Health Insurance Cross Sell Prediction

Amy Hung[1], Dominic Wei Yaw Ting[1], Kim Chung[1], Lin Qin[1] & Xinyuan Wei[1]

Abstract
Data science and machine learning are everywhere, even when we do not realize it. Less known to the public, but the insurance industry heavily relies on these. In this paper, we apply the modern machine learning techniques on the insurance policyholders' data to analyze and predict their behavior. Using Python language, our approach to the data resulted in exciting insights to help the insurance companies in modeling their businesses.

Keywords
Python — Insurance — Prediction

[1] *J. Mack Robinson, Georgia State University, Atlanta, United States*
The authors' name are listed in alphabetical order.

Contents

Introduction

Insurance is a means of protection from financial loss. An insurance company provides insurance to its customer, known as a policyholder or an insured. The loss may or may not be financial, but it must be reducible to financial terms.

The insured receives a contract, called the insurance policy, which details the conditions and circumstances under which the insurance company will compensate the policyholder. The amount of money charged by the insurance company to the policyholder for the coverage set forth in the insurance policy is called the premium. Since insurance operates through pooling resources, most insurance policies are provided for individual members of large classes, allowing insurance companies to benefit from the law of large numbers in which predicted losses are similar to the actual losses.

Here is an insurance company that has provided health insurance to its policyholders. It wants to predict whether the policyholders from the past year will also be interested in the insurance company's vehicle insurance.

This paper aims to build a model to predict the policyholders' response towards vehicle insurance. This is necessary for the insurance company to plan how to reach out to its customers and optimize its business model and revenue. In this model, the key is to maximize the recall score, so that the insurance company could promote insurances to all possible customers in an efficient way.

1. Data Description

The data used in this paper is from Kaggle.com. It contains information about the 381,109 policyholders' demographics, vehicles, and policies in 12 columns without any missing data. The summary of the variables are shown in Table 1.

2. Methods

Our primary goal is to estimate whether the policyholders from the past year will be interested in the insurance company's vehicle insurance, thus our optimization will focus on how to increase the precision of the models. Our main target variable to determine the precision of models is based on variable *Response*. The variable *Response* is defined as whether the customer is interested in buying vehicle insurance or not. We will classified a customer who possibly interested in purchasing the vehicle insurance as interested (Response =1) and vice versa. We will be using three main algorithms such as Logistic Regression,Decision Tree and Random Forest in modeling. In this paper, we will focus on estimating recall and choose the best model which has the highest recall score.

Table 1. Summary of variables

Variables	Description	Type
id	Unique customer ID	int64
Gender	Gender of the customer, Male, Female	object
Age	Age of the customer	int64
Driving_License	=1 if customer has driving license	int64
Region_Code	Unique code for the region of the customer	float64
Previously_Insured	=1 if customer has vehicls insurance	int64
Vehicle_Age	Age of the vehicle, <1 yr, 1-2yr, >2yr	object
Vehicle_Damage	=Yes if customer had vehicle damage in the past	object
Annual_Premium	Annual premium that customer needs to pay	float64
Policy_Sales_Channel	Anonymized code for the different sales channels	float64
Vintage	Number of days that customer has been associated with the company	int64
Response	=1 if customer is interested in vehicle insurance	int64

3. Data Preprocessing

After data cleaning process, we found that there is no missing value,however the original data has a response ratio of $interest$: $not\ interested$ equals to $1 : 7.159$.This imbalanced data will negatively impact algorithms such as Logistic Regression that optimizes across the entire training set. As a result, after using pure random sampling to get training data and test data set, we use oversampling and undersampling method to make a 50/50 split on the training data. Therefore, we have four data set to be used for our models building and analysis [1].

1. Original training data set-this data was used to act as a control data set for the purpose of testing the function of the weighted model like the weighted logistic regression model.

2. Oversampling data set – this data involves randomly duplicating examples from the minority class (response=1) and adding them to the training data set, and its ratio of response value is $1 : 0 = 267502 : 267502$.

