Stat 222: Fraud Detection Final Write-up

Daniel Huang

May 13, 2022

1 Problem Description and Introduction

Since the development of the internet, more and more transactions are processed through online portals without a physical storefront. The challenge posed by online transactions is especially evident with credit card fraud. Everyday, there exist customers who are plagued by fraudulent activities on their credit cards or debit cards. These illicit activities are nuisance not only for the customers, but also for the vendors. Whenever a customer files a complaint (this usually involves requesting a chargeback) with the credit card-issuing bank, the vendors have to pay a fee to their own acquiring banks, regardless of whether the complaint was eventually successful or not. In addition, if the vendors have already mailed out the merchandise, it is sometimes impossible for them to get the items back, which implies further loss. Building a classifier that accurately picks up the fraudulent transactions thus becomes more and more important. Moreover, understanding which predictors/features are more important than others is also integral to help curb future fraud. Additionally, we focus on simpler machine learning models and algorithms in order to create interpretable models and focus on business implications rather than purely predictive power. This focus is discussed more in depth later.

2 Previous Work

Credit card fraud has been a rising topic in recent years, and there has already been a significant amount of work done in the field. Currently, there are many papers utilizing machine learning models and algorithms to detect fraud. However, many of these papers [1] [2] [3] use a PCA transformed dataset, thus not allowing for interpretation of the model coefficients or parameters. We were only able to find one paper using non-PCA transformed data [4], however the focus is heavily on model performance rather than interpretability. Similarly, due to the nature of the Kaggle competition, most Kagglers only focused on model performance and used many complex machine learning algorithms and ensemble learning in order to get the best results.

3 Dataset

The data that we will be investigating comes from a Kaggle competition hosted by Vesta Corporation. The data contains information about credit and debit card transactions as well as the customers making the transactions. The company has guaranteed more than billions of dollars of transaction, and they provide the dataset in an effort to improve their ability to detect illicit transactions.

The data comes in the form of 2 tables - an identity and a transaction table. The identity table contains information about the users and has 144233 rows with 41 variables. The transaction table contains information about individual transactions and has 590540 rows with 394 variables. These tables can be joined on the common column called “TransactionID”.

Due to privacy concerns regarding the data, many of the variables are masked, so we do not have knowledge about their values. There are a few that we can identify, based on their values, but the vast majority have names such as “V1” or “C10”, and are thus difficult to identify.

One other issue with the data is missing values. After joining the identity table to the transaction table we are left with only the 144233 rows that have identity information, making it difficult to utilize. On top of this, both tables have many missing values, with about 40% of the transaction table and 35% of the identity table containing missing values.

Having the two tables available to us presents us with a choice: Either work with the full transaction table (as the majority of Kagglers do) or somehow try to incorporate identity information. We attempt to compare the performance of our models on both sets, the transaction only and the transactions with identity. We will refer to these as the transaction and merged datasets henceforth.

4 Exploratory Data Analysis, Data Pre-processing, and Feature Engineering

4.1 EDA

To understand the dataset, we perform some EDA. First we plot the number of credit card companies (**card4** in the transaction table) separately for the legitimate and fraudulent group,

