ALLSTATE INSURANCE CLAIMS SEVERITY PREDICTION

Final Report (spring 2017)

How Severe is an Insurance claim?

Name- Revanth Reddy Katanguri

Instructor- Michael Hahsler

Southern Methodist University

Contents

List of figures:

EXECUTIVE SUMMARY

When you've been devastated by a serious car accident, your focus is on things that matter most: family, friends and other loved ones. The last thing you wanted to do is to spend your time or mental energy with your insurance agent doing all the paperwork. Allstate, a personal insurer in the United states, posted this project in Kaggle as the company is seeking some fresh ideas to improve their claims service for over 16 million households they protect. The idea to predict the claims of the people during their tough times found interesting to me and went ahead to solve this problem.

This report aims to provide a clear understanding of working with a dataset which contains categorical and continuous variables, understanding and preparing the data to build a model, evaluating the models by metrics like RMSE, R-squared to give a method which will be the best fit to predict the outcomes. As the outcome, which is to be predicted is a continuous variable(LOSS), this whole problem comes under a prediction problem. Linear Regression and Xtreme Gradient Boosting (XgBoost) are the two methods used to build the models. Data is well prepared by applying the transformations on the original given data, K-fold cross validation technique is used to check the performance measures of the two methods which are used.

Findings indicate that Xgboost (tree booster) performs well when compared to other models which are built in this study. This report follows the format of CRISP-DM. The training error (R.M.S.E) was reduced by 20.5% after creating a model based on Xgboost(tree boosting), the results of Cross- validation also adds to the fact that xgboost is the best performing model on the given data. The report ends with forecasting the solution and concluding the study with some limitations.

• Problem Description

This report aims to solve the problem of predicting the claims of the customers of Allstate Insurance company, and thus proving the answer to the question "How severe is an insurance claim?". This solution to this problem has a lot of importance as there will be a lot at stake like the lives of the customers, Money, time. So, if this problem can be addressed as good as possible, we can save a lot of precious things like money, time and life of a customer. I have tried as much as possible to get the best possible solution.

• Business Goal

With the help of data science and with this report I wish I could provide two best models for predicting the severeness of an insurance claim to all the people who are interested in this topic and to the insurance companies.

- Clients: Allstate Insurance Company, other personal insurers.
- Stakeholders: Customers, Insurance Companies.
- Opportunity: To get in depth analysis on Linear regression, Xtreme gradient Boosting and to apply several packages in Rstudio to improve the visualizations.
- Challenges: To provide two best models to predict the claims.
- Humanistic implication: Can save lot of time, money, mental energy during tough times for the customers of an insurance company.

• Data description

The Data is collected from Kaggle (https://www.kaggle.com/c/allstate-claims-severity), training and testing data sets were provided. The data set contains of 188318 observations with 116 categorical variables, 15 continuous variables in which the variable "Loss" which is of our interest (variable to predict) is present.

	id	cat1	cat2	cat3	cat4	cat5	cat6	cat7	cat8	cat9	cat10	cat11	cat12	cat13	cat14	cat15	cat16	cat17
1	4	A	B	A	A	A	A	A	A	B	A	B	A	A	A	A	A	A
2	6	A	B	A	B	A	A	A	A	B	A	A	A	A	A	A	A	A
3	9	A	B	A	B	B	A	B	A	B	B	A	B	B	B	A	A	A
4	12	A	A	A	A	B	A	A	A	A	A	A	A	A	A	A	A	A
5	15	B	A	A	A	A	B	A	A	A	A	A	A	A	A	A	A	A
6	17	A	A	A	A	B	A	A	A	A	A	A	A	A	A	A	A	A
7	21	B	A	A	A	B	B	A	A	A	A	A	A	A	A	A	A	A
8	28	B	B	A	A	A	A	A	A	B	A	A	A	B	A	A	A	A
9	32	A	B	A	A	A	A	A	A	B	A	A	B	A	A	A	A	A

Figure 1 screenshot of Categorical variables in the data

cont1	cont2	cont3	cont4	cont5	cont6	cont7	cont8	cont9	cont10	cont11	cont12	cont13	cont14	loss
0.7263	0.245921	0.187583	0.789639	0.310061	0.718367	0.33506	0.3026	0.67135	0.8351	0.569745	0.594646	0.822493	0.714843	2213.18
0.330514	0.737068	0.592681	0.614134	0.885834	0.438917	0.436585	0.60087	0.35127	0.43919	0.338312	0.366307	0.611431	0.304496	1283.6
0.261841	0.358319	0.484196	0.236924	0.397069	0.289648	0.315545	0.2732	0.26076	0.32446	0.381398	0.373424	0.195709	0.774425	3005.09
0.321594	0.555782	0.527991	0.373816	0.422268	0.440945	0.391128	0.31796	0.32128	0.44467	0.327915	0.32157	0.605077	0.602642	939.85

Figure 2 screenshot of Continuous variables in the data

• Data preparation

The datasets which are given doesn't contain any missing values, but few continuous variables contain outliers which are fixed by following several steps.