3. Undersampling data set- this data randomly selecting examples from the majority class to discard from the training dataset to match the minority class and its ratio of response value is $1 : 0 = 37385 : 37385$.

4. Test data set – this data keeps the same distribution as the source data and is used to calculate recall, precision and accuracy of models.

4. Model

4.1 Logistic regression

Logistic Regression is a linear model using a sigmoid function for classification. It is being used to predict binary outcome from a linear combination of predictor variables.

To solve the binary classification problem, there are two output probabilities. One is the target probability $P(Y = 1 \mid X)$ and another is non-target probability $P(Y = 0 \mid X)$.

$$y = \sigma(f(x)) = \frac{1}{1+e^{\beta^\top x}} \tag{1}$$

Next, use maximum likelihood estimation to estimate the parameter and in order to simplify, people make a slight twist toward MLE by using Log Loss. After simplified the Log Loss function,

$$L(\beta) = \sum_{i=1}^{n} log(1+e^{\beta^\top x}) - y_i\beta^\top x \tag{2}$$

There are several methods to minimize $L(\beta)$ such as coordinate descent (CD) algorithm (liblinear), Newton's method in optimization (newton-cg), Limited-memory BFGS (lbfgs), Stochastic Average Gradient (sag) and unbiased version of Stochastic Average Gradient (saga).

In our case, we firstly use Limited-memory BFGS to optimize the problem. We use original dataset, oversampling dataset and undersampling to train and test the model. The training and testing datasets have nearly the same recall. For oversampling and undersampling dataset, recall are 94%; however, for the original dataset we only have 0.05% of recall.

Since the original dataset has an imbalanced problem and weighted logistic regression has better ability to solve the imbalanced dataset by weighing each class

$$L(\beta) = -\frac{1}{n}\sum_{i=1}^{n} w_0(y_i\beta^\top x) + w_1(1+y_i)log(1+e^{\beta^\top x}) \tag{3}$$

where w_0 is class weight for class 0
w_1 is the class wight for class 1
We tried to use weighted logistic regression to improve the performance of the model.

After adjusting the weights inversely proportional to class frequencies in the input data, we increased the recall of original dataset from 0.05% to 95%. From the table, we can observe that in general, weighted logistic regression has better performance, compared to logistic regression.

Table 2. Performance of logistic regression models with default parameters.

Datasets	Methods	Recall	Precision	Accuracy
Original	Unweighted	0.0005	0.3333	0.8776
	Weighted	0.9511	0.2595	0.6620
Oversampling	Unweighted	0.9422	0.2627	0.6695
	Weighted	0.9511	0.2595	0.6620
Undersampling	Unweighted	0.9438	0.2623	0.6685
	Weighted	0.944	0.2622	0.6682

However, we obtained the highest recall score in the Weighted logistic regression on oversampling data together with overfitting problem.Thus, we tried different regularization methods such as L1, L2(default) and elastic-net in order to solve the problem. Furthermore, we also adjusted the inverse of regularization strength, and the smaller the value is, the stronger the regularization.

It turns out that using the L1 regularization method and elastic-net regularization method seems to improve performance by a little. After adjusting parameters, in general, we obtained a better recall score, and the performance of the model by using these three regularization methods are similar.

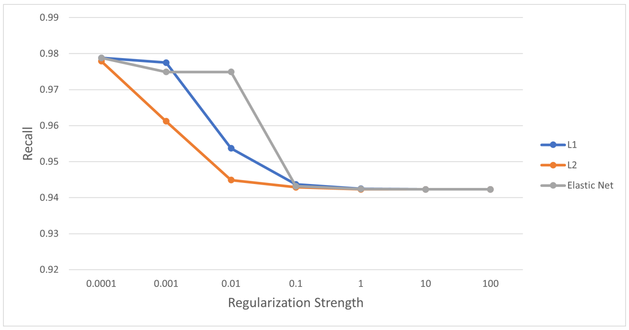

Figure 1. Logistics Regression with Regularization

4.2 Decision Tree

A decision tree is a tree-like collection of nodes intended to create a decision on values affiliation to a class or an estimate of a numerical target value. Each node represents a splitting rule for one specific attribute. For classification, this rule separates values belong to different classes; for regression, it separates them in order to reduce the error in an optimal way for the selected parameter criterion. And the building of the new nodes process will be repeated until the stopping criteria are met [2].

