Prediction of Insurance Claim Severity Loss Using Regression Models

Ruth M. Ogunnaike[1], Dong Si[2]

Computing and Software Systems
University of Washington Bothell
Bothell, WA. USA
tunrayo@uw.edu, dongsi@uw.edu

Abstract. The objective of this work is to predict the severity loss value of an insurance claim using machine learning regression techniques. The high dimensional data used for this research work is obtained from Allstate insurance company which consists of 116 categorical and 14 continuous predictor variables. We implemented Linear regression, Random forest regression (RFR), Support vector regression (SVR) and Feed forward neural network (FFNN) for this problem. The performance and accuracy of the models are compared using mean squared error (MSE) value and coefficient of determination (Rsquare) value. We predicted the claim severity loss value with a MSE value of 0.390 and a Rsquare value 0.562 using bagged RFR model. In addition where applicable, the final loss value was also predicted with an error of 0.440 using FFNN regression model. We also demonstrate the use of lasso regularization to avoid over-fitting for some of the regression models.

Keywords: Regression; Insurance claim severity; Machine learning; Regularization.

1 INTRODUCTION

A claim severity can be defined as the amount of loss associated with an insurance claim. The average severity is calculated by dividing the total amount of losses that an insurance company experiences by the number of claims that were made against policies that it underwrites. Loss is the amount paid or to be paid to the claimants under their insurance policy contracts. Currently, the details of computing a forecast of the paid claim loss is complicated [2].

Insurance companies rely on actuaries and the models that actuaries create to predict future claims, as well as the losses that those claims may result in . The models are dependent on a number of factors, including the type of risk being insured against, the demographic and geographic information of the individual or business that bought a policy, and the number of claims that are made. Actuaries look at past experience data to determine if any patterns exist, and then compare this data to the industry at large.

Claim severity loss forecasting has played a major role in determining auto insurance rate and premiums [7]. It is important to obtain accurate estimate of the losses that could arise from an insurance contract. Also in an event a car accident occurs, an insurance policy holder will prefer a fast and quality service from the insurance company when it comes to processing claims and it can also take a considerable amount of time to settle claims in some cases [5].

To provide quality claim service to millions of policy holders protected by insurance companies, and also to create an accurate forecast that predicts rates and premiums, it is necessary for insurance companies to have automated systems that can accurately predict claim severity loss given a set of input. This research work aim to predict the severity loss value of an insurance claim using continuous and categorical features from previously processed insurance claims.

In the domain of loss prediction model, the work in [10] attempts to use convolution approach to estimate loss severity distribution; using convolution of normal and exponential distribution for modelling a loss distribution of property insurance claims. The work in [12] demonstrates actuarial applications that uses hierarchical models to fit micro-level insurance data consisting of claims policy and payment files to predict loss type and accident frequency in automobile.

Researchers have also tried to use insurance claims data to build loss prediction models used for financial risk assessment in construction projects [9]. The research work uses regression analysis to explore the relationships among independent risk factors such as natural disasters, geographic information, and model construction and the dependent variable (percentage of loss) to build a loss prediction model based on the insurance payout records. A similar work [11] uses regression analysis to build loss estimation models for insurance companies that shows the correlation between post-earthquake damage of structural components to direct financial loss of residential buildings based on post-earthquake damage evidence and obtained a Rsquare value of 0.41 using quadratic regression function.

As far as we are aware, we are the first to publish results from a regression model that directly predicts severity loss value of an insurance claim. However, the work in [4] uses regression models (K-nearest neighbour, support vector regression and feed forward neural networks) to predict the overall cost of energy consumed during various categories of entertainment events. The work in [3] also used regression models to predict real estate property prices. Similarly, the work in [6] uses regression models (boosting, linear regression, support vector machine) to predict stock prices.

Also, the work in [1] uses regression models to predict the quality of signal transmitted by optical fiber across communication channels. Other works attempted to illustrates how to quantify the inherent uncertainty in fitting claims severity distributions and estimates the cost of high layered factors that contributes to the severity loss of a claim [5].

This study explores the use of four regression machine learning approaches to predict claim severity loss: Support Vector Regression (SVR), Linear Regression, Random Forest Regression (RFR), and Feed Forward Neural Networks (FFNN).

The four approaches generated varying prediction error rates and coefficient of determination values (R-square). Of the four approaches used for this study, random forest regression model generated the lowest mean square error.

