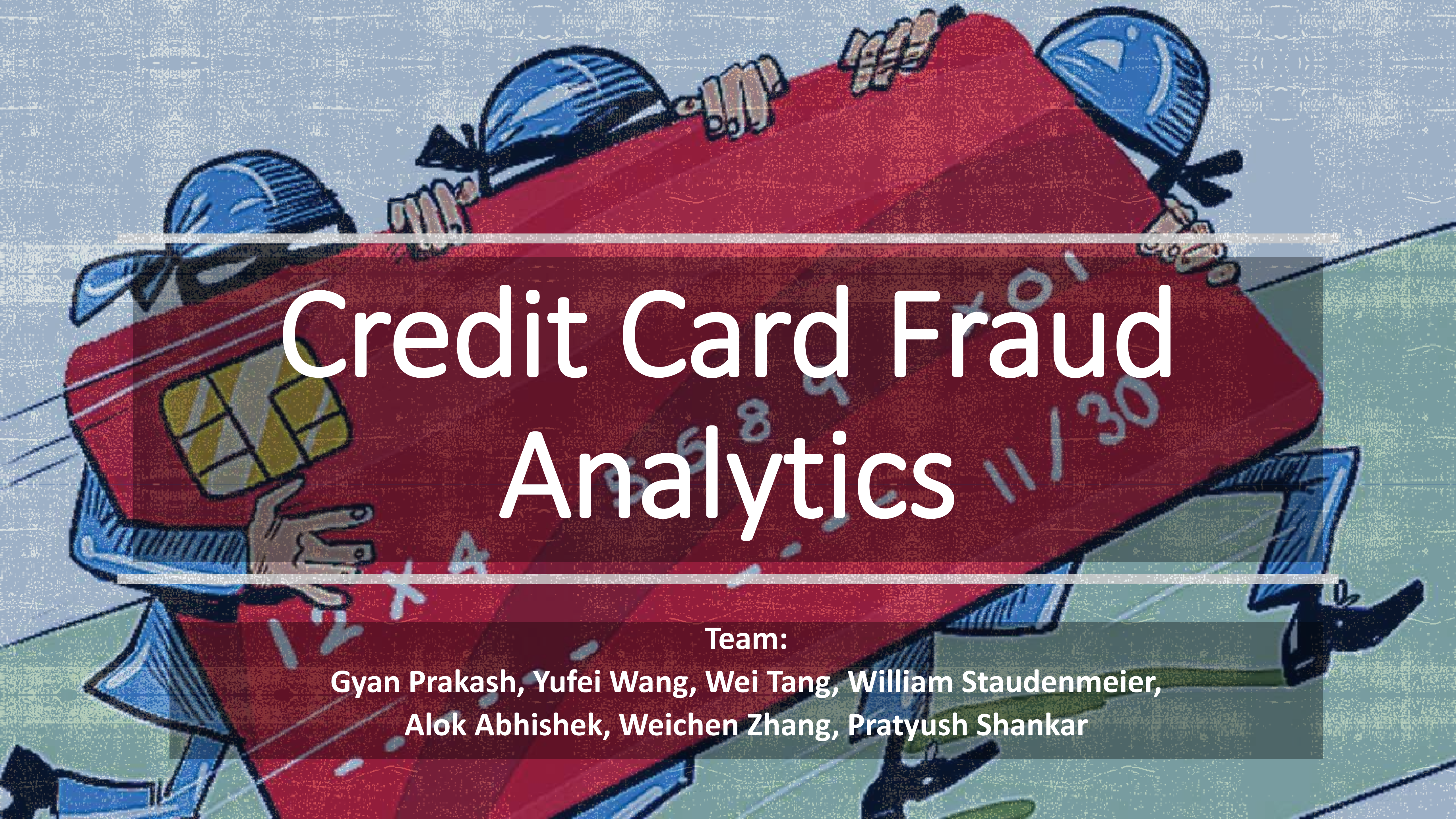


Credit Card Fraud Analytics

Team:

Gyan Prakash, Yufei Wang, Wei Tang, William Staudenmeier, Alok Abhishek, Weichen Zhang, Pratyush Shankar

Executive Summary

Objective

Build a supervised learning model to detect fraudulent credit card transactions

Data Analysis Methods

5 step approach from data exploration, imputation, expert variable creation, feature selection and data modeling

Conclusions

Our best model, Bootstrap Forest, gave an FDR at 2% of 52.37% on OOT dataset.

