CPE695-WS Final Report (Group 12)

Chongzheng Guo
CWID: 10468590

Juan Guo
CWID: 10471471

Zhiyang Deng
CWID: 10465446

***Abstract*—In order to solve credit fraud detection under card-not-present scenario, this paper investigate two ML algorithm to build credit detection models: Logistic Model and Random Forest. Compared with existent literature work, we apply four random sampling techniques to solve imbalance dataset problem and utilize multiple accuracy rates to evaluate model performance other than relying on single criteria of accuracy.**

I. Introduction

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.

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. However, due to confidentiality reasons, the data provider are NOT allowed to share the original dataset including the features or background information mentioned above, the dataset (https://www.kaggle.com/mlg-ulb/creditcardfraud/) used in our project contains only numerical input variables which are the result of a PCA transformation in which there are 28 principle components obtained with PCA. Even though this dataset has been cleaned by PCA techniques, we still need to emphasize that one typical characteristic of transaction data is class-imbalance, meaning that The percentage of fraudulent transactions is around 1%. Therefore, choosing appropriate learning strategies to deal with this issue is also significant.

Since credit fraud detection can be naturally categorized into classification problem, the supervised learning algorithms are firstly taken into consideration. At this stage, we chose to implement **logistic regression** and **random forest classifier** to detect credit fraud cases. Under the criteria of **confusion matrix accuracy**, the accuracy rate of two algorithms are 99.91% and 99.96% respectively. However, confusion matrix accuracy is not so meaningful for unbalanced classification problem. If we apply **average accuracy**, which is a criteria of accuracy designed for class-imbalance dataset, the accuracy rate of two algorithms are 50% and 75.6% respectively. The huge difference of accuracy rates of two algorithms implies that our choice of accuracy criteria is inappropriate, which requires more investigations. Our next step will focus on the choice of accuracy criteria, after settling down the accuracy rate, we will improve our machine learning algorithms based on our results.

II. Related Work

As we mentioned above, the class-imbalance problem and the choice of accuracy measure are significant for credit detection problem. Indeed, the work [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, such as balanced accuracy, Cohen's Kappa and Mathew's Correlation Coefficient. However, 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. In our project, we shall not only improve the existent ML algorithm, but also a combination of accuracy metrics for highly imbalanced dataset will be presented.

III. Our Solution

The results presented below are only the outcomes at this stage, which is supposed to be modified in final report.

A. Description of Dataset

Our dataset contains transactions made by European card-holders 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. It contains values from 28 "Principal Component Analysis (PCA)" transformation features, namely V_1 to V_{28}.

In addition, due to confidentiality issues, we are unable to obtain the original characteristics of the relevant data and more background information. Only two characteristic values which are 'Time' and 'Amount' have not been converted by PCA. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction amount, this feature can be used for exampledependant cost-sensitive learning, the characteristic value 'Class' is the response variable, and its value is 1 when fraud occurs, otherwise it is 0. Since There are no missing values in the data set, no missing values filling is needed.

Moreover, according to the **Fig. 1**, we may find out that the dataset is indeed highly imbalanced. In addition, the "time" feature (**Fig. 2**) looks very similar for both types of transactions. Normal transactions show regularity and have some certain periodic distribution, while fraudulent transactions are more evenly. On the other hand, from the perspective of amount, **Fig. 3** shows that most fraud transactions are less than $500 and most normal transactions are less than $10000.

Feature "Amount" is the transaction amount, and its value is not in the same order of magnitude as the anonymous variables V_1-V_{28}, in order to prevent the algorithm from taking special care of a single variable, it is necessary to reduce the order of magnitude of the Amount variable. The normalization method is used here. It is mapped to the standard normal distribution.

Feature "time" contains the seconds elapsed between each transaction and the first transaction in the dataset. It has nothing to do with whether each transaction is fraudulent, so this feature can be dropped. At the same time, 'Amount' is normalized to produce 'NormAmount', and the original feature "Amount" can be dropped.