- Step 1: To check the presence of outliers in the 14 continuous variables, boxplots are plotted on each of the variable to detect the outlier existence.

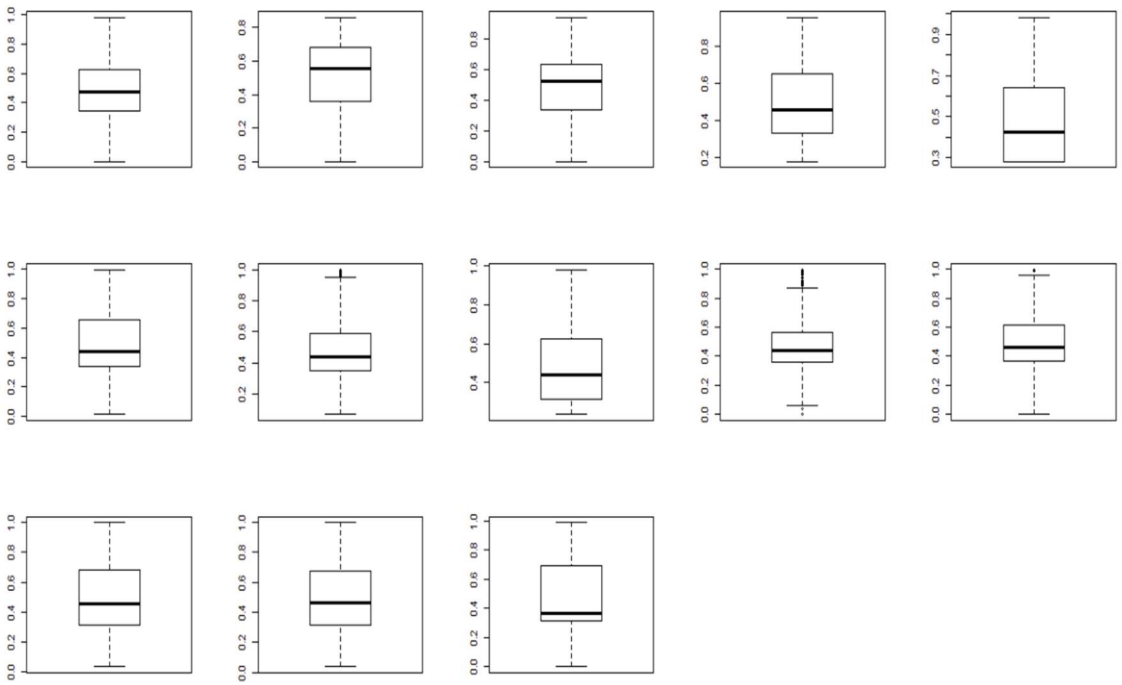


Figure 3 Boxplots for Cont. variables

From the above plot it can be understood that Cont. variables 7, 9 and 10 has the outliers.

