Supervised Learning Capstone

Predicting online credit card fraud: my first Kaggle competition

Jeremy Brezovan, Sept. 2019

The problem: online (card-not-present) fraud

"While EMV chips have sharply reduced counterfeit credit cards at point of sales (POS), e-commerce fraud where the card is not present (CNP) for the sale grew twice as fast as POS fraud."

- Eric Kraus, vice president of fraud management, Fidelity National Information Services (FIS)

The problem: online (card-not-present) fraud

- **Retailers stand <u>to lose about $130 billion in revenue on fraudulent card-not-present transactions</u> between now and 2023 as they fail to keep up with digital fraud detection and prevention measures.**
- **By 2023, companies supporting retail transactions are expected to spend about $9.6 billion yearly on fraud detection and prevention capabilities.**

https://www.retaildive.com/news/digital-card-not-present-fraud-to-hit-130-billion-by-2023/545171/

The Kaggle competition

- **Led by IEEE-Computational Intelligence Society (https://cis.ieee.org/)**

- **Data provided by Vesta Corporation (https://trustvesta.com/)**

"In this competition, you'll benchmark machine learning models on a challenging large-scale dataset...If successful, you'll improve the efficacy of fraudulent transaction alerts for millions of people around the world, helping hundreds of thousands of businesses reduce their fraud loss and increase their revenue."

https://www.kaggle.com/c/ieee-fraud-detection/overview/description

The Kaggle competition dataset

- **Metadata for this dataset was mostly nonexistent. However:**
  - **A handful of variables have descriptive names or contain easily identified data**
  - **There is an informative "Data Description" thread on the discussion board, guided by a Vesta representative**
  - **Exploration of the data also provided some useful insights**

https://www.kaggle.com/c/ieee-fraud-detection/overview/description

The Kaggle competition dataset

- **Datasets consist of two files: Transactions and Identity**

- **Tables are joined by a Transaction ID. Not all transactions have associated identity data.**

- **Kaggle made two sets of data available:**

   1. **"Training" set - includes "isFraud" variable, 0|1**

   2. **"Test" set - fraud TBD by the model**

- **Competition submissions must have the following variables: Transaction ID, and probability of fraud, represented as a decimal between 0 and 1.**

Exploring the dataset

Exploring the dataset: transaction data

- **Transaction file:**
  - **Identifiable variables include: transaction ID, transaction amount, transaction date[*], purchaser ZIP and country[*], purchaser and recipient email domains, product code[*]**
  - **Encoded/undocumented variables are grouped: card attributes, distances, related counts, time deltas, match indicators, Vesta-engineered features**

  * Data is transformed to obscure its actual value or meaning

Exploring the dataset: identity data

- **Identity file:**
  - **There are only two immediately identifiable variable names in this file: Device Type (primarily desktop or mobile), and Device Info, which appears to describe the purchaser's operating system.**
  - **Everything else I understand about this file comes from the Kaggle comments section, and my own exploration.**
  - **About 45% of transactions in the training set have associated identity data. This seemed good enough to make the identity data worth keeping.**

Exploring the dataset

I took a look at the distributions of all of the variables, broken down by group, to see how I could clean up and prep this data.

- What type of data each variable represents
  - Categorical/continuous?
  - Possible purpose? (informed by the Kaggle comment thread)
- Where NaNs/nulls exist, and which values we can use to replace them.

Exploring the dataset: card attributes

Card attributes are a mix of categorical and continuous data, including:

- **Card network**

- **Card type (debit/credit mostly)**

- **Probably credit limit**

- **Possibly APR**

- **Two additional, unknown continuous variables**

Exploring the dataset: addresses and distances

- **Address variables represent billing ZIP code and country**

  - ZIP is only 3 digits, 100-540; country is a number, 10-102

- **Purchaser and recipient email address domain**

  - Recipient domain is only populated when relevant--gifting a good or service

- **Distances between billing and shipping address, for example**

Exploring the dataset: counts

"How many addresses are associated with the payment card" is an example from the Vesta rep

- **All other context is missing for these variables**
- **Mode is often 0 or 1, but with some huge outliers**

Exploring the dataset: time deltas

Examples from the Vesta rep include:

- **The number of days since the previous transaction**

- **Last time this email address was seen**

- **Last time since this billing address was used**

Exploring the dataset: match indicators

Mostly binary, T/F values.

