Semester Project

Mange Chen(machen), Yining Wang(wangyini), Xing Wei(xw46), Wei Zhang(wz30)[1]*

Abstract
Introduction: In this project, we set up a model to predict the likelihood that drivers will initiate auto insurance claims next year. In order to explore new and more powerful methods, we use machine learning so that more accurate predictions allow them to further adjust prices and hope to make it easier for more drivers to get access to car insurance.According to the target class in the training dataset, this is a binary classification problem.

Keywords
Binary Classification — Class imbalance — Logistics regression

[1] *Computer Science, School of Informatics , Computing and Engineering, Indiana University, Bloomington, IN, USA*

Contents

1. Problem and Data Description

Problem Statement
This project is building a new and powerful model to predict the probability that a driver will initiate an auto insurance claim in the next year. A more accurate prediction will allow them to further tailor their prices, and hopefully make auto insurance coverage more accessible to more drivers.

Data Description
In the train data and test data, features which belongs to similar groupings are tagged as such in the feature names(e.g., ind, reg, car, calc).
In addition, feature names include the postfix bin to indicate binary features and cat to indicate categorical features. Features without these designations are either continuous or ordinal. Values of -1 indicate that the feature was missing from the observation. The target columns signifies whether or not a claim was filed for that policy holder.

2. Data Preprocessing & Exploratory Data Analysis

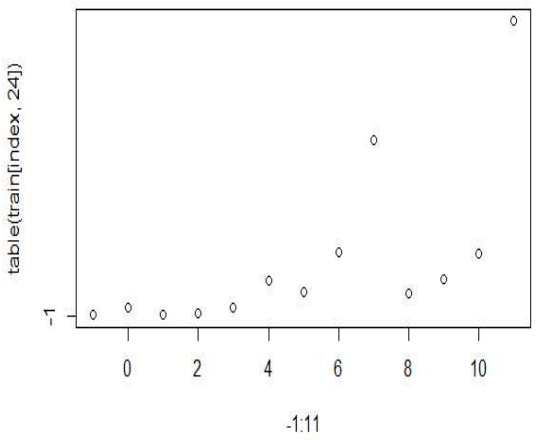
2.1 Handling Missing Values

Data Preprocessing

```
[1] "216 Missing value of 4 variable:
0.000362895909356666"
[1] "83 Missing value of 6 variable:
0.00013946113317608"
[1] "5809 Missing value of 7 variable:
0.00975954785857812"
[1] "107772 Missing value of 23 variable:
0.181064897885123"
[1] "107 Missing value of 24 variable:
0.000179767881023904"
[1] "5 Missing value of 25 variable:
8.40036827214505e-06"
[1] "411231 Missing value of 26 variable:
0.690898368984496"
[1] "266551 Missing value of 28 variable:
0.447825312661707"
[1] "11489 Missing value of 30 variable:
0.0193023662157349"
[1] "569 Missing value of 32 variable:
0.000955961909370107"
[1] "5 Missing value of 35 variable:
8.40036827214505e-06"
[1] "1 Missing value of 36 variable:
1.68007365442901e-06"
[1] "42620 Missing value of 38 variable:
0.0716047391517644"
index[1]: 4 6 7 23 24 25 26 28 30 32 35 36
38
```

Missing Value Processing
ps_car_03_cat has 69.09% of missing values.
ps_car_05_cat has 44.78% of missing values.

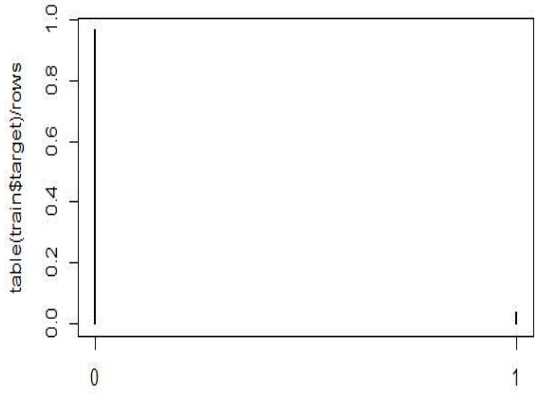
Because *ps_car_03_cat* and *ps_car_05_cat* have a large proportion of records with missing values, we should remove these variables.
For the other categorical variables with missing values, we can leave the missing value -1 as such.
For the column of missing value is **Continuous**, we should replace by the mean
For the column of missing value is **Ordinal**, we should replace by the mode.
We do not make change when the it is **Categorical**.



2.2 Exploratory Data Analysis

Data exploration(visualization and cleaning)
Features and target
Data type can be defined as categorical, continuous or ordinal.

