How severe is an insurance claim?

Li Ding
Goergen Institute for Data Science
University of Rochester
Rochester, NY 14627
lding7@ur.rochester.edu

Tianyuan Xie
Goergen Institute for Data Science
University of Rochester
Rochester, NY 14627
txie4@ur.rochester.edu

Siyu Nan
Goergen Institute for Data Science
University of Rochester
Rochester, NY 14627
snan@ur.rochester.edu

Abstract

This project aims at solving a *Kaggle* competition problem: *How severe is an insurance claim?*. Briefly, competitors are asked to use real data from *Allstate Insurance Company* to build a prediction model. Before fitting into model, all categorical attributes are transferred to numerical ones by One-Hot Encoding while numerical data are normalized to reduce the skewness. Linear regression methods and tree method are two main general approaches used in this project. Among linear regression method, we first tried linear regression models, which use l_2 loss as the objective function, without penalty, with l_1 penalty, with l_2 penalty and with both on the parameters. We then form our own method, taking l_1 loss as the objective function after found that the l_2 loss models mentioned above did not give us good results. Besides, tree-based methods, as another kind of general approaches in machine learning, give us much better results compared with linear regression. Among single tree, random forest with CART, XG-Boosting, Multi-layer Perceptron, neural network with XG-Boosting, finally the fine-tuned XG-Boosting gives the best result which ranks 18% among all more than three thousand competitors.[1]

1 Introduction

According to the National Highway Traffic Administration, car accidents happen every minute of the day. Motor vehicle accidents occur in any part of the world every 60 seconds. And if it's all summed up in a yearly basis, there are 5.25 million driving accidents that take place per year (1). Considering the huge amount of accident happened every day, it would be a waste of time and money to analyze the severity of the claims manually for insurance companies. Besides, the loss payments for many types of liability insurance claims can take many months or even years to complete. Approval process, assignment bureaucracy, and schedules of benefits for serious employer's liability claims are among the many reasons for lengthy claims settlement periods (2). This is why insurance companies are continually seeking automated methods to predict the accident cost to improve their claim service.

There are a number of economic methodologies available for accident cost estimation, such as Gross Output approach and Life Insurance approach (3). However, the substantially differ in approaches causes cost estimates vary to a great extent (4). Richard Price first applied statistical methods in insurance (5). Recently, Zhang et al. propose a Bayesian nonlinear hierarchical model that addresses some of the major challenges non-life insurance companies face when forecasting the outstanding

[1] Team Ranking can be found via *https://www.kaggle.com/zephyrd/competitions*. Code scripts can be found via *https://github.com/Zephyr-D/Kaggle-Allstate-Claims-Severity*.

claim amounts for which they will ultimately be liable (6). Sant estimated the expected loss in auto insurance by multivariate statistical procedures (7). Researchers also involved clustering technique to classify the policy holder's potential risk in the first stage and model the claim cost within each risk group (8). Guelman (9) suggested the gradient boosting algorithm to be a good candidate for insurance loss cost prediction, because it gives interpretable results and it is highly robust to not clean data. The comparison study with general linear model shows higher predictive accuracy.

Though there has been great improvement in model development, the accuracy of the predicted results is still too low to use without a risk. Thus, the main objective of this project is to compare the results of different machine learning approaches to get an accurate cost model. The insurance claim dataset used in this project is achieved from Kaggle data competition provided by Allstate Insurance Company (10). Training dataset has 188k claims. Each includes 116 categorical variables and 14 continuous variables. This dataset is preprocessed before fitting into model. The categorical attributes are turned to numerical by One Hot Encoding technique. Skewness correction is done on the continuous features. In this study, both linear algorithm and tree-based method are considered. The linear approaches include LASSO, Ridge Regression, Elastic Net Regression and Linear Regression by l1-loss, while the tree-based method contains CART, Random Forest, Multi-layer Perceptron, XG-Boosting, and Neural Network. Finally, the best model is suggested for prediction of the cost of accident in the future.