Examples from the Vesta rep include:

- Does the phone area code match the ZIP code?

- Do purchaser and recipient's first or last names match?

Exploring the dataset: Vesta-engineered features

- **Absolutely no idea what these are, but there are 339 of them.**

- **Many of these variables seem to have a similar distribution, which made me wonder how well PCA would work combining them into a much smaller number of components.**

Exploring the dataset: identity information

- **Mix of continuous and categorical variables**

- **Immediately identifiable values: operating system, browser user agent, desktop or mobile indicator, screen resolution**

Exploring the dataset: the target variable

Fraudulent transactions are a small number of the total transactions.

Non-fraudulent 96.5%
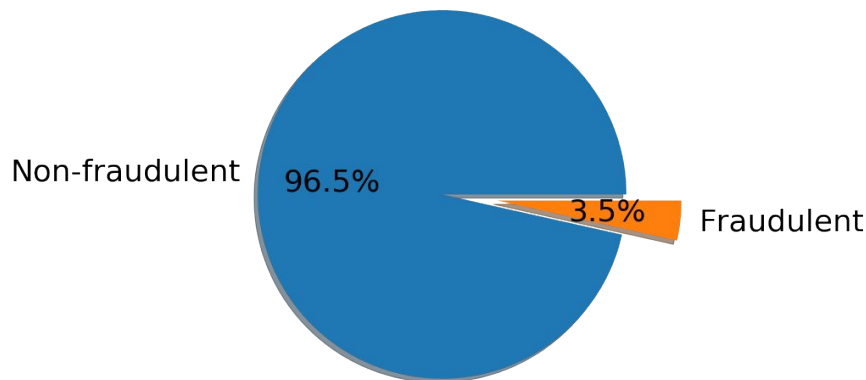
3.5% Fraudulent

In the training data, approximately 3.5% of the transactions are fraudulent. The Vesta rep on the Kaggle discussion board suggested this is representative of what they see in the wild as well.

This bias must be taken into account when training the model, or the model could easily say nothing is fraudulent, and still be correct 97% of the time.

Exploring the dataset: fraudulent transactions

Transaction Amount (USD) for fraudulent transactions

Transaction Amount (USD)
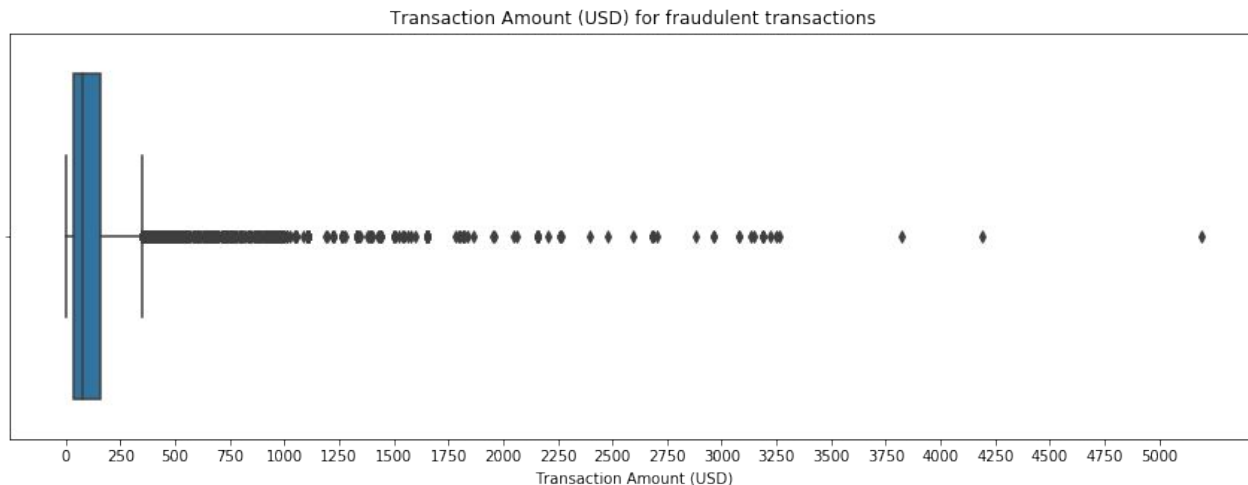
The mean transaction amount is only $149; the median is $75.

