ACTL4305 Individual James To

James To

12/11/2020

Contents

1 Executive Summary

The aim of this report is to analyse the predictive power of linear regression models to predict pure premiums for auto insurance. The report delves into four different "variations" of linear models and finds the best fit. In addition, the report analyses the issues with auto insurance data, but performing Exploratory Data Analysis and gaining insights through visualisations. One of the core issues tackled in the report is "skewed" data. Two methods were used to try and predict the pure premium values of the policy holders. The four models variations of models are as follows:

- Ordinary Least Squares (OLS) Regression

Often the first regression technique for any forecasting project, OLS regression is well talked about for its easy interpretability and low computational power. However, the model lacks where "Black Box" models do not, and that is in accuracy of it's predictive power. Surprisingly, OLS regression resulted in our best model with the lowest Test MSE for both methods used for tackling predictions of pure premiums.

- Ordinary Least Squares (OLS) Regression with Shrinkage Techniques

Shrinkage Techniques were used in hopes of improving the OLS model, but failed to do so. Though yielding similar results as OLS, the LASSO, Ridge, and Elastic Net all fell short of beating the prior model when it came to predicting pure premiums.

- Best Subset Selection

Best Subset Selection attempts to find the best combination of predictor variables by going through every possible combination. In total, the 35 predictors - binary and correlated removed - went through a multitude of combinations but still resulted in a model worse than both OLS and OLS with Shrinkage Techniques.

- Conditional Model

The Conditional Model was a unique model that attempted to address the heavily zero-inflated data. This model first performed a classification to predict if a policy holder would have a claim at all, and then perform regression on those who were predicted to find the value of that claim. However, this model was unsuccessful for our purposes due to a failed classification.

From these four variations and their sub-types, OLS regression still performed the best in both methods. This was reasoned to be due to the severe underfitting of the models to the data, as seen by low Adjusted R-Squared values, and lower Test MSE than Training MSE.

2 Introduction

My team has been provided a data set of auto insurance policy holder characteristics with the objective of finding the most appropriate model for predicting *pure premiums*. However, the data set provided did not include the desired response variable, and instead, provided the *exposure*, *claim count*, and *claim incurred* (*loss*) of each policy holder, which was could be used to calculate the pure premium. Each team member was delegated a type of model to use to predict the *pure premium*, and this report is focused on linear regression models and a variety of variations. The variations of linear models explored were as follows Ordinary Least Squares (OLS) regression, OLS with shrinkage techniques, best subset selection, and a special conditional model. This report is to be coupled with a presentation that compares all the types of models explored by the team.

Before the models, the team decided how to calculate the pure premium predictions. Given the formula as the following:

$$PurePremium = \frac{claim.count}{Exposure} \times \frac{Loss}{claim.count} = frequency \times severity$$

The team elected to employ a two method approach to the predict the response variable.

Method 1: Modelling *frequency* and *severity* using separate models then multiplying the best predictions together to form the *pure premium* prediction. This is known as the "Frequency-Severity" Method and is supported by various academics and insurers due to greater interpretability of predictor variables on each response and flexibility (Shi P., Feng X., Ivantvoa A., 2015). This methodology is achieved by taking parts of the formula above to calculate *frequency* and *severity*, and it should be noted that when $claim.count = 0$, the undefined *severity* was simply converted to 0 as well.

Method 2: Using the formula above and adding in a new response variable to the data set simply known as *premium* and running predictive models.

Both methods explored throughout this report.

3 Data Exploration

Before creating a predictive model, a preliminary analysis was performed to discover any hidden details or concerns within the data. The data set comprised of 40,621 data points (policies) and 23 variables (excluding the "Index" column). Furthermore, the data consisted of continuous, categorical, and discrete variables, which needed to be considered when making a model. One of the first concerns with linear models is multicollinearity, which can be explored through Figure 1.

The correlation between a variety of variables is quite high and must be examined in further detail. Firstly, *vehicle.age* and *vehicle.value* were found to be highly negatively correlated (-0.7). Also, *yrs.licensed* and *ncd.level* was also quite highly positively correlated (0.6). Furthermore, there are four variable that have high correlation with each other and these are *cubic.cent*, *horse.power*, *weight*, and *length*.

Table 1: Correlation of 4 Related Predictors

	cubic.cent	horse.power	weight	length
cubic.cent	1.0000000	0.6401437	0.7164241	0.7223396
horse.power	0.6401437	1.0000000	0.7784995	0.6604775
weight	0.7164241	0.7784995	1.0000000	0.8418331
length	0.7223396	0.6604775	0.8418331	1.0000000

The correlations of *cubic.cent*, *horse.power*, *weight*, and *length* were all highly correlated with all being above 0.64. These correlation values were contextually understandable due to engine size determining much of