B. Machine Learning Algorithms

(1) **Logistic regression**:

Since we are dealing with a binary classification, 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.

(2) **Random Forest**:

Even though this is a binary classification problem, Random forest, a multinomial classifier, is also a suitable candidate for our modeling. After introducing extra randomness, Random forest, compared with Logistic regression model, improves the prediction accuracy in the meanwhile the cost of computation is not significantly increased, whose prediction is relatively robust to an imbalanced dataset.

C. Implementation Details

To begin with, the data set is divided into a training set and a validation set (the ratio is 8:2), then we apply Logistic regression algorithm to then training set, which surprisingly shows that our algorithm does NOT converge, even after

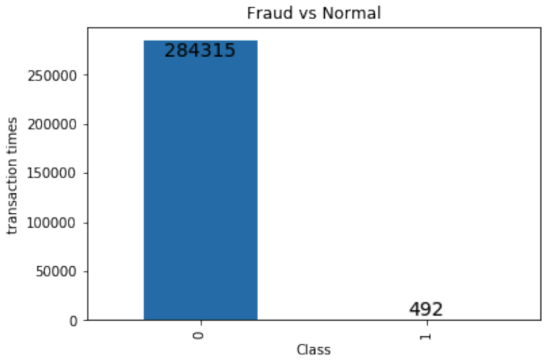

Fig. 1. The number of fraud and normal transaction

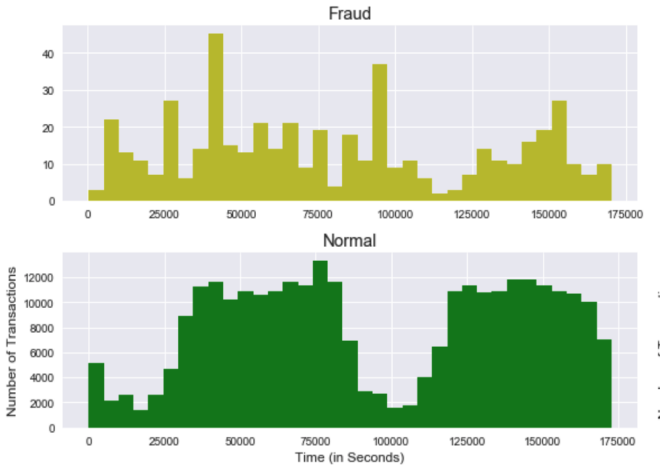

Fig. 2. Time compares across fraud and normal transactions

Fig. 3. Amount compares across fraud and normal transactions

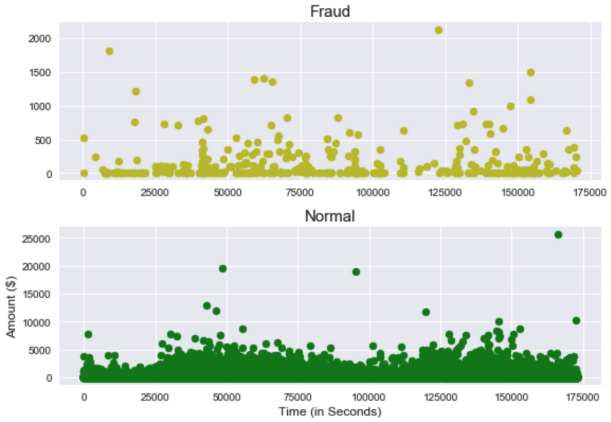

Fig. 4. Amount compares with Time for fraud and normal transactions

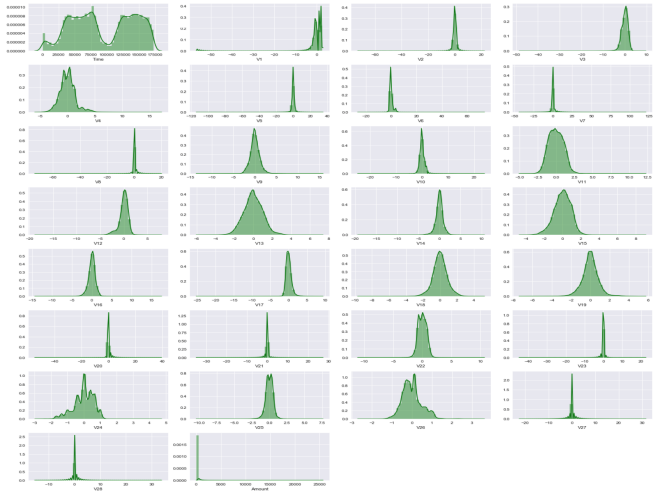

Fig. 5. feature V_1-V_{28} and Amount, Class, Time features distribution

the maximum iterations has been achieved. Therefore, we conclude that two groups of dataset need to be normalized respectively, then the dominance of features with relatively large absolute value will be weakened by normalization so that the speed of convergence of our model can be improved.

After normalizing the data set, Logistic regression modeling and random forest modeling are established on training set. By applying the models to the validation set, the prediction results and accuracy rates (based on confusion matrix) of the models are shown in the **Fig.6** and **Fig.7** and accuracy rate of Logistic Regression model & random forest model are 99.91% and 99.96% respectively.

However, the high accuracy rate does NOT indicate that our model performs very well on fraud detection. Indeed, it can be observed that the false negative blocks of two confusion matrices are both large, meaning that the ratio between positive cases labeled incorrectly and negative cases labeled correctly are significantly large, which is a direct consequence caused by class-imbalance. Moreover, confusion matrix shows that our prediction of majority class is significantly well, but the minority class (Fraud) is what model needs to detect. Therefore, the traditional accuracy rate CANNOT reflect the true performance of our model on fraud detection, which is overestimated.