Cleanup/prep for modeling

Cleanup/prep for modeling: cleanup issues

- **Many of the Vesta-engineered variables, and some of the identity variables, are poorly populated.**

- **Device Info, and identity columns that contain categorical information such as screen resolution and browser information, all have a relatively high number of unique values. This makes encoding them difficult.**

- **Some variables are categorical, with few categories**

Cleanup/prep for modeling: resolving issues

- **I tried dropping Vesta and identity variables that were > 50% NaNs/nulls. (Ultimately, keeping them and performing PCA was the better approach.)**

- **I cleaned up the Device Type, screen resolution, and browser variables, reducing the number of unique values (to 14, 4, and 8, respectively)**

- **Encoding T/F, Found/NotFound/etc. as numeric values**

Cleanup/prep for modeling: resolving issues

- **P_emaildomain and R_emaildomain** represent the email of the purchaser, and the email of the "recipient".

- Values differed between the test set and the training set, making it difficult to accommodate them either via encoding or by using a random forest model.

- In the end, they were also not very relevant to detecting fraud, so both columns were dropped.

Cleanup/prep for modeling: replacing nulls/NaNs

- **Other categorical variables: used "None", "other", or "NA" to replace NaNs**

- **Continuous variables: use either the median value, or 0**

- **Numeric variables that appear to be categorical in nature: Either chose a number outside of the range of the data (often 0), or choose the most common value. (Ex: addr1 and addr2)**

Cleanup/prep for modeling: applying PCA

- As I guessed, applying Primary Component Analysis reduced the feature set from around 400 variables to under 150

- Still retained 95% of the variance explained by the original feature set, with much less data to process during model training
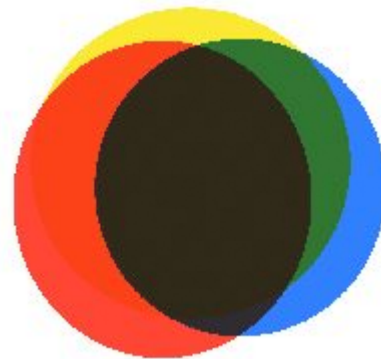
Building and choosing models

Building and choosing models

A single, dependent variable with only two possible outcomes sounds like a great case for a classifier.

1. A **random forest** is an obvious choice, as they handle both regression and classification issues rather well.
2. An **SGD classifier** may also work.
3. A **gradient boosting classifier** is my third option.
4. We were asked to consider how a KNN model could be used to detect fraud in the exercises, so let's look at a **KNN classifier** too.

Building and choosing models

Best performers, ranked:

1. **Random forest**
2. **Gradient boosting**
3. **SGD**
4. **KNN**

I built a final version of the random forest model using parameters suggested by GridSearchCV, and ran training/test/Kaggle test data through it.