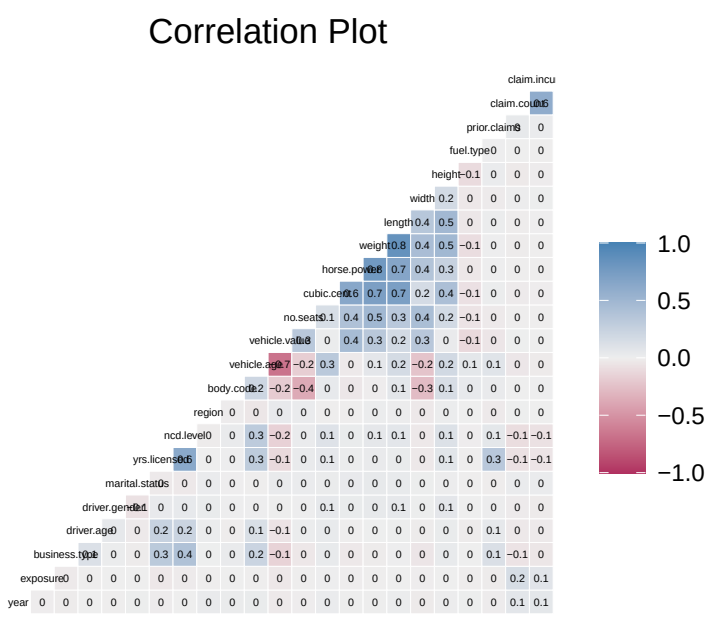


Figure 1: Correlation Matrix

the horse power and length of cars when engineered. By removing the highly correlated variables and only leaving one from each group, multicollinearity could be reduced for all of the models.

Another concern was the imbalance of data of the response variables. The histograms for *exposure*, *claim.count* and log(*claim.incurred*) were explored to find any poetential problems.

Both *claim.count* and *claim.incurred* conveys a large imbalance in data as shown by the skewed histogram. The zero-inflated data would negatively affect the accuracy of the models. The proportion was zero claim values was further analysed in the bar plot below.

As seen by the bar plots, there was a large disproportion between the number of policy holders with a claim and the number without. To be price, 92.21% of the policies had zero claims, and this negatively impacted many of the linear models as linear models work best with a normal distribution of the data.

4 Data Preparation

4.1 Data Splitting

A few steps were taken to prepare the data. Firstly, the team added columns for *severity* and *frequency* to using the formula in **Section 2**. To address the issue of NAs - from dividing by "zero"-valued claim count - in *severity*, all NAs were changed to zero as a zero *claim.count* would mean no *severity*. Then, the team had to ensure the training and testing sets used for all of the different models needed to be the same to ensure fair comparisons. A data partition of 70% was agreed upon by all team members. To do this, the team used the "set.seed(2020)" as shown in the code below:

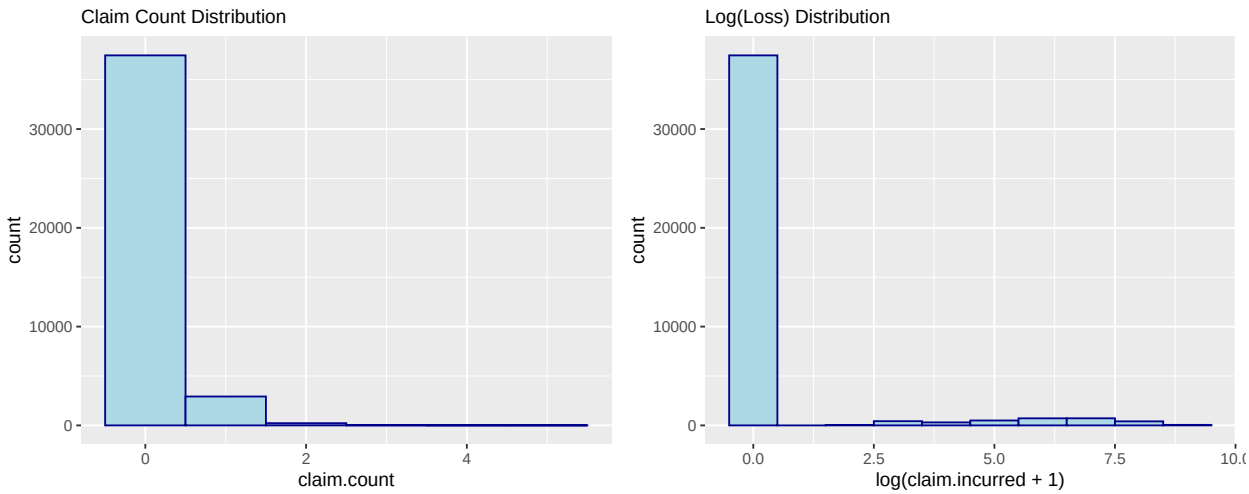


Figure 2: Histograms of Response Variables

```
set.seed(2020)
train_obs <- sample(nrow(A2_Data), floor(0.7*nrow(A2_Data)))
train_set <- A2_Data[train_obs,]
test_set <- A2_Data[-train_obs,]
```

4.2 Data Transformation

The categorical variables, which had data type "character", were changed into factors for analysis. Furthermore, separate data sets were created to make these categorical variable binary for use in certain models - for linear models this was Best Subset Selection. The code was as below:

```
Binary_Data <- A2_Data
Binary_Data[sapply(A2_Data, is.character)] <- lapply(A2_Data[sapply(A2_Data, is.character)], as.factor)
Binary_Data <- dummy.data.frame(Binary_Data)
```

5 Model Selection

The modelling method studied for the report was linear regression methods and different variations. A variety of variations were used to model *frequency*, *severity*, and *pure premium*, such as shrinkage techniques, and best subset selection. It should be noted that, when modelling, the variables *exposure*, *claim.count*, *claim.incurred*, *frequency*, and *severity* were removed from the train set as they were directly related to the computation to the response variables. The two main performance measures we looked at for our analysis were the Test MSE and the Adjusted R-Squared.

