Documentation of Allstate Purchase Prediction Challenge

Meijian Guan

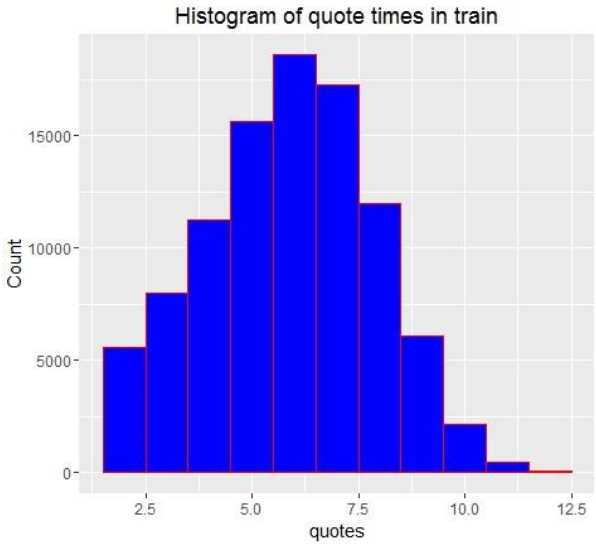
Introduction

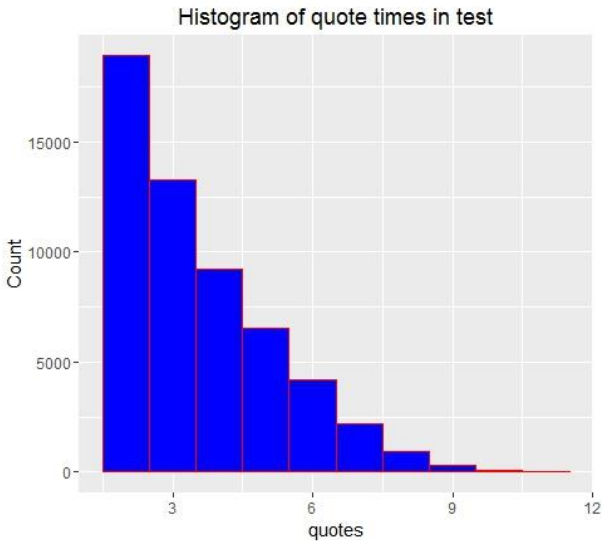
In this challenge, train and test data set containing customer characteristics and quote histories were provided to predict coverage choices for customers. Data exploration, feature engineering, data reformatting were performed using R before applying predictive algorithms. Since this challenge tends to be a nonlinear classification problem. I decided to use the neural network (NN) pattern recognition model for multi-class labels and a customized neural network model to predict binary outcomes using Matlab. The fundamental strategy is to combine multiple records for one customer into one row - purchasing point in train and the final quote in test data, with considering shopping time, quote history and final quote information. The training data set was split into three parts, 60% as training, 20% as cross-validation and 20% as test.

Data preparation

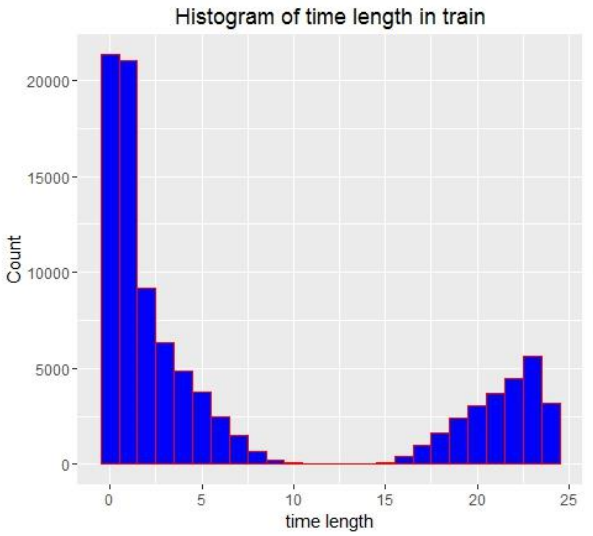
Overall, the data sets provided by Kaggle and Allstate are pretty clean and well-formatted. There are missing values in column “risk_factor”, “c_previous” and “duration_previous” in both data sets, and there are additional missing values in column “location” of the test data. The missing values were replaced or removed. In addition, six features were transformed or engineered into new variables that can better represent the original features or improve the performance of predictive algorithm. Mean normalization was also performed on six features in both data sets.