The rest of this paper is organized as follows; The research methodology is presented in section 2, section 3 explains algorithm selection optimization and an evaluation of the achieved results in Section 4. Section 6 concludes the paper.

2 METHODOLOGY

This section introduces the data set and describes the analyses and preparation (data pre-processing) that occurred before the machine learning approaches were applied. It also outlines features selection, regularization and model building.

Figure 1 describes the overview of our approach. It starts by designing data which involves data transformation, normalization and splitting the data into training, test and validation set. This is followed by feature extraction steps that figures out which subset of selected features are really informative for assessing the predicted value (severity loss). Model selection decides on which function will be used to fit the input data as well as the extraction of model coefficients. Model quality is evaluated using Rsquare, and MSE between the actual and the predicted values. The proposed framework uses a software package called MATLAB which is equipped with tools for training models and enables graphical representation of some of the experimental results.

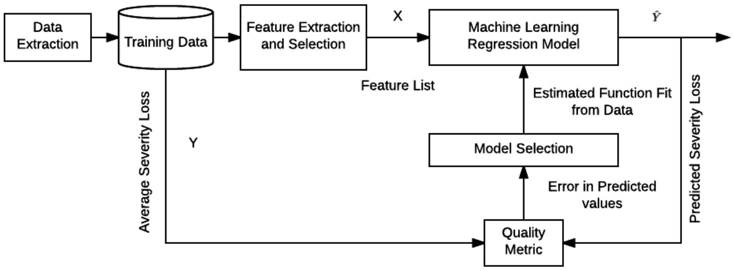

Fig. 1. Work Flow of the Proposed Regression Machine Learning Framework

2.1 Dataset

The dataset includes continuous and categorical features. The target value for this study is the severity loss value of an insurance claim. The dataset contains a total of 188318 records and a total of 130 features (116 categorical features and 14 continuous features). The data was provided by Allstate insurance company on Kaggle website for the purpose of implementing models that predicts a claim severity loss value in order to improve services for their customers.

The original dataset contains normalized continuous predictors while the categorical feature values are alphabet characters of length 1 and 2. The categorical feature values were converted to numeric values ranging from 0 to 205. Figure 2 shows the graph of the original data points in our dataset.

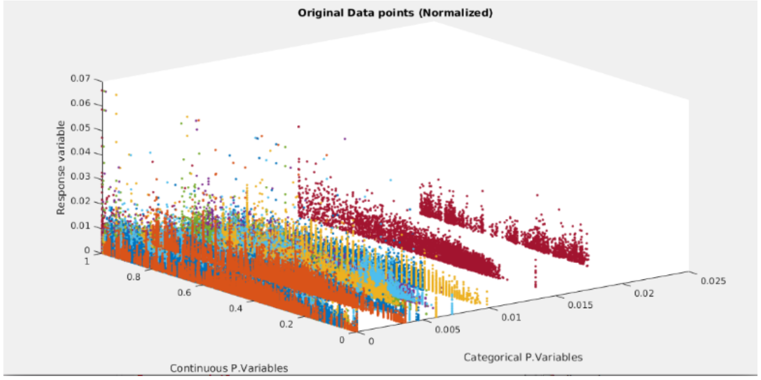

Fig. 2. Original Data points showing the Predictor and Response Variable
The x-axis is the average of the normalized value of categorical predictors, y-axis is the average of continuous predictor and z-axis shows the response variable (the severity loss value to be predicted).

2.2 Feature Design and Selection

To identify the important features i.e. the parameter with large variance with the response variable, we used ensemble function to compute the importance of each predictor variable.

Figure 3 shows the importance estimates of the predictor variables. As shown in figure 3, approximately 10 predictors have importance estimate value greater than 0.02. Using only the predictor with high importance value does not guarantee a high performance model. Only the predictors of high importance can be used if those predictors explains the variance of the data with a cumulative value of 0.9 ± 0.05. It is easy to identify the number of predictors to be used in training the regression models (the number of predictors that cumulatively explains the data variance).

