Wenke Huang

[whuang67@illinois.edu](mailto:whuang67@illinois.edu)

312-375-1454

Part 1:

1. How would I estimate?

For the premium, usually I think it is not necessary to predict it since customers usually pay first and then insurance companies cover their pets’ bills. Usually, the monthly premium of a customer is a fixed value because situations that may make people want to change their monthly payment do not regularly happen all the time, like having a new member. If there are situations that we have no access to the next premium amount, we can treat it as same as the premium amount of the most recent month.

For the cost, I would like to build some statistical models to predict the proportion of cost/premium since different customers may have different premium payment plans. The pet’s age and medical history, owner’s annual income and family size and some other information that we can legally have access to and use should be taken into consideration.

If we would like to make the prediction and then use the model to make some further strategies, I think linear regression or decision tree should be a good choice. Because we can easily interpret the quantitative results and correlations between or among different variables. If what we want is a “black box” with as perfect predictions as possible, complicated models like support vector machine or ensemble methods may give us a satisfactory result. But it usually takes longer time to train complicated models, and we are not able to interpret the correlation between or among different variables based on these kinds of models. Complicated models also easily give us over-fitted results which means that it works on training datasets only. The predicted proportion of cost/premium times the corresponding premium of this customer is the estimated cost here.

After we get the prediction of the cost, the difference between the premium and the cost will be remaining “value” that we would like to have.

1. How could we as a company use this estimate?

I think here we have many ways to use this estimate.

First, it can help us detect “good” and “bad” customers. The “good” and “bad” customers here stand for that we can and cannot make profits from these customers, respectively. We need to keep re-evaluating our customers. For example, a customer used to be eligible for a lower rate. But as time goes by, he/she may not be a “good” customer any more. We may think about raising his/her premium to balance our potential loss. On the other hand, becoming a “good” customer may result in a decreasing of monthly payment which can help us maintain or strengthen the customer loyalty.

Second, it can help us insurance agents detect new customers. Our agents are facing new customers every day. During the conversations between agents and customers, our agents can try to collect the data that we need to estimate the remaining “value” of this customer which will give our agents a clear understanding. Based on it, our agents can choose different talking strategies to sell the best insurance product that both fit the customers well and make more profits.

Like I mentioned in the previous question, some statistical models can help us find out the quantitative correlation between or among different variables. There are always some unknown situations that we may not predict correctly or on time, like a new virus that influences the health of dogs seriously. With this information, we can quickly detect that the remaining “values” of dog owners are going down or even negative in a short period of time. To decrease the loss, we can make some financial strategies that prevent us from getting involved too much.

Part 2:

In this part, I would like to choose b. Future claims paid out.

1. Dataset and Invalid Records

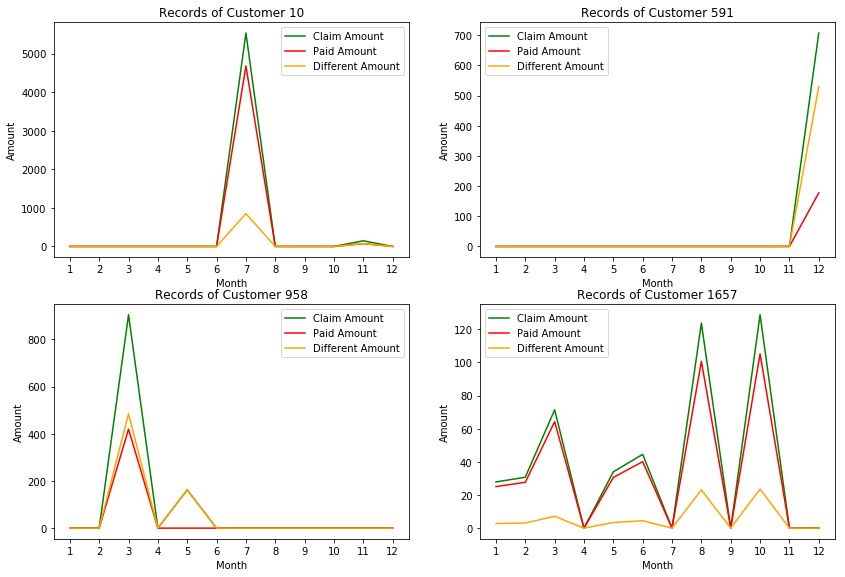
The dataset ClaimLevel.csv contains 144910 observations and 4 variables which are Policy ID, Claim Date, Claimed Amount and Paid Amount. We know that the Paid Amount will not be greater than Claimed Amount. But there are 9 records that is against this rule. Here, I choose to remove those 9 records since they only represent 9/144910 = 0.01% of the whole dataset.