2 Data Exploration

As mentioned above, the training dataset has 188k claims. Each includes 116 categorical variables and 14 continuous variables. "cat" and "cont" are the variable names in the dataset for categorical and continuous variables, respectively. For example, cat1 represents categorical 1. Besides, the accident loss is also given in the training set. So the total 130 features will be used to train our model and the predicted loss will be compared with the true loss to measure our model accuracy. There is no information available what each attribute means. So no domain knowledge can be applied for feature selection.

2.1 Categorical Attributes

cat1 to cat 72 only have two labels A and B. In most cases, B has every few entries, which indicates the data is highly unbalanced. Fig. 1 shows the boxplots of log(loss) on label A and B of the first twelve categorical variables. We can notice from this figure that in cat3, cat7, cat10, cat11 and cat12, B has higher average loss than A. From the dataset, we also know that B has much less count than A. So our assumption is B represents the rarely happened severe accident.

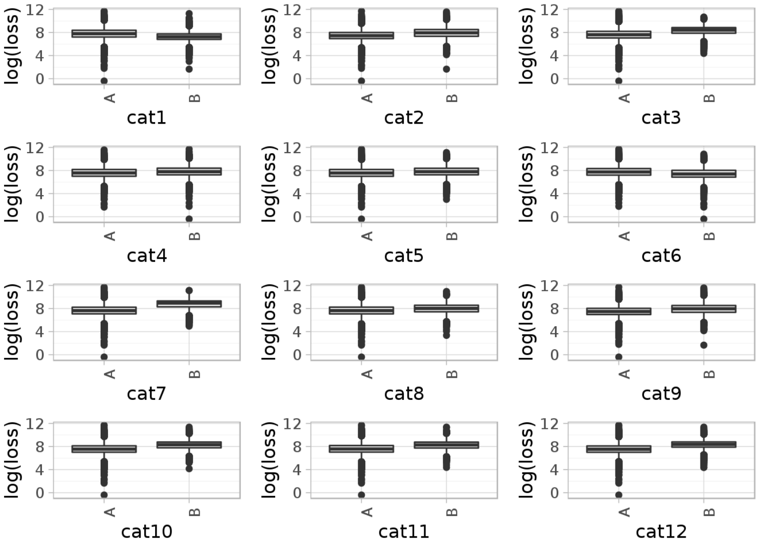


Figure 1: Boxplots of log(loss) on label A and B of the first twelve categorical variables.

cat73 to cat116 have more than two labels. Though it is hard to interpret the meaning of each attributes, we are able to decode one attribute, which is cat112. This attribute has 51 labels, which represent 50 states and Washington D.C in our opinion. Figure 2 shows the number of observations based on states in cat112. The warmer color implies higher count. It is obvious that California is well sampled followed by Texas, New York and Florida.

