Fraud Case Study

Authors: Karey Shumansky, Megan O'Rorke, Nick Halpern, Aymeric Flaisler
4/18/2017

Problem

In order to improve fraud detection and support customer service team triage efforts, the business needed a web interface with a ML model under the hood that flags events as high, medium or low probability of fraud.

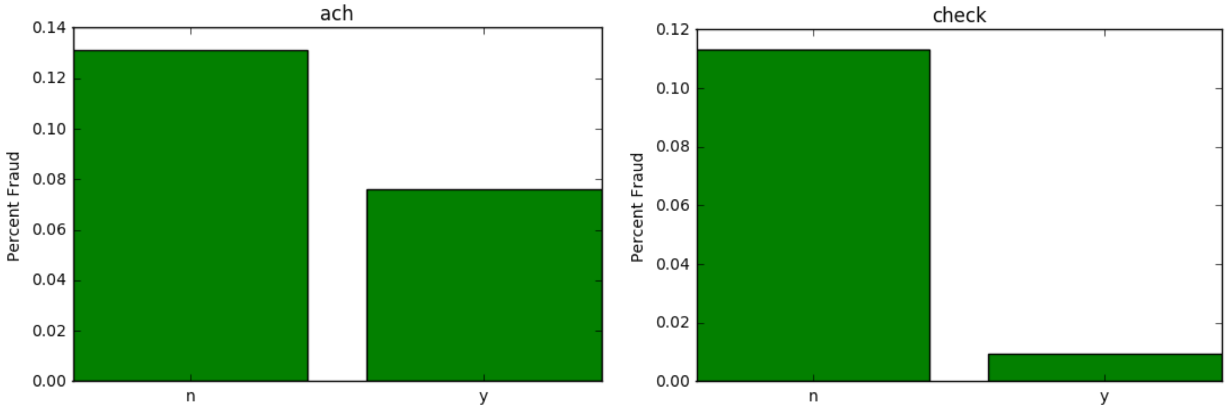
Data

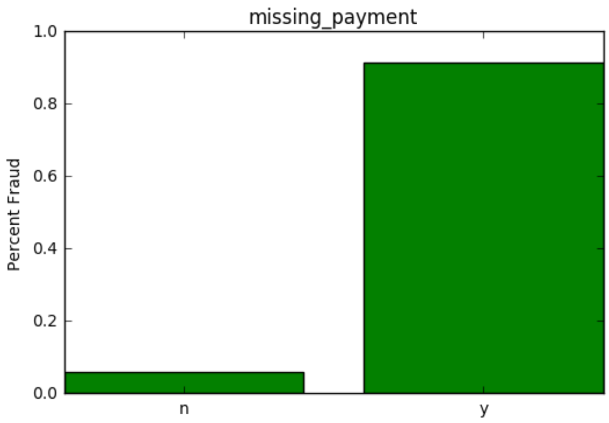
The data is 14,337 events from 2007-2013, of which 1,239 are fraudulent. We define events as fraudulent if the account type is labeled as **fraudster**, **fraudulent_event**, or **fraudster_att**. We do not classify spam events as fraudulent.

Data Analysis

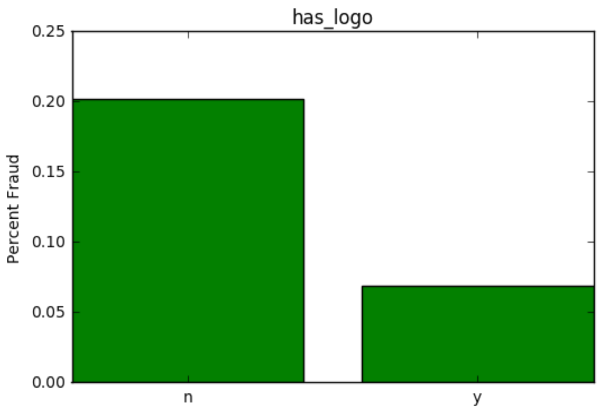
There are many fields which are highly (positively or negatively) correlated with fraud. Below are a few examples.

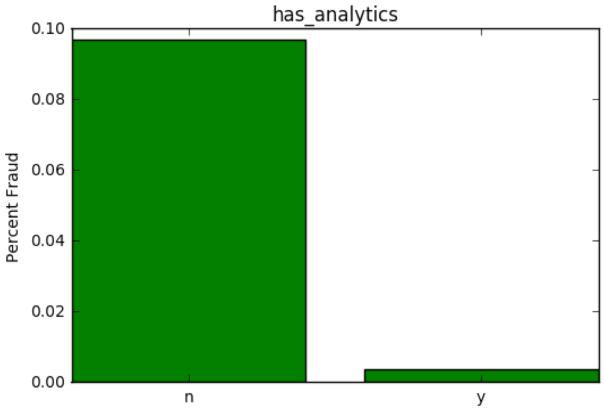
Payment type: checks are almost never fraudulent. ACH transactions are roughly equivalent to the average for the whole universe. Events with missing data are almost always fraudulent.