1. Imbalanced Class(target).



The plot shows the class(target) is imbalance.There are high percentage of 0 nearly 98%. Class balance has to be balanced in the training set.
The ways to deal with imbalanced class are undersampling, oversampling, synthetic data generation, cost sensitive learning.

- **Undersampling**
  This method works with majority class. It reduces the number of observations from majority class to make the data set balanced. This method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.

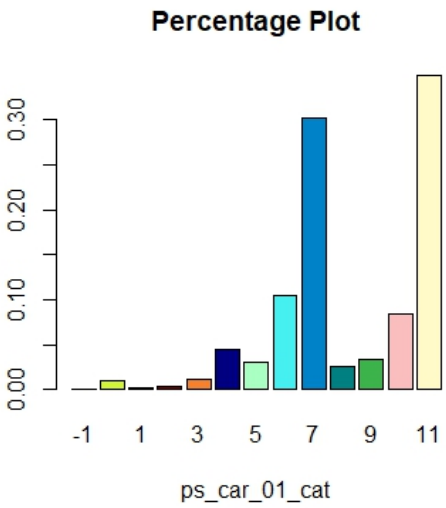
- **Oversampling**
  This method works with minority class. It replicates the observations from minority class to balance the data. It is also known as upsampling. Similar to undersampling, this method also can be divided into two types: Random Oversampling and Informative Oversampling.

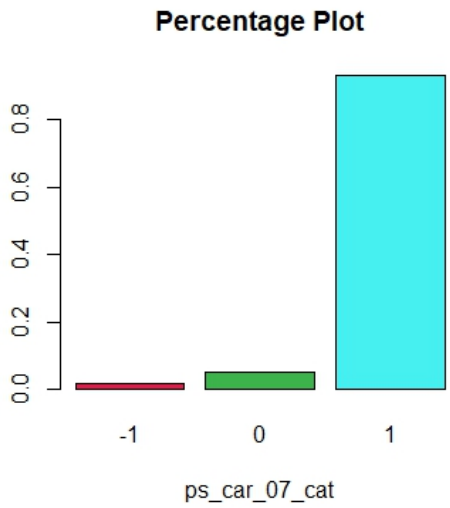
- **Synthetic Data Generation**
  Instead of replicating and adding the observations from the minority class, it overcome imbalances by generates artificial data. It is also a type of oversampling technique.

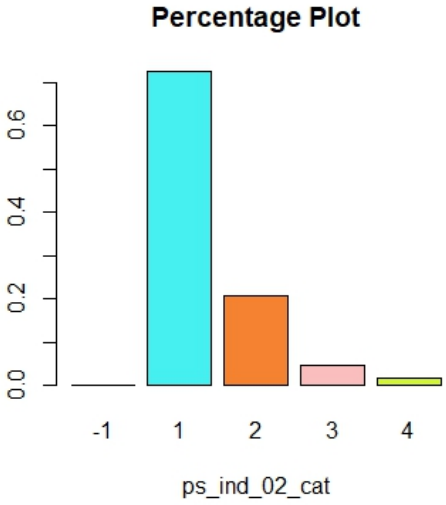
- **Cost Sensitive Learning (CSL)**
  This method is another commonly used method to handle classification problems by using imbalanced data. It's an interesting method. In simple words, this method evaluates the cost associated with misclassifying observations.

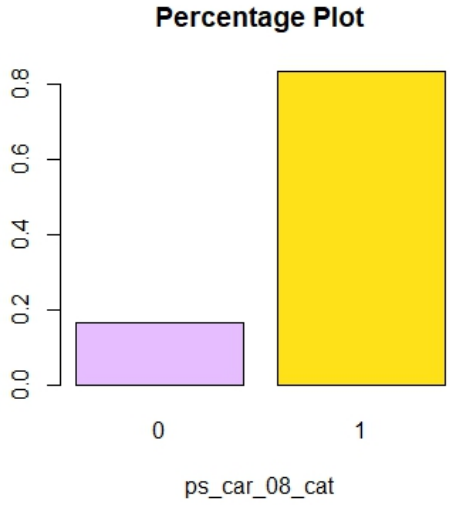
We picked undersampling to rebalance the class because this method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.
Explore the anomaly detection on the train + test data set is the same as the less frequent categories, which are more likely to have label 1, we could find "strange" samples by using unsupervised methods. For example, if we train a basic automatic encoder, the sample reconstruction error for the AUC score is much higher.