1. Data Preprocessing

There are only 31526 distinct Policy ID, which indicates that there may be multiple records for a single customer. They may claim multiple times during the year. Additionally, since we will use this data to predict the Paid Amount of January 2017, using a spread sheet based on month value of variable Claim Date should be a better idea. Each row stands for a distinct Policy ID while columns stand for their corresponding Claimed and Paid Amount of 12 months. If a customer made multiple claims in the same month, I would like to add them up. The new dataset contains 31526 observations and 25 variables which are Policy ID and the Claimed and Paid amounts of 12 months.

I would like to consider the Paid Amount of the last month (Dec.) as our response and the Claimed and Paid amounts of previous eleven months (Jan. to Nov.) as our predictors to train our model. The reason the Policy ID will not be used is because it is a unique key which indicates different records of this dataset. The Claimed Amount will not be used since it is unknown like Paid Amount if we would like to make future predictions.

The plot below indicates historical records of 4 random chosen customers (Policy ID: 10, 591, 958 and 1657). From these four subplots, we can read that some customer tend to claim their pretty frequently while some may not use it very often.



Z-score transformation will be used to scale our dataset. After transformation, all predictor columns will have 0 mean and 1 standard deviation. 80% of the observations will be used as training set while the rest 20% is put in the testing set. The dataset is split randomly.

1. Model Fitting

We would like to try Linear Regression, Decision Tree Regression, Random Forest Regression, K Nearest Neighbors and Multilayer Perceptron Neural Network Regression. Root mean squared error (RMSE) is used here. Because this metric not only share the same unit with our response variable, but also is more sensitive about greater difference between actual and predicted values. Both training and testing error will be calculated to help us check if we over- or under-fit the model.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Linear Reg. | Tree | Random For. | KNN | MLP NN |
| Train MSE | 268.96 | 126.38 | 154.67 | 266.31 | 267.14 |
| Test MSE | 251.24 | 312.88 | 251.48 | 248.56 | 250.81 |

Linear regression:

The performance of Linear Regression is OK. From the background of the dataset we can easily know that Claimed and Paid Amounts should be correlated. The Pearson correlation coefficient of Claimed and Paid Amounts in the original datasets is …. Hence, Linear Regression may be not a very good choice due to multicollinearity.

Decision Tree:

The train MSE is a little low, but the test MSE is high. Usually decision tree is not considered as a complicated model. Personally, I think the performance difference between it and Linear Regression is because Decision Tree model is non-parametric method but the other one is parametric one.

Random Forest:

I used 5-fold cross-validation here to help us find out the best tuning parameters. Here, the parameters that I tuned are ntree (number of trees that we train) and mtry (number of variables that we try at each split). We will have 250 trees and try 3 different variables at each split. The performance here is more satisfactory than both those of Linear Regression and Decision Tree.

KNN:

5-fold cross-validation is also used here to search for the best tuning parameter number of neighbors. Eventually, we choose 121 neighbors. For distance, we use Euclidean distance (Minkowski with p equal to 2). The training and testing performances are basically like that of Linear Regression here.

MLP Neural Network:

I would like to build a model with two hidden layers one of which has 11 nodes and the other one has 5 nodes. I notice that this Multilayer Perceptron model does not give us a very good result, either. The performance of this model is like those of Linear Regression and KNN.

1. Ensemble Models

Single models may not be able to give us a good result here. So, I would like to see if we can ensemble these models and see the performance of our new models will be better or not. Like the previous part, both training and testing MSE’s will be calculated.

1. Model 1: Arithmetic average of all five models
2. Model 2: Arithmetic average of all models except Decision Tree
3. Model 3: Linear Regression \* .2 + Random Forest \* .4 + KNN \* .2 + MLP Neural Network \* .2
4. Model 4: Linear Regression \* .1 + Random Forest \* .4 + KNN \* .3 + MLP Neural Network \* .2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Model 1 | Model 2 | Model 3 | Model 4 |
| Train MSE | 201.34 | 233.14 | 214.94 | 214.66 |
| Test MSE | 248.58 | 247.96 | 248.01 | 247.73 |

Based on the previous two tables, I noticed that the performances were not significantly improved after we ensembled those single models together.

1. Summary and Output

Based on the general performance here, I would like to make the prediction by using Model 4 since this one contains the lowest testing RMSE among all 9 models that we have attempt. The predicted value of Paid Amount of January 2017 will be saved in file output.txt. The first line of the file is the header which is “PolicyId,PaidAmount”. Starting from the second line we have the Policy ID’s and their corresponding predicted values. The delimiter here is comma “,”.

1. Further Thoughts

Backward Elimination, Regularization like LASSO or some other features selection can be considered here to help us find a Linear Model with potential better performance.

Running time and whether being interpretable or not can also be taken into consideration when performances of different models are alike.

Adding some other possible features like age, insurance premium, payment type, etc can be added into the dataset.