Mobile money transaction Fraud Detection

Authors-

Ankit Nanavaty(*ananava@ncsu.edu*)Aditya Shah(*ahshah4@ncsu.edu*)
Vishwas S P(*vsomase@ncsu.edu*) Ashish Pawar(*akpawar@ncsu.edu*)

Dataset - https://www.kaggle.com/ntnu-testimon/paysim1

Introduction and Background

Detecting a fraudulent mobile money transaction is the focus of our work. As mobile transactions continue to increase, online fraud detection continues to become a bigger issue. Although fraud via smartphones is increasing at a faster pace than general PC/laptop based fraud, smartphones have the potential to become as secure a channel as the web through the use of advanced encryption and authentication technologies [6]. By paying close attention to red flags and suspicious activities, you can avoid merchant services fraud. According to the 2018 Global fraud report[1], it is evident that out of the digital market place consumers 91% of customers use smartphone out of which 88% use for personal banking and it has been noted that 72% cite fraud as a growing concern over the past twelve months and 63% report higher level of fraudulent losses over that same period.

One such activity is cited in the rise of Synthetic identities. Synthetic identities come from accounts not held by actual individuals, but by fabricated identities created to perpetuate fraud [4]. These identities are created by combining the credentials and information of a mixed set of individuals to create a completely new ID. Criminals use this kind of technique to commit frauds in the area of healthcare, utility services and taxes. The research in the area of combatting such kind of frauds, motivates us to find a robust system to detect fraudulent transactions.

Smartphones have been an easy target for fraudsters as it lacks the security level that other mobile devices have. Fraudsters know that it is generally easier to take over an account by phishing, spear phishing (targeting an individual) or Smishing (phishing via a mobile device), than to open a new account using a real or 'synthetic' identity, which is why the risk of account takeover is one of the most alarming trends in fraud[5].

Method

Approach:

As per the proposal, we have preprocessed the data, compared and applied the appropriate sampling technique and then applied the below given algorithms, along with tuning of the parameters.

1. The steps that were followed during the preprocessing were as follows:

- Checking the nature of the data:
  - Took 1/4th random sample of the data which approximately represented the actual dataset, the ratio of fraudulent to non-fraudulent transaction was 0.001 as of which we concluded the dataset to be highly imbalanced.
- Selecting the relevant transactions:
  - There are 5 types of transactions, CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER, of these types we checked which type contributed towards being fraudulent.
  - TRANSFER and CASH-OUT types turned out to contribute towards being fraudulent.

2. Sampling techniques comparison:

- Prediction with random sampling:
  - Data was split randomly such that 2/3rd data was used as training dataset and 1/3rd was used to test. When logistic regression was applied recall rate was found to be 0.7229
- Prediction with SMOTE (Synthetic Minority Over Sampling technique)
  - SMOTE is an oversampling method. It works by creating synthetic samples from the minor class instead of creating copies. The algorithm selects two or more similar instances (using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances. When logistic regression was applied recall rate was found to be 0.9723
- Prediction with NearMiss(Under Sampling technique)
  - NearMiss is an under-sampling technique. Instead of resampling the Minority class, using a distance, this will make the majority class equal to minority class. When logistic regression was applied on this recall rate was found to be as same as SMOTE i.e. 0.9723, but the accuracy of SMOTE is found to be better. Considering Recall rate, Precision rate, Accuracy rate we choose SMOTE Sampling technique.

3. Algorithms and parameter tuning-
   - **Logistic Regression-**
     - This approach is primarily used when the dependent variable is categorical. We divided the train and test data in the ratio of 0.77:0.33 and then applied logistic regression on them. The parameter tuned for logistic regression is 'C', which is the regularization parameter. Smaller the value of C, the better is the stronger regularization. The novelty of the approach is given by the ROC curve which is enlisted in the Rationale.

   - **KNN-**
     - In KNN classification, predictions are made for a new instance by searching through the entire training set for the K nearest neighbors and summarizing the output variable for those K instances.
     - For KNN, the parameter tuning of the k-values was carried out as per the cross_val_score() method with a cross-validation value of 10 and scoring done based on accuracy. The final k-value was chosen as the maximum value obtained for the above mentioned function.
     - The novelty of the approach was a trade-off between the accuracy and recall by choosing the appropriate k-value.

   - **Random Forest -**
     - Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Therefore, in Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).
     - The tuning parameters are n_estimators, max_depth and learning_rate.