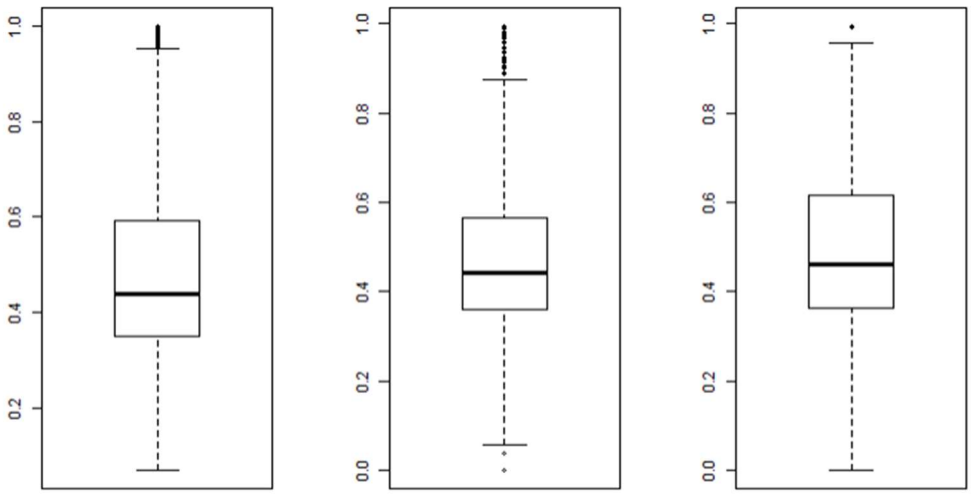


Figure 4 Cont. variables 7,9,10 which has outliers

- Step 2: Now to fix those continuous variables 7,9 and 10, I've used a technique called capping, which is nothing but all the outlier values are capped to its nearest possible value. This can only be done after having an understanding on the distribution of the variables. To get an idea of distribution "quantile" measure is used.

```
t2<-quantile(allstatetrain$cont9,prob=c(0.01,0.05,0.93,0.94,0.95,0.96,0.97,0.985,0.99,1.0))
t2
```

```
     1%      5%     93%     94%     95%     96%     97%   98.5%     99%    100%
0.21374 0.28066 0.86398 0.90411 0.91644 0.92444 0.93383 0.94438 0.96909 0.99540
```

In this way after getting the idea of distribution the outliers are capped by using the following code:

```
allstatetrain$cont9_new<-ifelse(allstatetrain$cont9<t2[1],t2[1],ifelse(allstatetrain$cont9>t2[3],t2[3],allstatetrain$cont9))
```

In this way, all the three variables are fixed and are now without any outliers.

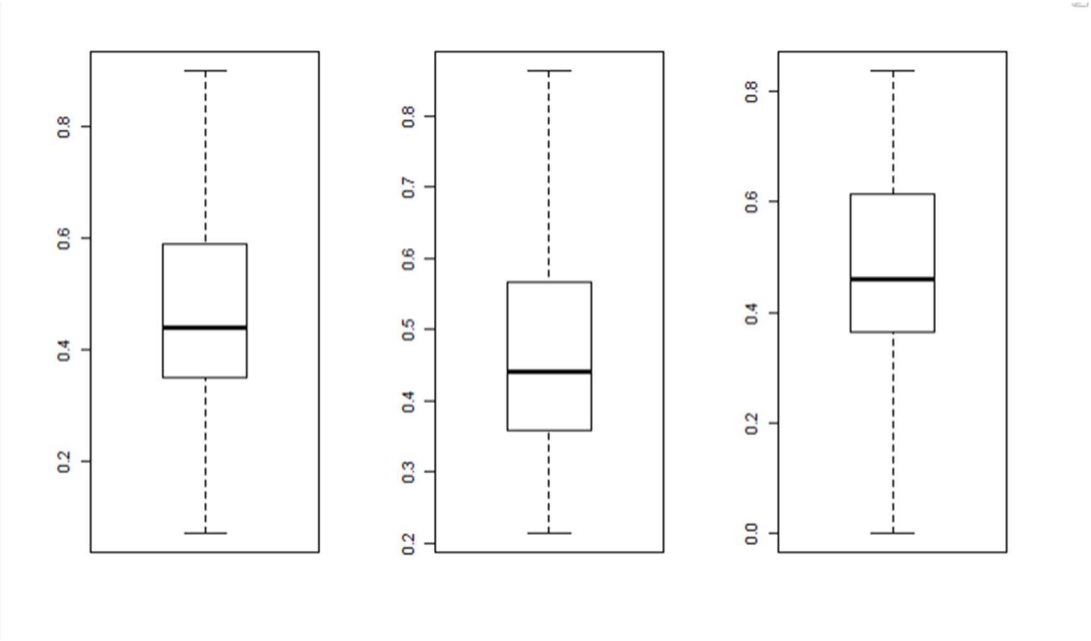


Figure 5 fixed Cont. variables 7,9,10

- Step 3: The distribution of Loss variable (which is to be predicted) is checked.