More precisely (see **Fig.8**), See that the category of a small number of samples named as positive and the category of a large number of samples named as negative. The crosses represent positive samples, that is, a small number of samples. The squares represent negative samples, that is, most samples. Assume that both are generated by Gaussian distributions with the same variance. Because the number of samples of the minority class is small, in its Gaussian distribution, the minority samples have more high probability, and the small probability parts will only have a few minority samples. Considering that there are many samples in most classes, and there is sample crossover. Therefore, in the Gaussian distribution of minority samples, there will be many majority samples in the small probability part. Therefore, the predicted result will be more biased towards to the majority category.

Therefore, we decide to apply three more appropriate criteria of accuracy rate: (1) **Average Precision Score**, (2) **AUC-ROC curve** and (3) **Precision-Recall Curve**.

(1) Note that the confusion matrix provides with the value of TP, FP, FN, TN, the value of $recall = TP/(TP + FN)$ and the value of $precision = TP/(TP + FP)$. The main idea of Average Precision Score is to summary a precision-recall curve as the weighted mean of precision achieved at each threshold, that is, $\sum_n (recall_n - recall_{n-1}) * precision_n$.

(2) For the 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.

(3) 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.

As it can be seen, The Accuracy Rate and Confusion Matrix of two algorithms are presented, which provides us with a relatively positively high results. However, the results of Average Accuracy Rates, AUC-ROC curve and Precision-Recall curve have a significantly shrinkage, which appropriately reflect a basic fact that the false negative of logistic regression is higher than the false negative of random forest (see **Fig.9**, **Fig.10**, **Fig.11**). Therefore, these three criteria of accuracy rate are more appropriate for imbalance dataset.