5.1 Ordinary Least Squares Regression

Ordinary Least Squares (OLS) Regression is essentially finding the line of best fit between the predictor variables and the response variable. This line of best fit is described by the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$$

OLS Regression is often praised for it's interpretability and low computational cost. It is also the first model performed for any predictive analytics, and there are a large amount of real world applications of this model.

5.1.1 Simple Linear Regression Model

The first linear model we studied was an OLS regression with all the "X" variables.

5.1.1.1 Frequency As shown in the summary in **Appendix 5.1**, the OLS *frequency* linear regression yields an interesting output and performance measures. This can be interpreted from the Adjusted R-squared which is 0.01196, meaning the variables only explain 1.19% of the model. The MSE for the model is 0.5930657, which is usually considered a good thing as the error is so low, however, this is due to the small range of possible *frequency* values being from 0 to 12 and the data is heavily zero inflated. The following variables were all statistically significant (p-value < 0.05) *year*, *business.typeRB*, *marital.statusMarried*, *yrs.licensed*, *body.codeB*, *vehicle.age*, *vehicle.value*, *no.seats*, *width*, and *prior.claims*, which which meant they had a high probability of actually explaining the Response Variable. Of these variables, cars with *body code B* got into less frequent accidents than other cars as shown by the relatively high coefficient (0.11). The residuals were then studied..

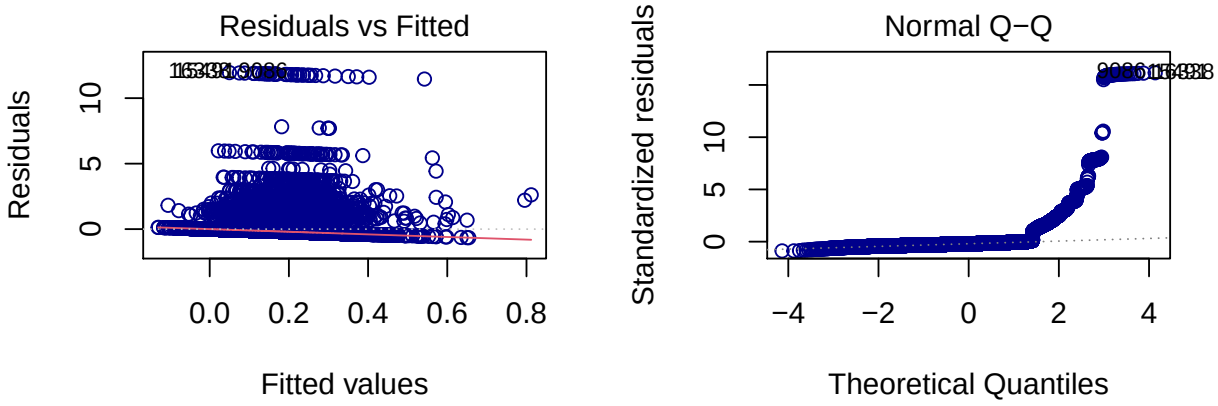
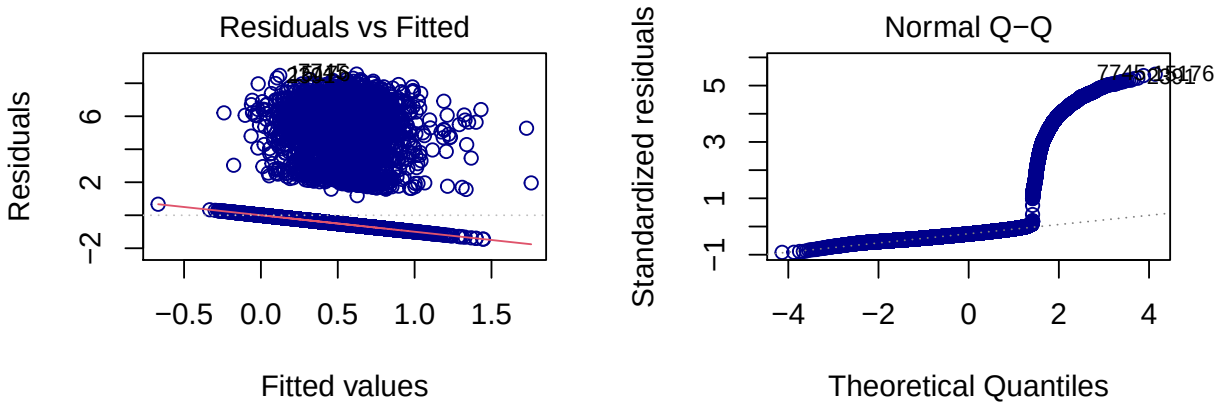


Figure 3: *left* - Frequency Residual vs Fitted. *right* - Frequency Q-Q Plot.

The left plot shows the models overfitting of the zero values, depicted by the red line going through the zero values. Above this red line is all the values that it failed to predict correctly (aka all nonzero values). The right plot is the Q-Q plot, which shows that the data is heacily skewed, as indicated by the sudden inflexion. This is reaffirming our data is not normally distributed.

5.1.1.2 Severity Looking at the summary of OLS for *severity* in **Appendix 5.2**, the model performs poorly on all metrics. There are 11 significant variables to the model but, the Adjusted R Squared and Test MSE demonstrate that this is a poor mode as they are 0.01043 and 127,126 respectively. Similar to the *frequency* model above, the model fits the zero values correctly, but anything above zero failed to be correctly predicted. This is further emphasised when taking the the model of *log(severity + 1)* and looking at the figure below.