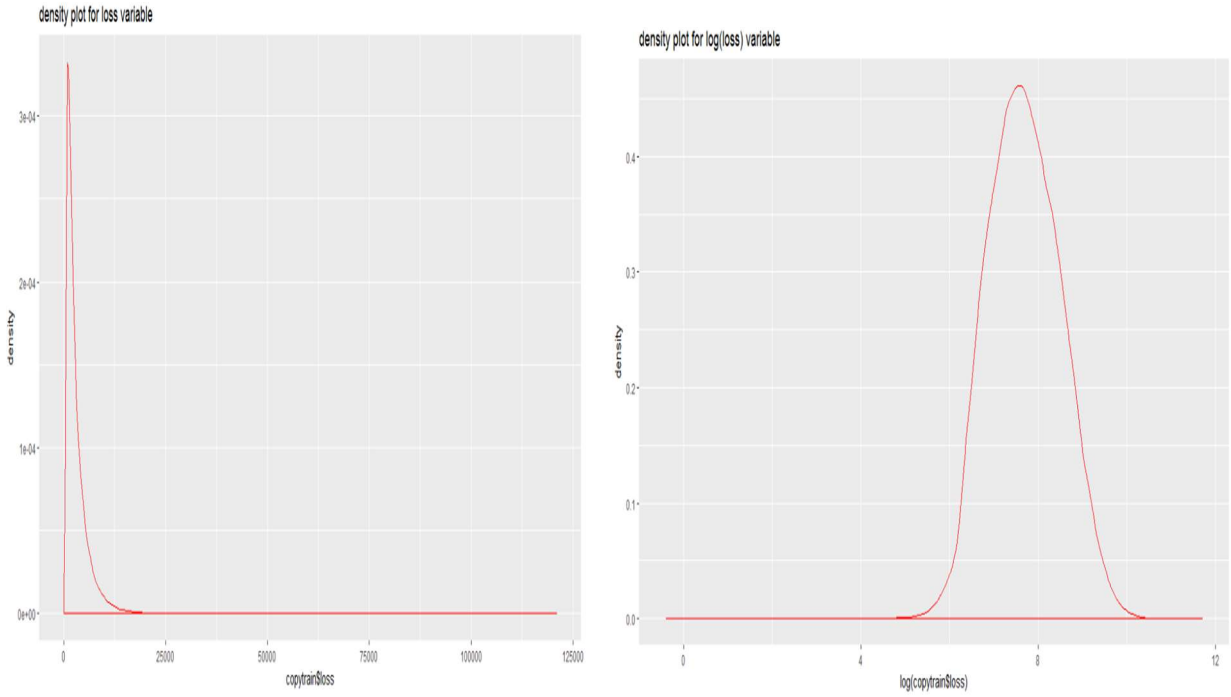


Figure 6 log transformation of Loss variable

The loss variable is transformed from skewed distribution to normal distribution by log transformation.

• Data Exploration

Correlation plot:

Though not much information is given about the variables in the dataset, to get the understanding of the relationship between the variables, a correlation plot has been plotted.

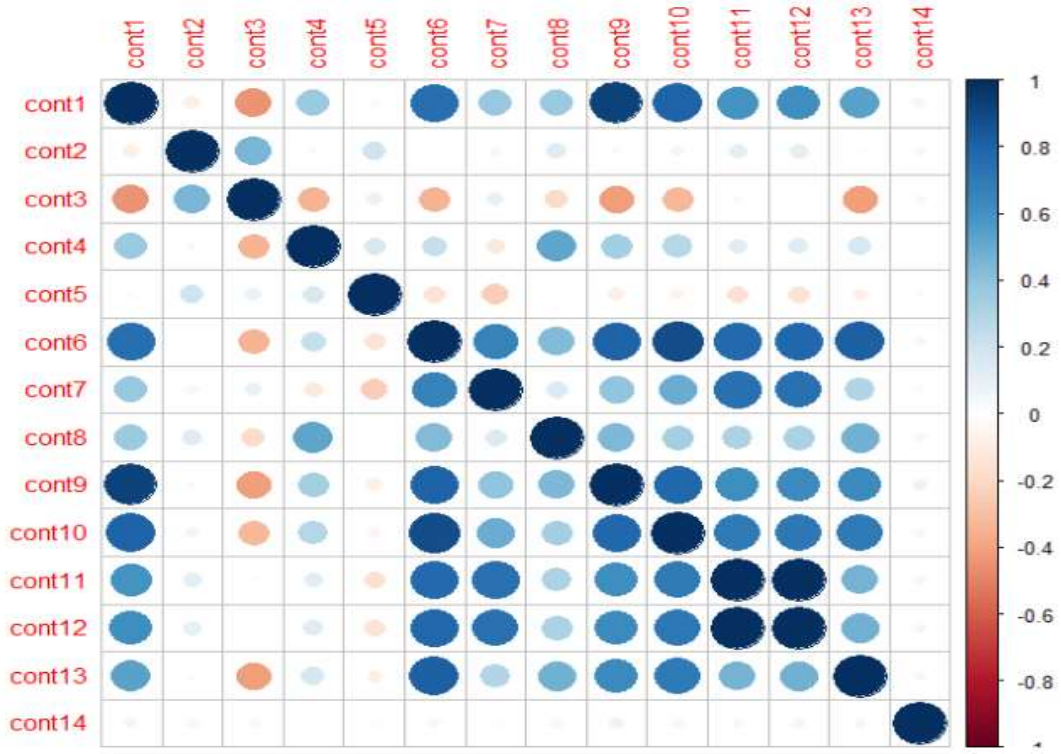


Figure 7 correlation plot of cont. variables

From the above plot it can be understood that:

Cont11 and Cont12—has a very strong positive correlation (+).