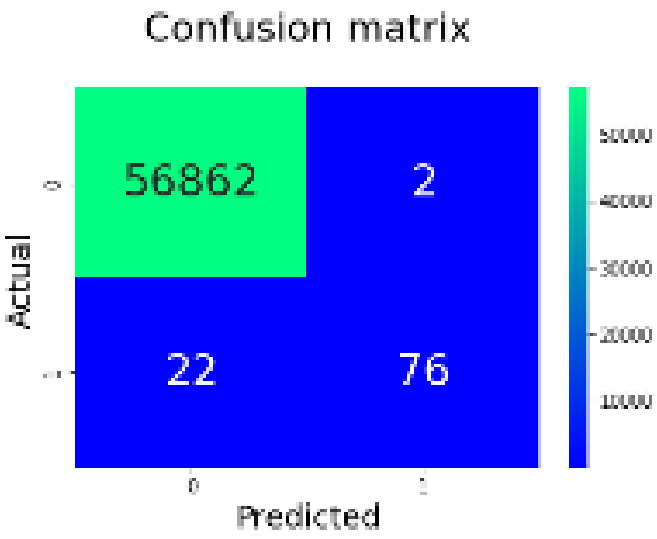

Fig. 6. Heatmap of confusion matrix (Logistic Model)

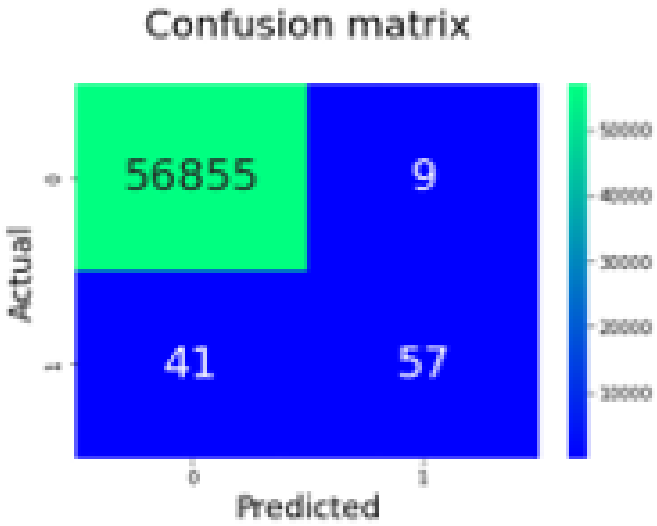

Fig. 7. Heatmap of confusion matrix(Random Forest)

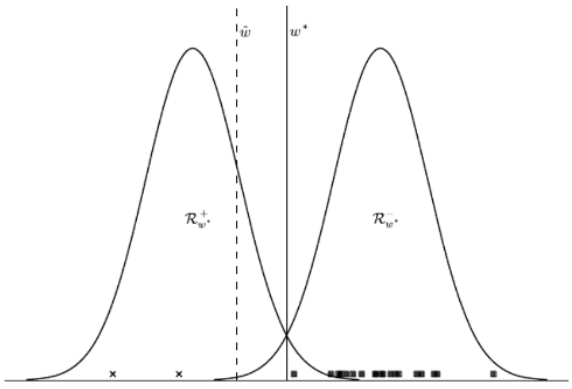

Fig. 8. Heatmap of confusion matrix(Random Forest)