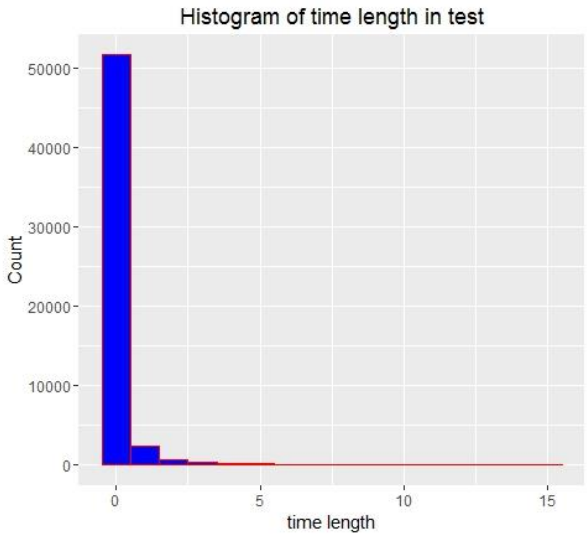
Data exploration. There are 665249 quote records from 97009 unique customers in the training data, and each customer has 24 features. Each customer placed ~6 quotes on average before purchasing a policy, ranged from 2 to 12 (excluded purchase point). The number of quotes before purchasing was close to normal distribution in training data but skewed in test data due to data trimming (see following plots).



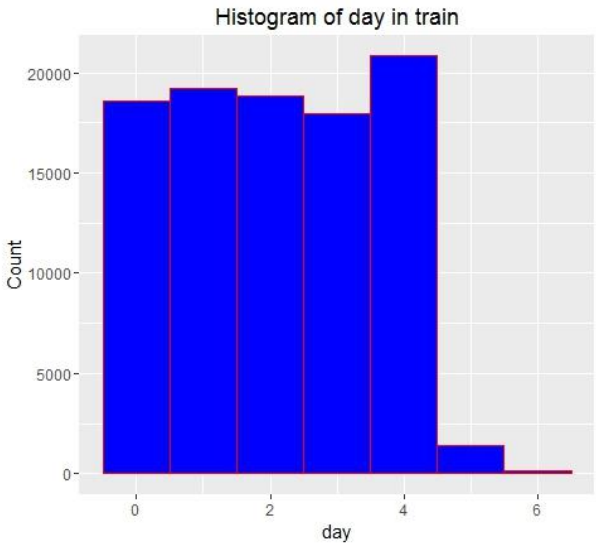


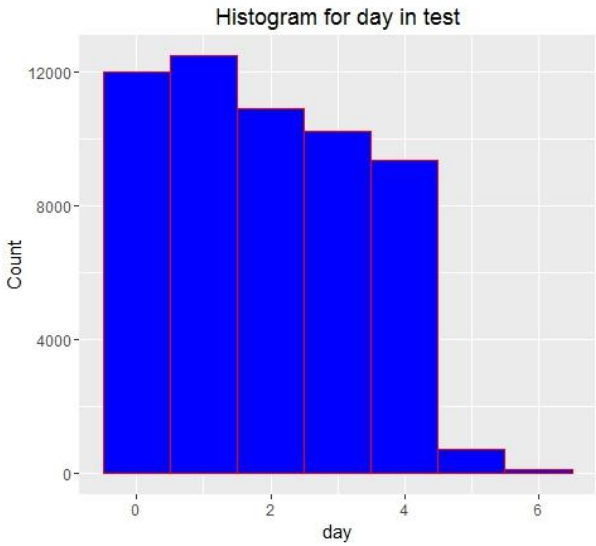
The length of time that spent by each customer was recalculated based on the “time” variable (purchasing time point – first time point). The distribution is showing below for both data sets. Again, the data trimming in test data might have caused the significant pattern difference in two data sets (see following plots).