   - **XG Boost -**
     - XGBoost is an implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

- The tuning parameters are n_estimators, max_depth and learning_rate. In this kind of dataset, it was observed that boosting algorithm like XGBoost proves to be of greater significance as compared to bagging algorithm like Random forest when tuned with the same parameters.

Rationale:

Sampling

We incorporated the above sampling techniques to measure the best results of sampling that could be achieved on the given imbalanced data. It was evident that the above-mentioned sampling techniques provided better results on highly imbalanced datasets compared to other sampling techniques. And out of the experimental results that we obtained, the SMOTE sampling technique is chosen as it provides the best combination of recall and accuracy among others.

Approach in choosing best classification algorithm

1. Logistic Regression- This was primarily chosen as our class was categorical in nature and the area under ROC curve observed was good enough (AUC), to consider this as the first approach for our dataset. However though the accuracy was high, the recall was low, hence we chose KNN over logistic regression.

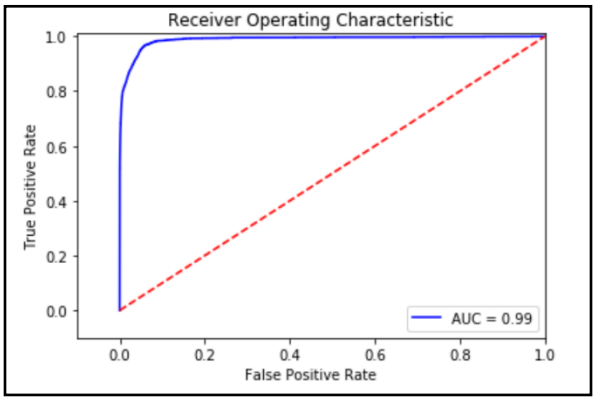

Figure 1: ROC curve for logistic regression

2. KNN- We observed that the knn gave good accuracy for k=1. But just to figure out whether our model was overfitting, we trained our model over different values of k, and observed that there was no significant change in the accuracies. And thus, we came to a conclusion that knn was not able to distinctly classify as a result of overfitting.

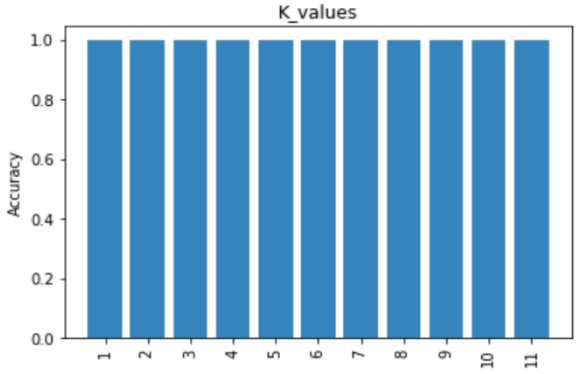

Figure 2: plot for k-values

3. Random Forest and XGBoost – To avoid the above issue and to get a higher recall score, we used the random forest and xgboost method as proposed in the research papers that we have referred. By hyper-tuning on the same parameters, it was found that, xgboost performed best in terms of accuracy and recall combined, which was the requirement of the task.

Experiment

Dataset:

The dataset is generated using the simulator called PaySim. PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The original logs were provided by a multinational company, who is the provider of the mobile financial service which is currently running in more than 14 countries all around the world. PaySim uses this data to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods.

There are 11 features for each transaction and one such transaction entry is shown below-
(1, CASH_OUT, 416001.33, C749981943, 0, 0, C667346055, 102, 9291619.62, 1, 0)

- step - maps a unit of time in the real world. In this case 1 step is 1 hour of time. Total steps 744 (30 days simulation).
- type – can take 5 values CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER. (CASH_OUT)
- amount - amount of the transaction in local currency. (416001.33)
- nameOrig - customer who started the transaction. (C749981943)
- oldbalanceOrg - initial balance before the transaction. (0)
- newbalanceOrig - new balance after the transaction. (0)
- nameDest - customer who is the recipient of the transaction. (C667346055)
- oldbalanceDest - initial balance recipient before the transaction. Note that there is no information for customers that start with M (Merchants). (102)
- newbalanceDest - new balance recipient after the transaction. Note that there is no information for customers that start with M (Merchants). (9291619.62)
- isFraud - This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customer's accounts and try to empty the funds by transferring to another account and then cashing out of the system. (1)
- isFlaggedFraud - The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction. (0)

Hypothesis:

The proposed approach in our case is the SMOTE sampling technique along with XGBoost classification for handling the imbalanced dataset.