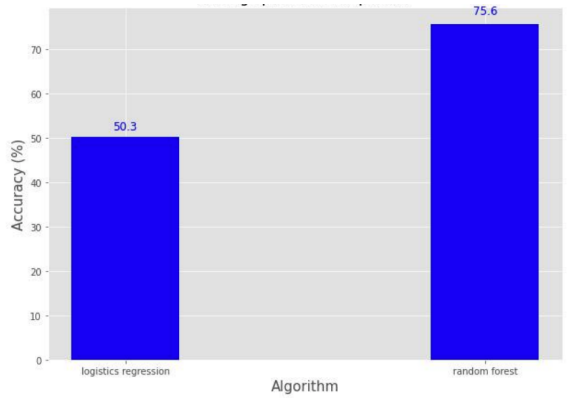

Fig. 9. Average accuracy results: Logistic Model v.s. Random Forest

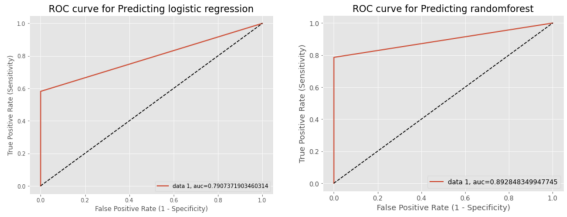

Fig. 10. AUC-ROC curve results: Logistic Model v.s. Random Forest

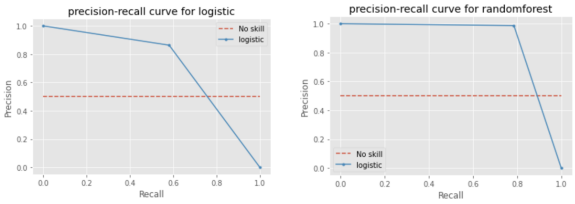

Fig. 11. Precision-Recall curve results: Logistic Model v.s. Random Forest

D. Techniques for Performance Improvement

Since we are dealing with imbalanced dataset, we need to apply some techniques to improve the performance of two algorithm.

(1) **Random Undersampling & Oversampling**:

The first choice is **Random Sampling Techniques**. The random sampling techniques consists of two kinds: random undersampling and random over-sampling. In principle, 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.

(2) **Synthetic Minority Oversampling Technique**:

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

The approach is effective because new synthetic examples from the minority class are created that are plausible, that is, are relatively close in feature space to existing examples from the minority class. The detailed algorithm is shown below:

(1) 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

(2) 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

(3) 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 $\delta \in (0, 1)$ is a random number.

(3) **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. In ADASYN, we consider a density distribution of minority class which thereby decides the number of synthetic samples to be generated for a particular point, whereas in SMOTE, there is a uniform weight for all minority points. The detailed algorithm is shown below:

(1) 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.

(2) 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.

(3) 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.

(4) For each minority class data example x, generate g_i synthetic data examples according to the following steps:
Do the Loop from 1 to g_i:

(i) Randomly choose one minority data example, $\bar{x}$, from the k nearest neighbors for data x_i.
(ii) Generate the synthetic data example: $\hat{x} = \delta|\bar{x} - x_i| + x_i$ where $|\bar{x} - x_i|$ is the difference vector in n-dimensional spaces, and $\delta \in (0, 1)$ is a random number.