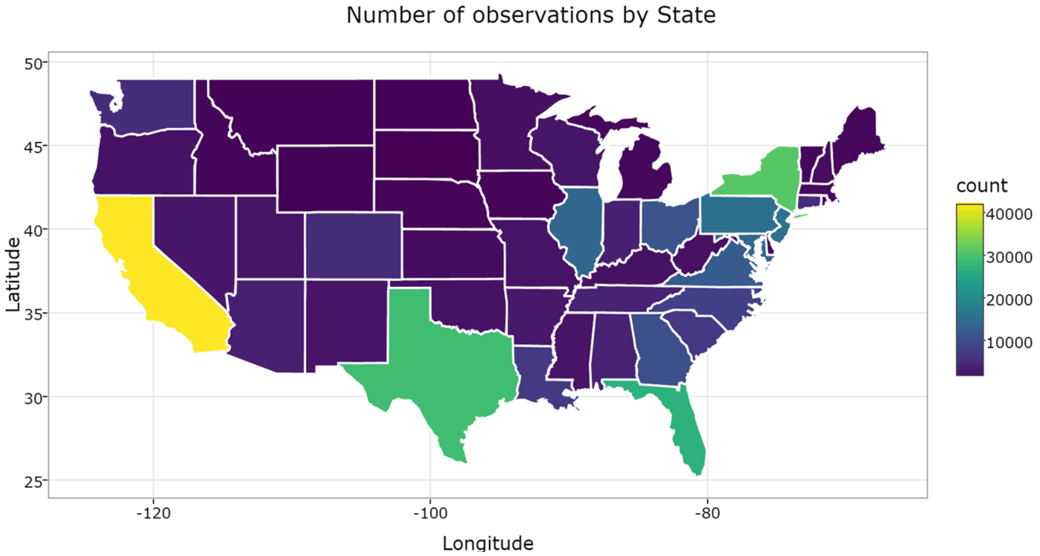


Figure 2: The number of observations based on states in cat112.

2.2 Numerical Attributes

The correlations of all continuous attributes are investigated first and the results are exhibited in Figure 3. The cooler color represents higher correlation. The values of correlation coefficient over 0.5 are listed on the right of Fig. 3. It is clear that some feature are highly correlated, which will be a problem for linear model.

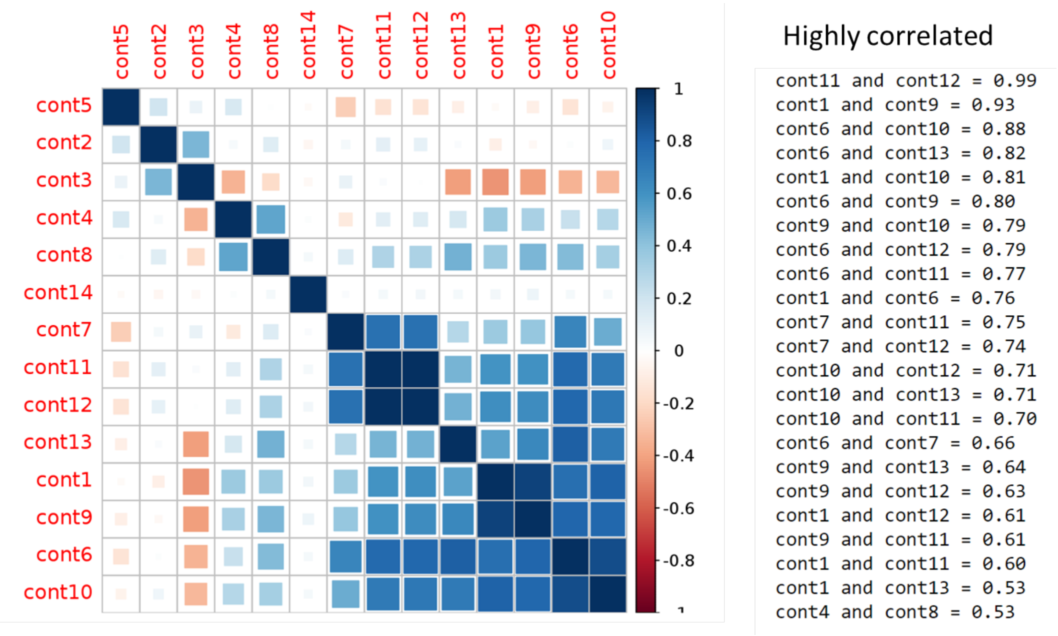
Besides, the skewness of the attributes is also studied. Figure 4 shows the density distribution of all continuous attributes together with their skewness coefficient. All the features are highly left or right skewed. However, the distribution that we would like to have is Gaussian. Thus, data preprocessing is necessary before fitting into model.

3 Data Preprocessing

3.1 One-Hot Encoding Technique

Since we would like to apply linear regression as our baseline, all the categorical data need to be converted to numerical data. One way is to convert levels to number by alphabetic order. However, the nature of the categorical feature is not numeric. So it would be confusing to algorithm. Thus, the methodology used in this project is One Hot Encoding.