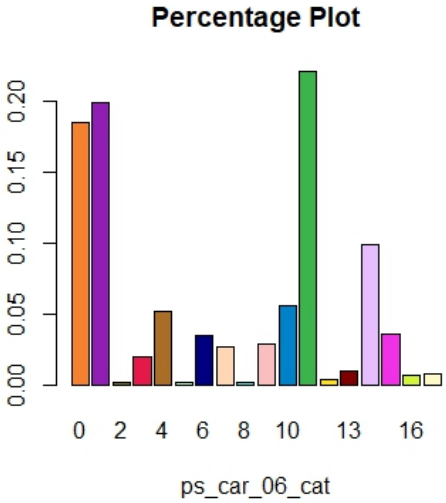
2. The Relationship: The analysis of each category's percentage.

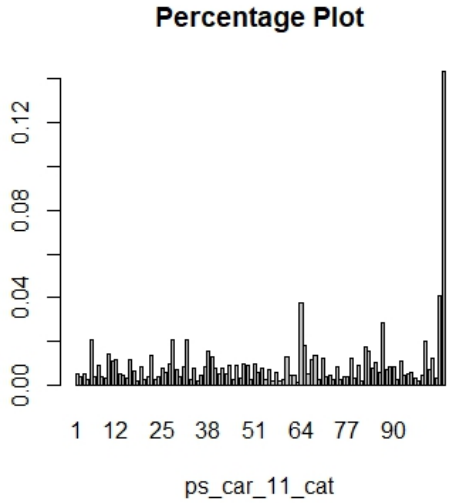

Percentage Plot
0.00
0.10
0.20
0.30
-1
1
3
5
7
9
11
ps_car_01_cat


Percentage Plot
0.0
0.2
0.4
0.6
0.8
-1
0
1
ps_car_07_cat

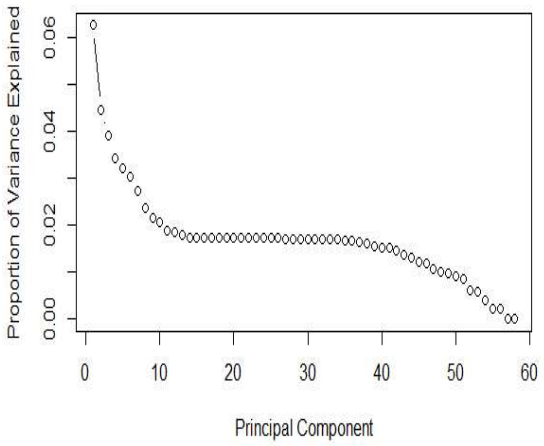

Percentage Plot
0.0
0.2
0.4
0.6
-1
1
2
3
4
ps_ind_02_cat


Percentage Plot
0.0
0.2
0.4
0.6
0.8
0
1
ps_car_08_cat


Percentage Plot
0.00
0.05
0.10
0.15
0.20
0
2
4
6
8
10
13
16
ps_car_06_cat


Percentage Plot
0.00
0.04
0.08
0.12
1
12
25
38
51
64
77
90
ps_car_11_cat

3. PCA or not
   Yes. Principal Component Analysis (PCA) identifies the combination of components (directions in the feature space) that account for the most variance in the data.



   According to loadings of PCA, the PC57 and PC58 are dropped because of low explanations for the target variable

4. Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. We decided to use 10 folds to partition the dataset.

3. Algorithm and Methodology

Briefly explain the algorithms in this section, i.e., linear regressions. You can add more subsections if needed, i.e., regression trees etc.

1. **Logistic Regression**
   Logistic regression is a machine learning algorithm for classification. The probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

   Advantage: It is designed for classification, and is most useful for understanding the influence of several independent variables on a single outcome variable.

2. **Naive Bayes**
   Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.

   Advantage: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

3. **K-Nearest Neighbours**
   Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point.

   Advantage: This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.

4. **Decision Tree**
   Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

   Advantage: Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data.

5. **Random Forest**
   Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of data sets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

   Advantage: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

6. **Support Vector Machine**
   Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.
   Advantage: Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

4. Experiments and Results

Experiments:
By designing 6 different models, we can apply those models to test dataset. After that, we will get 6 different accuracy number and compare all vary values for picking the largest accuracy result.
This result belonging to the algorithm will be the answer.
By testing different models, we will get result in order to choose the best model and algorithm.

1. Normalization This method is based on the grade conversion. The first step is to allocate a space for the ordering features from 0 to 1, and then apply the inverse function ErfInv of the error function to shape them like Gauss, and then subtract the average value. This trafo does not involve binary features. The result is usually better than the standard average/standard scaler or min/max.
2. Naive Bayes model The result from Naive Bayes is not good. The reason we suspect is because of 0 probability. To figure this out, we need add Laplace factor and m estimation.