Problem Statement

- About 31.8 million U.S. consumers had their credit cards breached in 2014
- As per studies in 2007 for every $100 of transaction $0.07 was lost due to fraud
- As part of this project we are building a supervised learning model to identify fraudulent credit card transactions

Process Overview

- Data Discovery and initial analysis
- Data Visualization and understanding

- Expert variable creation using training data set
- Data imputation

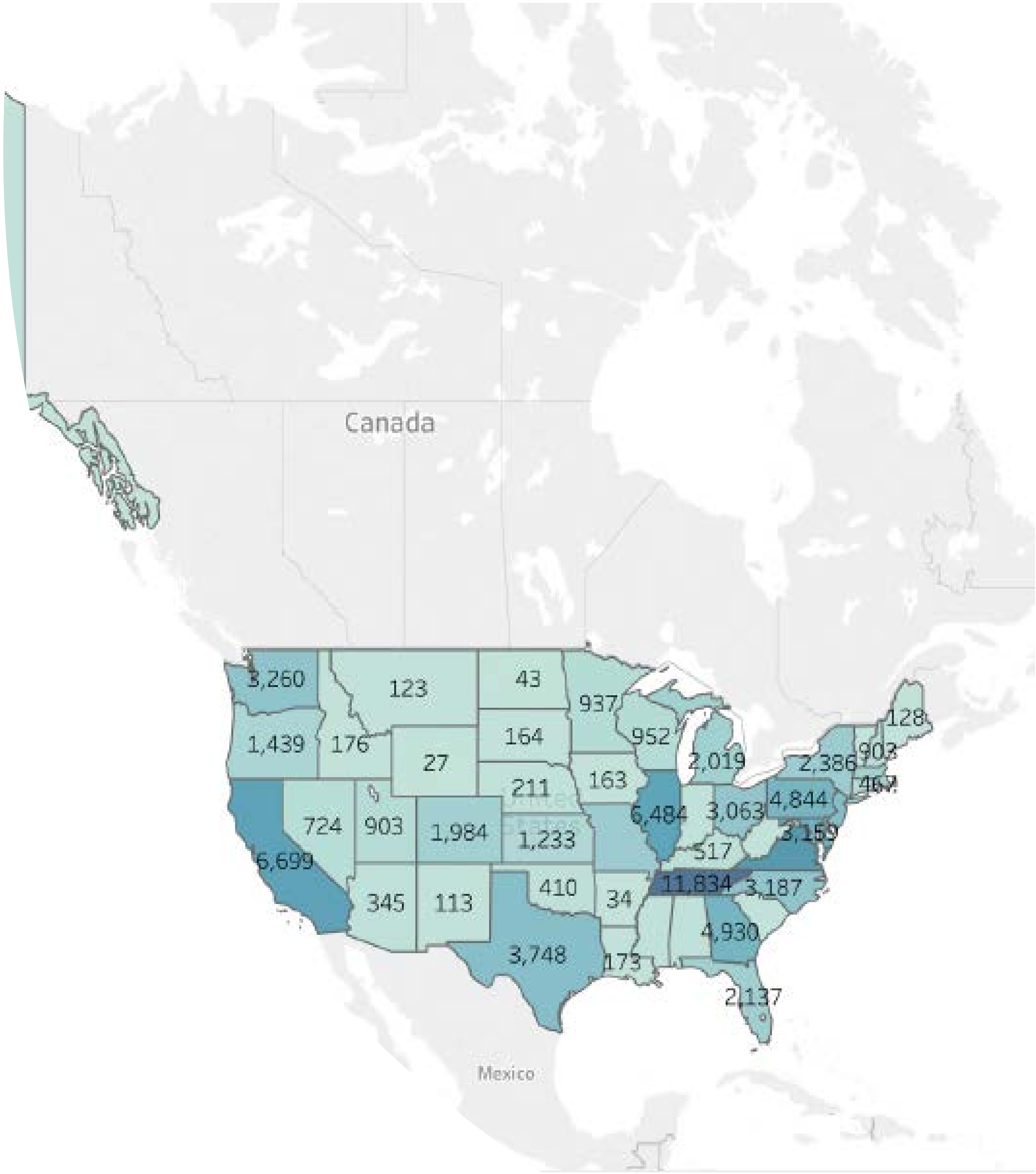
- Feature Selection (KS + LASSO)