Cat1 is taken as an example. It has two labels, A and B. One Hot Encoding splits the cat1 column into two columns, cat1A and cat1B and put dummy variables in. For observations with label A in cat1, “1” is put into cat1A and “0” is put into cat1B and vice versa for observations with label B. Same procedure is applied to all categorical features. After converting, the number of features increases from 130 to 1176.



Highly correlated

```
cont11 and cont12 = 0.99
cont1 and cont9 = 0.93
cont6 and cont10 = 0.88
cont6 and cont13 = 0.82
cont1 and cont10 = 0.81
cont6 and cont9 = 0.80
cont9 and cont10 = 0.79
cont6 and cont12 = 0.79
cont6 and cont11 = 0.77
cont1 and cont6 = 0.76
cont7 and cont11 = 0.75
cont7 and cont12 = 0.74
cont10 and cont12 = 0.71
cont10 and cont13 = 0.71
cont10 and cont11 = 0.70
cont6 and cont7 = 0.66
cont9 and cont13 = 0.64
cont9 and cont12 = 0.63
cont1 and cont12 = 0.61
cont9 and cont11 = 0.61
cont1 and cont11 = 0.60
cont1 and cont13 = 0.53
cont4 and cont8 = 0.53
```

Figure 3: The correlations of all continuous attributes.

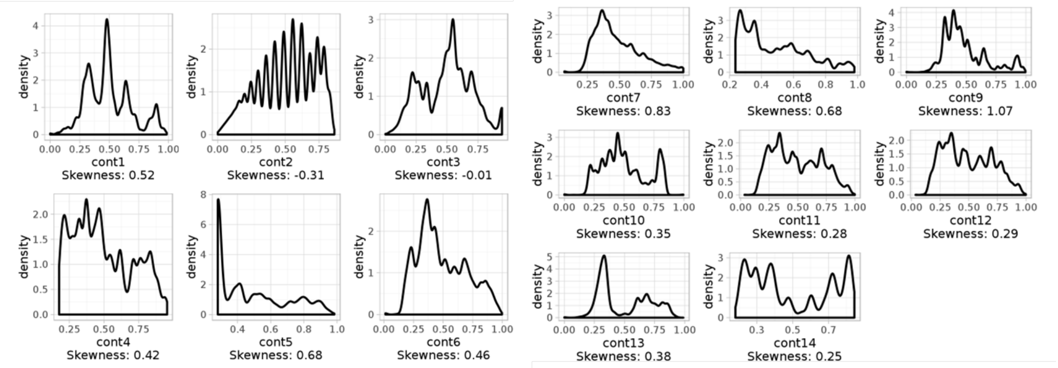


Figure 4: The density distribution of all continuous attributes together with their skewness coefficient.

3.2 Skewness Correction

The skewness coefficients of 14 continuous attributes and loss are summarized in Table 1. All features needs to be normalized to Gaussian distribution. Among all, “loss” shows the highest skewness. So it is taken as an example for skewness reduction. The distribution of “loss” is shown in Figure 5 (a), the shape of which confirms the skewness value we calculated. To alleviate the skewness, all the values are first add a factor and take logarithm with base 10. The corrected distribution is shown in Figure 5 (b). After normalization, the distribution becomes a Gaussian. All the continuous data are transferred to decrease the skewness. The resulting dataset is ready for linear regression.

4 Methodology and Implementation

4.1 Linear Algorithms

In order to minimize the mean absolute error, one general idea is to use linear algorithm method to solve the problem. In our project, we first tried some algorithm learnt in class including l_2 loss

Table 1: Skewness coefficients of 14 continuous attributes and loss.