We can see that there was an even clearer showcase of the skewed data. Regardless, taking the predicted values of *frequency* and *severity* and combining them for Method 1, it was found that the Test MSE was 896,096.

5.1.1.3 Premium We then inspected Method 2, where we simply ran the OLS regression on the *premium* response variable. The coefficients were than compared to those of *severity* and *frequency*:

The Test MSE for Premium was 880,376, and the Adjusted R-Squared was less than 1%. In addition, there were only 6 variables of statistical significance in this model. There was a marginal difference in the Test MSE of the Method 1 and Method 2, but Method 2 was the lower Test MSE. However, one major concern found from the models was the correlated variables discussed in prior sections. These variables will lead to over weighting of the predictors, which will cause over fitting and inaccuracy.

5.1.2 Null Model

Considering the poor results from the basic linear regression models, the null model was used to test the potential of linear models for this data set. The null model is derived from the null hypothesis. This was considered one of the benchmarks for the linear regressions methods as it represented randomness of the relationship between predictors and response.

Table 2: Null Model Test MSE Benchmarks

Response	MSE
Frequency	0.6
Severity	128273.0
Method1	899018.0
Method2	887136.0

If the models had higher Test MSEs than the Null Model, then the model would be deemed as no better than random.

5.1.3 Multicollinearity Removed

In **Section 2**, it was noted that there were highly correlated variables that would need to be addressed due to multicollinearity being an issue for linear models. The variables *vehicle.age*, *ncd.level*, *horse.power*, *weight*, and *length* were removed from the data set with the intention of improving the linear model.

The correlated variables were removed, the *frequency* test MSE (0.593342) was higher than the previous model's, but the *severity* test MSE was lower. However, most importantly, when looking at the individual *premium*, the model had an improved test MSE of 880,331, which is a difference of 45. Regardless of these improvements, the R-Squared for these models were still lower than 0.01, conveying unreliability for predicting *premiums*. In addition, the train MSE (972,114) was still higher than the test MSE, indicating there is underfitting for the model. Given the *premium* result and the underlying concerns multicollinearity can have for many linear models and that my team came to the conclusion that they could be detrimental to the overall model, the correlated variables were removed by re-adjusting the train set and test set (with the following code). Though the shrinkage methods can deal with multicollinearity, the performance measures indicated that they performed better without the correlated variables at all.

```
train_set <- train_set %>% select(-vehicle.age, -cubic.cent, -horse.power, -length, -ncd.level)
test_set <- test_set %>% select(-vehicle.age, -cubic.cent, -horse.power, -length, -ncd.level)
```

5.2 Shrinkage Techniques: LASSO, Ridge, and Elastic Net

Shrinkage techniques are used to reduce overfitting of the model to the training data by penalizing the coefficients with by introducing a tuning parameters/hyperparameters (lambda) to manipulate the values of the coefficients. There are three types of shrinkage techniques commonly used and they are LASSO, Ridge, and Elastic Net, which all have different capabilities. To find the optimal tuning parameter, the training data was partitioned into "10-Folds" and cross-validation (CV) was used to find the lambda with the lowest CV error. One of the major advantages of shrinkage techniques is their ability to address multicollinearity by effectively reducing coefficient size. However, for the purposes of more reliable modelling and comparison amongst team members (as discussed in the previous section), correlated predictor variables were removed prior to this.

5.2.1 LASSO

The first shrinkage method investigated was LASSO Regression. This method is know for its innate ability to perform feature selection by reducing variables to zero value coefficients - known as absolute shrinkage. The LASSO formula is as follows:

$$minimize\{SSE + \lambda \sum_{j=1}^{p} |\beta_j|\}$$

LASSO models were fitted for all three response variables and the these were the training MSE and testing MSE values:

Table 3: LASSO MSE Comparisons

Type	Train_MSE	Test_MSE
Frequency	0.5475002	0.5936
Severity	138989	127044
Combination	980125	899004
Premium	972544	880552

In all cases, the test MSE was lower than the train MSE, depicting underfitting of the model, a common trend in all the shrinkage techniques. In addition, *frequency* and *premium* both had higher test MSE than the OLS with correlated variables removed, however, the *severity* LASSO model resulted in a lower test MSE than all prior and future models.

The following plots depict the feature selection of LASSO performed:

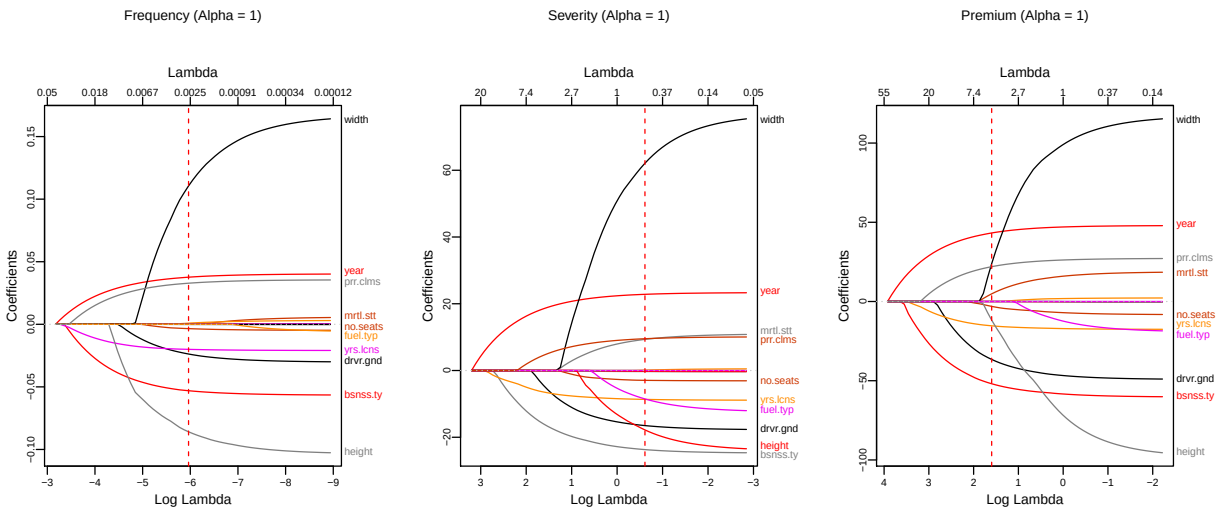


Figure 4: The plots show the relationships between tuning parameter (*lambda*) and the predictor coefficients. As LASSO performs natural feature selection, the variables are reduced to zero as the *lambda* become larger. The red line marks the optimal *lambda* value.

The plots above in Figure above demonstrate the behavior of the coefficients as the lambda value is increased. The red line marks where the lambda with the lowest cross validation occured - from left to right the minimum lambda values were 0.00257, 0.54218, and 4.93790. The *frequency* model removed 2 variables *region* and *fuel.type*, and *severity* removed 1 variable *region*; indicating they were not relevant for the predicting the response variable. However, LASSO Regression with response variable *premium* removed a total of 6 variables, *driver.age*, *region*, *body.code*, *vehicle.value*, *weight*, and *fuel.type*. These 6 variables were not correlated, however, they increased the CV error, indicating they were reducing the model accuracy if left in the model. One disadvantage of LASSO regression is that there are no p-values to study, meaning we cannot determine the statistical significance of these variables.

5.2.2 Ridge Regression

Similar to LASSO, Ridge regression penalizes the coefficients by adding a tuning parameter, however, cannot reduce the coefficient of a variable to zero, and hence does not possess inherent feature selection. As lambda is pushed to infinity, the coefficients are also pushed towards zero, but will never be zero. In the same fashion as LASSO, the optimal lambda value is found by finding the lambda with the lowest CV error.

$$minimize\{SSE + \lambda \sum_{j=1}^{p} \beta_j^2\}$$

Ridge models were fitted for all three response variables and the these were the training MSE and testing MSE values:

In all cases, the test MSE was lower than the train MSE, demonstrating underfitting of the model. Furthermore, ridge regression underperformed compared to Ordinary Least Squares Regression, accentuating the underfitting of the model.

The following plots depict the coefficients of Ridge performed:

Table 4: Ridge MSE Comparisons

Type	Train_MSE	Test_MSE
Frequency	0.5475	0.5937
Severity	138990	128341
Combination	991355	898994
Premium	991355	880481

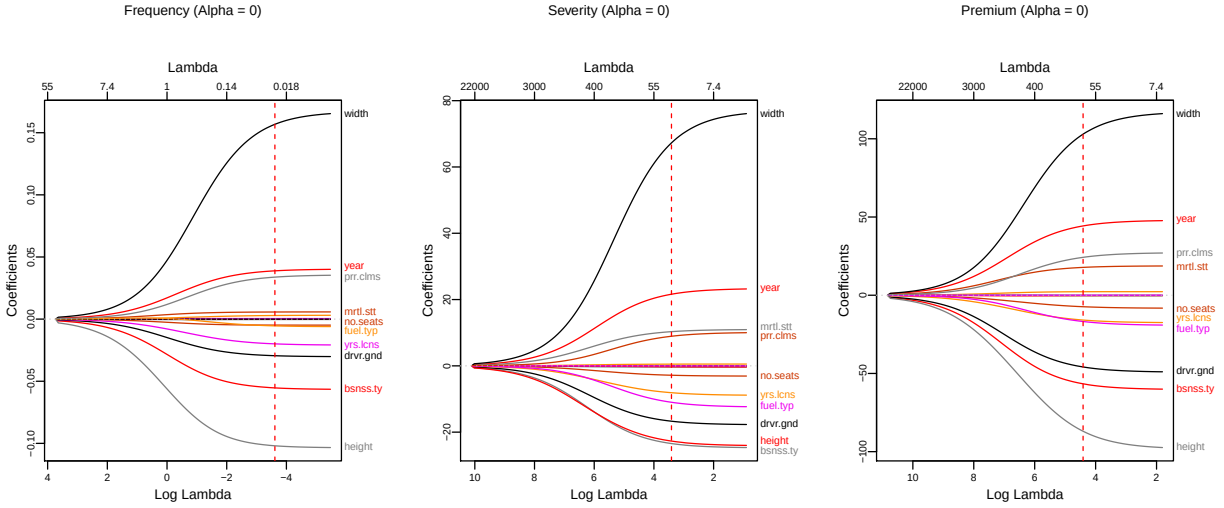


Figure 5: The plots show the relationship between the tuning parameter (*lambda*) and the predictor coefficients.

