Fraud detection analysis on online transactions and credit cards

vijaylaxmi.nagurkar@sjsu.edu *monica.dommaraju@sjsu.edu* *srisruthi.chilukuri@sjsu.edu*

Abstract—As the financial world moves from traditional currency exchange to online transactions, credit card and online usage has become very popular and are widely in use now. With this increase in transaction volumes and the amount of money involved, fraudsters are finding numerous ways to exploit the system. Detecting these fraudulent transactions is challenging for reasons such as transactions data being heavily skewed and the volatility of the normal and fraudulent behavioral profiles. Features selection, sampling techniques and choosing the right algorithms play a crucial role in identifying these transactions. We intend to use naïve bayes, k-nearest neighbor and logistic regression-based approaches to find the better performing algorithm to detect the fraudulent transactions. Factors such as precision, accuracy, sensitivity, AUC are considered to evaluate the performance of the above-mentioned algorithms**.**

Index Terms—Credit cards, Machine learning, Machine learning algorithms, Cleaning, Adaptation models, Real-time systems, credit card frauds, fraud detection system, fraud detection, confidential disclosure agreement, real-time credit card fraud detection, skewed

1. INTRODUCTION

In 2018 for United States alone, there were 41 billion credit card transactions based on cards from four major networks – Visa, Mastercard, American Express, and Discover that accounted for about $3.8 trillion dollars [1]. Online retail sales for Feb 2019 has surpassed general merchandise stores [2]. These two indicate the potential for fraudulent activities in the online and the credit card transactions. To mitigate credit card and online frauds, e-commerce companies and banks have been implementing different fraud detection mechanisms and as a preventive maintenance taking actions such as blocking credit cards, blacklisting and placing a temporary hold on accounts etc. These mechanisms may help financial institutions and banks save costs that may incur by these fraudulent transactions. However, they are not adequate to completely eradicate these frauds as fraudsters are quickly coming up with new innovative ways. Identifying the threshold to tag the transaction as fraud is sensitive and require good predictive modelling algorithms. Incorrectly tagging the transaction as a fraud may result in losing customers loyalty and trust. For instance, consider a customer planning to book an air travel reservation from a country different than that issued a credit card. Though the transaction is genuine, credit card company may temporarily place a hold on the account until the customer authorize that transaction. This scenario may be considered reasonable for an infrequent flyer. However, certain frequent flyers may find this inconvenient to get assistance from customer service to unblock their credit cards. This is an example of treating the same parameter in different ways based on the customer types and their behavioral profile. There can be numerous different scenarios that can happen while performing online transactions too. Considering the volatility of usage patterns and behaviors, identifying fraudulent transactions is a difficult problem and is considered a suitable candidate to solve using data mining and machine learning techniques.

2. ABOUT THE DATASET

For this project, we are planning to work on two different datasets. The first one being the credit card transactions dataset that was obtained from Kaggle.It contains European credit card transactions for two days from September 2013 [4]. However, most of the features contained in this dataset have been vectorized by performing Principal Component Analysis(PCA),where the real attributes are hidden except time and amount.

The second one that is online fraud transactions dataset was obtained from Vesta corporation, a payment service company. This dataset was posted by IEEE and contains 590K transactions and associated user identities [3].

The major reason for us choosing the second dataset was to understand and perform various data pre-processing tasks like data cleaning, feature engineering, etc that weren't otherwise possible on the vectorised dataset.This dataset has numerous attributes, and at the same time some of these attributes have missing values, which is common in the real world.

2.1 Operations that have been performed on each of the datasets

Credit card transactions dataset

- Performed visualizations to understand how each of the features are distributed with respect to class label.
- Applied various kinds of sampling techniques like upsampling, downsampling, hybrid sampling to handle heavily biased data.

- Ran multiple machine learning algorithms to understand how each of them performed on our dataset.
- Analyzed the performance of each of the models using various performance metrics like classification report, confusion matrix, AUROC and AUPRC techniques.

Online Transactions dataset

- Performed exploratory data analysis
- Performed data preprocessing and Feature engineering
- Applied PCA as a part of dimensionality reduction
- Applied AutoML using TransmorgrifAI on the preprocessed dataset to find the best performing models.