Decision trees are among the most popular machine learning algorithms because the decision tree can be used to visualize the decision making. Besides that,it works well on a large dataset.

With the default setting, the decision tree model has the best performance by using the undersampling dataset. It has 0.7129 for the recall score of class 1. By contrast, the model which uses the original dataset and oversampling dataset has the recall score less than 0.3 for class 1. However, there are some overfitting problems by using the undersampling dataset. Thus, we tried to adjust the maximum depth of the tree and the maximum the leaf nodes. We used 5, 10 and 20 as values to adjust the parameters.

Overall, by adjusting those parameters, all the recall scores achieved recall score of 0.97, and setting maximum the leaf nodes to 5 makes the performance best.

Table 3. Decision tree performance after tuning parameters

Max. Leaf	Max. Depth	Recall	Precision	Accuracy
5	5	0.9794	0.2446	0.6274
5	10	0.9794	0.246	0.6274
5	20	0.9794	0.2446	0.6274
10	5	0.9777	0.2454	0.6294
20	5	0.9777	0.2453	0.6293

4.3 Random Forest

Random Forest(RF) is one of the most successful ensemble learning techniques which has been proven to be popular and powerful techniques in the pattern recognition and machine learning for high-dimensional classification and skewed problems. [3]

With default parameters, the best RF is based on the under sampling training data set. For the training data set, the recall is very high at 99%, but the recall is 86.98% for the test data set. It is lower than the best default weighted logistic regression model with recall of 0.9511.

Table 4. Forest Tree performance with default parameters

Dataset	Recall	Precision	Accuracy
Original	0.1447	0.3286	0.8592
Oversampling	0.2861	0.3051	0.8329
Undersampling	0.8698	0.27	0.6963

Because the trees that are grown very deep and learn highly irregular patterns, they overfit their training sets. Having more trees in the bag reduces the variance. We fine-tuned the best default RF model on the under sampling training data set by gradually varying the number of trees, and depth to minimize overfitting error and increase recall.

Based on the result of Table 5 , we found that the recall increased a lot from 87% to about 95% after tuning. With max depth =5 and tree numbers=5 , we can get the highest recall of 97.79%, which is similar with the highest recall in the logistic regression models with adjusted parameters. Even though the overfitting problem still exists for the precision and accuracy, the recall remains high for training and test data set.

Table 5. Forest Tree performance after tuning parameters

		Test data set		
Tree numbers	Max depth	recall	precision	accuracy
Default	None	0.8698	0.2700	0.6963
5	5	0.9779	0.2512	0.6406
10	5	0.9731	0.2545	0.6480
15	5	0.9762	0.2520	0.6426
5	10	0.9617	0.2607	0.6617
10	10	0.9606	0.2613	0.6630
15	10	0.9609	0.2620	0.6641
5	15	0.9460	0.2629	0.6689
10	15	0.9522	0.2644	0.6701
15	15	0.9541	0.2647	0.6701

5. Performance

In this paper, we used three algorithms: Logistic Regression, Decision Tree and Random Forest. As shown in Table 6 , for the models with default parameters, we could see that the recall scores of models on the original data set are poor. But for weighted logistic regression models, after adjusting weights inversely proportional to class frequencies in the input data, we increased the recall of original dataset from 0.05% to 95%, which means the class weight parameter of 'balanced' helps in solving the imbalanced data problem. And after we do oversampling and undersampling, we could see that scores of three models increased a lot, and the recall score of the logistic regression models is quite similar with that of the weighted logistic model on original data.

