC suggests the lowest number of variables, 17, and it is reasonably low.

3.2.4 Models Comparison

Let us now compare the results of the above described algorithms. First, we should consider which predictors were selected by the models as important. The table below provides a list of these features.

Best	**Forward**	**Backward**
CoverageExtended	CoverageExtended	CoverageExtended
CoveragePremium	CoveragePremium	CoveragePremium
EmploymentStatusEmployed	EmploymentStatusEmployed	EmploymentStatusEmployed
GenderM	GenderM	GenderM
Marital.Status Single	Marital.StatusSingle	Marital.StatusSingle
Monthly.Premium.Auto	Monthly.Premium.Auto	Monthly.Premium.Auto
Number.of.Open.Complaints 1	Number.of.Open.Complaints1	Number.of.Open.Complaints1
Number.of.Policies2	Number.of.Policies2	Number.of.Policies2
Number.of.Policies3+	Number.of.Policies3 +	Number.of.Policies3 +
Vehicle.ClassSUV	Vehicle.ClassSUV	Vehicle.ClassSUV
Vehicle.ClassSports Car	Vehicle.ClassSports Car	Vehicle.ClassSports Car
		Vehicle.ClassTwo-Door Car
		Vehicle.ClassFour-Door Car
	CoverageExtended:Monthly.Premium.Auto	CoverageExtended:Monthly.Premium.Auto
	CoveragePremium:Monthly.Premium.Auto	CoveragePremium:Monthly.Premium.Auto
	EducationCollege	Policy.TypeSpecial Auto:Sales.ChannelBranch

To begin with, all the models considered coverage, gender, “single” marital status, monthly premium, the fact of a complaint, and number of policies as significant factors influencing CLV. However, there are some differences as well. While best subset selection concluded that only “unemployed” employment status counts as important, forward and backward selection considered all the categories in the “Employment.Status” variable to be significant. Additionally, backward selection also incorporated all the vehicle classes, while forward and exhaustive models treated only the possession of a sports car or a SUV to be relevant. Moreover, forward selection, unlike other algorithms, regarded college education to be a valuable factor. Lastly, both forward and backward stepwise selection considered common interaction to be influential, i.e., the interaction between coverage plan and monthly insurance premium. Backward selection, in turn, also treated an interaction of “Special” policy type with “Branch” sales channel to provide substantial LTV information, which is exactly as expected.

Then, let us compare the goodness-of-fit of the models. Root mean squared error (RMSE), mean absolute deviation (MAE), and R-squared will be used as the quality assessment metrics. The table below provides a comparison within the measures.

Model	Subset	RMSE	MAE	R2
Best	train	0.2114	0.1229	0.8948
Forward	train	0.2074	0.1140	0.8987
Backward	**train**	**0.2068**	**0.1112**	**0.8993**
Best	test	0.2060	0.1201	0.9016
Forward	test	0.2025	0.1118	0.9049
Backward	**test**	**0.2019**	**0.1087**	**0.9055**

Based on performance, backward selection shown the best results on both sets. Most likely it pertains to the fact that the backward selection model is the most complex out of the three considered. Moreover it has the highest number of predictors.

3.3 Shrinkage Methods

3.3.1 Ridge Regression

The first shrinkage method employed was ridge regression. It does not perform feature selection, however, is still useful, since it applies penalty to the coefficients and therefore provides robustness to overfitting. Prior to fitting ridge regression model, numeric features were standardized. The search for optimal lambda was carried out with 10-fold cross-validation on the train set. In general, there are two ways to determine optimal value of lambda, either by selecting the one which leads to the lowest mean cross-validation error, or according to one standard error rule. In order to select the one out of the two, we compared the train and test set metrics for the refitted models with respective lambdas in the table below.

Model	Subset	RMSE	MAE	R2
Ridge_Min	train	0.2554	0.1707	0.8817
Ridge_1SE	train	0.2554	0.1707	0.8817
Ridge_Min	test	0.2511	0.1684	0.8878
Ridge_1SE	test	0.2511	0.1684	0.8878

Based on the figures, the same lambda was chosen by both rules, and there is no difference in which model to select.

3.3.2 Lasso Regression

Lasso applies a penalty to the coefficients such as ridge, but it can make the coefficients equal to zero, which is why it also conducts feature selection. The same procedure as for ridge regression was repeated for the lasso model, and the results are displayed in the table below.

