Credit Fraud Detection Model

Chongzheng Guo, Zhiyang Deng, Juan Guo

CPE-695WS Final Project Presentation
Stevens Institute of Technology

October 31, 2022

Overview

1. Inroduction
2. Description of Data Set
3. Model Formulation
4. Model Improvement
5. Conclusion

Introduction

Problem Formulation

Since the interest in online shopping becomes dominant in our market, the use of credit cards has been rapidly expanded in recent years. Fraud behavior that someone steals the credit card information to perform online transactions has also seen more frequently. Note that credit card fraud can be categorized into two cases: **card-present (CP) scenario** and **card-not-present (CNP) scenario**, which are important to be distinguished since the fraud detection techniques will vary, depending on whether a physical copy of credit card would be used. Especially in the online shopping case, fraudsters are more likely to exploit the pitfalls of CNP scenarios. According to the 2019 Nilson report [1], there was 54% of all fraud cases in 2018 belonging to the CNP scenarios. Therefore, our project will aim to develop machine learning (ML) algorithm for fraud detection under the CNP scenarios, whose data set will contain the online shopping information recorded by banks.

Introduction

Typical Data Set Choice

In the credit fraud detection research, transaction data is typically used for research dataset, which is collected by for example a bank. In most cases, transaction data can be divided into three categories [2, 3]: (i) account-related features; (ii) transaction-related features; (iii) customer-related features.

P.S. **Due to confidentiality reasons, the data provider are NOT allowed to share the original dataset including the features or background information mentioned above**

Literature Review

Related work

As we mentioned above, the class-imbalance problem and the choice of accuracy measure are significant for credit detection problem. The paper [4] points out that relying on one single performance metric could generate misleading result when dataset is highly imbalanced and the final decision in predictive model should consider a combination of several performance metrics.

Most existent literature [5, 6, 7, 8, 9, 10] in supervised learning algorithm of credit fraud detection either only proposes a novel algorithm but not considering appropriate accuracy performance metric, or focus on discussing a general network-based or automatic framework of credit card fraud detection system (FDS) but not caring about the algorithm details.

Description of Data Set

Our dataset contains transactions made by European cardholders via credit cards in September 2013, which shows transactions that occurred within two days, of which 492 were frauds out of 284,807 transactions. The dataset is highly imbalanced, with positive (fraud) accounting for 0.172% of all transactions. This data set only contains numeric input variables, which are the result of PCA conversion.