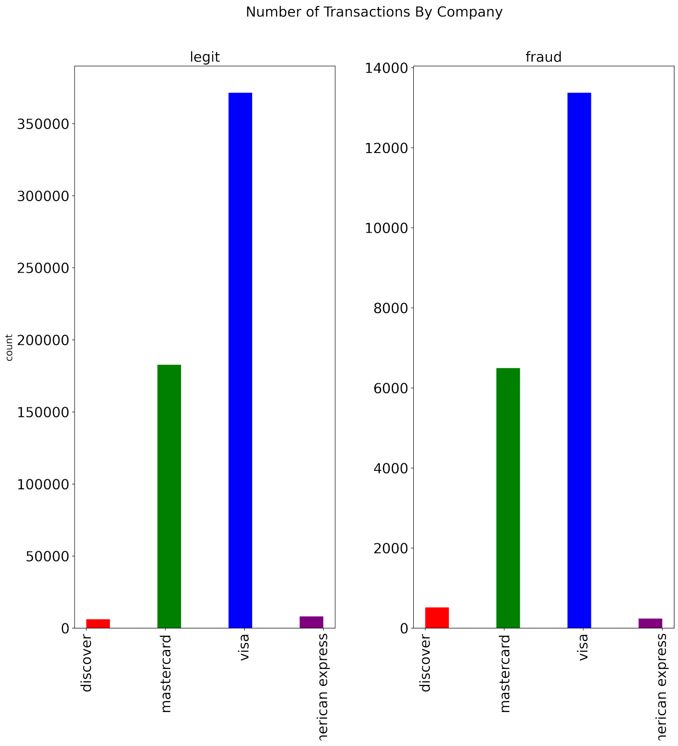

Figure 1: Credit card company distribution across legitimate and fraudulent transactions.

Figure 1 shows that across both legitimate and fraudulent transactions, Visa is the most frequently appeared company, and overall the distributions across both groups are really similar. This indicates that the credit card company might not be a good predictor for predicting whether a transaction is fraud or not. We see this behavior across many other variables as well, which leads us to believe that it will require some combination of all of our predictors in order to accurately detect fraud.

One other variable we investigate is whether the distribution of product codes associated with each transaction is different between legitimate and fraudulent transactions.

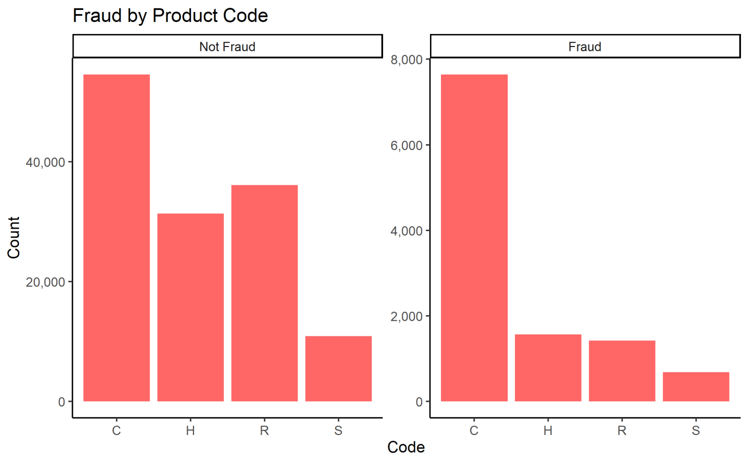

Figure 2: Product code counts for Fraudulent and Legitimate group in the transaction table.

Figure 2 shows that product type C, appears much more often in the fraudulent group, as compared to the legitimate group. This implies the product code might be useful in predicting whether a transaction is fraudulent or not. We ultimately see this show up later on.

4.2 Data Pre-processing

In light of the many missing values, we start by dropping columns where more than 70 percent of the values are missing. We then use missing imputation on the remaining columns, filling the remaining missing categorical columns with their individual modes and the missing numerical columns with their individual medians.

To avoid overly increase the dimensions of the features due to high cardinality of some of the categorical features, we summarize the information based on grouping: for purchaser email domains of the transactions, we reduce them to their suffix, and we do the same for recipients. However, this method does not work for other categorical variables such as credit card type, which has thousands of categories.

4.3 Feature Engineering

Taking inspiration from some of the top performers on Kaggle, we engineer some features in order to try and improve our model performance. Some of the variables are categorical, such as Billing Zip Code and Billing Country. However, there are so many categories that it does not make sense to simply use one-hot encoding to deal with them. In order to not lose the information, we calculate the TransactionAmt divided by the mean of these variables, grouped by category. Thus, we are able to retain the information from these predictors without having to deal with a huge amount of features. We engineer this feature for Billing Zip Code, Billing Country, Card type, and a few other key variables as well. While some Kagglers created more complex features, we decided not to include those as it was much more difficult to interpret the meaning of those variables as compared to these simple aggregations.

5 Model Evaluaton

The main metric we are using is the area under the ROC curve (AUC); this is a better metric than accuracy because a naive classifier that classifies every transaction as legitimate can easily attain an accuracy of more than 90 percent, but that is not what we want. A random classifier achieves 0.5 AUC, and the best model we find on Kaggle achieves 0.94; we want to compare our models with these numbers.