2.2 Software libraries and tools used:

Core libraries:

Core libraries that we have used for Data wrangling, visualizations and building machine learning models are

- Jupyter
- Numpy
- Pandas
- Scikit-Learn
- Matplotlib
- Seaborn
- GraphViz
- Findspark
- SqlAlchemy
- mysqlclient

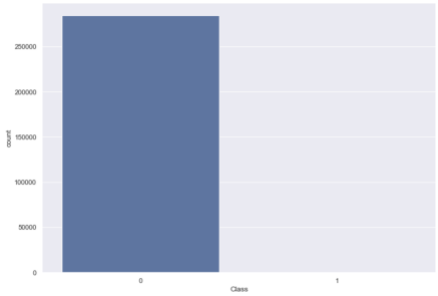
Tools and technologies used:

- Amazon RDS
- Elastic search
- Kibana
- Tableau
- TransmogrifAI AutoML
- MySqlWorkBench

3. HANDLING BIASED DATA

3.1. Sampling Techniques

The data source that we considered was highly skewed towards non-fraudulent transactions. Hence, we decided to use resampling techniques to balance the data.




```
Percentage of fraud counts in original dataset:0.17274856306200346
Percentage of fraud counts in the new data:50.0%
```

Essentially, we wanted to process that data to have 50-50 ratio. To achieve this process, we carried out various sampling techniques like Oversampling, Undersampling and hybrid sampling. Each of them works differently:

3.2. Oversampling:

This refers to adding the copies of underrepresented class (i.e Fraudulent Class) to the dataset. It can be carried out either by SMOTE (Synthetic Minority Over-Sampling Technique) or by random sampling

- SMOTE

  It is a technique that combines oversampling and undersampling; with a difference that the oversampling approach is not by replicating the minority class but by constructing new minority class data instance via an algorithm.

- Random Oversampling

  It is a technique to

3.2 Under Sampling:

Undersampling works by choosing samples from the majority class to match the size of the minority class. It can be carried out by using multiple techniques, out of which we have chosen Random Sampling and NearMiss Sampling.

- Random Undersampling

  Random undersampling works by just randomly selecting the samples of majority class until it has reached the size of minority class. We have used no replacement feature of sampling to avoid selecting the same sample multiple times.

- NearMiss Algorithm

  NearMiss algorithm working using the following principle. When two samples of different classes are very close to each other, the algorithm removes the samples of majority class. This it helps in the classification process by increasing the spaces between two classes

3.3. Hybrid Sampling:

Hybrid sampling takes the best of both undersampling and oversampling techniques. It balances the classes by removing the majority class samples and adding synthesized samples to the minority class. We can use many approaches like SMOTE, Random UpSampling to boost the minority class and NearMiss, Random Downsampling to reduce the Majority class. In our project we have used Random UpSampling and Random Downsampling simultaneously to achieve the balance in the dataset.

4. MACHINE LEARNING ALGORITHMS

We ran multiple machine learning algorithms on the balanced data. Some of the algorithms that we used were logistic regression, KNN, SVM, Decision Tree, Gaussian Naive Bayes' and Random Forest Classifier.

Multiple models which were run using various parameters using gridsearchCV. This method takes input in the form of a tuple comprising of "model" and "parameters". It later combines an estimator with a grid search to tune hyper-parameters and returns the best estimator by trying out every parameter in combination to output the model which performed its best using a certain parameter.

We have run multiple models on our undersampled data; here are our tabulated accuracy scores for each of the models used:

Algorithm	Mean cross validation score	Test Accuracy
Logistic Regression	0.9425	0.9773
K Nearest Neighbour	0.9377	0.9823
Random Forest Classifier	0.9362	0.9805
Support vector Machines	0.9368	0.9898
Decision Tree	0.9149	0.9333
Gaussian Naive Bayes	0.9225	0.9651

4.1 Model Training and Design Decisions

As the data is heavily biased towards non fraud data, models should be trained in such a way that it should be able to classify the fraud samples more accurately. In this project we have followed many multiple approaches such as

- We split the original dataset into Train and Test samples using stratification split.
- We held 20% of the dataset as hold out set, to finally test the model. This approach helped us to accurately measure the performance metrics.
- Used StratifiedKFold approach with K set to 5 for splitting the train and test samples.
- For each fold iteration we have run the GridSeachCV fit across multiple parameters to find the best estimate.
- We cross validated the best estimate that we acquired from GridSeachCV using the Sklearn cross_validate_score method. This helped us look at the mean accuracy scores instead of getting into the accuracy traps.
- We have selected the best model from K iterations by looking into the recall scores.
- Even though we have achieved accuracy scores more than 99% for most of the model runs, our goal was to choose the model with good recall scores as the target we want to achieve is to not miss the fraudulent transaction.
- We have also plotted ROC and PRC curves to also evaluate the precision scores.
- We ran the best model selected from above to run predictions on our hold out dataset.