Model	Subset	RMSE	MAE	R2
Lasso_Min	**train**	**0.2061**	**0.1116**	**0.8999**
Lasso_1SE	train	0.2111	0.1226	0.8955
Lasso_Min	**test**	**0.2016**	**0.1096**	**0.9057**
Lasso_1SE	test	0.2056	0.1170	0.9024

In case of lasso, the lambda based on minimal CV error yielded significantly better results compared to the lambda chosen according to 1 standard rule. However, the value of lambda itself was extremely close to zero. Hence, very little feature selection was performed, as only 8 out of 64 variables had coefficient equal to zero. Therefore, it was decided to opt for a higher lambda so that lasso applies a higher penalty and therefore conducts a more discriminative selection. The results of this approach are outlined in the table below.

Model	Subset	RMSE	MAE	R2
Lasso_Grid_Min	train	0.2236	0.1439	0.8859
Lasso_Min	**train**	**0.2061**	**0.1116**	**0.8999**
Lasso_Grid_Min	test	0.2184	0.1377	0.8932
Lasso_Min	**test**	**0.2016**	**0.1096**	**0.9057**

The “Lasso_Grid_Min” is the model with $\lambda = 0.01$, while the “Lasso_Min” was fitted with $\lambda = 4.543294e^{-5}$ .All in all, the model with the lower penalty performed better on both training and test data sets. However, it is important to mention that higher lambda selected only 8 predictors, whereas lower “stopped” at 56. According to the “Lasso_Grid_Min” model, important variables were income, number of policies, monthly premium and its interaction with premium coverage, as well as extended coverage and SUV vehicle class.

3.3.3 Models Comparison

Finally, let us compare the best lasso and ridge regression models in order to determine which method provides a better fit. Consider the table below.

Model	Subset	RMSE	MAE	R2
Ridge_Min	train	0.2554	0.1707	0.8817
Lasso_Min	**train**	**0.2061**	**0.1112**	**0.8999**
Ridge_Min	test	0.2511	0.1684	0.8878
Lasso_Min	**test**	**0.2017**	**0.1096**	**0.9057**

Lasso regression significantly outperforms ridge in terms of RMSE and MAE. The difference in R-squared is minor, which is due to the fact that both models have a very high dimensionality, 56 and 64 predictors respectively, which inevitably leads to a high value of R-squared.

3.4 Other Models

As another step in the research, we wanted to find out if there is possible to achieve a more accurate fit than for the methods implemented above. The methods included Decision Tree, Random Forest (RF), Gradient Boosting Machine (GBM), K Nearest Neighbors (KNN). In the subsequent paragraph a short overview of each method will be provided.

3.4.1 Decision Tree

Decision Tree is a machine learning algorithm that builds a tree-like model by recursively splitting the feature space into subsets based on the values of input variables [1]. Each internal node of the tree represents a test on a specific feature, and each leaf node represents a class or a numeric value. The decision of which feature to split on and where to make the split is based on a measure of impurity or information gain.

The algorithm starts with the entire dataset at the root node, and at each step, it selects the best feature and threshold value that maximizes the information gain, which is defined as the reduction in impurity achieved by splitting on that feature. The most commonly used impurity measures are entropy and Gini index. Once the split is made, the dataset is partitioned into two or more subsets, each of which is recursively processed in the same way until a stopping criterion is met. This criterion could be a maximum tree depth, a minimum number of instances per leaf, or a minimum improvement in impurity.

In case of the problem at hand, decision trees have an advantage in terms of interpretability. It is possible to visualize the tree, and extract the features used to construct it, which can be then considered important. However, because of high dimensionality and substantial noise, the model is likely to overfit.

3.4.2 Random Forest

Random Forest algorithm is largely based on decision trees. Specifically, it is an ensemble algorithm of bagging type. These algorithms fit the model on several bootstrap samples and then produce the prediction by averaging (in case of regression) the outcomes of each individual predictor. RF in particular fits a defined number of trees, where each tree is fitted on a random number of features.

Since both samples and features are randomized, given a large enough number of trees per iteration, random forest is more robust to the issues which individual decision tree faces. Nevertheless, these strengths come at a cost of losing interpretability. Despite it is still possible to extract feature importance, RF is already largely a black-box model.

3.4.3 Gradient Boosting Machine

Gradient Boosting Machine is also an ensemble algorithm and is based on decision trees as well. The boosting ensemble algorithms work in a fixed number of iterations, and basic idea behind them is to give more attention to the instances, which were poorly predicted at a previous iteration. As mentioned, GBM fits simple decision tree at the first iteration, and then produces more trees, which are fit to the negative gradient of the loss function, representing the direction of steepest descent for the optimization problem [2]. The predictions of all individual trees are then averaged (regression) with weighted average.