In addition, we also compute the Recall score and the Precision score; the first score refers to the proportion of true positives (true fraud) that actually gets identified by the classifier, and the second score refers to to the proportion of true positives within the group that the classifier identifies as fraud. We care more about the Recall in this problem, since we believe that false negatives are more costly than false positives (it is worse to label a fraudulent transaction as legitimate rather than label a legitimate transaction as fraudulent).

Before fitting out models, we leave out 20% of the data to use as a test set in order to evaluate our results. By using our own test set and not the one on Kaggle, we are able to see the number of false positives and false negatives rather than simply the AUC score of our classifier. We use the same train and test set for our Logistic and Random Forest classifiers.

6 Models

Because our focus is on utilizing simpler machine learning models as opposed to more complex ”black box” learning algorithms, we only train logistic regression and random forest models on both transaction and merged data. This allows us to compare the performance of these simpler algorithms across the transactions with and without identity information.

6.1 Logistic Regression

We first consider the simple logistic model,

$$P(Y_i = 1|X_i) = \frac{1}{1 + e^{-X_i \cdot w}}, \tag{1}$$

where $Y_i = 1$ indicates that a transaction is fraud, and X_i is the corresponding feature vector with intercept added.

Before fitting to the model, we apply the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset due to the presence of many more legitimate transactions than fraudulent ones in our dataset. This leaves with a 50/50 split between the classes.

In addition, we cut down the number of features (original 200+) to only about 60 using feature selection. The method we use is logistic regression with L2 regularization, where we remove coefficients that are smaller than the mean value. As mentioned before, because we are centering and normalizing each of the covariates, the magnitude of the model coefficients now correspond to the strength of that predictor in the model.

The AUC score for logistic regression with just the transaction table is 0.82, and the AUC score with the combined dataset is 0.90, which is reasonable because we have more information with the identity table. These results can be found in 1.

6.2 Random Forest

We next fit the random forest model, again with both the merged and the transaction only dataset. The AUC score for random forest with just the transaction table is 0.909 and the

AUC score for random forest with the merged table is 0.95, which is close to the highest score seen on Kaggle. Again, these can be found in 1.

Since we have already done feature selection using logistic regression, here we are really just training a random forest model on the remaining predictors. Using grid search, we also find the best hyperparameters for the model and we again validate using validation on the test set.

7 Main Results

7.1 Model Comparison

Table 1: Main Result

Model	Precision	Recall	AUC
Logistic: Transaction	0.101	0.713	0.82
Logistic: Merged	0.368	0.748	0.898
Random Forest: Transaction	0.318	0.675	0.909
Random Forest: Merged	0.493	0.852	0.95

Comparison of Logistic and Random Forest models over both Transaction only and Merged (Transaction with identity) datasets. Precision is (TP)/(TP + FP) and Recall is (TP)/(TP + FN). Both are calculated at the standard threshold of 0.5.

The results from 1 show that the models trained on the merged dataset perform better in all categories (precision, recall and AUC) when compared to the models trained on purely the transaction data. Specifically when looking at the best model for both datasets, the random forest, the merged dataset gives us an AUC of 0.95 as compared to 0.909 from the transaction only dataset. Following our intuition, this increase in performance is due to having more information about each transaction now, specifically identity information. While again we are unable to specifically identify these variables due to them being masked, these results indicate that having this identity information is improving our ability to detect fraud.

7.2 Variable Importance

We are also interested in which variables are the best at predicting fraud, and if these are common across the learning algorithms.

For logistic regression, because we centered and normalized the features before we fit the model, each feature is roughly on the same scale. Using this fact, the magnitude of each coefficient corresponds to how important that variable is.

Figure 3 shows the top 10 coefficients. Negative signs indicate that the increase in a feature decreases the probability of fraud, keeping all other constant, vice versa. Red error bars are estimated with bootstrapping. By nature of the dataset, most variables are masked to protect the privacy of the customers. But we know that $C's$ are important features that the company should invest in collecting information from (these are information related to counts:e.g., number of addresses associated with the credit card). Surprisingly, the card company appears in the top 10 (specifically, being in the discover card group increases the probability of fraud), which is counter-intuitive because we conclude that company might not be that important in our EDA.