How to measure the accuracy and error:

1. Divided the training dataset to train dataset and test dataset:
   (a) Train our model according to the training dataset.
   (b) Predict the result (probability to claim the insurance) over test dataset by trained model.
   (c) Use mean(pred) to divide the pred to two parts.
   (d) Compare the predicted result with the class tagged by the original test dataset by using this equation:

   $$pred\_class = ifelse(pred > 2 * median(pred), 1, 0)$$

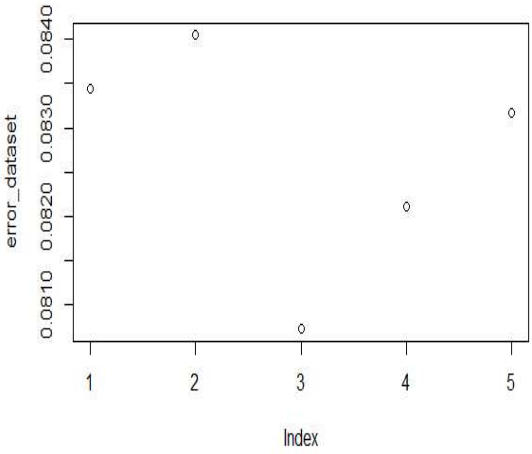
   (e) Use $2(2*)$ to improve the performance of accuracy.
2. Measure error:
   According to the result, we can find there is a relationship between error and accuracy:

   $$error = 1 - accuracy$$

Result:

```
accuracy     algorithm        score
0.8966351 -- navie bayes
0.9035424 -- decision tree -- 0.24245
0.9347734 -- xgboost       -- 0.24245
0.9685502 -- logitic       -- 0.24656
```

So we choose logistic regression which has the highest accuracy. Using K-fold cross validation to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. (k = 5) Below is the image about the error over 5 folders or 5 different dataset.



5. Summary and Conclusions

1. **Summary**
   In this Kaggle project, we work on several workstations and run Ubuntu Linux 16.04. We have tried many ML algorithms including cattorost, deepboost, randomForest, glmnet, lightgbm, mxnet, rerf (Randomer Forest), rgf (normalized greedy forest), neural networks in pytorch/nn (accessed via mesh packets in R) and xgboost. But in the end, we ended up using only logistic regression. According to other competitors reults on Kaggle, neural network seems to be the fastest and best performing of all enhancement algorithms. The result we got from this algorithm becomes the highest number. In addition, we manually changed these algorithms using our own optimization algorithm and changed them in the order of shorter retention times.
   By training this data and competing, we learned and applied multiple techniques such as handling missing values, plot, biplot and a whole bunch of algorithm (PCA), Naive Bayes, logistical regression we leaned in class.
   We divided this project to four parts.
   (a) Understand and identify the problem: binary classification.
   (b) Data relationship and preprocessing the data
   This step is significant. Because when we predicted the model later, we changed the algorithm

and did not really improve performance based on the accuracy and score results.
Essentially, we handled the missing data by replacing with mode for categorical data and mean for continuous data. Then we use PCA to reduce the dimension (we pick the first 48 principle component to explain the 95% variable) and reproduce the training dataset and test dataset.

(c) Apply the model
This is relatively simple step, we have a lot of preparations and just picked the algorithms we want to use and predict the result (probability).

(d) Compare the result and submit the csv file to Kaggle
After the prediction with several model, we compare the accuracy to pick the best one, according to the result in 4, logistic regression is picked.
Unfortunately, we put effort but did not improve our scores a lot.
There are couple things we can discover more. First, we can use feature selection to improve the quality of data. Then we may try to use Neural Network to improve our output.

2. **Conclusion**
The given data set is very unbalanced. We use random forests, classifiers and gradients to increase the accumulation, then logistic regression appears in the first phase.
Logistic regression is a benchmark method for this type of task. Normally we cannot see the values of successive left-side variables and represent them as noise in the form of some binary observations that reflect the results of that variable's value.
In logistic regression, we use the sigmoid function to simulate the probability of our hypothesis $\in 0, 1$. We trained a logistic regression model on the training set.
There is no formalization. For the output vector y, we use the original probability generated by the sigmoid function instead of converting the value to binary decision 0 or 1.
Obviously, we need to try more advanced methods such as neural networks and create a better model for the current task. But logistic regression can provide at least one paradigm and help us to think about which methods are most likely to succeed.

Acknowledgments

We thank our gorgeous instructor Hasan Kurban from who provided insight and expertise that greatly assisted the research.

References

https://analyticsindiamag.com/7-types-classification-algorithms/

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

https://www.kaggle.com/bertcarremans/data-preparation-exploration

https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/discussion/44608

https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/discussion/44629