Name	Skewness	Name	Skewness	Name	Skewness
cont1	0.516	cont6	0.461	cont11	0.281
cont2	-0.311	cont7	0.826	cont12	0.292
cont3	-0.01	cont8	0.677	cont13	0.381
cont4	0.416	cont9	1.072	cont14	0.249
cont5	0.682	cont10	0.355	loss	3.795

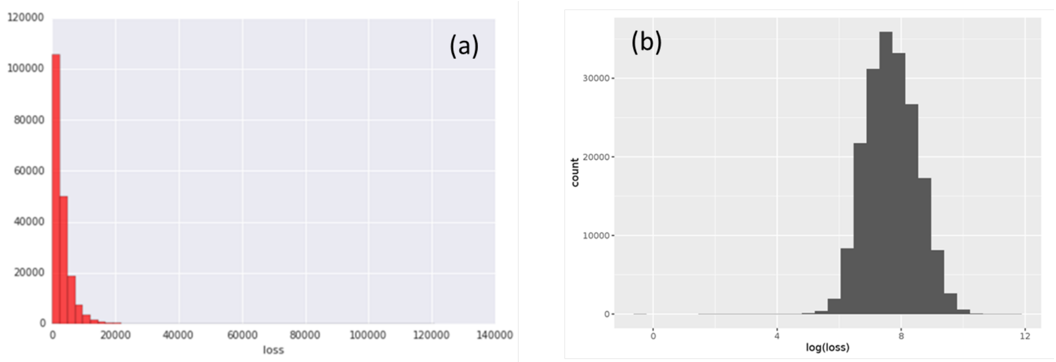


Figure 5: The distribution of "loss" before (a) and (b) after skewness correction.

without penalty: linear regression and l_2 loss with l_1 penalty: LASSO (11); After that, we tried some similar method as expansion including l_2 loss with l_2 penalty: Ridge Regression (14); l_2 loss with l_1 and l_2 penalty:Elastic Net Regression (13). After all the trials, results were still not good enough. Thus we improved taking l_2 loss into taking Linear Regression with l_1-loss as the objective function which gave best result in all linear algorithm.

The baseline we set is linear regression. Linear regression takes the form of: $\hat{y} = Ax - b$. Data matrix of sample and attributes is A, b is the intercept and $\hat{y}$ is the predicted cost. Our objective function is:$\min_{x \in R^{1176*1}} \frac{1}{2}||y - \hat{y}||^2$ where $\hat{y} = Ax + b$. This formation is minimizing l_2 without penalty. Solve for x directly, we get: $x = (A^T A)^{-1} A^T (y - b)$ Plugging in our matrix, we get mean absolute error: 1278.

After the first trial, adding l_1 penalty for denoise come to our mind, which is LASSO we learnt in class. LASSO takes the form of: $\min_{x \in R^{1176*1}} \frac{1}{2}||y - \hat{y}||^2 + \lambda||x||_1 \; where \quad \hat{y} = Ax + b$. This did a little better job, giving us mean absolute error:1262.5. We also tried Ridge Regression which added l_2 penalty It takes the form of: $\min_{x \in R^{1176*1}} \frac{1}{2}||y - \hat{y}||^2 + \lambda||x||_2 \; where \quad \hat{y} = Ax + b$. This gave us mean absolute error: 1267, a bit worse than LASSO. We then tried adding both l_1 and l_2 lost, named Elastic Net Regression. This takes the form of: $\min_{x \in R^{1176*1}} \frac{1}{2}||y - \hat{y}||^2 + \lambda_1||x||_1 + \lambda_2||x||_2 \; where \quad \hat{y} = Ax + b$. By doing this, we get mean absolute error: 1260. Among these existing linear regression method, Elastic Net Regression gave the best result.