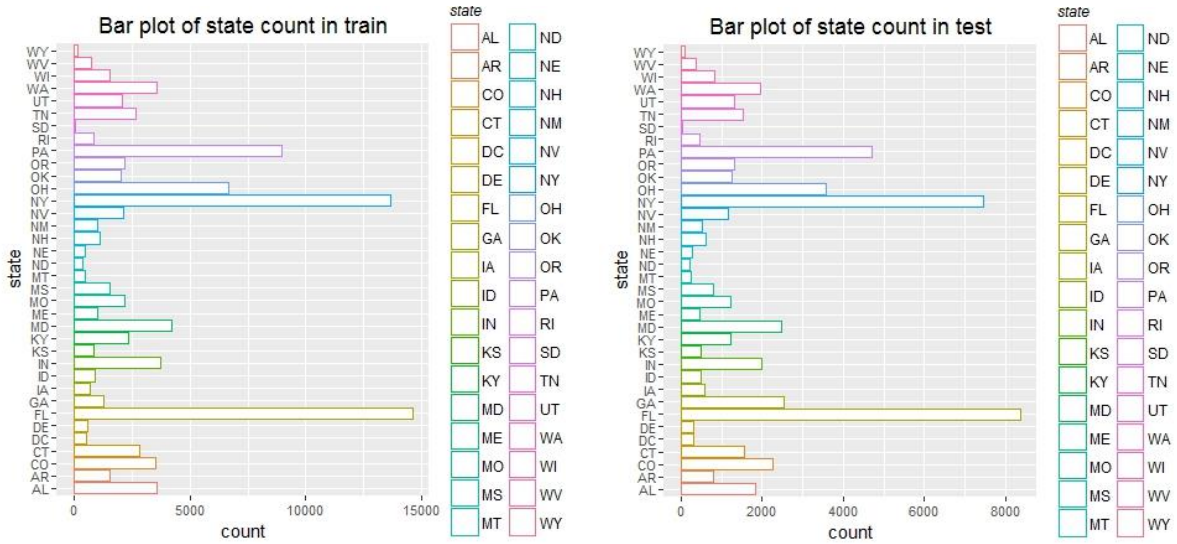


Most customers purchased their cars during weekdays and it’s consistent in both data sets (see following plots). And it’s reasonable to convert it to binary variable.