Unlike LASSO regression, ridge regression does not perform feature selection as it cannot shrink coefficients to 0. From Figure 4.4 we can see it can shrink coefficients extremely close to zero. However, we can see that the lambda value where CV error is lowest is actually where coefficients are larger than OLS regression for both *severity* and *premium*, which essentially means the hyperparameter is effectively NOT regularizing the variables. Studying the coefficients, we can see *width* was considered the variable with the biggest impact as it had the largest coefficient, and *no. of seats* seemed to have very little importance. However, one major disadvantage of shrinkage techniques is that there is no academically established measure of significance (like p-value for OLS) of the predictors.

5.2.3 Elastic Net Regression

The Elastic Net shrinkage technique is a combination of LASSO and Ridge regression as depicted by the formula:

$$minimize\{SSE + \lambda_1 \sum_{j=1}^{p} \beta_j^2 \ + \lambda_2 \sum_{j=1}^{p} |\beta_j|\}$$

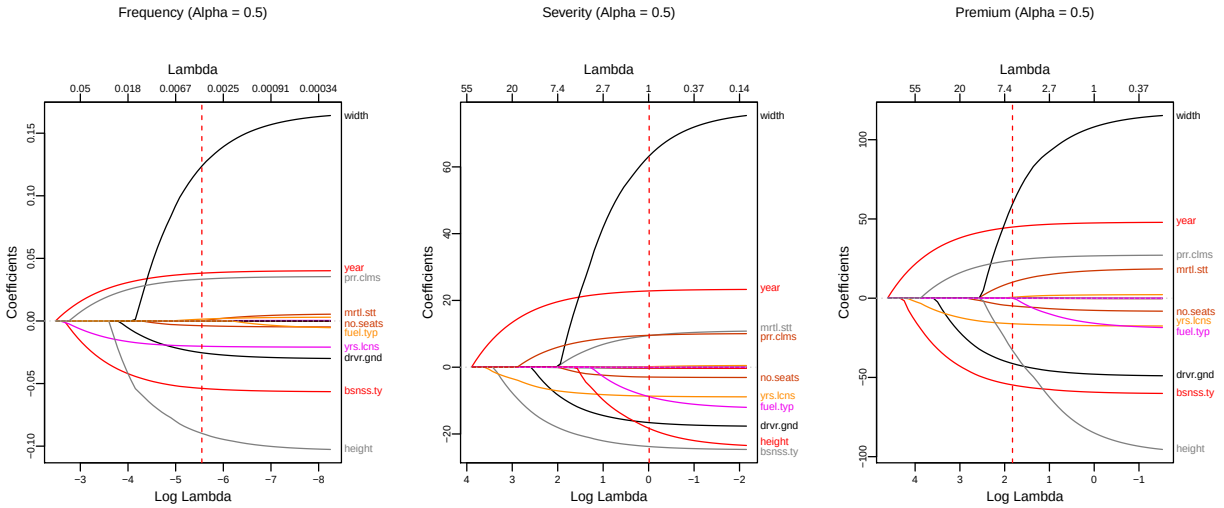
Elastic Net can still perform feature selection like LASSO. Furthermore, one of the benefits of this is when there are two highly correlated variables, one of the variables could be reduce to zero whilst the other will still be kept, but with a coefficient close to zero.

The results of Elastic Net were as follows:

Table 5: Elastic Net MSE Comparisons

Type	Train_MSE	Test_MSE
Frequency	0.5936	0.5475
Severity	138996	127065
Combination	991376	896631
Premium	972523	880710

Another aspect to Elastic Net is the "alpha" term, which essentially dictates whether the model will act more like LASSO regression (alpha = 1) or Ridge regression (alpha = 0). For the purposes of this study a balance combination was used with alpha = 0.5. The plot below conveys the effects of the shrinkage technique:



The figure above shows that the mix of LASSO and Ridge is working effectively in *frequency* as we see both the values going directly to zero (LASSO) and those approaching zero (Ridge), and the goodness-of-fit is seen through the Elastic Net model yielding the lowest MSE for *frequency* of all shrinkage types as well as OLS. Observing the coefficients, we can see that feature selection has been used to remove *region* from all three response variable models, identifying the variable as ineffective at measuring the response variables.

5.2.4 Summary of Shrinkage Techniques

The most successful shrinkage techniques was the LASSO for *severity* and the combination method, Ridge for *premium*, and Elastic Net for *frequency*. The advantages of shrinkage techniques are the ability to shrink, and, in the case of LASSO and Elastic Net, remove variables that negatively impact the accuracy of the model through cross validation. However, this is a slight trade off for interpretability due to the absence of p-values. Studies found potential in a new technique, but is still being explored. Furthermore, these shrinkage techniques underperformed compared to OLS without shrinkage techniques, and this could be due to a few reasons. One potential reason is that the OLS model is already severely underfitting the training data, as seen by the Train MSE > Test MSE, and shrinkage techniques are used to prevent overfitting. Another potential reason is the predictors being poor variables for forecasting the response variable. This will be further discussed in the Section 6.

5.3 Best Subset Selection

Best Subset Selection (BSS) is known as the "all possible regression" model as it takes every combination of predictor variables and find the combination that yields the lowest error score. This is notably better than its counterparts Stepwise function that simply removes and and adds functions to a model one at a time based on statistical significance. BSS requires that the categorical variables are all changed to have dummy variables, leading to a total of 35 predictors. This means there are 34,359,738,368 (2^{35}) possible combination, which is very computationally intense. Furthermore, the Best Subset selection can be based on a variety of performance measures such as Adjusted R-Squared, Cp, and BIC. For the purposes of this report Cp was used to as the BSS measure.