Description of Data Set

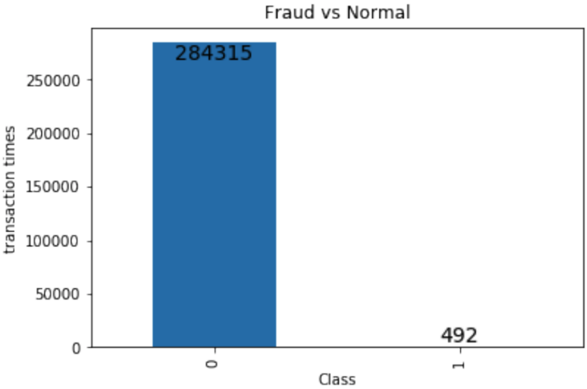


Figure: The number of fraud and normal transaction

Description of Data Set

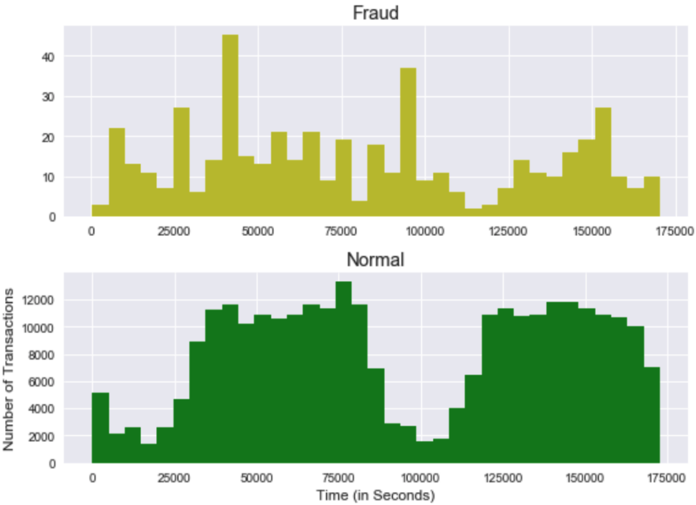


Figure: Time compares across fraud and normal transactions

Description of Data Set

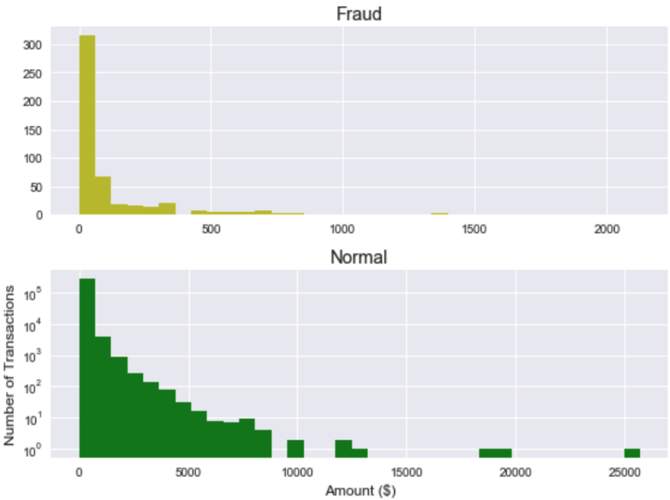


Figure: Amount compares across fraud and normal transactions

Description of Data Set

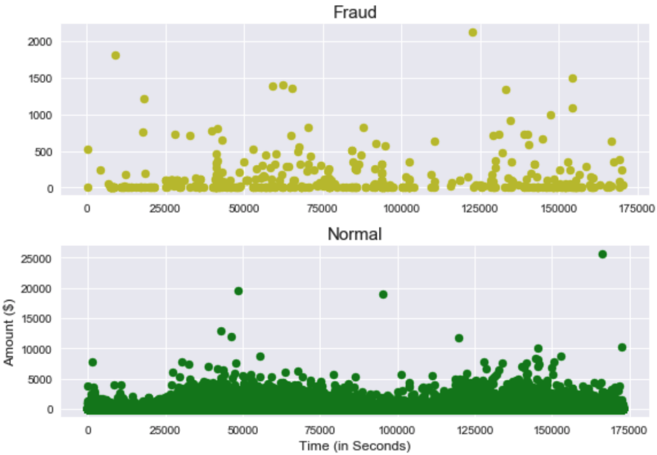


Figure: Amount compares with Time for fraud and normal transactions

Description of Data Set

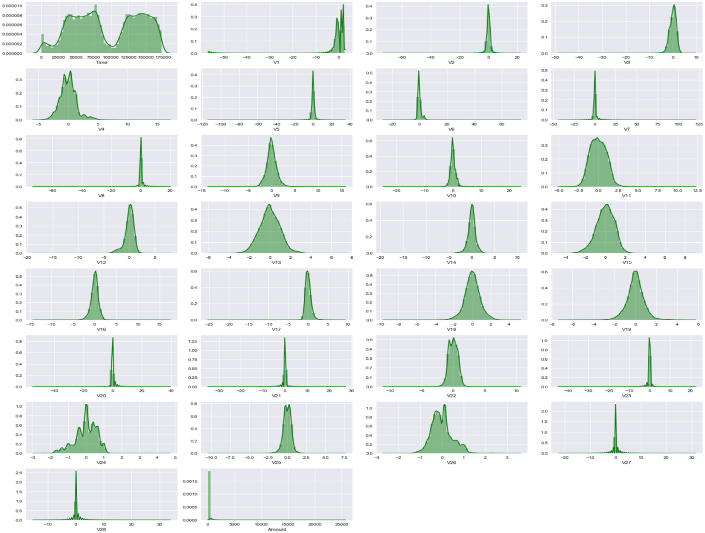


Figure: feature V_1-V_{28} and Amount, Class, Time features distribution

Model Formulation: ML Algorithm

Logistic Model

Logistic regression model with Sigmod function $\phi(z) = (1 + e^{-z})^{-1}$ is firstly applied for this credit fraud detection problem, in which data of each columns need to be normalized, that is, $\bar{x}_i = (x_i - \mu)/\sigma$ where μ is the sample mean of each column and σ is sample standard deviation of each column.