Cont9 and Cont1------has a positive correlation (+).

Cont3 and Cont1------has a negative correlation (-).

Zero variance and Near zero variance predictors-

To check the variance in the variables nearZeroVar() has been used and the variables which have the highest frequency ratio haven't been used in the model creation.

```
head(zero.var[zero.var$nzv==TRUE,])
```

	freqRatio <dbl>	percentUnique <dbl>	zeroVar <lgl>	nzv <lgl>
cat7	40.17140	0.001062033	FALSE	TRUE
cat14	81.70444	0.001062033	FALSE	TRUE
cat15	5537.76471	0.001062033	FALSE	TRUE
cat16	28.08386	0.001062033	FALSE	TRUE
cat17	142.86402	0.001062033	FALSE	TRUE
cat18	189.79838	0.001062033	FALSE	TRUE

After going through all the variables cat15, cat22, cat62, cat70 has the highest frequency ratio i.e. they have less variance when compared to other variables. So, theses variables weren't used in the model creation.

- ## Modelling

Linear Regression***:***

Linear regression has been performed by using all the variables except zero-variance predictors, and the "Loss" variable is chosen as the one to predict:

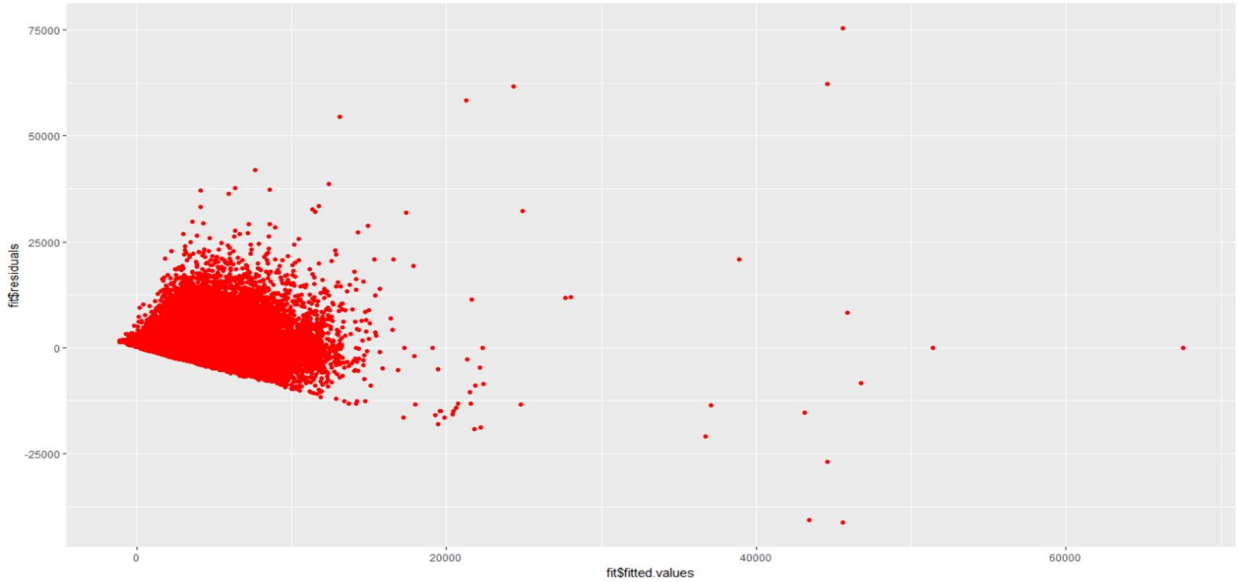


Figure 8 fitted v/s residual plot without log transformation

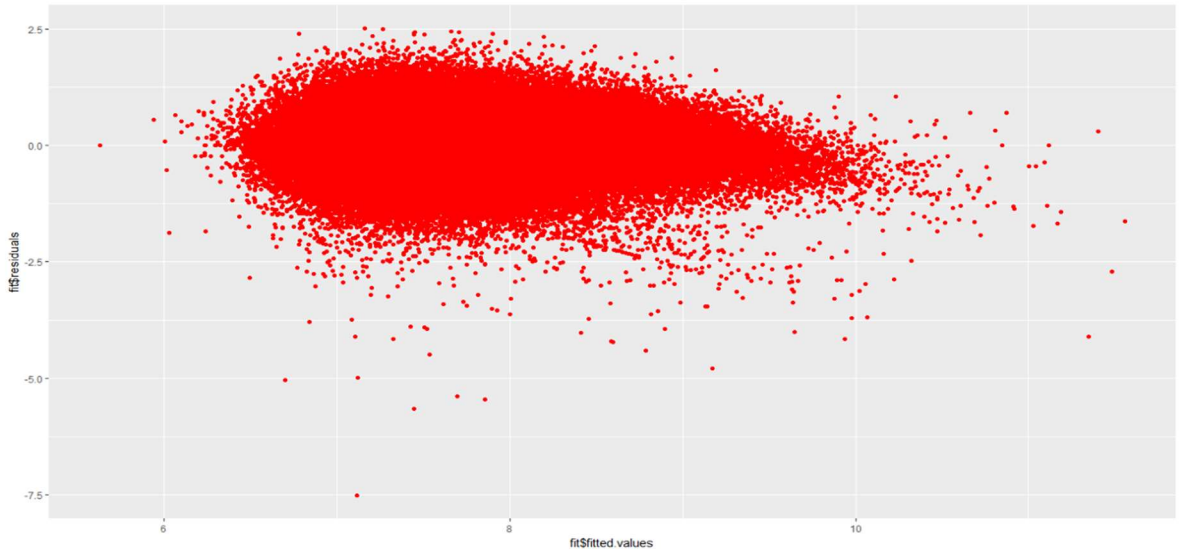


Figure 9 fitted v/s residual plot with log transformation

From the above two generated plots Fig 9 makes more sense than Fig 8 as more residual values are around the value "0" in fig 9 which is not the case with fig 8 (without log transformation of Loss Var).

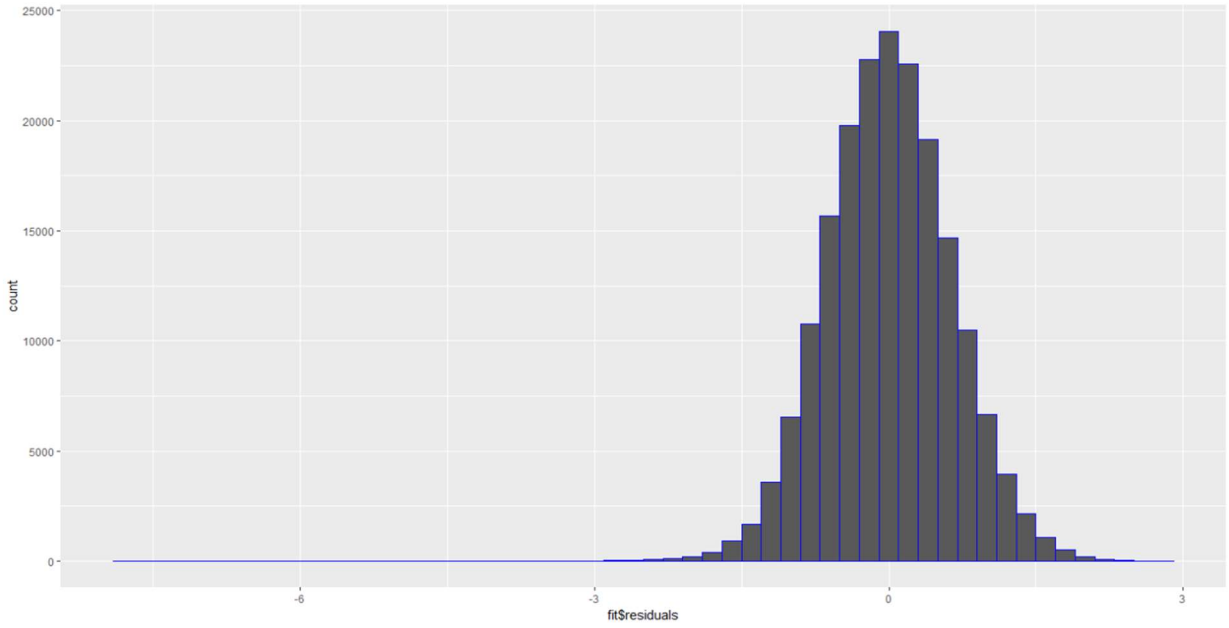


Figure 10 residual Freq plot for Log(loss) model

The above plot gives a clear understanding of the frequency of the values of residuals for the model with "Log(Loss)" as the variable to predict. Most of the

residuals have the value “0” or around zero, which makes this model reliable and genuine.