The BSS model was run and the results were as follows:

For *frequency* (**Appendix 5.3.1**), there were nine coefficients that yielded the lowest Cp Error and these were *year*, *driver.age*, *marital.status (Divorced)*, *body.codes (A, B, E)*, *horse.power*, *length*, and *fuel.type (LPG)*. According to this model, cars with *body.code* type "E" would get into the least accidents, and cars with *fuel.type* "LPG" and younger cars (*year*) were more likely to get into a car accident. However, it's important to note that the R-Squared value is 0%, indicating that our predictor variables do not explain our response variable. Furthermore, the Test MSE has been the highest thus far for *frequency* at 0.5974.

Similarly, *severity* (**Appendix 5.3.2**) had 9 variables it selected and these were *year*, *business.type (NB)*, *marital.status (Divorced)*, *body.codes (B)*, *horse.power*, *weight*, *length*, *prior.claims* and *fuel.type (LPG)*. The largest positive coefficient was policy holders that were "New Business". Contextually, this could make sense as those getting new insurance could be because they are new drivers. Unfortunately, the Adjusted R-Squared value is 1%, and the Test MSE is relatively high compared to other models (127,624), meaning that this is not a good model for severity.

Looking at Method 1: Combining the results of *frequency* and *severity* resulted in a Test MSE of 897,818 for the *premium*, which was one of the higher Test MSE values.

Looking at Method 2 (**Appendix 5.3.3**): Seven variables were identified as the best combination of predictors to forecast premium and they were *year*, *driver.age*, *marital.status (Divorced)*, *body.codes (B)*, *length*, *prior.claims*, and *fuel.type (LPG)*. The resultant Test MSE for this model was 884,562, the highest of all the models thus far.

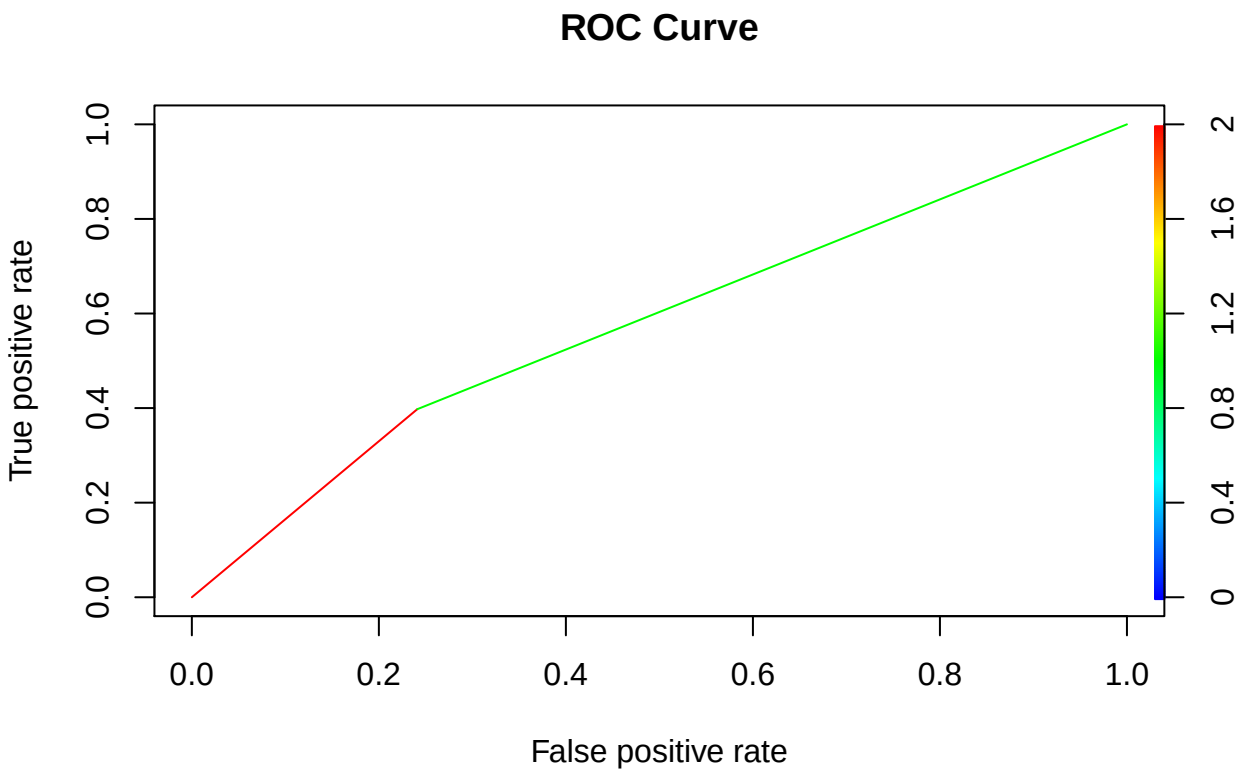
In summation, the Best Subset Selection Method did not find us the best model for any of our response variables, and the method underperformed. Common practice in the workplace is that BSS is a worse version of LASSO, but many argue that BSS can still be better due to selecting unbiased coefficient estimates for its selections

5.4 Conditional Model

Due to the overall poor results of the previous model, research was conducted to find ways to improve the model. As discussed in Section 3 and Section 5.1, the linear models heavily suffered from the zero-inflated values, hence new methods were looked into. One method that was considered was removing all the zero-values from the data, however, this would be removing nearly 85% of the data from our models.