Usually, GBM provides even better performance than RF, since it can be executed for a large number of iterations and actually focuses on improving the prediction of instances with high residuals. Additionally, the boosting is able to capture non-linearity. However, it requires careful choice of parameters, computationally more costly, and is black box.

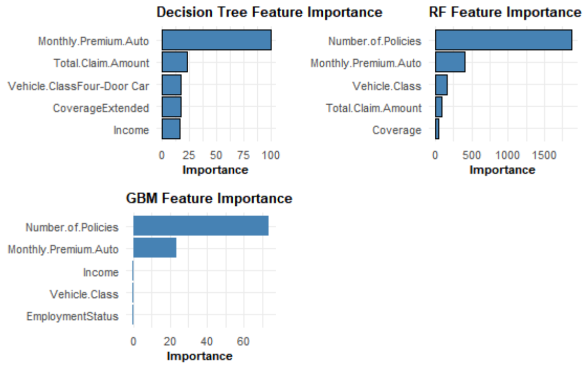
3.4.4 K Nearest Neighbours

KNN is the most straightforward algorithm among all the methods mentioned before. The idea is as follows. For each individual data point, the algorithm finds k closest neighbors based on a distance metric, and then uses weighted average to produce the prediction. Since the distances are employed, it is necessary to make sure the features are scaled beforehand.

Despite the simplicity, KNN is not expected to perform well in the context of our problem due to high number of predictors. The algorithm is very prone to a so-called "curse of dimensionality". In high dimensions, distances between data points become smaller, making it difficult for the model to make an accurate prediction. In this case, more neighbors are required, but this leads to a higher variance, and poorer generalization as a result.

3.4.5 Models Insights

Among the 5 models mentioned above, it was possible to extract feature importance from the 3 of them. The chart below represents the top-5 important features with respect to each algorithm.



Generally speaking, monthly premium and number of policies can be considered the most important factors based on the figure. Gradient boosting was highly discriminative in its choice of predictors and largely made predictions based on these two variables. Other models somewhat favored also total claim amount, coverage, vehicle class, and income. All in all, the difference with the subset selection results is not that substantial, as they included these variables into the optimal models as well.

3.4.6 Performance Comparison

Finally, let us evaluate the performance of all 5 algorithms and compare it with each other, as well as best models from the subset selection and shrinkage methods sections. The table below provides a comparison within the frame of RMSE, MAE, and R-squared.

Model	**Subset**	**RMSE**	**MAE**	**R2**
Backward	train	0.2068	0.1111	0.8993
Backward	test	0.2019	0.1087	0.9055
Lasso_Min	train	0.2061	0.1116	0.8999
Lasso_Min	test	0.2016	0.1096	0.9057
Tree	train	0.2167	0.1395	0.8894

Tree	test	0.2204	0.1421	0.8873
RF	**train**	**0.0887**	**0.0422**	**0.9825**
RF	**test**	**0.1922**	**0.0920**	**0.9150**
GBM	train	0.1955	0.1021	0.9101
GBM	test	0.2002	0.1038	0.9071
KNN	test	0.4332	0.2644	0.5739

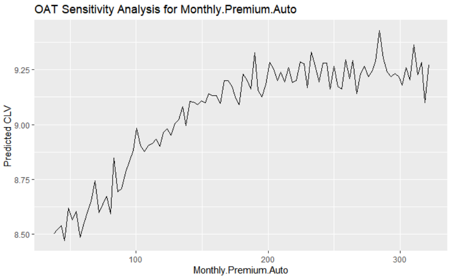
The model with the best fit turned out to be Random Forest. Even though there is a clear overfitting situation due to substantial difference between train and test metrics, the model still provided the best approximation of LTV. However, lasso regression and backward selection did not perform much worse, while possessing lower complexity. Backward selection performance is actually about the same as of lasso, which is notable given the former used only 17 predictors, while the latter needed 56. Lastly, as expected, KNN provided a drastically poorer fit than any other model due to the effect of the "curse of dimensionality".

3.5 One at a time sensitivity analysis

We performed a One-at-a-Time (OAT) sensitivity analysis on the Random Forest regression model since it had the best performance. This analysis helps us understand how changes in each numeric input variable individually impact the model's predictions. We analyzed five numeric variables: Income, Monthly Premium Auto, Months Since Last Claim, Months Since Policy Inception, and Total Claim Amount.