- Data Partition – Training, Testing and OOT

- Predictive Modeling

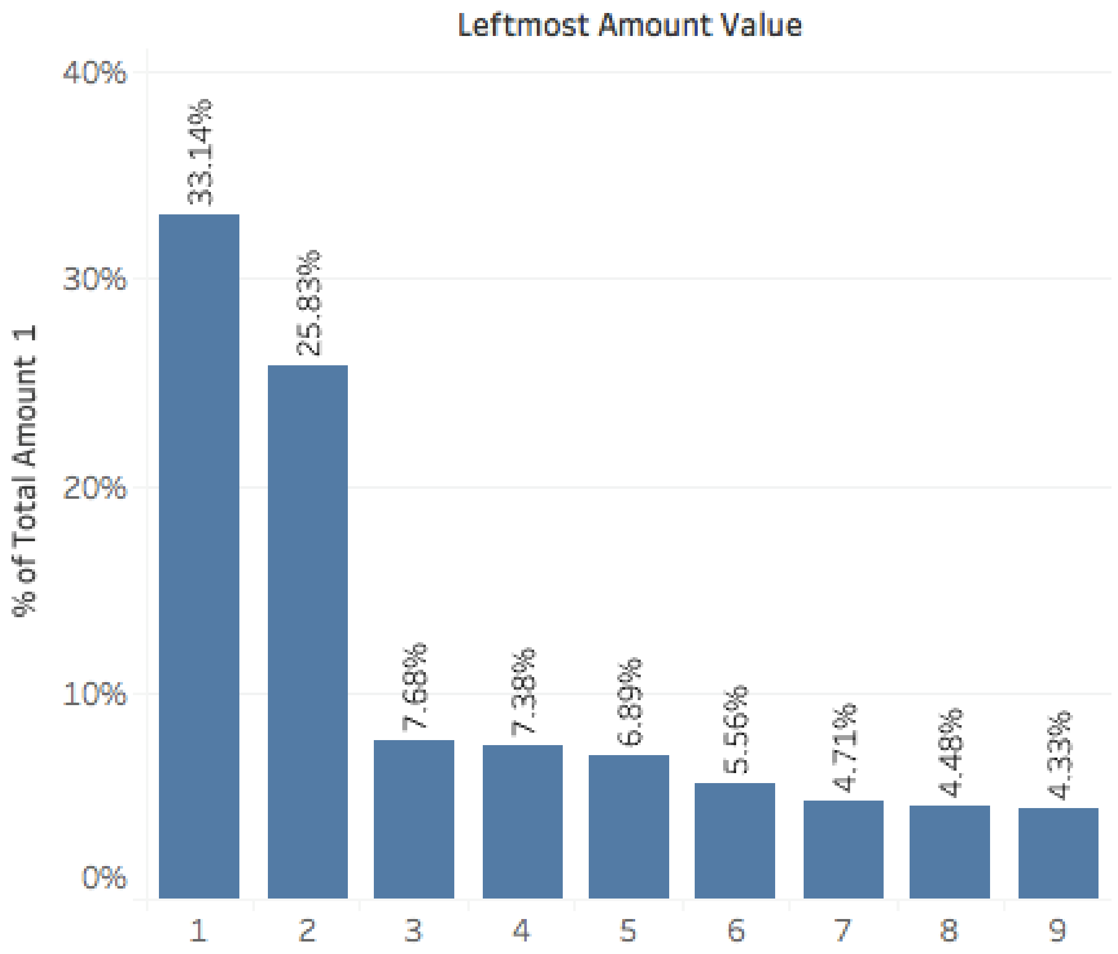
Data Description

- One year of credit card transaction data for Gov. Agency
- 96,707 sample transaction labeled with fraud/not fraud
- 6.26 MB and 10 columns
- The data contains following information:
  - Card number
  - Date of transaction
  - Merchant number
  - Merchant description
  - Merchant state
  - Merchant zip
  - Transaction type
  - Transaction amount
  - Fraud label



