Detecting a Niche of Fraudsters

Sharon Colson, Breanna Moore, Gavin Mcduffee, Seth Tompson

1.
Problem Statement and Background

The problem is to find an optimal way to process and analyze data from retailer partners in order to detect fraud cases, so that fraudsters can be found and refused service in the future. The dataset is a list of financial purchases from retailers. The list is filled with categorical and numerical variables such as the item purchased, cost of the item, code for the item, number purchased, and many other features. The success measure for this project would be accuracy on predicting which purchases are fraud without misclassifying too many real transactions as fraud. Companies, banks, and even consumers all care about fraudsters being caught. This type of solution would allow banks to save quite the amount of money, with companies having less issues with fraudsters. Of course, consumers also benefit with better methods. With better methods, there are far less chances of customers being falsely accused of being a fraudster with a false positive. Nobody wants to be stuck in another state with their cards frozen due to fraud suspicion, so better algorithms will help save money (and keep anxiety down) substantially. Related work with this data problem is credit card companies. They take the data of transactions that a person has over a period of time, and then they can take new transactions from the person to see if they are real, or if they believe based on the data that it is fraud.

2.
Data and Exploratory Analysis

The data is split into 4 sections corresponding to training and testing sets with the target variables for each already having been split from the feature datasets. The training data represents eighty percent of the total with the final twenty percent having been partitioned as a part of the analysis setup. The training set alone consists of 92,790 data points and 148 features with a combination of character, double, and integer data types. Each data represents one shopper's "cart" with the ability to add up to twenty-four unique items, each unique item has a corresponding feature indicating how many of the item has been placed into the cart, the price, model, make, and goods code. In addition to specific information for each unique item there is a feature for the total number of items purchased. The target feature, "fraud_flag" is represented in the training data set by 0 for non-fraudulent purchase and 1 indicating a fraudulent purchase. However, the target feature presented by the testing set is presented as a ratio. We feel that this points to an initial algorithm of logistic regression.

Due to the nature of the data presentation there are MANY empty strings and NA values in the data set as the majority of the data represent less than ten unique

items in the cart. All of the data features imported as double may be converted to integer but the more concerning features are the twenty-four goods code features. The majority of these are integer data however there are values for “fullfilment”, “service”, and “warranty” preventing these features from being interpreted as integer.

Analysis of the data was completed using tidyverse, dplyr, and ggplot2 beginning with a summary (summary()) of the data and scatter plots, bar charts, and box plots in an effort to understand the data shape and confirm outliers. As might be imagined with data consisting of numbers bought and prices, the data will need to be scaled. Some features hit their max at less than fifty while others see maximums in the tens of thousands. We are still trying to fully understand the visualizations we’ve created and feel that others may be warranted but preliminary analysis indicates that most carts tend toward lower dollar and lower amounts purchased. Additionally, there are FAR fewer fraudulent data points (1319 of the total 92,790) than non-fraudulent data points so this will most likely need to be considered. Fraudulent data points do not seem to be specific to low or high numbers or items, number of unique items, type of item, or low or high price. In other words, we have not found a specific pattern yet in our analysis.

3.
Methods

[Describe the methods you explored (usually algorithms, or data cleaning or data wrangling approaches). Justify your methods in terms of the problem statement. What did you consider but *not* use? In particular, be sure to include every method you tried, even if it didn't "work". When describing methods that didn't work, make clear how they failed and any evaluation metrics you used to decide so.]

4.
Tools

[Describe the tools that you used and the reasons for their choice. Justify them in terms of the problem itself and the methods you want to use. Tools will probably include machine learning, and possibly data wrangling and visualization. Please discuss all of them. How did you employ them? What features worked well and what didn't? What could be improved? Describe any tools that you tried and ended up not using. What was the problem?]

5.
Results
[Give a detailed summary of the results of your work. Here is where you specify the exact performance measures you used. Usually there will be some kind of accuracy or quality measure. There may also be a performance (runtime or throughput) measure. Please use visualizations whenever possible. Include links to interactive

visualizations if you built them. You should attempt to evaluate a primary model and in addition a "baseline" model. The baseline is typically the simplest model that's applicable to that data problem, e.g. Naive Bayes for classification, or K-means on raw feature data for clustering. If there isn't a plausible automatic baseline model, you can e.g. compare with human performance by having someone hand-solve your problem on a small subset of data. You won't expect to achieve this level of performance, but it establishes a scale by which to measure your project's performance. Compare the performance of your baseline model and primary model and explain the differences.]

6.
Summary and Conclusions

[In this section give a high-level summary of your results. If the reader only reads one section of the report, this one should be it, and it should be self-contained. You can refer back to the "Results" section for elaborations. This section should be less than a page. In particular, emphasize any results that were surprising.]

7.
Appendix

Include the link to your github/gitlab repository (that I can access) containing your R programs/scripts, and link to the data.

https://github.com/SColson82/Detecting_Fraudsters_ML.git
https://challengedata.ens.fr/participants/challenges/104/