Results of Linear regression for two models: (Loss= variable to predict)

Table 1 linear regression results for two models

	Log(Loss)	Loss
Residual standard error (R.M.S.E)	0.5627	1999
Multiple R-squared	0.522	0.5288
Adjusted R-squared	0.5196	0.5264

The results suggest that the model with Logarithmic transformation has better results as R.M.S.E and R-squared measures are good for the first model.

Note – even though the R-squared measure is almost equal for both the models, after going through the residual models and R.M.S.E, model with Log transformation looks good.

Xtreme Gradient Boosting (XgBoost):

XgBoost is one of the most used and successful method in several Data science competitions. I have tried to use this method to get a better model than which I got with linear regression. Two boosting approaches have been used in this method a) Tree boosting b) Linear boosting.

A table showing the results of the two boosting methods, the model is made to run several times by changing the parameters.

Results for XgBoost:

Table 2 xgboost results for both the models

S.no		Tree boosting	Linear boosting
1	R.M.S.E-	0.509	0.56235
2	R.M.S.E-	0.512	0.56274
3	R.M.S.E-	0.457	0.56278
4	R.M.S.E-	0.447	0.5627

Results indicate that the model with R.M.S.E =0.447(tree boosting) fits well with the training data.

```
[196]   train-rmse:0.449835
[197]   train-rmse:0.449349
[198]   train-rmse:0.449141
[199]   train-rmse:0.448441
[200]   train-rmse:0.447875
```

Figure 11 o/p of the tree boosting model at the end of 200 rounds.

parameters used for this model are

[nrounds=200, eta=1, max_delta_step=100, max_depth=10,

min_child_weight=100, subsample=0.75].

Note - In both the methods number of rounds= 200 are kept constant.

Another thing which is interesting is that results of tree boosting are better than the results of linear regression but the results of linear boosting, in every case the R.M.S.E ~ 0.5627 which is the same with the R.M.S.E of linear regression in table 1.

Figure 12 plot for the tree boosting in XgBoost

The above plot is just to show how the tree boosting looks in xgBoost. As there is a lot of Data, number of features, the above plot is just the tree after completing 2 rounds. The final model is built after completing 200 rounds.

- ## Cross validation (Evaluation)

Table 3 cross validation results of three methods

Metrics	Linear Regression	Xgboost(Tree boosting)	Xgboost (linear boosting)
R.M.S.E	0.57987	0.54608	0.5676
R-squared	0.4986	---------	------

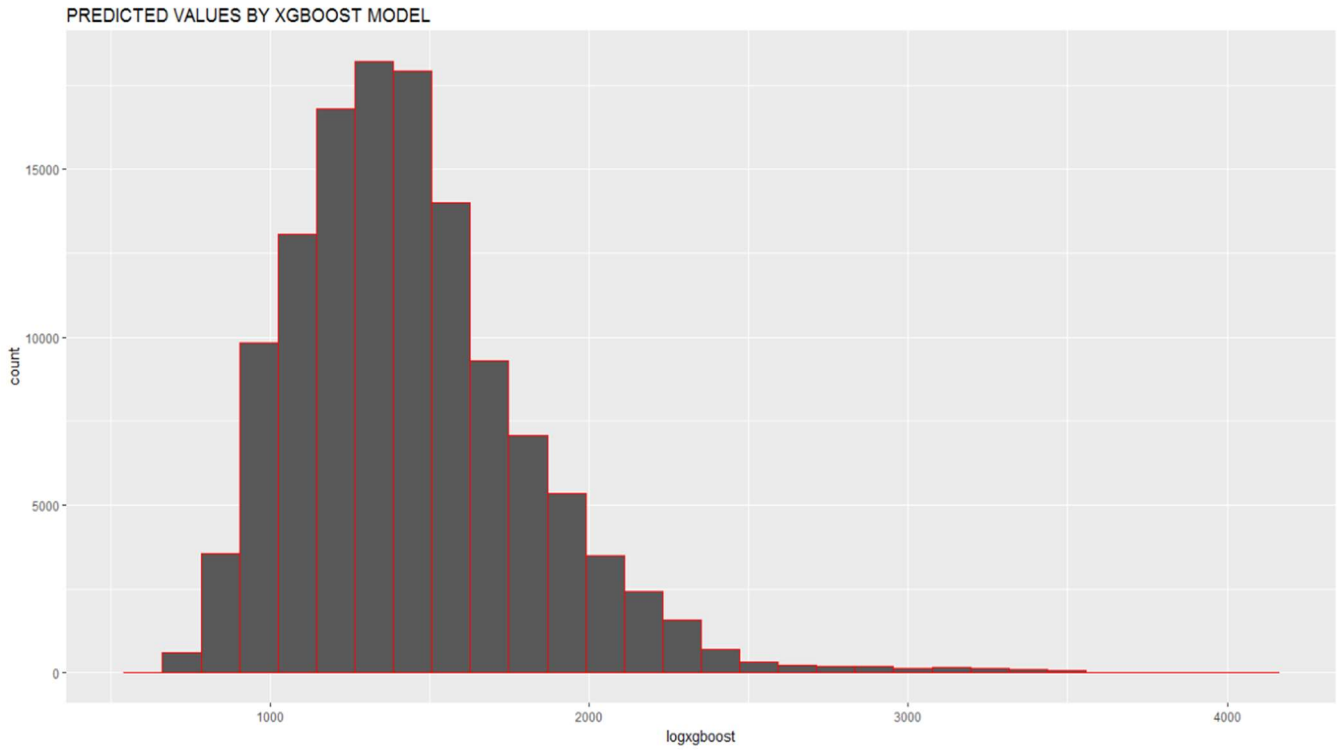
From the above results Xgboost model performs well on the given training data set, 0.54608 is smaller than the other two methods. Xgboost doesn't have the feature to input the R-squared metric. By using XgBoost (tree boosting) the R.M.S.E value has been decreased by 6%.