In addition, least absolute shrinkage and selection operator (LASSO) was used to select important predictor variables. Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the model it produces. Lasso was also introduced to minimize the magnitude of the coefficients of each of the predictor variables in order to avoid over-fitting and solves the problem by;

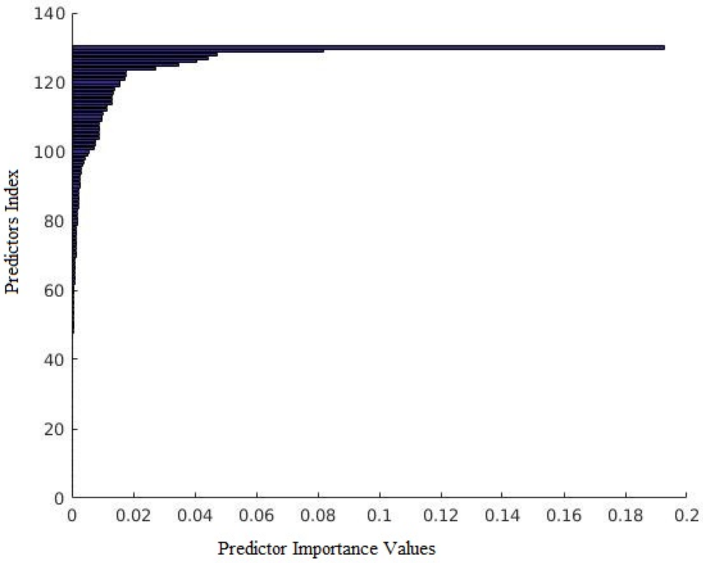

Fig. 3. Importance of Predictor Variables

$$LASSO : min \sum_{i=1}^{m} (y_i - w.x_i - b)^2 \ + C \sum_{j=1}^{n} \|w_j\| \ , \tag{1}$$

where x_i is the predictor variable value, y_i is response variable, w is the coefficient of a specific predictor variable and b is the fitted least-squares regression coefficients for a set of regularization coefficients, Lambda.

Figure 4 shows the cross-validated MSE of lasso fit plot. The plot identifies the minimum deviance point with a green circle and dashed line as a function of the regularization parameter Lambda (λ). The green circle points are the lambda values that minimizes the cross validated MSE and the blue circled points are the lambda values with the greatest amount of shrinkage whose cross-validated MSE is within one standard error of the minimum. We selected the predictors between the blue and the green line, however the performance of the models were really poor. Hence, the primary training uses the entire predictors for training.

3 ALGORITHM SELECTION AND IMPLEMENTATION

We selected linear regression, support vector regression, random forest regression and feed forward neural networks regression models to predict insurance claim severity loss value.

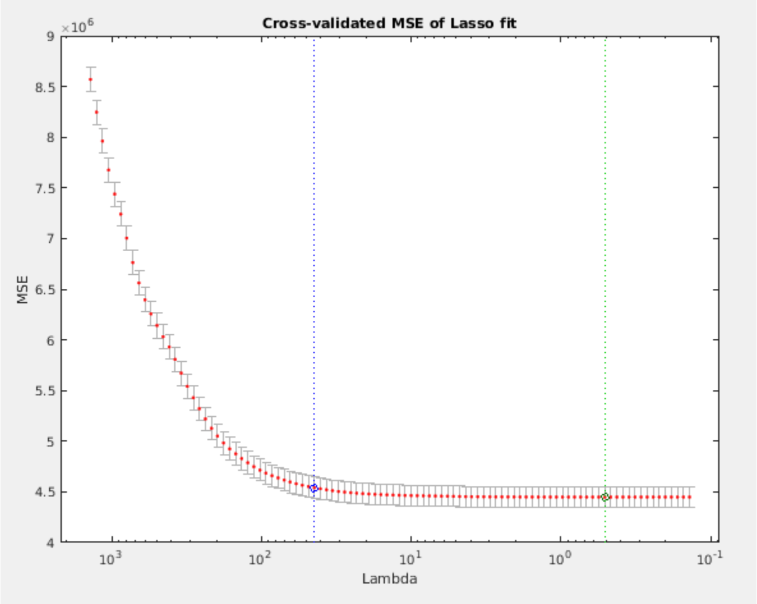

Fig. 4. Cross-validated MSE of Lasso fit

3.1 Linear Regression

To establish baseline performance, we used linear regression to model the severity loss values, Y, as a linear function of the data, X, the predictor variables.

$$f_w(X) = w_0 + w_1 x_1 + ... + w_m x_m = w_0 + \sum_{i=1}^{m} w_j x_j \tag{2}$$