1. Hypothesis that uses this technique a lot is the healthcare sector to check the occurrence of rare diseases. This approach generates new synthetic positive examples similar to the original data. For each subsampled partition, a different oversampled set of deleterious variants is added, resulting in balanced data sets.
2. Evidence of the kind of transactions are embedded in the dataset as the difference between fraudulent and genuine transactions which is obtained by examining their respective correlations in the heatmaps.
3. Applying these kinds of data for classification has proved to work best with boosting approach, with tuning of its depth and learning rate.

This approach appropriately works on such datasets, as in cases of highly imbalanced data, wherein the number of frauds are less, XGBoost proves to detect the frauds appropriately, with less number of true negative values resulting in a high recall score.

Experimental Design:

From our experiment, we have obtained the imbalanced data from Kaggle for the credit card fraud detection.

We used below 3 techniques for sampling the data-

- Basic train and test split.
- Using SMOTE for sampling. (Oversampling)
- Near Miss technique. (Under sampling)

Results for normal sampling-

Accuracy: 0.9972

Recall: 0.7229

Results for SMOTE sampling-

Accuracy for SMOTE: 0.8967

Recall for SMOTE: 0.9745

Results for Near-Miss sampling-

Accuracy for Near-Miss: 0.8667

Recall for Near-Miss: 0.97451

Proof of hypothesis 1-

So, we can see, that the combined results of accuracy and recall measures for the SMOTE sampling technique are much better and thus, as per the hypothesis this technique works best for sampling.

Proof of hypothesis 2 -

From the below heatmaps generated, we can see that the co-relations among the dependent variables allows us to know if the transactions are genuine or fraud. The red portions in the grid depict the best co-relations coefficients which can be used as the features for predicting transactions.

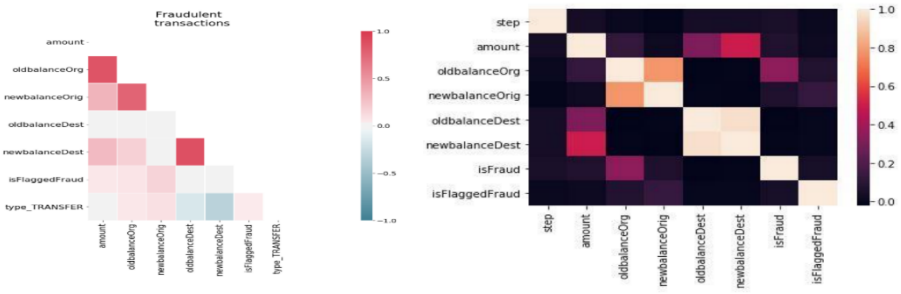

Figure 3: Heat Maps for correlation

Proof of hypothesis 3 -

From the below figures, we could see the performance of the chosen algorithms for imbalanced and the balanced datasets. We could observe that the xgboost would perform best for Balanced dataset on applying SMOTE.

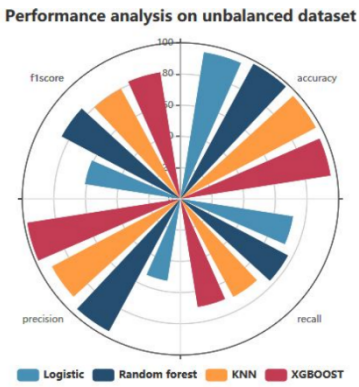

Figure 4: Unbalanced data analysis

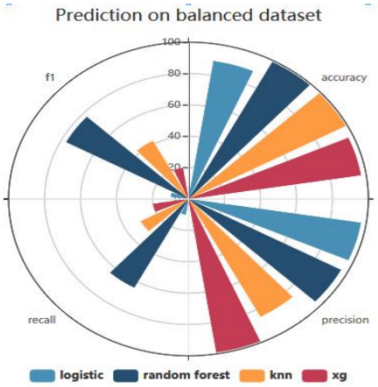

Figure 5: Balanced data analysis

Results

The below results are for the imbalanced data. The plot for the number of non-frauds is divided by 100 for visibility of both type of records in the plot.