Submission to Kaggle

Submission to Kaggle: the output of my model

Probability of credit card fraud, Kaggle submission
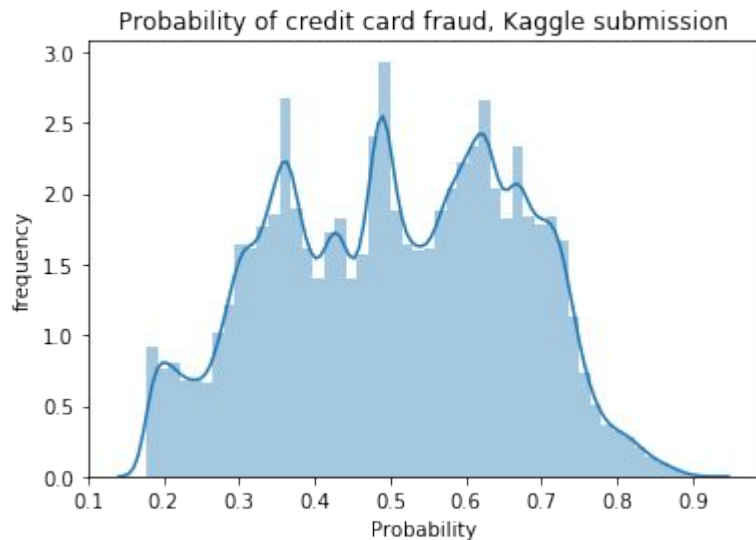
Assuming Kaggle's test set has a distribution similar to what Vesta finds in practice, I was concerned at first that not enough of this distribution was < .5

Submission to Kaggle: evaluation
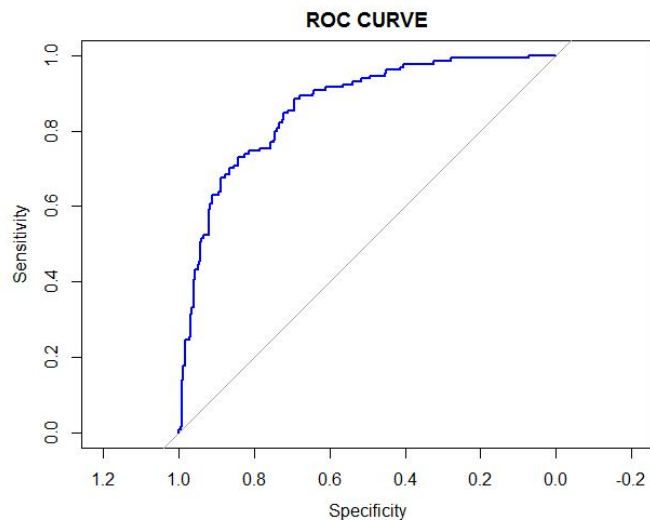
ROC CURVE

"**Submissions are evaluated on area under the ROC curve between the predicted probability and the observed target.**"

Image from https://towardsdatascience.com/implementing-binary-logistic-regression-in-r-7d802a9d98fe

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	Konstantin Yakovlev			0.9642	181	21h
2	alijs			0.9642	68	3d
3	ML Keksika			0.9639	124	20m
4	Fatih Ozturk			0.9637	104	21m
5	[ka.kr] boradori			0.9634	292	5h

In the money   Gold   Silver   Bronze

So, I won't be winning any monetary prizes, or presenting this model at a conference...

4425	Jeremy B			0.8521	1	6m
4426	Leo Breiman			0.8517	2	2d
4427	Salty Gold Fish			0.8516	1	1mo
4428	DataWookie			0.8514	2	2mo
4429	Eivind			0.8508	4	2d
4430	Chirag Gomber			0.8506	10	1mo

https://www.kaggle.com/c/ieee-fraud-detection/leaderboard

Submission to Kaggle: my rank

IEEE-CIS Fraud Detection
Can you detect fraud from customer transactions?
Research · 17 days to go · 🏷 tabular data, binary classification

4432/5035
Top 89%

I'm in the bottom half of the leaderboard, but for a first submission, landing in the top 89% seems pretty good. I'll take it!

Future improvements

Future improvements

Improve the model

- **I read about [using a random forest to learn imbalanced data](#).**
- **Spend more time understanding and tuning hyperparameters. I tried using GridSearchCV to automate some tuning, but it took forever.**
- **I read about [a Bayesian Optimizer](#) that is supposed to be much faster than GridSearchCV. I wanted to learn to use it to tune parameters, but time became a concern.**

Future improvements

Other performance improvements

- **Train on a machine with more resources than my laptop--maybe a virtual machine in the cloud (AWS/Azure/Google Cloud).**
- **Transform the code in my notebook into a Python script that can run without the overhead of the browser and Jupyter Notebooks.**
- **In a production environment, breaking this pipeline into rerunnable components would also help with future support/development efforts.**