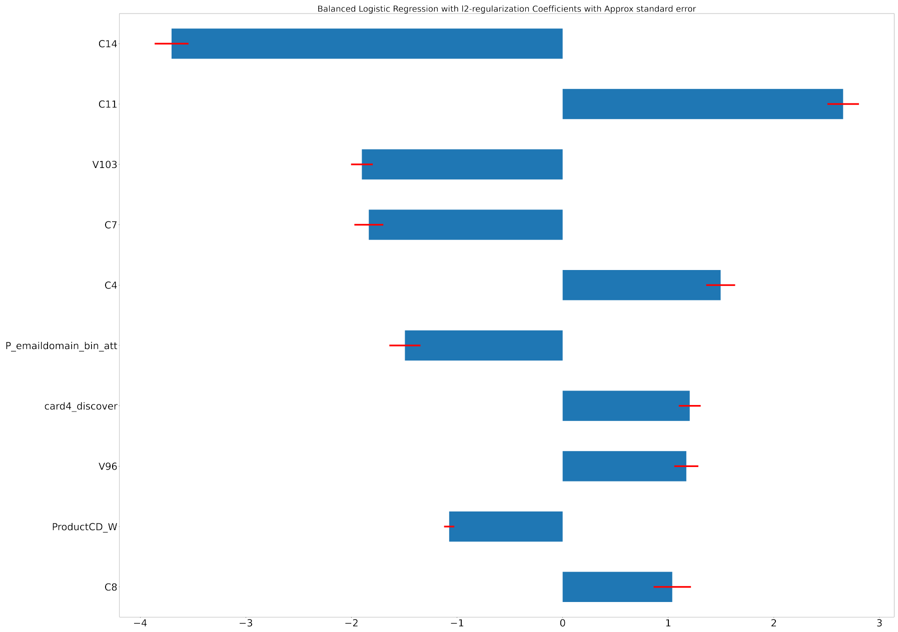

Figure 3: Top 10 Logistic regression Coefficient for Transaction Only Dataset

Now looking at the random forest model, the measure of variable importance in this case is the mean decrease in impurity contribution from each feature. Features with high feature importance should have a high mean decrease in impurity. Figure 4 shows the feature importance for the random forest using just the transaction table. The red error bars are for one standard error. Again we see the C's are the important features. The transaction amount appears as important features, a possible explanation is that mean decrease impurity favors features that have high cardinality, i.e., a lot of distinct values, which is common for a numerical columns like transaction amount. Similar analysis applied to the merged dataset shows that some of information unique to the identity table appears in the top features, which indicate that the information in the identity table can definitely help us identity fraud.