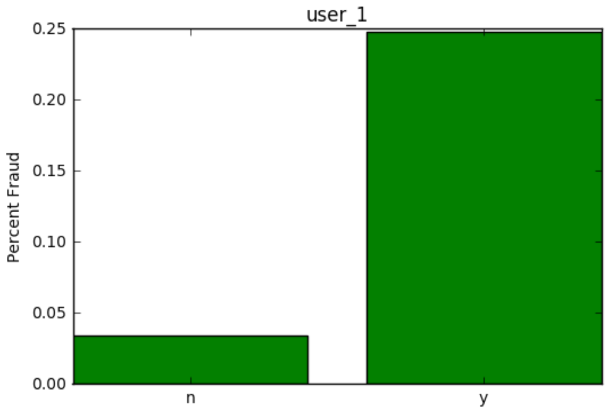


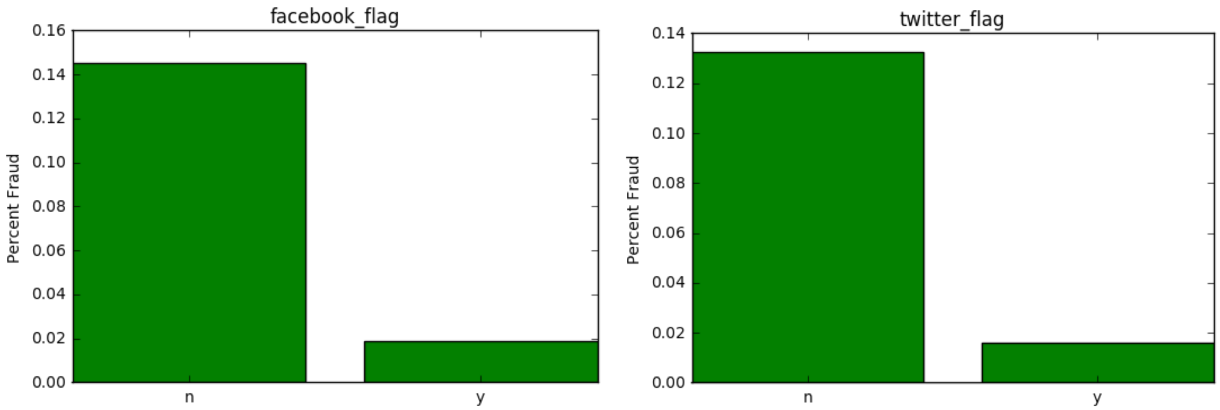


Many fields which indicate how active an event planner has been promoting the event / making a detailed post are highly predictive of fraud.









Engineered Features

Currency flags: EUR, GBP
Social media usage: Twitter_flag, Facebook_flag
Information in the posting: Venue null/not null; Organization null/not null; Previous Payouts 0/non-zero; Payment method ACH, Check or Missing; Logo yes/no

Metrics

As with many classification problems, we recognize that false positives and false negatives are not equally costly. We make the assumption that in the case of false positives, the company forgoes fees of 2.5% (5.5% commission - 3% credit card fees). In the case of false negatives, the company pays out revenues to the event and then refunds its customers, which costs the company 100% of the ticket amount. Therefore, our cost function is:

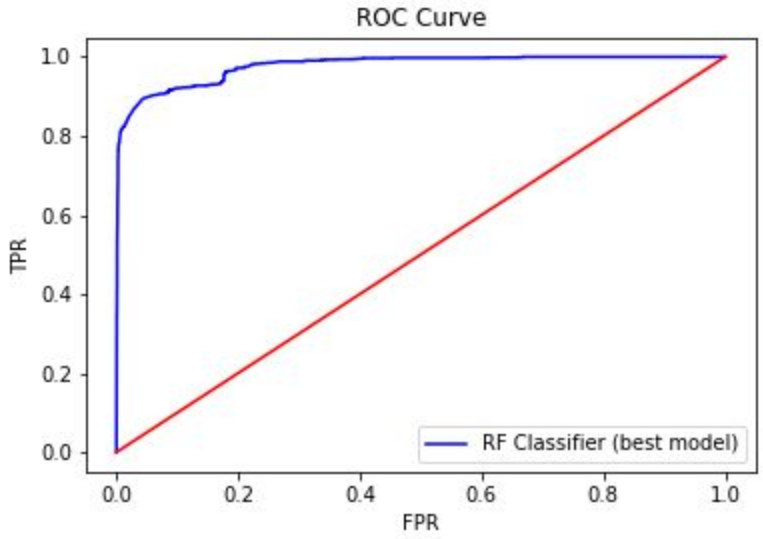
c = 0.025 * ticket sales * false positives + ticket sales * false negatives