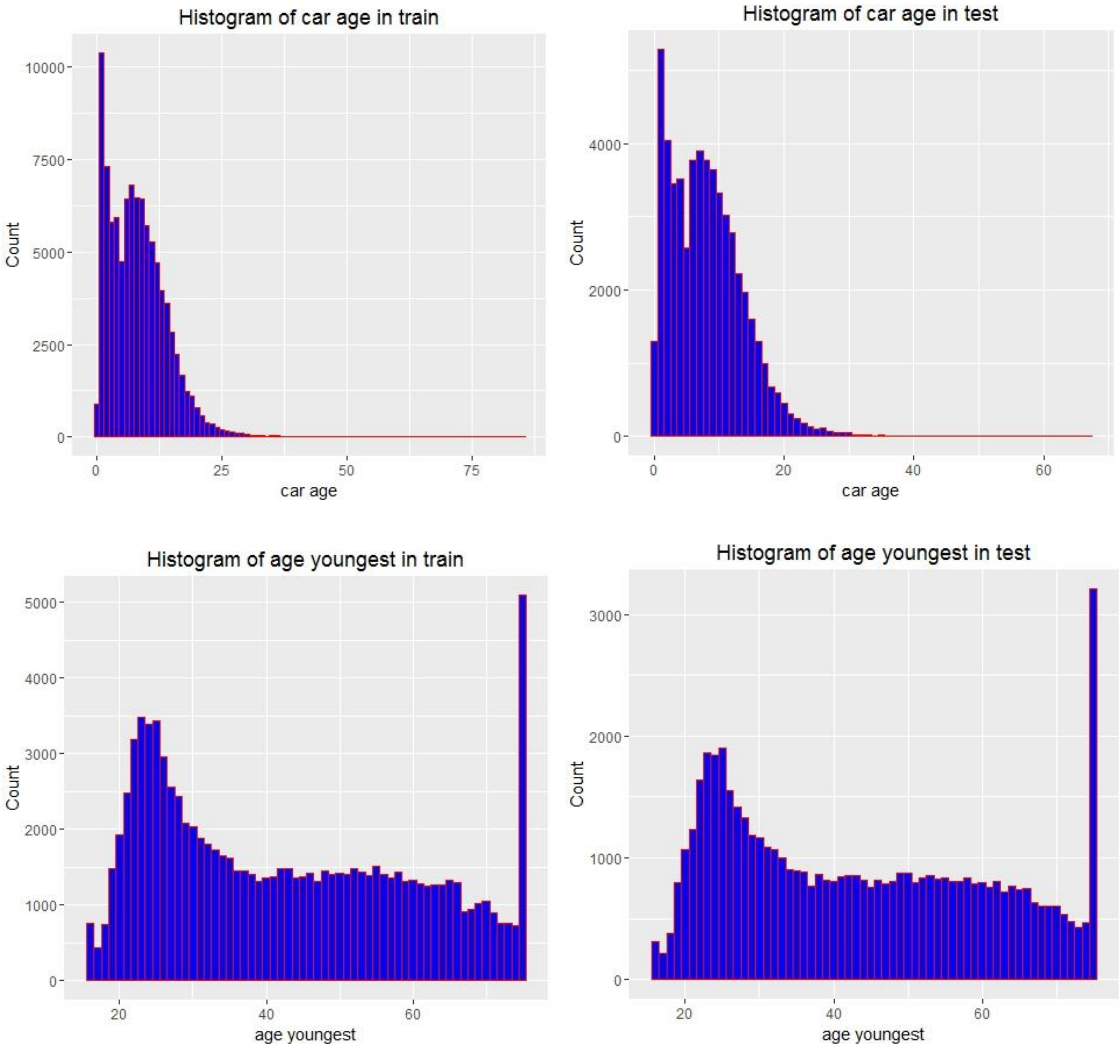


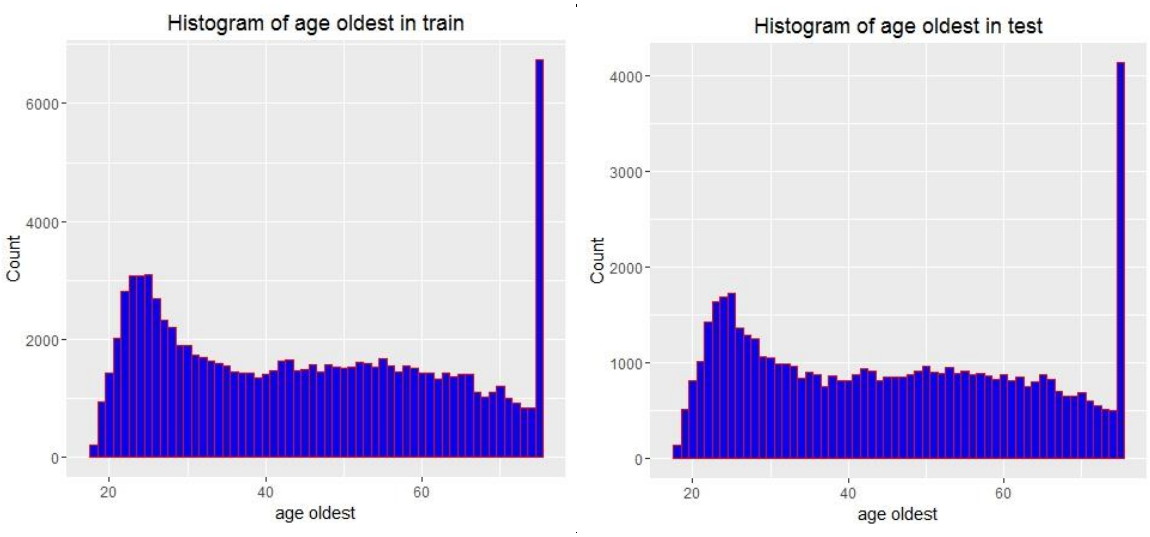


There were 36 states had quote histories in either data set. Customers from New York and Florida had the most quote records while South Dakota and Wyoming had the smallest number (see following plots). Other interesting findings including that customers from ME and GA never purchased policy C class 1; no customers from GA purchased policy D class 1; customers from FL only purchased policy G class 3 and 4; customers from ND were only interested in policy G class 2; there was no one in OH purchased policy G class 1, SD only had a few people bought policy G class 2.



Car age, youngest age in group and oldest age in group are similar in both data sets.





Missing value. 34346 customers don't have "risk_factor" and 836 individuals are missing "c_previous" as well as "duration_previous" in training data set. In the test data, there are 20931 individuals don't have "risk_factor"; 198 customers are missing location; 2197 customers are lacking both "c_previous" and "duration_previous". In addition, there are 138 individuals in test data are missing "car_value" but without "NA" placed. Solutions to treat missing value are listed as following.

"risk_factor": missing risk factors were replaced with the average value of risk factor in both train and test data.

"c_previous": missing values were replaced with 0 (no policy) in both train and test.

"duration_previous": missing values were replaced with average duration in both train and test.

"car_value": 138 records with empty car values were removed from analysis in test data since there might be problems during the data processing.

"location": 198 individual with missing location were removed.

```
> sapply(tmp, function(x) sum(is.na(x)))
     customer_ID       shopping_pt       record_type               day              time             state          location        group_size
               0                 0                 0                 0                 0                 0                 0                 0
       homeowner           car_age         car_value       risk_factor       age_oldest     age_youngest    married_couple        C_previous
               0                 0                 0             34346                 0                 0                 0               836