4.3 Performance Metrics

We also performed K-folds and cross-validation. To capture more fraudulent transactions, we derived precision-recall score, accuracy. As we need to minimize the false negatives, we were interested in the recall score to capture this. When we try to increase recall, it tends to decrease precision. However, in our case, if we predict that a transaction is fraudulent and turns out not to be, is not a massive problem compared to the opposite. KFold was used to identify optimal configuration for Logistics classifier. We plotted the ROC curve to check if the model is working correctly. We tried to get higher AUC as possible.

- *Cross validation score*

It is a performance evaluation technique which splits the data into folds and ensures that each fold is being used for testing at least at one point.
In our project, we have used cross-validation from sklearn to make sure our model is not over-fitted.

- *AUC-ROC curve*

It stands for Receiver Operating Characteristics and tells how well the model has distinguished between the classes. Higher the value of AUC, better is the model at distinguishing between i.e. between fraud and non-fraud cases.

- *Confusion Matrix*

Confusion matrix for Credit card transactions database

Fraudulent transaction identified as fraud (True Positive)	Non-fraudulent transaction identified as fraud (False positive)
Fraudulent transaction identified as not-fraud (False negative)	Non-Fraudulent transaction identified as not-fraud (True negative)

In our project, as the data is heavily skewed towards the non fraud samples, accuracy is not the valid metric to consider. We aim to focus more on detecting the frauds sample which are treated as non fraud.

So, False negatives are more important to detect in this scenario. Similarly, the false negative cases are also equally important to detect because it is also undesirable to detect a genuine transaction as a fraudulent one. This degrades the customers trust and bankability on the financial institutions.

We consider two important metrics Precision and Recall understanding how the accuracy of the classification model has to be defined.

1. Precision= TP/(TP+FP); in an ideal case, the false positives are 0 i.e. no genuine transactions are incorrectly identifying as fraudulent transactions.
2. Recall = TP/(TP+FN); in an ideal case, the false negatives are 0 i.e. every fraudulent transaction is correctly identified as fraud.

For our use case, it is more important to observe the false negatives to detect every fraudulent transaction, not missing out any.
However, False negatives are also to be given importance as they may mislead financial institutions and annoy customers.

Hence, it is understood that both precision and recall are important and the optimal threshold value to differentiate a fraud from non-fraud can be found out using precision-recall curve; defined using F-measure.

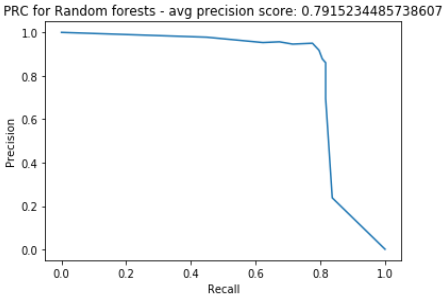
F-measure = 2* precision*recall/(Precision+recall)

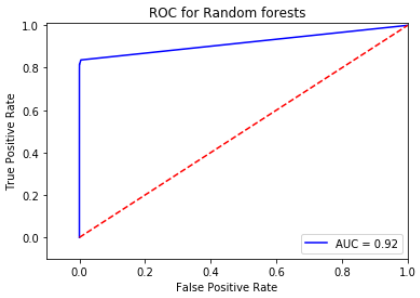
- Classification report: This report more precisely gives the count of classification metrics like precision, recall and f1-score on a per-class basis.
-

```
Accuracy score for the real test set:
0.9995435553526912
confusion matrix for the real test set:
[[56860     4]
[   22    76]]
Classification report for the real test set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.95      0.78      0.85        98

    micro avg       1.00      1.00      1.00     56962
    macro avg       0.97      0.89      0.93     56962
 weighted avg       1.00      1.00      1.00     56962
```



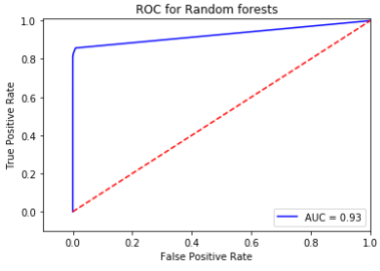


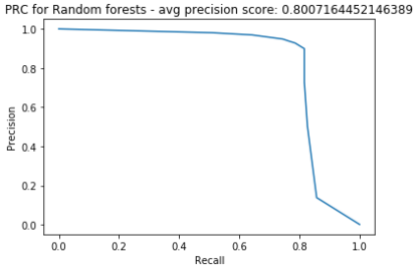
- Random Forest classifier algorithm generated best results when Hybrid sampling was carried out:

```
Accuracy score for the real test set:
0.9995259997893332
confusion matrix for the real test set:
[[56858     6]
[   21    77]]
Classification report for the real test set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.93      0.79      0.85        98

    micro avg       1.00      1.00      1.00     56962
    macro avg       0.96      0.89      0.93     56962
 weighted avg       1.00      1.00      1.00     56962
```