Random Forest Classifier

Random forest, a multinomial classifier, is also a suitable candidate for our modeling, which has satisfied the prediction accuracy but the cost of computation is not significantly increased and whose prediction is relatively robust to an imbalanced dataset.

Model Formulation: Preliminary Result

The accuracy rate of Logistic Regression model & random forest model are 99.91% and 99.96% respectively and confusion matrix of two models are shown in the next slides.

However, the confusion matrix and traditional accuracy rate are not so meaningful for imbalanced dataset, we have to doubt whether this result is the true.

Model Formulation: Preliminary Result

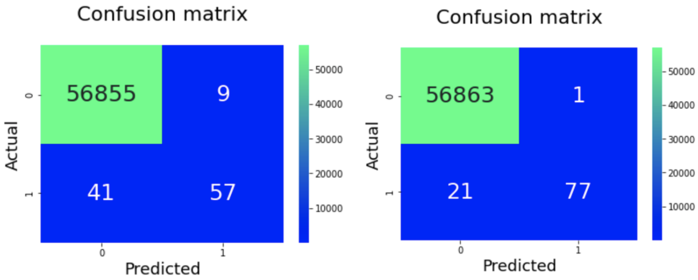


Figure: Confusion Matrix of Logistic Model and Random Forest Classifier

Model Formulation: Introducing New Accuracy Criteria

Average Precision Score

Based on previous slide, the confusion matrix gives us the value of TP, FP, FN, TN, the value of $recall = TP/(TP + FN)$ and the value of $precision = TP/(TP + FP)$. The principle of Average Precision Score is to summarizes a precision-recall curve as the weighted mean of precision achieved at each threshold, that is, $\sum_n (recall_n - recall_{n-1}) * precision_n$.

Model Formulation: Introducing New Accuracy Criteria

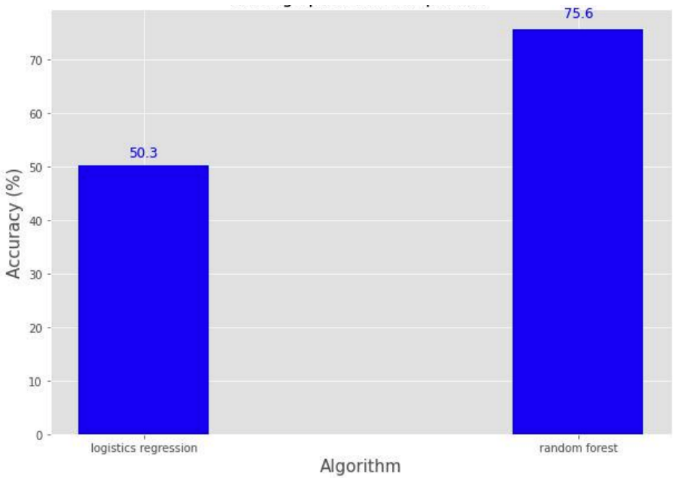


Figure: Average precision scores: Logistic Model v.s. Random Forest

Model Formulation: Introducing New Accuracy Criteria

AUC - ROC Curve

ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.

More precisely, FPR is the x-axis and TPR is the y-axis. FPR refers to the probability that the actual negative sample is incorrectly predicted as a positive sample. TPR refers to the probability that the prediction is correct in the actual positive sample. The more convex to the upper left is the curve , the better is the accuracy.

Model Formulation: Introducing New Accuracy Criteria

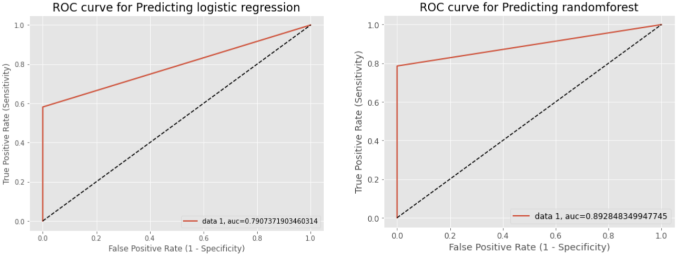


Figure: AUC-ROC curve results: Logistic Model v.s. Random Forest

Model Formulation: Introducing New Accuracy Criteria

Precision-Recall Curve

The precision-recall curve shows the trade-off between precision and recall for different threshold. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall.

In the PR curve, Recall is the x-axis and Precision is the y-axis. In the PR space, the more convex to the upper right is the curve, the better is the accuracy.