duration_previous                 A                 B                 C                 D                 E                 F                 G
             836                 0                 0                 0                 0                 0                 0                 0
            cost
               0
```

```
> sapply(testFinal, function(x) sum(is.na(x)))
     customer_ID       shopping_pt       record_type               day              time             state          location        group_size
               0                 0                 0                 0                 0                 0               198                 0
       homeowner           car_age         car_value       risk_factor       age_oldest     age_youngest    married_couple        C_previous
               0                 0                 0             20931                 0                 0                 0              2197
duration_previous                 A                 B                 C                 D                 E                 F                 G
            2197                 0                 0                 0                 0                 0                 0                 0
            cost
               0
```

```
> table(testFinal$car_value)

          a     b     c     d     e     f     g     h     i
  138    84   143  1756  9202 18112 15160  8323  2477   321
```

Feature engineering

"shopping_pt": column **"**shopping_pt" was converted to the number of quotes (column "quoteTimes") placed by a customer before purchasing a policy, where "quoteTimes" = "shopping_pt" (purchase point) – 1.

"day": Since the "day" in training data basically fall into two categories, weekdays and weekends, column day was changed to a binary variable named "dayBinary", where 1 denotes weekdays (0-4) and 0 denotes weekends (5-6).

"time": column **"**time" of all quotes for each customer was converted to a single value in column "timeLen", which is the length of time between first quote and purchase point. This transformation may better carry the information of time points in quotes. Mean normalization was performed on "tiemLen".

"state": column "state" was converted to a new column "stateID", with state names changed to categorical values from 1 to 36.

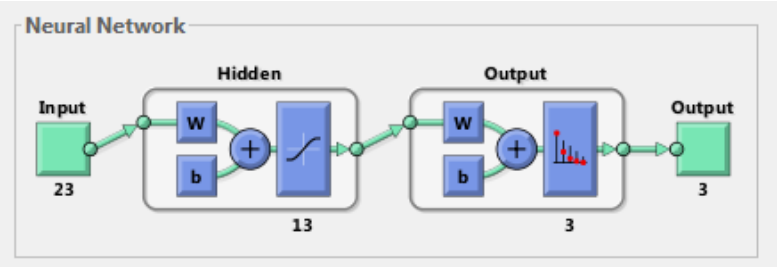
"car_value": column "car_value" was converted to a new column "car_value_categ" with integers 1 to 9 represent nine car values.