Where w_j are the weights of the features, m is the number of features and w_0 is the weight to the bias term, $x_0 = 1$. The weights of the linear model can be found with the least square solution method, we find the w that minimizes error. Writing in matrix notation, we have:

$$Err(w) = (Y - Xw)^T \ (Y - Xw) \tag{3}$$

where X is the n x m matrix of input data, Y is the n x 1 vector of output data, and w is the m x 1 vector of weights. To speed up computation, the weights can be fitted iteratively with a gradient descent approach. Given an initial weight vector w_0, for $k = 1, 2, ..., m$, $w_{k+1} = w_k - \alpha_k \delta \ Err(w_k)/\delta \ w_k$, and end when $|w_{k+1} - w_k| < \epsilon$
Here, the parameter $\alpha_k > 0$ is the learning rate for iteration K. The performance of linear regression is shown in Table 1

3.2 Support Vector Regression (SVR)

We used the linear and gaussian kernel SVR to predict severity loss. The linear SVR estimates a function by maximizing the number of deviations from the actual obtained targets y_n within the normalized margin stripe. The SVR algorithm is a convex minimization problem that finds the normal vector of the linear function as follows [8]

$$min_{w,\gamma}\ (\frac{1}{2}\left|w\right|^2 + C\sum_{n=1}^{N}\gamma_n\ + \gamma_n^*) \tag{4}$$

Where γ_n, γ_n^* are slack variables allowing for errors to cross the margin. The constant C>0 determines the trade off between the flatness (minimized of the function and the amount up to which deviations larger than margins stripes are tolerated. The results of the SVR can be found in figure 5 and Table 1

3.3 Random Forest Regression (RFR)

The Random forest regression is an ensemble algorithm that combines multiple Regression Trees (RTs). Each RT is trained using a random subset of the features, and the output is the average of the individual RTs. The sum X of squared X errors for a tree T is:

$$S = \sum_{c\epsilon leaves(T)\ i\epsilon C} \sum_{i\epsilon C} (y_i - m_c)^2 \tag{5}$$

where $m_c = \frac{1}{n_c}\sum_{i\epsilon C}\ y_i$, the prediction for leaf c, y_i is the actual loss, T is a regression tree, c are leaves of trees T and S is the sum of squared errors. Each split in the RTs is performed in order to minimize S. The basic RT growing algorithm is as follows:

- **Step 1**: Begin with a single node containing all points. Calculate m_c and S.
- **Step 2**: If all the points in the node have the same value for all the independent variables, then stop. Otherwise, search over all binary splits of all variables for the one which will reduce S the most. If the largest decrease in S would be less than some threshold δ , or one of the resulting nodes would contain less than q points, then stop. Otherwise, take that split, creating two new nodes.
- **Step 3**: In each new node, go back to step 1. One problem with the basic tree-growing algorithm is early termination. An approach that works better in practice is to fully grow the tree (i.e., set q = 1 and $\delta\ = 0$), then prune the tree using a holdout test set.

For our implementation we used the LSBoost and bagged decision trees.

1. ***Bagged***: Bootstrap-aggregated (bagged) decision trees combine the results of many decision trees, which reduces the effects of overfitting and improves

generalization. Individual decisions trees tend to overfit. Tree bagger grows the decision trees in the ensemble using a bootstrap samples of the data. It selects a random subset of predictors to use at each decision split in the random forest algorithm

2. ***LSBoost***: The ensemble methods use multiple learning algorithms to obtain better predictive performance that could be obtained from any of the constituent learning algorithms alone.

3.4 Feed-forward Neural Network

Neural networks (NN) consists of interconnected neurons, or nodes, and have the ability to approximate nonlinear relationships between the input variables and output of a complicated system. Feed Forward Neural Networks (FFNN) are one of the most frequently used NNs for value forecasting and was chosen for this study. Feed forward neural network is composed of an input layer, one or more hidden layers of neurons, and an output layer. Each layer contains a chosen number of neurons, which are then individually interconnected with adaptable weighted connections to neurons in the succeeding layer (with the exception of the output layer). The output of each neuron in the hidden layer is determined using;

$$f_j(x) = \varphi(\sum_{i=1}^{n} w_{ij} x_i + \theta_i) \tag{6}$$

where $f_j(x)$ is the output of the jth neuron, φ is transfer function (such as a Gaussian or sigmoid function), x_i is the ith input to the neuron, w_{ij} is the connection weight between the ith neuron in the input layer and the jth neuron in the hidden layer, and θ_i is the bias or threshold.

The neurons in the output layer also have weighted connections, exclusively with the last hidden layer in the network. Training the network involves adjusting the weights between neurons so that the neural network can produce desirable results when given a set of inputs. A variety of training algorithms can then be used to minimize the network error function. This study uses a feed forward network with a single hidden layer and Levenberg-Marquardt learning algorithm.