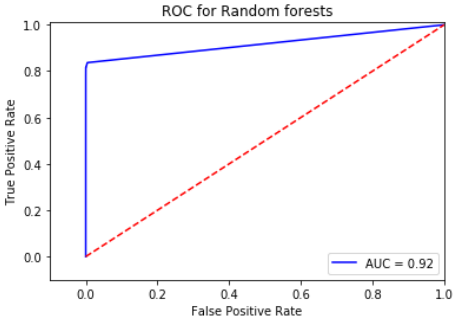


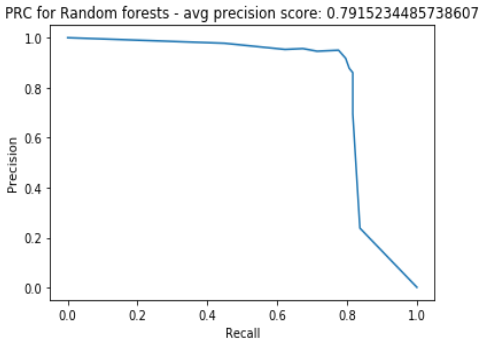
- Random Forest classifier algorithm generated best results when Under sampling using Near miss algorithm was applied:

```
Accuracy score for the real test set:
0.9861662160738738
confusion matrix for the real test set:
[[56090   774]
[   14    84]]
Classification report for the real test set:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     56864
           1       0.10      0.86      0.18        98

    micro avg       0.99      0.99      0.99     56962
    macro avg       0.55      0.92      0.58     56962
 weighted avg       1.00      0.99      0.99     56962
```



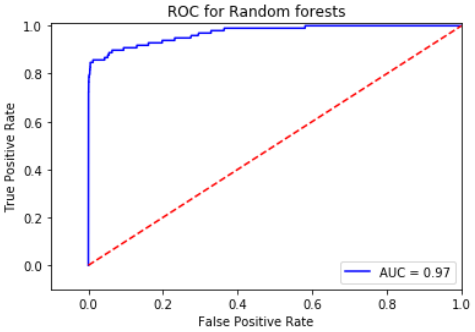


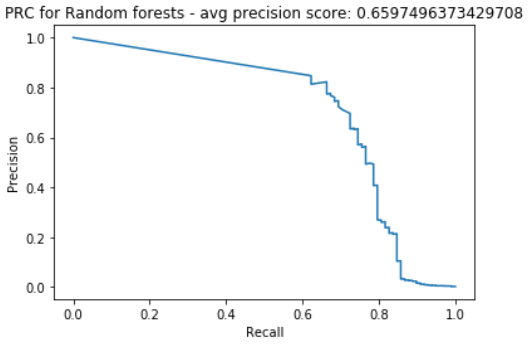
- Random Forest classifier algorithm generated best results when Under sampling using Near miss algorithm was applied:

```
Accuracy score for the real test set:
0.9632562058916471
confusion matrix for the real test set:
[[54785  2079]
[   14    84]]
Classification report for the real test set:
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     56864
           1       0.04      0.86      0.07        98

    micro avg       0.96      0.96      0.96     56962
    macro avg       0.52      0.91      0.53     56962
 weighted avg       1.00      0.96      0.98     56962
```



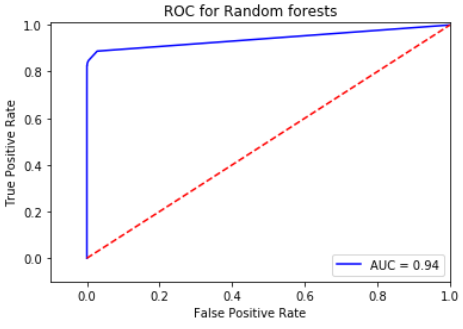


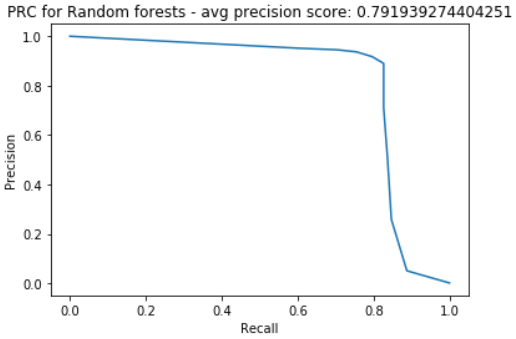
- Random Forest classifier algorithm generated best results when SMOTE was applied:

```
Accuracy score for the real test set:
0.9995259997893332
confusion matrix for the real test set:
[[56854    10]
[   17    81]]
Classification report for the real test set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.89      0.83      0.86        98

    micro avg       1.00      1.00      1.00     56962
    macro avg       0.94      0.91      0.93     56962
 weighted avg       1.00      1.00      1.00     56962
```