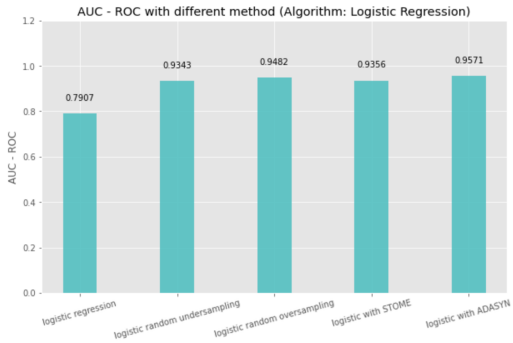

Fig. 12. Accuracy Rate of Different Techniques for Logistic Model under AUC-ROC

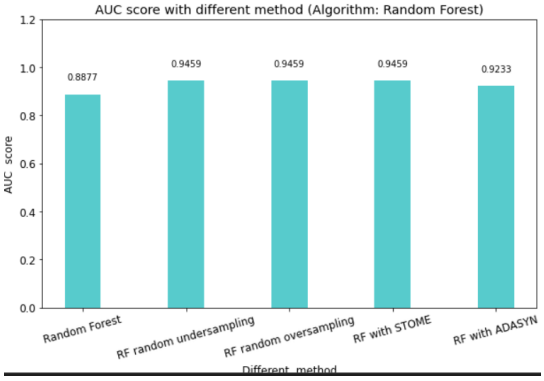

Fig. 13. Accuracy Rate of Different Techniques for Random Forest under AUC-ROC

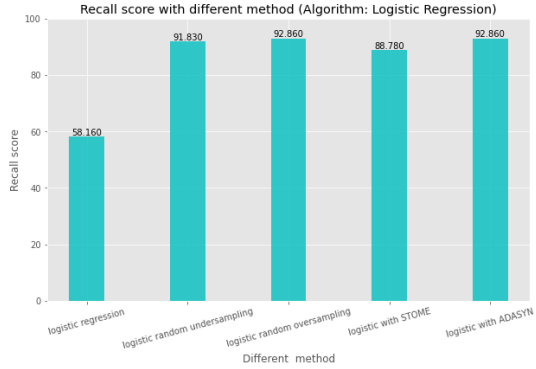

Fig. 14. Accuracy Rate of Different Techniques for Logistic Model under Recall Score

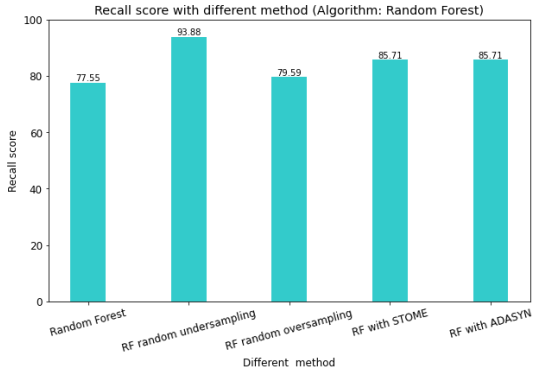

Fig. 15. Accuracy Rate of Different Techniques for Random Forest under Recall Score

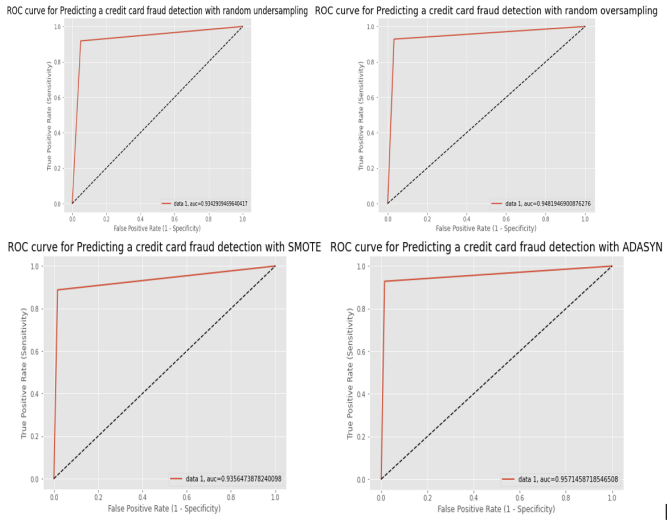

Fig. 16. AUC-ROC curve

IV. Comparison

After applying four techniques mentioned above, we could find out there exist significant improvements in Logistic model (see **Fig.12 & Fig.14**). After decreasing the influence of imbalance, our model can be trained to be less sensitive to the large difference of data imbalance. Also, the accuracy of random forest algorithm is slightly enhanced (see **Fig.13 & Fig.15**), which may be due to the good performance of original model, which leave less space for further improvement.

Moreover, we choose Recall Score and AUC-ROC curve, because they behave differently in the face of imbalanced data. When the data is imbalanced, the Recall Score is sensitive. As the ratio of positive and negative samples changes dramatically, the precision will change strongly. The AUC-ROC curve is insensitive to imbalance, and its curve can remain basically unchanged.