4 RESULTS

The regression models used both 5-fold and 10-fold cross validation unless otherwise stated. We analyzed the MSE and Rsquare values of each models with the values ranging from 0 to 1. Smaller values of MSE indicates a better model fit while a larger rsquare value indicates that the model well explains the variability of the response data around its mean.

1. **Mean Squared Error (MSE)** is the most important criterion for this study since the main purpose of the regression models used is prediction.

MSE is an absolute measure of fit and it indicates how closely the predicted values from the model match the response variable (severity loss) the model is intended to predict. Lower values of MSE indicate better accuracy and it is calculated as follows;

$$\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{y_i})^2}{y_i} \tag{7}$$

where n is the number of observations, y_i is the actual loss and $\hat{y_i}$ is the predicted loss.

2. **Coefficient of determination (Rsquare)** indicates the proportionate amount of variation in the response variable y explained by the independent/predictor variables X in the regression model. Rsquare is the square of the correlation between the response values and the predicted response values. A higher rsquare value indicates a better model fit and is calculated as follows;

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2} \tag{8}$$

where y_i is the actual loss, $\hat{y_i}$ is the predicted loss and $\bar{y_i}$ is the mean of the target value (severity loss).

Table 1 shows the summary of the performance of each model using only categorical predictors, only continuous predictors and using the entire predictors. Tests was also done to identify the effect of using only the predictor that minimizes the mean square error. However, there was drop in the performance of the regression models.

Table 1. Performance (MSE and R-square value) of Regression Models

	All		Categorical		Continuous	
Model	**MSE**	**R-Square**	**MSE**	**R-Square**	**MSE**	**R-Square**
Random Forest(Bagged)	0.391	0.562	0.420	0.527	0.419	0.527
Neural Network	0.441	0.519	0.411	0.484	0.441	0.529
Linear Regression (Simple)	0.477	0.463	0.491	0.449	0.491	0.4495
SVR (Linear)	0.498	0.439	0.510	0.452	0.546	0.467
Random Forest(LSBoost)	0.790	0.108	0.832	0.066	0.832	0.066
Linear Regression (CV)	0.899	0.038	0.773	0.131	0.772	0.131
SVR (Gaussian)	0.970	-0.033	0.970	-0.033	0.990	-0.031

4.1 Support Vector Regression

Both linear and Gaussian kernels were used for training the SVR algorithm. 5-fold cross validation was used to train both models and the linear kernel achieved a better performance with a MSE value of 0.498 while the gaussian kernel MSE value of 0.970. As shown in table 1, the linear kernel SVR model also explains the variability of the response data around its mean better than the Gaussian kernel model with rsquare values 0.439 and -0.033 respectively.

The overall performance of support vector machine is average. Based on our observations the linear kernel functions fits better with the dataset used. Figure 5 shows the mean squared error of the linear partitioned SVR model across the 5 partitions. The minimum MSE was achieved on the fifth partitioned model. The MSE reported for this model is the average of all the partitioned data models.

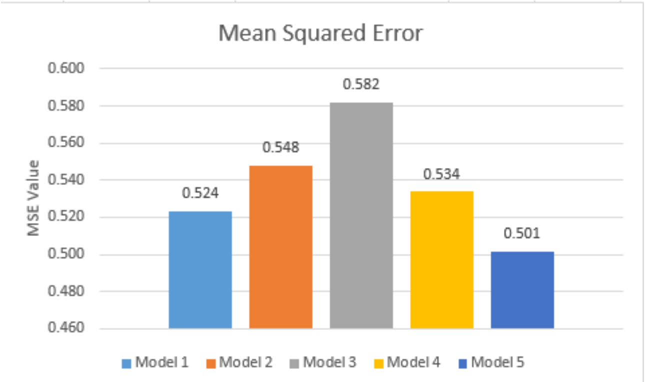

Fig. 5. MSE values of Linearly Partitioned SVR model

4.2 Linear Regression

Using 5-fold cross validation and lasso regularization that returns a Lambda value that helps to minimize MSE, we obtained MSE value of 0.899. Training the model using predictors with higher coefficients did not improve the performance of the model.