"cost": for each customer, the average "cost" of the policy they quoted or purchased were calculated as a new column "cost_avg" in both data. Mean normalization was performed on the new variable.

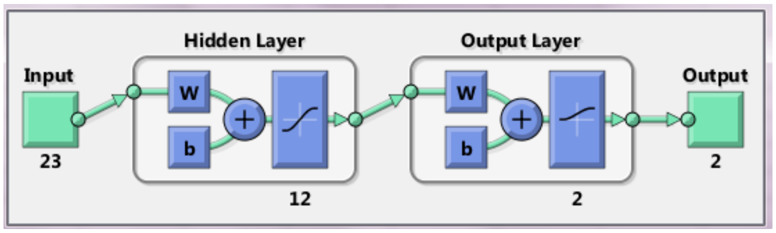
In addition, "car_age", "age_oldest", "age_youngest", "duration_previous_new" ("duration_previous" with NA replaced)

Model construction

The NN models were constructed using Matlab R2014a. For policy A, C, D, F, and G, which had more than two classes to predict, I used NN pattern recognition & classification model to perform the analysis. This model has a two-layer feed-forward network, with sigmoid hidden and "softmax" output neurons. An example diagram is showing below with 13 hidden neurons and 3 classes to predict.



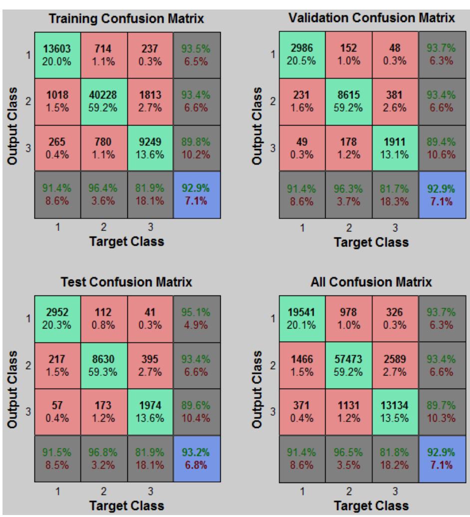
For policy B and E, NN pattern recognition & classification model didn't work very well. So I customized a NN with a hyperbolic tangent sigmoid transfer function for the first layer and a sigmoid transfer function for the second layer. Scaled conjugate gradient backpropagation was used as training algorithm. (example diagram showing below).