With existing algorithm not giving a satisfactory result, we tried to form the objective function in l_1 lost form since we are minimizing the mean absolute error, which is an l_1 formation. Therefore, our objective function looks like $\min_{x \in R^{1176*1}} ||y - \hat{y}||_1$ where $\hat{y} = Ax + b$. In class, we learned that by separating x into x^+ and x^-, we can directly use linear programming to solve this problem. However our matrix is too big to directly use linear programming. Therefore we solve for its closed form by Stochastic Gradient Descent, we get

$$x = \begin{cases} x - \gamma A_i \quad when \quad y - A_i x - b < 0 \\ x + \gamma A_i \quad when \quad y - A_i x - b > 0 \\ [-A_i, A_i] \quad when \quad y - A_i x - b = 0 \end{cases} \tag{1}$$

By this formation, we get mean absolute error: 1239 which performs better than all the l_2 lost formations.

4.2 Non-linear Algorithms

We also use some non-linear methods to analyze the data as well as build the prediction model. Basically, we use the tree-based methods because the decision tree always gives a good performance on analyzing such kind of data.

4.2.1 CART

We start from a single tree model. CART (15), short for Classification and Regression Tree, can be used to handle regression problems as we're going to use the numeric 'smiling' score as the aim feature. The model is determined by node split, which is given by the *Gini Impurity* for a set of items with J classes, suppose $i \in 1, 2, ..., J$ and let f_i be the fraction of items labeled with class i in the set.

$$I_{Gini}(f) = \sum_{i=1}^{J} f_i(1 - f_i) = 1 - \sum_{i=1}^{J} f_i^2 = \sum_{i \neq k} f_i f_k \tag{2}$$

By controlling the depth and prune the tree by cp (complexity parameter) value, we can build a good tree model for both analysis and prediction.

4.2.2 Random Forest

Random Forest (16) is a well-known ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging (17), to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$:

- Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
- Train a decision or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x') \tag{3}$$

4.2.3 XGBoost

XGBoost (19) is short for "Extreme Gradient Boosting", where the term "Gradient Boosting" (18) is proposed by J. H. Friedman. Based on this original model, XGBoost is fast, efficient and scalable end-to-end tree boosting system, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges.

Briefly, it do the additive training for tree boosting. For a single tree model f_i, the prediction value at step t is given by $\hat{y}_i^{(t)}$, so we have

$$\hat{y}_i^{(0)} = 0 \tag{4}$$

$$\hat{y}_i^{(1)} = f_1(x_i) \tag{5}$$

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{6}$$

Using some objective function and some gradient descent algorithm to optimize it, we can get the final model.

4.2.4 Multi-layer Perceptron

A multilayer perceptron (MLP) (20) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. MLP utilizes a supervised learning technique called backpropagation for training the network.

We use Relu as the activation function to provide a non-linear approach. We tried different structures including the three-layer 1024 - 2048 - 1, 1024 - 4096 - 1, and the four-layer 1024 - 4096 - 128 - 1. The number represents the number of nodes in each layer.

4.2.5 Neural Networks with XGBoost

We propose a new method which combines Neural Networks with XGBoost. Basically, we use well-trained neural networks to extract features from the original information, and then use them as the input of XGBoost. The idea comes from some work in computer vision and natural language processing which is popular in recent years, using pre-trained deep CNNs to extract features and use for some classifiers. Specifically, we use the last fully connected layer before the output layer of the three-layer MLP 1024-2048-1, which contains 2048 nodes, as the input of XGBoost. The intuition of doing this is to raise the amount of input because we are dealing with 180k observations. There is an obvious unbalance between features and observations.

4.2.6 Results

Table 2: Results of Non-linear Methods

Methods	Resulting MAE
CART	1741
Random Forest	1228 (Benchmark)
XG-Boosting (default)	1169
Multi-layer Perceptron	1168
NN+XG-Boosting	1143
Fine-tuned XG-Boosting	1106 (Our Best)

The results of all non-linear methods that we implement is shown in table 2. We achieve our best result using XG-Boosting with some fine-tuning of the attributes. Our proposed Neural Networks with XGBoost method performs well at first, but we find that it is hard to improve anymore. Finally our ranking is 533/3055 (18%).