When using a partitioning algorithm to train a model, it randomly divides the data into K which is 5 in this case. It trains the model using $K-1$ partitioned data and test the model using the outstanding partitioned data set. This is done iteratively for each partitions and computes the MSE for each partition. The blue line in figure 6 shows the MSE values across the partitioned data model . We also ran a test using a simple linear regression in which all the predictors variable are fit linearly and prediction is done using the coefficient value of each predictor variable and a lower MSE value 0.477 was achieved.

4.3 Random Forest Regression

Bagged random forest and boosted (LSBoost) decision tree were the algorithm used. We used 10-fold cross validation to decide the optimal number of regression

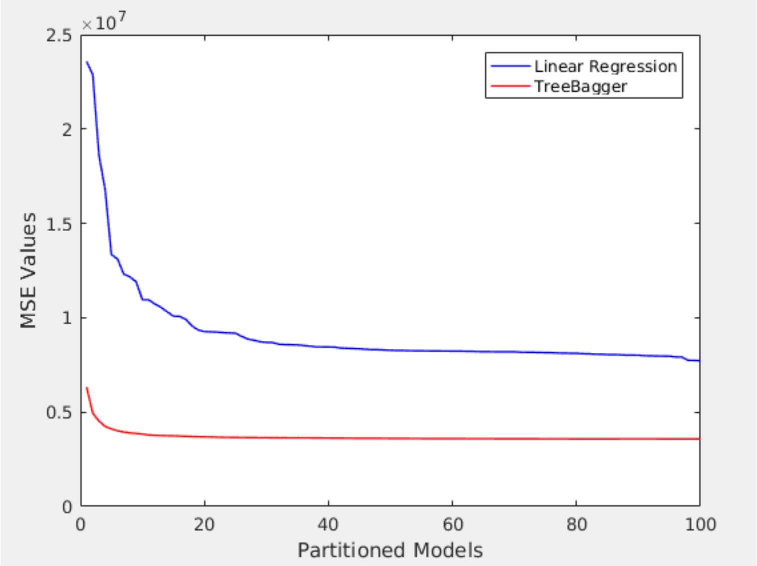

Fig. 6. Graph showing the Mean Squared Error of Linear regression model and the Bagged regression model

trees for both models. For our implementation, we used 100 compact regression trees because we obtained the minimum MSE with that value (100). Bagged RFR model achieved a better performance with MSE value 0.391 and 0.790 for LSBoost. Bagged RFR achieved the best result in terms of its MSE and it also best explains the variability between the response variables around its mean as shown in table 1. On the other hand, LSBoost model did not perform well compared to other models.

Figure 7 shows a plot of the actual and predicted severity loss of 100 data points from the test data. It can be observed that the residuals (actual - predicted) of some of the data point are small while some have a large residual margin. The residuals of the prediction determines the accuracy of the model. As shown in figure 6, the MSE value falls as we move across the bagged regression tress. Each tree predicts the loss value and the average of all the tree's prediction is returned as the final predicted value. Figure 7 shows a major margin between the MSE of the linear regression and the bagged regression model.

4.4 Feed Forward Neural network

Levenberg-Marquardt back propagation algorithm was used to train the neural network. This algorithm is an efficient technique for calculating the output variable. The initial weights are configured and random weights are associated with each transition. Using the feed forward propagation, the network progresses further and predicts an output based on these weights. This model achieved the

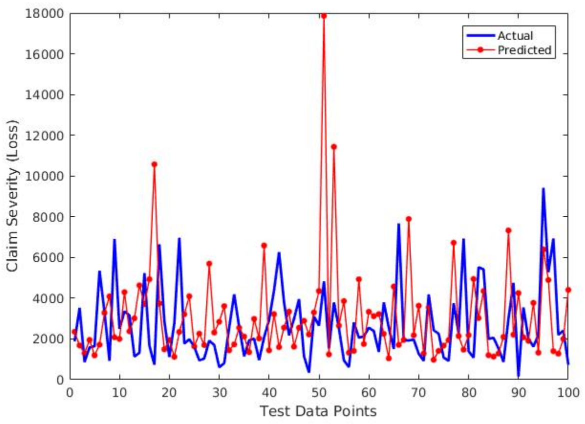

Fig. 7. Graph showing the actual loss value versus the predicted loss value using the random forest (tree bagger) model

second best performance with MSE value 0.416 and R-square value 0.519 (see Table 1).