* Number of Non-Frauds: **6354407** Number of Frauds: **8213** Percentage frauds: **0.1292**

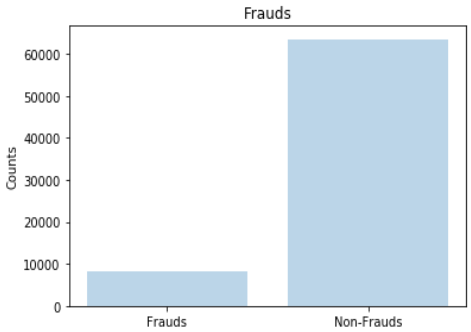

Figure 6: Fraud data distribution

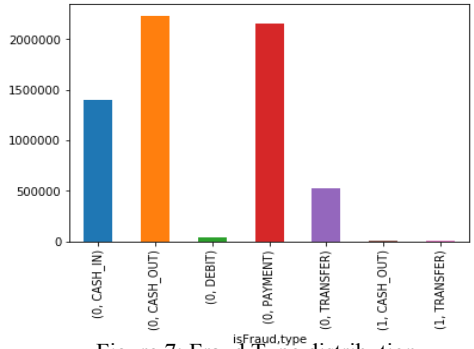

Figure 7: Fraud Type distribution

* Results for the confusion matrices produced

Model		Predicted	
		+	-
Actual	+	909766(TP)	1722(FN)
	-	761(FP)	1986(TN)

Figure 8: Confusion matrix for unbalanced data

Model		Predicted	
		+	-
Actual	+	818339(TP)	93149(FN)
	-	70(FP)	2677(TN)

Figure 9: Confusion matrix for balanced data

* Results for the algorithm performcances

	Accuracy	Precision	Recall	F1 score
Logistic	95.238	53.466	72.861	61.234
Random forest	99.924	96.972	77.138	85.928
KNN	99.897	92.866	71.892	80.892
Xgboost	99.902	98.632	70.986	82.557

Figure 10: Classification results for unbalanced data

	Accuracy	Precision	Recall	F1 score
Logistic	89.88	2.705	97.37	5.264
Random forest	99.83	64.67	80.37	77.391
KNN	99.39	30.042	64.52	42.45
Xgboost	97.27	19.38	99.38	17.84

Figure 10: Classification results for balanced data

Discussion:

- The above 2 tables represent the final result for both imbalanced and balanced dataset for all the four specified algorithms. In the first table, it can be seen that although the accuracies are high, the recall values are not satisfactory to give a clear distinction.
- As per our references, both the accuracies and recall must be given equal weightage for deciding the best classifier algorithm.
- This leads us to perform the classification by first balancing the data for which we have used SMOTE, the results for which are in the second table.
- In the second table, the recall value of the XGBoost algorithm is the best maintaining the high accuracy.
- The results are populated in the table after running the algorithms after performing hyper tuning on each of them.
- The hypothesis as per the referenced papers was that the xgboost algorithm gives out the best scores of accuracy and recall.
- Thus, as per the results we see that the best performance is given by xgboost, thus proving our hypothesis.

Conclusion

What have we learnt?

- As per the previous readings it can be conclude that recall and accuracy is the relevant parameter for balanced data and F1 score for imbalanced data.
- Smote leverages the performances of all the algorithms tested by balancing the data and provides a clear distinction which cannot be clearly concluded on the unbalanced data.
- Xgboost performs best in terms of recall (99.38) and accuracy(97.27) on balanced data.

Why does the result matter?

- As per a surge in online mobile transaction , it is the need of the hour to precisely detect fraudulent transactions.
- As most of the real world financial data is unbalanced smote proves to be a significant oversampling technique that can be used to balance the data for clear evaluation.

- Xgboost can lead to significant precise classification on unbalanced dataset in general when used with smote.

References

[1] The 2018 Global Fraud and Identity Report, Exploring the link between customer recognition, convenience, trust and Fraud risk- https://www.experian.com/assets/decisionanalytics/reports/global-fraud-report-2018.pdf

[2] E. A. Garcia and H. He, "Learning from Imbalanced Data," in IEEE Transactions on Knowledge & Data Engineering, vol. 21, no. , pp. 1263-1284, 2008.

[3] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas." Handling imbalanced datasets: A review, " in GESTS International Transactions on Computer Science and Engineering, Vol.30, 2006

[4] Synthetic identities getting real with customers, locate and minimize identity-based frauds with the right tools.

[5] Fighting Mobile Fraud Protecting Businesses and Consumers from Cybercrime- https://s3.amazonaws.com/content.iovation.com/white-papers/PDF/iovation-mobile-fraud-white-paper.pdf

[6] ONLINE PAYMENT FRAUD WHITEPAPER 2016-2020 https://www.experian.com/assets/decision-analytics/white-papers/juniper-research-onlinepayment-fraud-wp-2016.pdf

[7] Veni, C.V.. (2018). On the Classification of Imbalanced Data Sets. 10.13140/RG.2.2.14964.24961.

[8] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer, "SMOTE : Synthetic Minority Oversampling Technique", 2002.

Official code from - https://github.com/ankkit24/banking_transaction_fraud_detection