Table 6. Recall with default parameters

	Original data set	Oversampling	Undersampling
LR	0.0005	0.9422	0.9438
WLR	0.9511	0.9511	0.944
DT	0.2963	0.3043	0.7129
RF	0.1447	0.2861	0.8698

To improve the performance of models, we tried to tune parameters of the best models with default parameters in few conditions. For Logistic Regression, we pick weighted logistic regression on oversampling. After tuning the regularization method and strength, the recall score of the logistic regression model increased from 0.9511 to 0.9788. For the Decision Tree, we pick the decision tree model on the Under Sampling data set. After tuning the parameters of Max. Leaf and Max. Depth, the recall score of the Decision Tree model increased from 0.7192 to 0.9794. For Random Forest, we pick the random forest model on the under sampling data set. After tuning the parameters of n estimators (tree numbers) and Max. Depth, the recall score of the Decision Tree model increased from 0.8698 to 0.9779. As seen in Table 7, after tuning, the three models are similar with conditions where Weighted Logistic Regression with penalty= L1 or Elasticnet C=0.0001, Decision Tree with max leaf=5, max depth=5, and Random Forest with tree number=5, max depth=5.

Table 7. Best models performance after tuning

	Recall	Precision	Accuracy
WLR	0.9788	0.2498	0.6378
DT	0.9794	0.2446	0.6274
RF	0.9779	0.2512	0.6406

At the same time, the increase of recall score is at the cost of decrease of precision and accuracy. But our primary aim is to increase recall score in this condition.

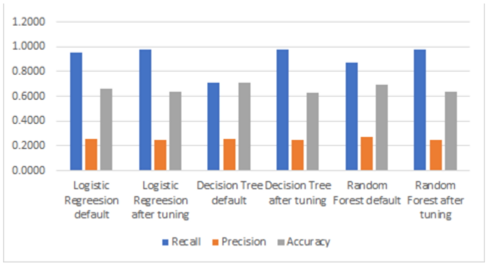

Figure 2. Model Performance

6. Lesson Learned

1. We need to understand the meaning of evaluation scores.
2. Learn to tune the parameters of models to improve the performance of models.
3. We found out that depending on the data's characteristics, we had to use different measurements to test our models' performance.

7. Conclusion

Our primary objective is to maximize the recall score, which measures how much our model could find interested customers among all estimated interested customers. With this method, the insurance company can implement the most efficient way of approaching potential future vehicle insurance policyholders. At first, We had highly imbalanced data in the data set. To avoid the over-fitting issue, we used over/undersampling data. We employed three machine learning algorithms on this task, and Weighted Logistic Regression with the elastic net regularization method outperformed Logistic Regression, Decision Tree, and Random Forest, without tuning parameters. Furthermore, after tuning the parameter, all the models had similar results.

8. Future Work /Reflection

1. We would try boosting models, for example, AdaBoost and XGBoost, to reduce the over-fitting problems.
2. We could apply our models to time series data and see how they perform over the years since the data was limited to policyholders' information from the previous year.

3. Comparing our result to the insurance company's work in the future to see how building a model and executing in a real-life environment has a similarity and difference.

9. Team Contribution

Table 8. Contribution of team members

Name	Contribution
Amy Hung	Code, Report
Dominic Ting	Code, Report, Presentation
Kim Chung	Code, Report
Lin Qin	Code, Report
Xinyuan Wei	Code, Report, Slide

Acknowledgments

We thank Professor Xiangshi Yin for his hard work and patience in providing us the knowledge in Python coding.

References

[1] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

[2] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[3] Ahmad Taher Azar, Hanaa Ismail Elshazly, Aboul Ella Hassanien, and Abeer Mohamed Elkorany. A random forest classifier for lymph diseases. *Computer methods and programs in biomedicine*, 113(2):465–473, 2014.