Basically, we trained the Random Forest on the whole dataset and then for each numerical variable, we split it into a sequence from min to max to 100 interval values. Then we got 100 rows from the dataset, switched the variable from those to the interval value we want to test and found the mean CLV. After doing it 100 times we can see how slowly incrementing the variable effects Log CLV.

Unfortunately, most numerical variables did not produce conclusive results. The only ones that did was Monthly Premium Auto and slightly Total Claim Amount.



As you can see as MPA went up, the mean CLV Log of those 100 rows seems to increase. Moreover, it increases in a much bigger way than Total Claim Amount and Income.

4. CONCLUSION

The study was concerned with the analysis of the factors, influencing customer lifetime value of an insurance company for the car insurance product. The methods used for the analysis included Multiple Linear Regression, Subset Selection algorithms, Shrinkage methods, as well as popular machine learning models, such as Decision Tree, Random Forest, Gradient Boosting Machine, and K Nearest Neighbors.

The first and the main goal of the analysis was to determine factors influencing LTV. Due to the violation of the OLS model assumptions, it was not possible to fit a linear regression and determine important predictors based on p-values. Instead, influential variables were identified with the use of subset selection methods, lasso regression, and feature importance of tree models. Based on the aggregated results of these methods, it was possible to outline several factors, influencing CLV the most. The table below provides these factors, as well as corresponding coefficients from lasso and backward selection, two best-performing linear models.

Predictor	**Lasso**	**Backward**
Monthly Premium	0.3664	0.0117
Two Policies	1.4022	1.4009
Three or More Policies	0.6925	0.6915
Extended Coverage	0.0247	0.1960
Premium Coverage	0.0727	0.4182
Vehicle Class: Two-door	0.2261	0.3838
Vehicle Class: Four-door	0.2217	0.3800
Vehicle Class: Sports Car	0.2170	0.3862
Vehicle Class: SUV	0.2666	0.3641

The magnitude of coefficients does not make a big difference, it is sufficient to mention that both models assign the same sign to them, which is positive. Therefore, the most influential predictors are considered those, which serve as lifetime value drivers. The conclusion is, however, rather trivial. It is evident that customers with higher monthly premium, holding more than 1 policy, and applying for extended or premium rather than basic coverage are more valuable. They simply pay for more expensive services, and therefore generate more revenue. What is not as straightforward is that LTV of luxury cars owners is lower than for owners of any other vehicle class. The main reason is that luxury car drivers have two to three times higher median total claim amount compared to the rest, which is why the company loses quite a lot on the payouts to these customers.

The second goal of the study was to determine if accurate prediction of customer lifetime value is possible. For regression problem it is not as straightforward to understand if the error is low on an absolute scale, since there are few standardized quality assessment metrics. One option is to use R-squared, however it inevitably increases merely with an increase in predictors quantity, and even adjusted R-squared is to an extent prone to such tendency. Nevertheless, we believe the predictive capacity of the employed models is sufficient. Machine learning algorithms surely performed better, but simpler lasso and backward selection models also provided a decently accurate estimation of LTV.

The outcomes of the research can be employed by the insurance firm for maximizing the aggregate lifetime value. Based on the analysis, it is difficult to segment customers other than by the class of the vehicle they own. The recipe for success is fairly simple and lies in a higher number of sales.

5. REFERENCES

[1] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman, “Decision trees: an overview and their use in medicine”, 2002, Journal of medical systems, 26, pp.445-463.

[2] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine”, 2001, Annals of statistics, pp.1189-1232.

[3] R. W. Nahhas, 5.24 Sensitivity analysis | Introduction to Regression Methods for Public Health Using R. Accessed: Apr. 22, 2023. [Online]. Available: https://bookdown.org/rwnahhas/RMPH/mlr-sensitivity.html

[4] S. Salleh, “Sensitivity Analysis: One at a Time or All Together?,” Analytica, Aug. 16, 2013. https://analytica.com/sensitivity-analysis-one-at-a-time-or-all-together/ (accessed Apr. 23, 2023).

[5] “Regression using k-Nearest Neighbors in R Programming,” GeeksforGeeks, Jul. 25, 2020. https://www.geeksforgeeks.org/regression-using-k-nearest-neighbors-in-r-programming/ (accessed Apr. 23, 2023).

[6]