The Conditional Model took the following approach: 1. Add a new binary variable that assigned value '0' when the response variable was the over inflated value (in this case zero), and '1' if greater than this value (any value greater thean zero) 2. Run a Logistic Regression to classify the test response variables into 0 or 1 with a reduced threshold to make up for the imbalance 3. Run a regression on just the values identified as 1 to predict the values

This method was attempted attempted for the project, however, there was no luck due to failed classification.



As depicted by the ROC curve and the AUC of 0.58, the logistic regression failed to classify the values correctly. Considering the prior success with the model as well as other academics, the Conditional Model isn't a bad model, but for our purposes it did not succeed.

6 Model Assessment

The best model among the linear methods was then deliberated upon based on prior analysis and Test MSE. The table below encapsulates the results.

Table 6: Test MSE Comparison

Model	Frequency	Severity	Method1	Method2
OLS	0.59310	127126	896096	880376
OLS Correlated Variables Removed	0.05933	127099	896350	880331
LASSO	0.59360	127044	896548	880552
Ridge	0.59370	128341	898994	880481
Elastic Net	0.54750	127065	896631	880710
Best Subset Selection	0.59740	127624	897818	884562

The key results were: - The best linear regression model for *frequency* was found to be the Ordinary Least Squares Regression with Elastic Net model. - The best linear regression model for *severity* was found to be the Ordinary Least Squares Regression with LASSO. - The best linear regression model for Method 1 - combining *frequency* and *severity* models - was Ordinary Least Squares with all variables. - The best linear regression model for Method 2 - response *premium* alone - was Ordinary Least Squares with correlated variables removed.

Though Shrinkage Techniques appeared better at modelling *frequency* and *severity* alone, they underperformed when it came to the *premium* response variable. In general practice, Shrinkage Techniques should improve the base model, but what we found here was that the model was severely underfitting the data, and Shrinkage techniques caused even further underfitting. Furthermore, all of the Adjusted R-Squared values were approximately 1%, meaning the predictor variables only explained approximately 1% of the response variable.

When deciding between Method 1 and 2 for linear methods, Method 2 prevails as it resulted in a lower Test MSE. However, it is not recommended to use OLS regression methods for predicting pure premiums due to the overall poor results - at least for this data set.

7 Conclusion

In conclusion, the linear methods did not perform well with our data set, and was not the recommended model to use.In saying this, OLS without any adjustments performed the best out of our models, accentuating it's potential usage over the "improved" versions. Auto insurance has often used GLM as their predictive model as GLMs are better at dealing with skewed data, something OLS regression fails to do.

8 Appendix

8.0.1 Appendix 4.1

8.0.2 Appendix 5.1.4

For *frequency*:

```
## [1] 0.6000142
```

For *severity*:

```
## [1] 128272.7
```

For *premium*

```
## [1] 899018.2
```

```
## [1] 887136.4
```

Appndix 5.3.1

Observations	28434
Dependent variable	freq
Type	OLS linear regression

F(9,28424)	13.00
R^2	0.00
Adj. R^2	0.00

	Est.	S.E.	t val.	p
(Intercept)	-80.54	8.51	-9.46	0.00
year	0.04	0.00	9.50	0.00
driver.age	-0.00	0.00	-3.62	0.00
marital.statusDivorced	0.03	0.03	0.84	0.40
body.codeA	-0.01	0.01	-0.64	0.52
body.codeB	-0.10	0.04	-2.94	0.00
body.codeE	-0.01	0.01	-1.11	0.27
horse.power	-0.00	0.00	-0.15	0.88
length	-0.01	0.02	-0.78	0.44
fuel.typeLPG	0.04	0.06	0.77	0.44

Standard errors: OLS

Appendix 5.3.2

Appendix 5.3.3

Observations	28434
Dependent variable	sev
Type	OLS linear regression

F(9,28424)	21.48
R^2	0.01
Adj. R^2	0.01

	Est.	S.E.	t val.	p
(Intercept)	-47242.26	4277.36	-11.04	0.00
year	23.45	2.12	11.06	0.00
business.typeNB	36.07	4.76	7.58	0.00
marital.statusDivorced	-1.42	14.95	-0.10	0.92
body.codeB	-24.82	17.71	-1.40	0.16
horse.power	-0.19	0.18	-1.05	0.30
weight	0.01	0.02	0.24	0.81
length	-2.61	11.49	-0.23	0.82
prior.claims	6.35	1.67	3.79	0.00
fuel.typeLPG	-16.14	29.11	-0.55	0.58

Standard errors: OLS

Observations	28434
Dependent variable	prem
Type	OLS linear regression

F(7,28426)	18.94
R^2	0.00
Adj. R^2	0.00

	Est.	S.E.	t val.	p
(Intercept)	-97176.21	11306.46	-8.59	0.00
year	48.23	5.61	8.60	0.00
business.typeNB	82.17	12.57	6.54	0.00
marital.statusDivorced	-8.43	39.51	-0.21	0.83
body.codeB	-53.19	46.27	-1.15	0.25
length	-6.62	16.50	-0.40	0.69
prior.claims	19.98	4.43	4.51	0.00
fuel.typeLPG	-0.44	76.86	-0.01	1.00

Standard errors: OLS