The model was trained using all predictor variables and the selected predictor variables (49 predictor variables identified by the lasso regularization) which resulted in a MSE value 0.442. The difference in the performance is insignificant. The model fits up to 75 percent of the test data and shows the feasibility of neural network regression model (See figure 10). This shows further computations can improve the performance of this model. Figure 8 shows the Error Histogram for 20 layer neuron. The result is satisfactory as the maximum frequency of instances in the minimum error region denoted by the yellow vertical line in figure 8 . As shown in figure 9, the minimum MSE achieved is 0.441 when normalized.

5 DISCUSSION

Random forest regression and feed forward neural network performed significantly better than the baseline linear algorithms (linear regression and linear SVR).This is possible due to their ability to account for nonlinear interactions between the predictor variables and the severity loss. As shown in Table 1, RFR and neural network has the best overall performance in terms of a minimum mean squared error and coefficient of determination for all test scenarios.

It was observed that, the higher the rsquare value of a model, the smaller the MSE of that model, since the rsquare value of a model indicates how well it explains the variability between the response variable (severity loss) and it surrounding mean. A model is considered a good fit if its rsquare value is high and its mean squared error is low. Models that well explains the variability between its independent variable (predictors) and its dependent variable (severity loss), predicts the response variable more accurately.

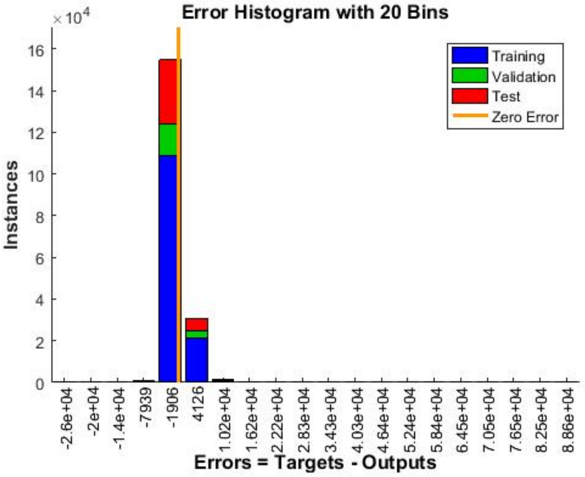

Fig. 8. Error Histogram for 20 Hidden Neurons

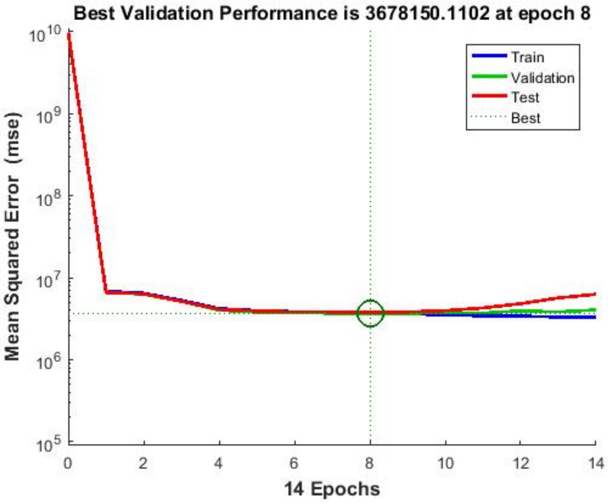

Fig. 9. Mean Squared Error for 20 Hidden Layers

Gaussian kernel SVR model has the worst performance for our study and almost performs as a naive model. Having a negative r-square values shows it is totally unfit for our data prediction. Although cross validated models improves performance for random forest models, cross validation on linear regression did not result in a good performance in this study.

The top two models in table 1, explains more than 50% of the covariance of the predictors and response variable. None of the models performed exceptionally