iter <dbl>	train_rmse_mean <dbl>	train_rmse_std <dbl>	test_rmse_mean <dbl>	test_rmse_std <dbl>
41	0.5290856	0.0010107774	0.5470196	0.002376255
42	0.5286662	0.0010276128	0.5468454	0.002304601
43	0.5281824	0.0009725220	0.5467476	0.002330282
44	0.5277174	0.0009389302	0.5466634	0.002297699
45	0.5272064	0.0008483623	0.5465560	0.002350118
46	0.5268588	0.0007544880	0.5464392	0.002402036
47	0.5264370	0.0007384354	0.5462982	0.002435684
48	0.5259920	0.0007686245	0.5461584	0.002450468
49	0.5255906	0.0007877882	0.5461226	0.002470721
50	0.5252120	0.0006970435	0.5460084	0.002531731

Figure 13 output for xgboost cross validation with 50 rounds

- ## Forecasting Solution

From the above study after finding that the Xgboost model performs better than the other models, values have been predicted for the given test data set.



In the above plot X-axis indicates the values of the "Loss" variable ($) i.e. the claims of the customers. Y-axis indicates the count. From the plot, it can be understood that most of the claims fell in between the values 1000$ and 2000$(count is above 15000). This indicate that most of the claims might be of the car insurance, as the avg. car insurance in many states of U.S.A is around $1200 to 1300.

- ## Conclusion

As stated earlier, the main aim of this study is to predict the severity of the claims of the customers of Allstate insurance company, the claims have been predicted for the given test data set, by applying the best available method i.e. Xgboost. This study clearly states the different steps that have been followed to prepare the data for building the models, performed Linear regression, built two models based on tree boosting, linear boosting by applying xtreme Gradient boosting approach, and finally evaluation is performed by applying 10-fold cross validation with R.M.S.E as evaluation metric. Out of the 4 models that have been created Xgboost (tree boosting) has been choosed for predicting the final claims.

Limitations:

Though this study has been performed with all the care, few things like-

1. Due to a large training dataset, cross-validation of linear regression couldn't run on the Rstudio, which is why only a part of the training data set is used for the evaluation purpose.
2. XgBoost model is interesting and has great features, but it is complicated, huge part of time has gone in understanding the model building, its behavior. The results would definetly be a lot better if XgBoost is used to its full potential.
3. This report has been written by thinking that, it is a report which is prepared for a client, which is why no detailing is given on some of the things.

References

1. http://www.galitshmueli.com/student-projects
2. https://www.kaggle.com/c/allstate-claims-severity
3. https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf
4. https://www.r-bloggers.com/near-zero-variance-predictors-should-we-remove-them/
5. http://www.r-tutor.com/elementary-statistics/simple-linear-regression/residual-plot
6. http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit
7. http://r-statistics.co/Linear-Regression.html