5 Conclusion

The objective of this project is to compare performance of different machine learning algorithm on this insurance claim dataset and to get the best model to predict the accident loss. Since meaning of the attributes are unknown, domain knowledge cannot be applied to process the data. A proper data preprocessing methodology plays an important role in achieving accurate model. One Hot Encoding technique is used to convert categorical data to numerical. All continuous features are corrected to have an approximate Gaussian distribution.

Different machine learning algorithms are implemented to get the best model. The baseline is linear regression, which shows MAE value of 1278. When the objective function is changed to l_1 loss, the MAE can be reduced to 1239. On the other hand, the non-linear models, especially tree-ensemble models performs much better than those linear models. The XG-Boosting method exhibits MAE value of 1106, which is 13.5% enhancement, compared to our baseline. This result also led us to be top 18% rank in the whole competition.

From the algorithm comparison study, we know that tree-based methods, especially XG-Boosting, works more efficiently for this special data structure with limiting information. In our opinion, this is mainly because the basis of decision tree has a sense of probability thesis, but linear models do not. This case is very close to real world problem, so the ideal linear models are not so suitable.

However, the final mean absolute error is still not satisfactory in the term of commercial application. Thus, our future work will base on the XGBoosting and further optimize this algorithm. Besides, the multicollinearity and continuous attributes processing will be taken into account as well.

References

[1] http://www.usacoverage.com/auto-insurance/how-many-driving-accidents-occur-each-year.html

[2] http://www.mondaq.com/unitedstates/x/366524/Insurance/Why+It+Takes+So+Long+For+Insurance+Carriers+To+Respond+To+A+Claim+And+What+You+Can+Do+About+It.html

[3] Jones-Lee, Michael W. "The value of life: an economic analysis." (1976).

[4] Partheeban, Pachaivannan, Elangovan Arunbabu, and Ranganathan Rani Hemamalini. "Road accident cost prediction model using systems dynamics approach." Transport 23.1 (2008): 59-66.

[5] Hacking, Ian. The taming of chance. Vol. 17. Cambridge University Press, 1990.

[6] Zhang, Yanwei, Vanja Dukic, and James Guszcza. "A Bayesian non-linear model for forecasting insurance loss payments." Journal of the Royal Statistical Society: Series A (Statistics in Society) 175.2 (2012): 637-656.

[7] Sant, Donald T. "Estimating expected losses in auto insurance." Journal of Risk and Insurance (1980): 133-151.

[8] Yeo, Ai Cheo, et al. "Clustering technique for risk classification and prediction of claim costs in the automobile insurance industry." Intelligent Systems in Accounting, Finance and Management 10.1 (2001): 39-50.

[9] Guelman, Leo. "Gradient boosting trees for auto insurance loss cost modeling and prediction." Expert Systems with Applications 39.3 (2012): 3659-3667.

[10] https://www.kaggle.com/c/allstate-claims-severity

[11] Tibshirani, Robert. “Regression Shrinkage and Selection via the Lasso: A Retrospective.” Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73.3 (2011): 273–282.

[12] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning Data Mining, Inference, and Prediction: With 200 Full-Color Illustrations. 4th ed. New York: Springer-Verlag New York, 2009.

[13] Zou, Hui, and Trevor Hastie. “Regularization and Variable Selection via the Elastic Net.” Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67.2 (2005): 301–320.

[14] Fan, Jianqing, and Runze Li. “Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties.” Journal of the American Statistical Association 96.456 (2001): 1348–1360.

[15] Breiman, Leo, et al. Classification and regression trees. CRC press, 1984.

[16] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[17] Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

[18] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.

[19] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." arXiv preprint arXiv:1603.02754 (2016).

[20] Rosenblatt, Frank. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961

[21] Haykin, Simon (1998). Neural Networks: A Comprehensive Foundation (2 ed.). Prentice Hall. ISBN 0-13-273350-1.