Model Formulation: Introducing New Accuracy Criteria

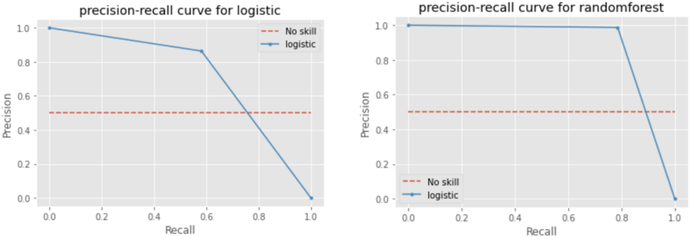


Figure: Precision-Recall curve results: Logistic Model v.s. Random Forest

Model Improvement: Random Sampling Techniques

Random Undersampling & Oversampling

Random over-sampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset. On other hand, random under-sampling involves randomly selecting examples from the majority class and deleting them from the training dataset.

Model Improvement: Random Sampling Techniques

Synthetic Minority Oversampling Technique

The SMOTE method is an extension of random over-sampling method. Other than simply duplicating examples from the minority class in the training dataset, SMOTE chooses to synthesize new examples from the minority class.

- Setting the minority class set A, for each $x \in A$, the k-nearest neighbors (k is fixed) of x are obtained by calculating the Euclidean distance between x and every other sample in set A

- The sampling rate N is set according to the imbalanced proportion. N examples $(x_1, x_2, \cdots, x_N)$ are randomly selected from its k-nearest neighbors, and they construct the set A_1

- For each example $x \in A$ and randomly choose $x_i \in A_1, i = 1, 2, \cdots, N$, we use the formula $\hat{x} = \delta|x - x_i| + x$ to generate a new example, in which $\delta \in (0, 1)$

.

Model Improvement: Random Sampling Techniques

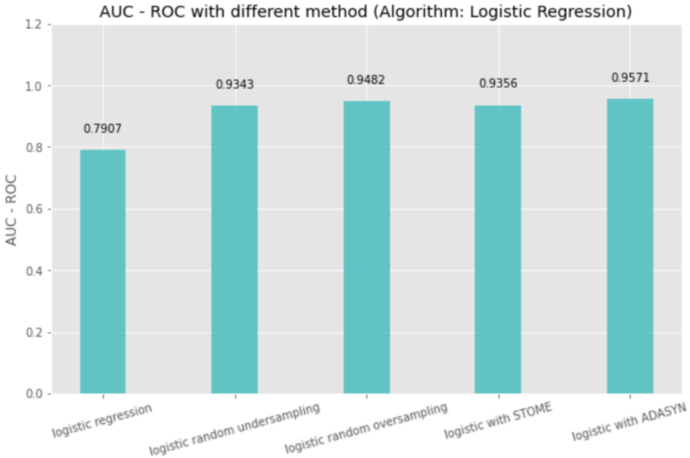
Adaptive Synthetic Sampling Method

The ADASYN is also an extension of random over-sampling method. However, the major difference between SMOTE and ADASYN is the difference in the generation of synthetic sample points for minority data points.

- Calculate the amount of new sample that needs to be generated: $G = (S_{majority} - S_{minority}) * \beta$ with $\beta \in (0, 1)$, in which $S_{majority}$ is the number of samples in majority class, and $S_{minority}$ is the number of samples in minority class.
- For each $x \in S_{minority}$, the k-nearest neighbors (k is fixed) of x are obtained by calculating the Euclidean distance between x and every other sample in set $S_{minority}$. Define that Δ_i is the samples in k-nearest neighbors belonging to the majority class and Z is a normalized factor so that $T_i = (\Delta_i / k)/Z$ is a probability distribution.
- Calculation of synthetic sample generated for each minority data point $g_i = G * T_i$ where G is the total number of synthetic data examples that need to be generated for the minority class.
- Then applying SMOTE method for each g_i to generate the new sample.