- Inferences:

For each of the oversampling, undersampling and hybrid sampling techniques that were carried out, random forest classifier has shown least values for type 2 and type 1 errors i.e. the number of False Negatives and False Positives have been the least. As Recall and precision are more crucial to us, we are looking for a maximum PRC score; which is 0.79 for all of these sampling techniques gives us the best way to train our models so as to detect fraud more efficiently.

5. Exploratory Data Analysis: Online Fraud Transactions Dataset

5.1 Insights from visualizations

1. We wanted to analyze what the average amount was in most of the fraud cases but as the attribute "amount" is a continuous variable; we had to discretize them into 20 equal buckets using qcut. From the generated count plot, we were able to analyze that the amount ranging from 75 to 100 had more fraudulent transactions.
2. Performed a count plot on "Device Type" and deduced that more fraud transactions have happened on credit cards as compared to the debit cards.
3. We could infer from some other visualizations that the fraud transactions were committed more via mobile rather than Desktop.

5.2 Data preprocessing

- Our dataset contains four csv files: Train and Test Identity, Train and Test Transaction.
- Initially, we first combined our csv files and merged them into training and test datasets based on their Transaction ID which is a common attribute in Identity and Transaction datasets.
- Train dataset contains 590540 rows and 434 columns and in Test data we have 506691 rows and 433 columns in which the class label wasn't provided.
- Our dataset contains 96.5% non fraud cases and 3.5% of fraud transactions which means we have a highly imbalanced data.
- We had 413 columns with null values, and we had to remove those which had null values greater than 90%; the same strategy applied to the columns which had homogenous values.
- While removing both these columns, the ones that were present in common in both train and data sets only had to be removed.
- This reduced our attributes list from 434 to 364.

5.3 Feature Engineering

- We tried to extract new features using categorical attributes "card#" using mean and standard deviation; using groupby method.
- Filled the missing values in the categorical columns with Nan.
- Performed Label Encoding on the categorical columns to change them to numerical.
- Filled the rest of the missing values in numerical attributes with median as mean is sensitive to outliers.

6. Conclusion

We have looked into various sampling techniques and data pre-processing strategies that we could use to classify a fraudulent transaction from a non-fraudulent one. Having learnt the importance of various performance metrics, we were able to analyse how each use case in machine learning demands a different level of understanding and analysis of those metrics. We have studied scenarios of when Type 2 errors become critical and when the Type 1 errors become so and concluded that it's not just the train and test accuracy scores that define the goodness of the model but depending on the situation, other performance metrics too can fall in picture.
In an attempt to understand how each of the models work on highly skewed data, we believe we were able to draw meaningful insights about the general behaviour of a fraudulent transaction.

7. Learnings

- We have analyzed about how each of the different sampling techniques work to handle the highly skewed data.
- Learnt about the parameter tuning to get best estimator on gridsearch using GridSearchCV
- Out of all the sampling techniques that were performed, SMOTE algorithm has given us the best results when used with Random Forest Algorithm.
- Hands-on experience on how autoML works
- New strategies like Label encoding, binning during preprocessing and data analysis
- Big data tools like spark, used by transmorgrifAI

8. REFERENCES

1. https://www.creditcards.com/credit-card-news/market-share-statistics.php
2. https://www.cnbc.com/2019/04/02/online-shopping-officially-overtakes-brick-and-mortar-retail-for-the-first-time-ever.html
3. https://www.kaggle.com/c/ieee-fraud-detection/data
4. https://www.kaggle.com/mlg-ulb/creditcardfraud
5. J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, 2017, pp. 1-9. doi:10.1109/I
6. M. I. Alowais and L. Soon, "Credit Card Fraud Detection: Personalized or Aggregated Model," 2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, Vancouver, BC, 2012, pp. 114-119. doi: 10.1109/MUSIC.2012.27 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6305834&isnumber=63058 04
7. SMOTE: Synthetic Minority Over-sampling Technique https://doi.org/10.1613/jair.953