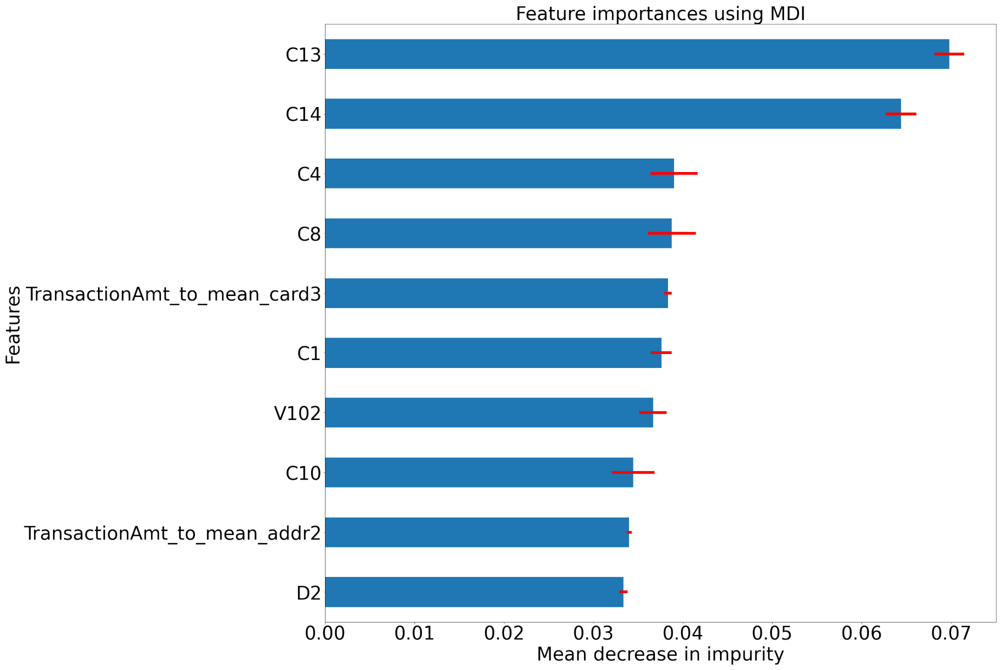

Figure 4: Mean Decrease in Impurity for the top 10 features for random forest using transaction only dataset

7.3 Caveats

It is important to note the assumptions and limitations on our work. After merging the identity table with the transaction table, it is possible that the distributions of the data are no longer the same. Thus, it may be inappropriate to compare these models. However, due to the limited scope of the data, we believe that this assumption can be appropriate in this scenario. Furthermore, we assume that the model will perform similarly on future data that we have not yet seen. Due to the timeframe of the training data, it is possible that our model will not perform well on future data, however we believe the insights gained from our analysis can be beneficial regardless.

8 Business Implications

8.1 Cost Analysis of Identity Information

We have seen that including identity information improves the performance of our classifier, improving our AUC by nearly 8% for logistic regression, and by 4% for the random forest. In fact, with just a simple random forest on the rebalanced data, we are able to achieve AUC scores comparable to the best ones in the Kaggle competition. However, how much is this identity information actually helping? In other words, how can we quantify the dollar value of this identity information? We use the following equation in order to come up with a 95% confidence interval on the average amount saved per transaction by having identity information:

$$
\begin{aligned}
\text{(Amt Saved Per Transaction)} &= \text{(Fraud Rate)} \times \text{(Recall of Transaction Only Model}- \\
&\qquad\qquad\qquad\quad \text{Recall of Merged Model)}\times \\
&\quad \text{(95\% Confidence Interval on Average Transaction Amount)} \\
&= 3.5\% \times (0.852 - 0.675) \times [74.8, 78.8] \\
&= [0.485, 0.488]
\end{aligned}
$$

This confidence interval tells us that we are 95% confident that on average, having identity information for a given transaction will save the company between $[0.485, 0.488]$, or around \$0.49 for that transaction. This means that if it costs the company less than \$0.49 on average to obtain identity information for a given transaction, they should do so since it will save them money in the long term.

8.2 Model Implementation

The models we have seen in this analysis use AUC as the primary metric for model evaluation. However, when actually implementing the model, decisions need to be made around which threshold to employ for the model. Because AUC considers the total area under the ROC curve, when actually predicting fraud, we must select some threshold. The default value is obviously 0.5, however due to the trade-off between false positives and false negatives, companies may choose to adjust this value in order to obtain desired values of precision and recall.

Additionally, companies should factor in the administrative costs associated with each of these decisions. For example, predicting any positive, false or not, comes with some administrative cost because a transaction must be canceled, thereby incurring a chargeback or other type of fee. In addition to these tangible costs, companies also must consider the potential risk of losing customers if given too many false positives. It is imperative that companies take these factors into account when implementing these fraud detection models.

9 Conclusion

Credit card fraud has quickly become one of the top areas where machine learning is being utilized. However, most of the prior work done has been focused solely on predictive power, and fails to create interpretable or insightful models. Our work in this paper shows that by utilizing identity information associated with transactions, we are able to use simple machine learning algorithms such as logistic regression and random forest to achieve comparable results to the best algorithms only using the transaction table. Furthermore, we quantify this improvement in performance, showing that companies can potentially save up to $0.49 per transaction if they invest in obtaining identity information for that transaction.

References

[1] Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. Credit card fraud detection - machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5, 2019.

[2] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics (ICCNI)*, pages 1–9. IEEE, 2017.

[3] Fayaz Itoo, Satwinder Singh, et al. Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, 13(4):1503–1511, 2021.

[4] Anuruddha Thennakoon, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga, and Nuwan Kuruwitaarachchi. Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 488–493. IEEE, 2019.