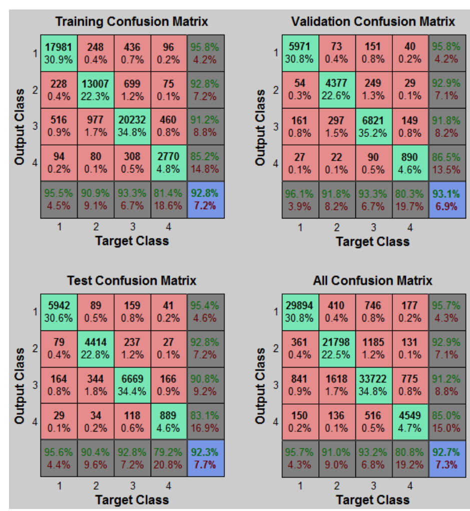


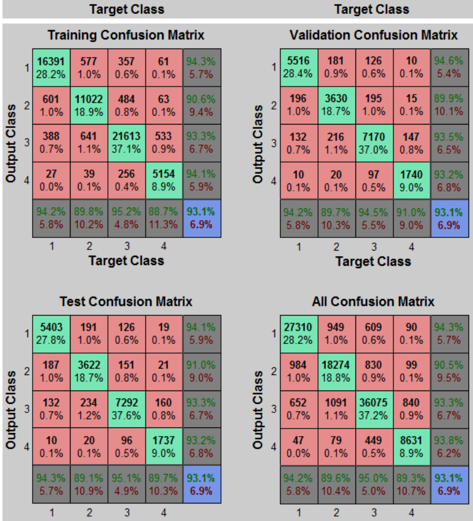
In both models, train data was split into three parts, 60% training, 20% cross-validation and 20% testing. The training set is used to teach the network. Training continues as long as the network continues improving on the validation set. The test set provides a completely independent measure of network accuracy. After train the NN, the test data was used to predict unknow policy purchases.

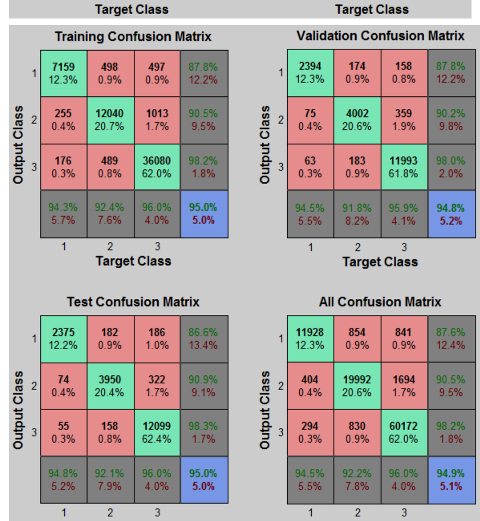
Results

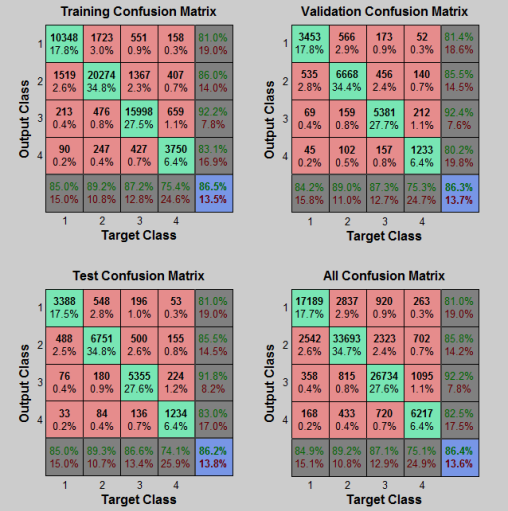
Overall, the training processes went very well. The NN modes for all policies were well trained; the prediction accuracies for multi-class outcomes were usually higher than 85% in the test set of train data (see following confusion plots of A, C, D, F, G).











However, despite the good performance of the NN classifier on multi-class outcomes, their performance on new test data was poor – only one category was successfully predicted for each policy.

Due to the poor performance of the NN pattern recognition & classification model on all policies, a customized NN model was selected to predict binary outcomes, policy B and E. It successfully predicted two categories for B and E. However, it didn't solve the problems of multi-class issues, therefore I kept the results from NN pattern recognition & classification model for A, C, D, F and G.

Discussion

Due to the short period of time, many data explorations and algorithms were not performed. The following things could be done for further improvement.

1. Feature selection algorithms may be applied to remove the redundant features. LASSO regression or elastic net might be good options to play with.
2. Some clustering algorithms can be used to discover the substructures of the customers, for instance, K-means or principle components analysis.
3. Missing values were treated in a very simple and naïve way in my analysis. Missing value prediction methods (e.g MICE package in R) can be introduced to replace the values with predicted values.
4. There were some errors in the "car_value" column in test data. It would be great if there is original data set to look at and try to rescue the 138 samples instead of removing them.
5. The NN models were not very well performed on multi-class policies. Some options may help to improve the model, for example, split data differently; use different combinations of transfer algorithm, increase the number of neurons in hidden layer, increase the number of layers, use a more sophisticated training algorithm such as Bayesian regularization backpropagation (it takes much longer training time and more memory).
6. Try more machine learning algorithms, such as logistic regression, SVM and deep learning.