We can also be more sophisticated at include long term value into the cost function. The theory is that if an event is incorrectly labeled a fraud, the user won't return to the site and their future business has been lost. In this case, we include estimated LTV into the cost function:

c = 0.025 * (estimated lifetime ticket sales) * false positives + ticket sales * false negatives

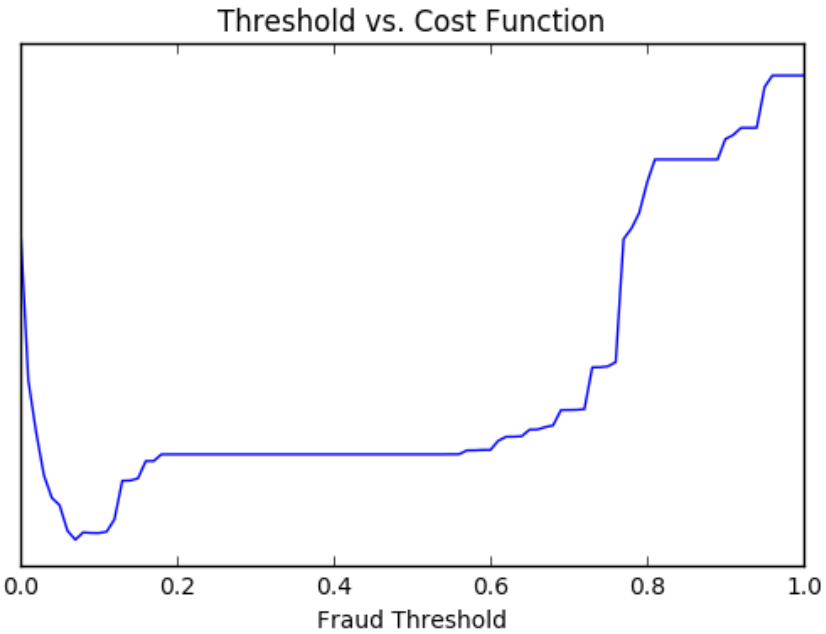
Modeling and Tuning

We got very good results from a Random Forest model. We choose this as our model after deciding against Logistic Regression (we have too many correlated features) and K Nearest Neighbors (too slow to predict).

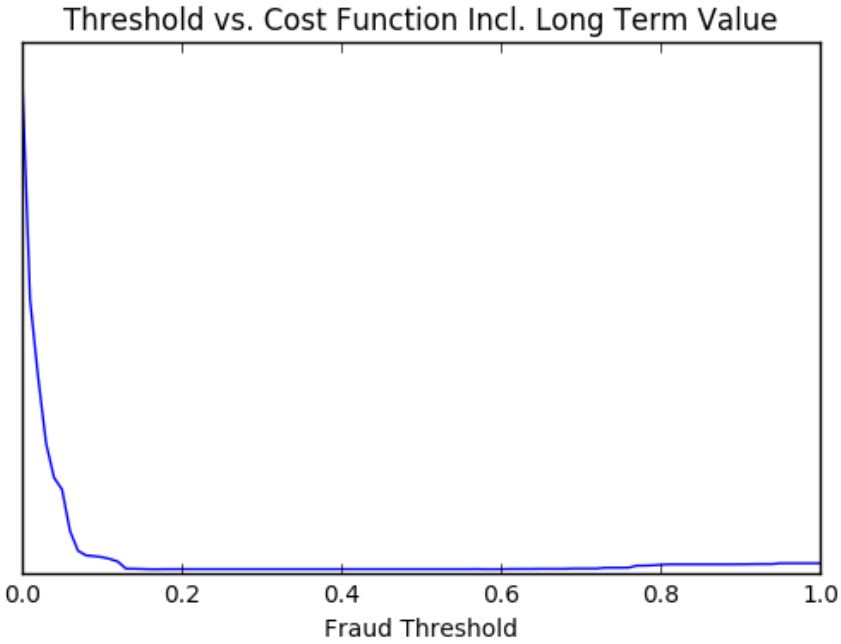


Hyperparameters: {‘max_depth’: 5, ‘min_samples_split’: 4, ‘n_estimators’: 50}

Not taking into account LTV, choose a threshold of 7%.



If we take LTV into account, choose a threshold of 16%!



Next Steps

- Refine the Consumer Long Term Value model. The threshold we want to set for our fraud detection is highly sensitive to this calculation.
    - How do gross ticket sales change over time for a single user? If they grow, and we see a possible fraud event with low GTS perhaps it is smart to not to label as fraud since the cost to refund is low compared to possible future event commissions.
    - On the other side, what is the impact on ticket sales if a consumer is sold a ticket to a fraudulent event? This is an argument for increasing the penalty in the cost function for false negatives.
- Look at how the feature importances change through time. Do fraudsters get wise to the fraud detection algorithm?
- Look into the population of spammers. What is the difference between a spammer and a fraudster? If we don't block spammers do they become fraudsters in the future?