The consistency of AUC-ROC in the face of imbalanced data shows that it can measure the predictive ability of a model itself, and this predictive ability is independent of the positive and negative ratio of the sample. But this insensitive characteristic makes it more difficult to see how a model predicts when the sample proportion changes. Because recall score is sensitive to the sample ratio, it can see the effect of the classifier changing with the sample ratio, and the actual data is imbalanced. This helps to understand the actual effect and functional effectiveness of the classifier, and can also use this make improvements to the model.

In summary, in actual learning, we can use AUC-ROC to judge the excellence of the two classifiers, and then select the classifier, and then we can measure the ability of a classifier to classify imbalanced data based on the results of Recall Score. So we can continue this two-step procedure to optimize the model.

V. Future Research Directions

(1) Due to the limitation of the data source, our model does not combine with some knowledge in finance, such as involving some financial trust system into the model to reflect how likely each person are who they say they are. (2) Unsupervised learning techniques should be also involved into the model to discover the new fraud pattern, however, we need to consider the dynamic nature of transaction data. (3) Constructing new combination of accuracy rate measurement

VI. Conclusion

In this paper, we firstly recognize the credit fraud detection problem is a classification problem so that Logistic Model and Random Forest Model has been chosen as our preliminary model. Then, the confusion matrix and traditional accuracy score shows our preliminary model performs well on fraud detection, which is a misleading caused by imbalance data. Therefore, multiple accuracy valuation criteria: average precision score, AUC-ROC curve, precision-recall curve are the new options. After doing detailed analysis, the combination of AUC-ROC curve and recall score is the best match for our problem

Then, We apply four random sampling methods to decrease the effect of imbalance data: under-sampling, over-sampling, STOME, and ADASYN. Without random sampling, the Logistic Model preforms relatively worse than Random Forest. However, its performance enhances 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. But, as we can see, Logistic model with random sampling becomes a good classifier for the credit fraud detection (see **Fig.16**).

For the future research, we wish that the original data set can be used for the further improvement of our model. Since PCA techniques may throw away two much imformation.

References

[1] Nilson report. *issue 1164 November 2019*. https://nilsonreport.com/upload/content_promo/The_Nilson_Report_Issue_1164.pdf/

[2] Yvan Lucas and Johannes Jurgovsky. *Credit card fraud detection using machine learning: a survey*. arXiv preprint arXiv:2010.06479,2020.

[3] Aderemi O Adewumi and Andronicus A Akinyelu. *Survey of machine-learning and nature-inspired based credit card fraud detection techniques*. International Journal of System Assurance Engineering and Management, 8(2):937–953,2017.

[4] Josephine S Akosa *Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data*. https://support.sas.com/resources/papers/proceedings17/0942-2017.pdf

[5] Suraj Patil, Varsha Nemade, PiyushKumar Soni *Predictive Modelling For Credit Card Fraud Detection Using Data Analytics*. Procedia Computer Science 132 (2018) 385–395

[6] Ying Meng, Zhaohui Zhang, Wenqiang Liu, Ligong Chen, Qiuwen Liu, Lijun Yang, Pengwei Wang *A Novel Method Based on Entity Relationship for Online Transaction Fraud Detection*. ACM Turing Celebration Conference - China Chengdu China May 17 - 19, 2019

[7] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, Fellow, IEEE, and Gianluca Bontempi, Senior Member, IEEE *Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy*. IEEE Transactions on Neural Networks and Learning Systems (Volume: 29, Issue: 8, Aug. 2018)

[8] Roberto Saia, Salvatore Carta *Evaluating the benefits of using proactive transformed-domain-based techniques in fraud detection* Future Generation Computer Systems Volume 93, April 2019, Pages 18-32

[9] Hassan, Doaa. *The impact of false negative cost on the performance of cost sensitive learning based on Bayes minimum risk: a case study in detecting fraudulent transactions*. International Journal of Intelligent Systems and Applications; Hong Kong Vol. 9, Iss. 2, (Feb 2017): 18.

[10] Artikis, A., et al. *A prototype for credit card fraud management: industry paper* Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems, pp. 249–260 (2017)