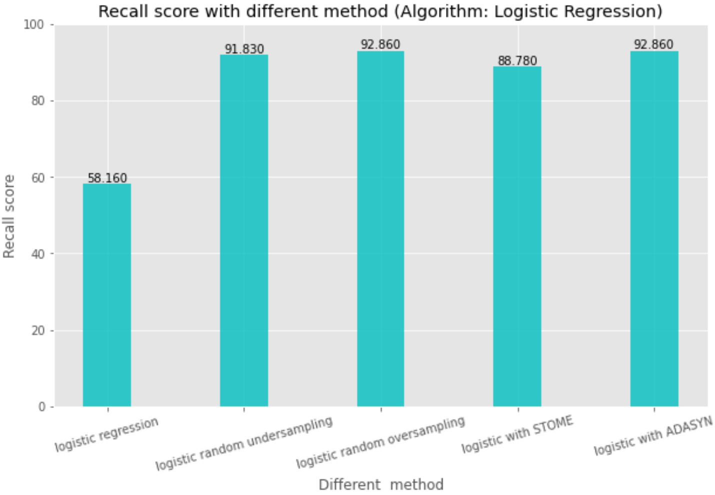
Model Improvement: New Result

Logistic Model with Different Random Sampling Method



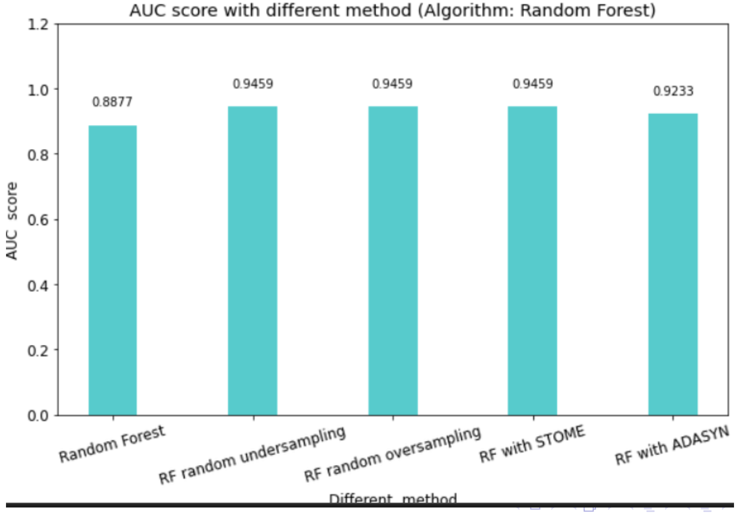
Model Improvement: New Result

Logistic Model with Different Random Sampling Method



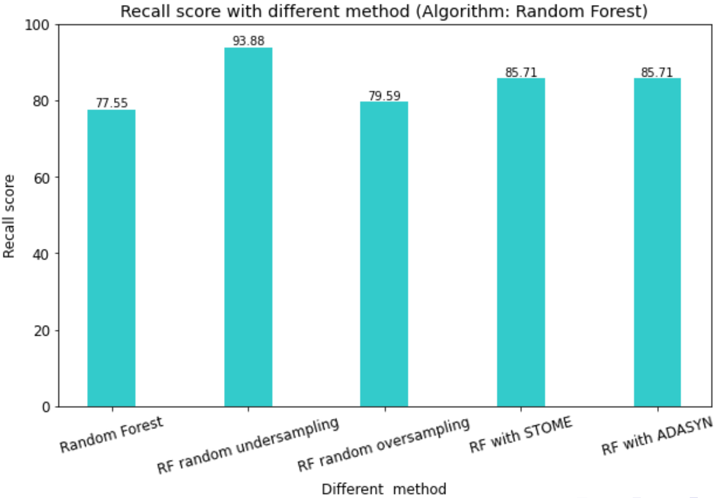
Model Improvement: New Result

Random Forest Classifier with Different Random Sampling Method



Model Improvement: New Result

Random Forest Classifier with Different Random Sampling Method



Conclusion

- In addition to accuracy score and confusion matrix, we also analyze the model through multiple accuracy valuation methods: average precision score, AUC-ROC curve, precision-recall cure. Among those criteria, ROC-AUC curve shows that our model is better.
- We apply four random sampling methods: under-sampling, oversampling, STOME, and ADASYN for imbalanced data. For the **Random Forest Classifier** and the **Logistic Model**, the results of under-sampling are always the worst among the four methods. We think it is caused by losing some of data.
- Without random sampling, the Logistic Model preforms relatively worse than Random Forest. However, its performance enhance dramatically after applying random sampling. In the meantime, the improvement of Random Forest is not significant. We think it is because the original Random Forest model performs good enough, so there is no much space for further improvement.

Thank You!