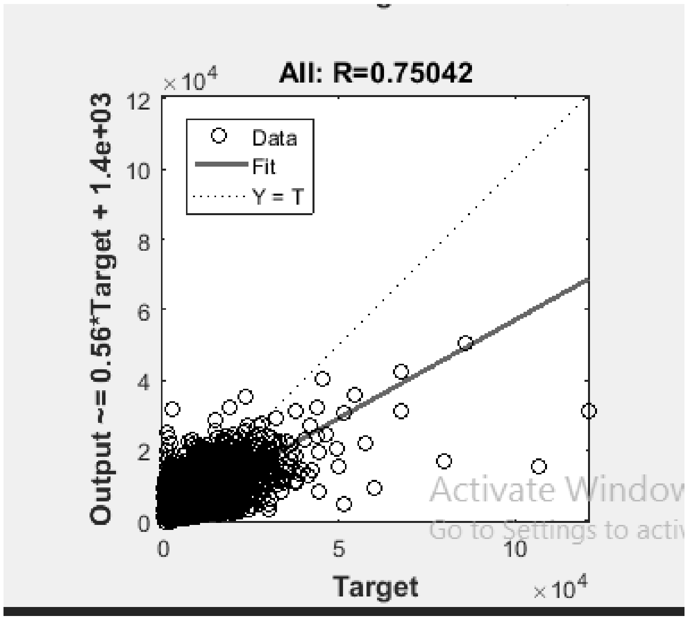

Fig. 10. Regression Plot for 20 Hidden Neurons

well in this study, although a more refined data could lead to better performance of the models. The original dataset does not give insight into the type of feature being used as a predictor; The features provided where just tagged as continuous and categorical variable. We implemented some feature engineering such as using the lasso regularization and principal component analysis, however these did not improve the prediction errors. Ensembling two models for future work could improve the accuracy of the proposed framework such as ensembling random forest with K-nearest neighbour regression models.

Although K-nearest neighbour are most commonly used for classification tasks, theoretically they can perform equally well in regression tasks since any arbitrary functions can be fitted with a multilayer perceptron [3].

Future efforts could be spent into improving the performance of the models and also apply other machine learning models such as the K-nearest neighbour, fast forest quantilem Bayesian linear.

6 CONCLUSION AND FUTURE WORK

Our prediction of insurance claim severity loss using categorical and continuous set of features with regression techniques provides a baseline for further studies. Current study results in a mean square error 0.391 and rsquare value 0.562. Although the implemented models needs further improvement, this application can be useful for insurance companies to create severity loss forecast that enables the insurance companies set premium rates and policies.

While this study focused on using different regression machine learning models to predict insurance claim severity loss, only four regression models was used in this study. Future studies can be done to improve the performance of the models and also apply other machine learning models such as the K-nearest neighbour, fast forest quantile and Bayesian linear. A redefined dataset that gives insight to the types of features can also be used. This will aid in feature engineering in addition to the regularization techniques used in this study.

Adopting well-used datasets in machine learning and improving the error by 0.01 with a particular algorithm can be considered a significant breakthrough. Since insurance companies usually involve large monetary claim transactions, improving the the prediction error by 0.01 can lead into the development of interesting future applications for insurance companies for computing severity loss that helps to determine premium and insurance rate

7 Appendix

The dataset and implementation codes used for this project can be found on the link provided. https://github.com/ruthogunnaike/lossPredictor

References

1. Rudra, R. and Biswas, A. and Dutta, P. and Aarthi, G.: Applying regression models to calculate the Q factor of multiplexed video signal based on Optisystem. 2015 SAI Intelligent Systems Conference (IntelliSys), 201–209 (2015)
2. Cummins, J. D. and Griepentrog, G.: Forecasting automobile insurance paid claims using econometric and ARIMA models. Research Gate journal
3. Nissan P., Emil J. and Liu D.: Applied Machine Learning Project 4 Prediction of real estate property prices in Montreal. Accessed 2016-12-06
4. Andrea Z., Katarina G., Miriam C., Luke S.: Energy Cost Forecasting for Event Venues. EPEC 2015
5. Glenn M.: On Predictive Modeling for Claim Severity. Science Direct journal.
6. Vatsal H. S.: Machine Learning Techniques for Stock Prediction
7. Philipe H., Glenn G. M.: The Calculation of Aggregate Loss Distributions from Claim Severity and Claim Count Distributions.
8. Smola A. J., Scholkopf B.: A Tutorial on Support Vector Regression, "Statistics and Computing": Vol. 14, no. 3, Pages (199 -222) (2004)
9. Hanguk Ryu, Kiyoung Son and Ji-Myong Kim Loss Prediction Model for Building Construction Projects using Insurance Claim Payout Journal of Asian Architecture and Building Engineering, Pages (441 -446) (2016)
10. Ro J. Pak Estimating Loss Severity Distribution Convolution Approach Journal of Mathematics and Statistics, Vol 10. 3, Pages (247 -254) (2014)
11. Qiang Xue, Cheng-Chung Chen, Kuo-Ching Chen Damage and loss assessment for the basic earthquake insurance claim of residential RC buildings in Taiwan Journal of Building Appraisal, Vol. 6, Pages (213 -226) (2011)
12. Edward W. Frees, Peng Shi, Emiliano A. Valdez Actuarial Applications of Hierarchical Insurance Claims Model