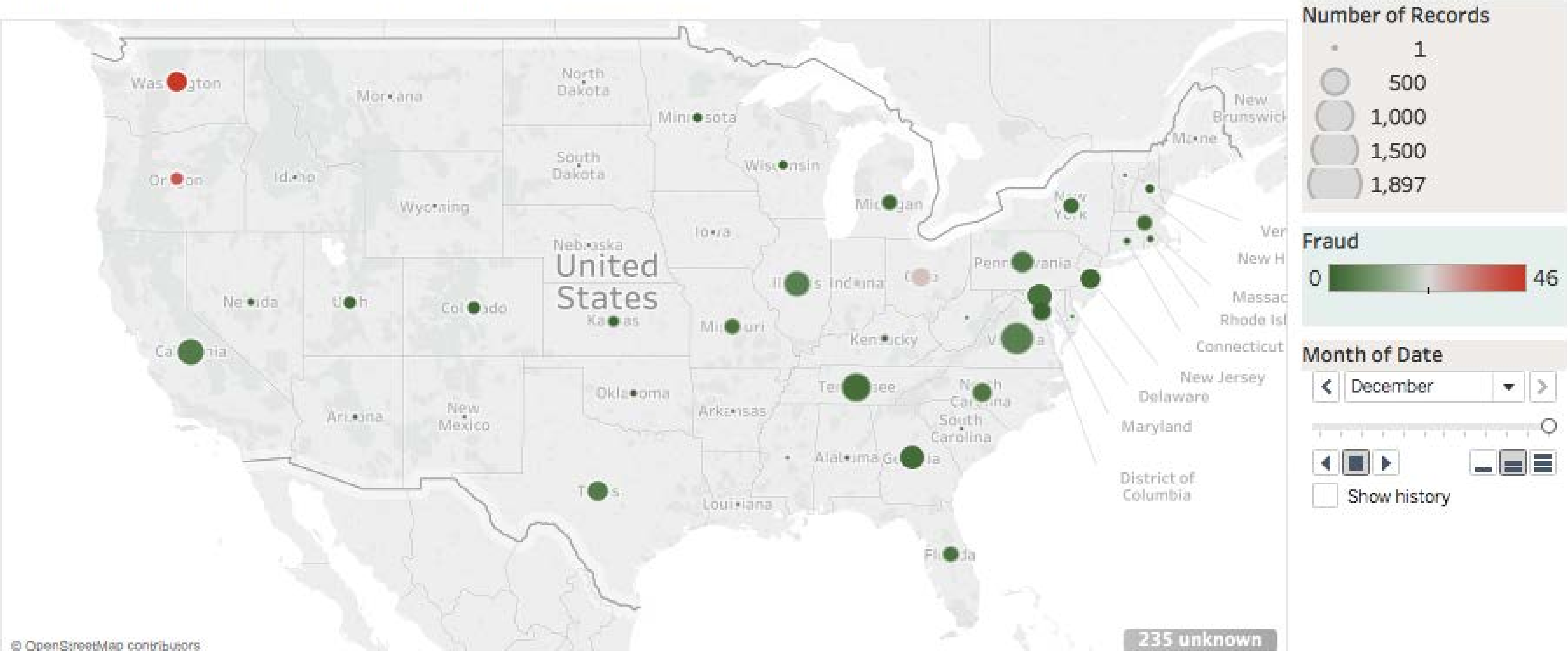
Data Visualization - 1

Benford's Law



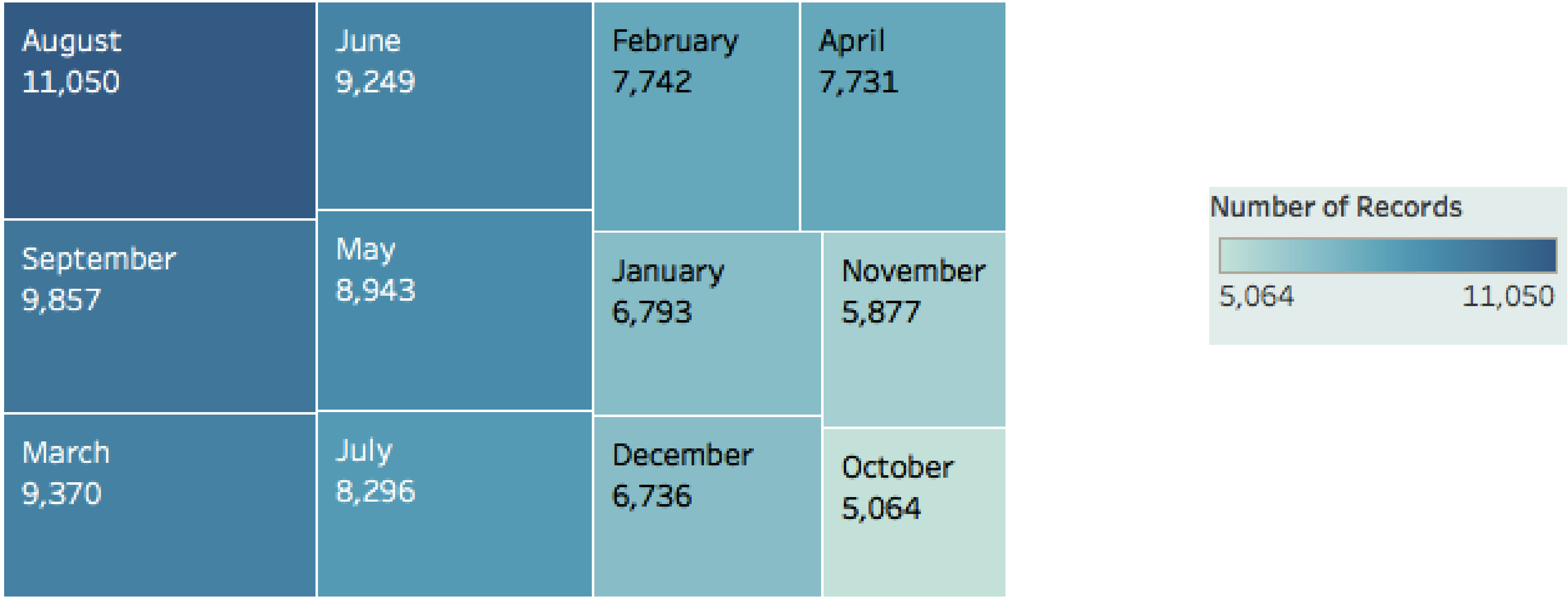
Data Visualization - 2

Transcation Counts - Fraud Map - *December*

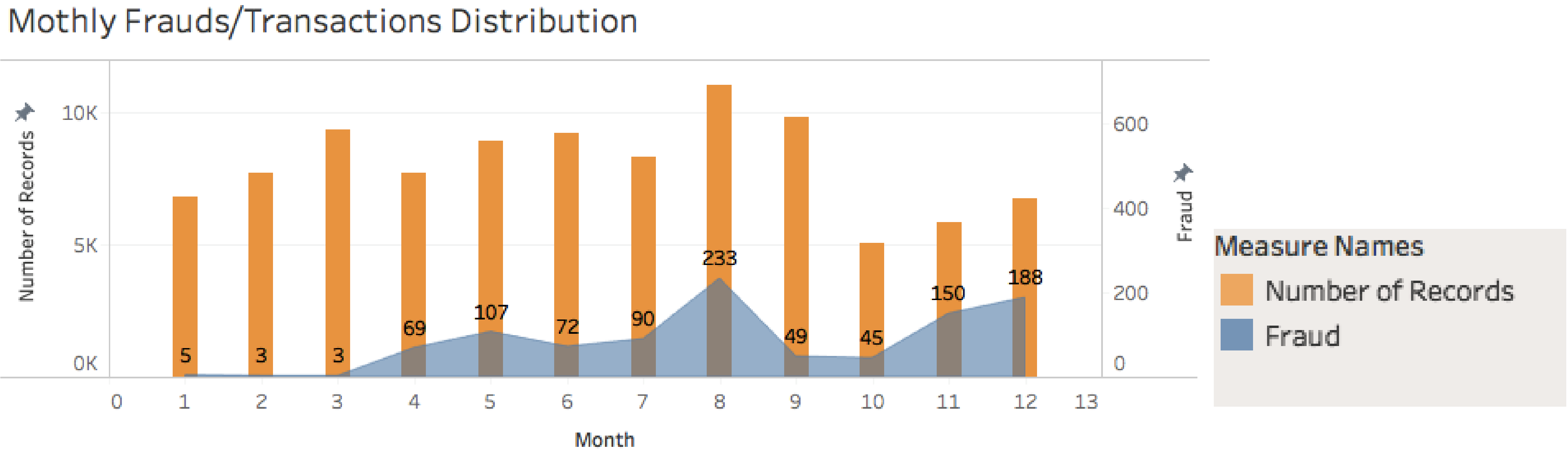


Data Visualization - 3

Monthly Transactions Distribution



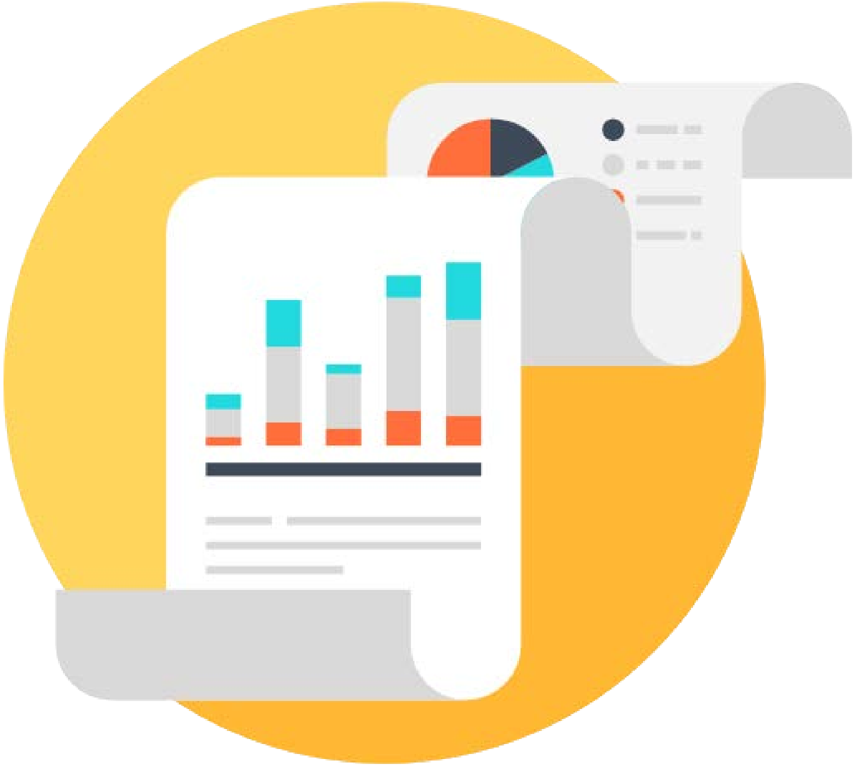
Data Visualization - 4



Data Cleaning

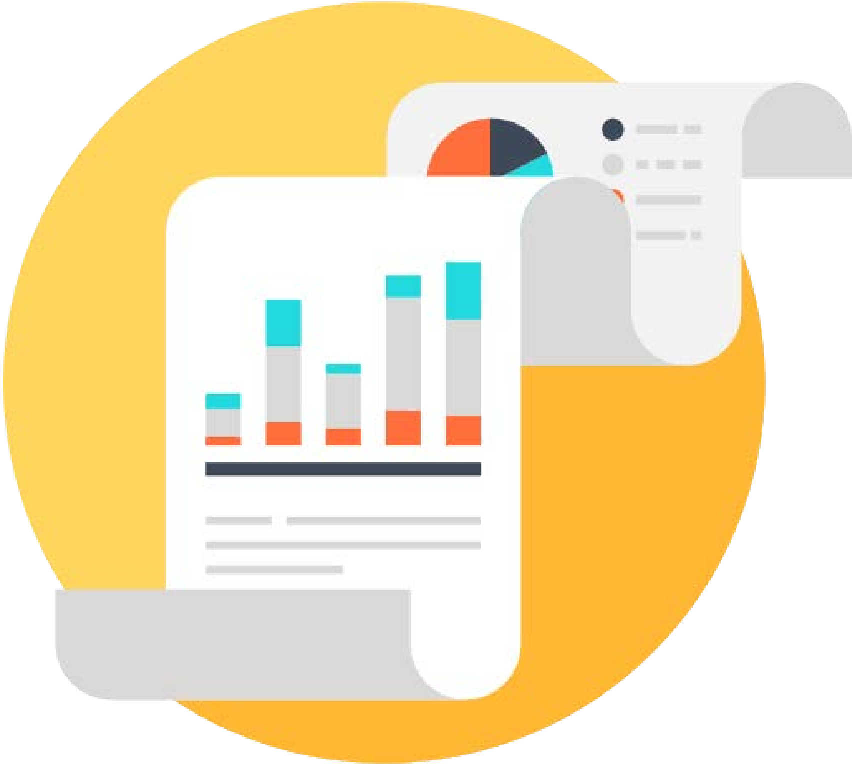
- Three variables -- merchnum, merch.state and merch.zip are not 100% populated, with missing values and 0's.
- We assigned values to these fields based on merch.description. We considered records with different merch.description as different merchants, and assumed that each merchant should have a unique merchant number, be in one state, and only have one zip code.
- We skipped missing values while counting linkages to create expert variables
- Since all these three variables are categorical, we assigned unique values in these fields according to merch.description, and those values were designed to be quite different from other existing values

Expert Variable Creation



- Since this analysis involves time, with limited data, we chose four different time windows 1, 3, 7, 15 and 30 days.
- The rationale is to capture more (and different types of) fraudulent records that might be detected in those time windows.
- We did not use fraud label to create expert variable to represent more real world scenerio

Expert Variable Creation



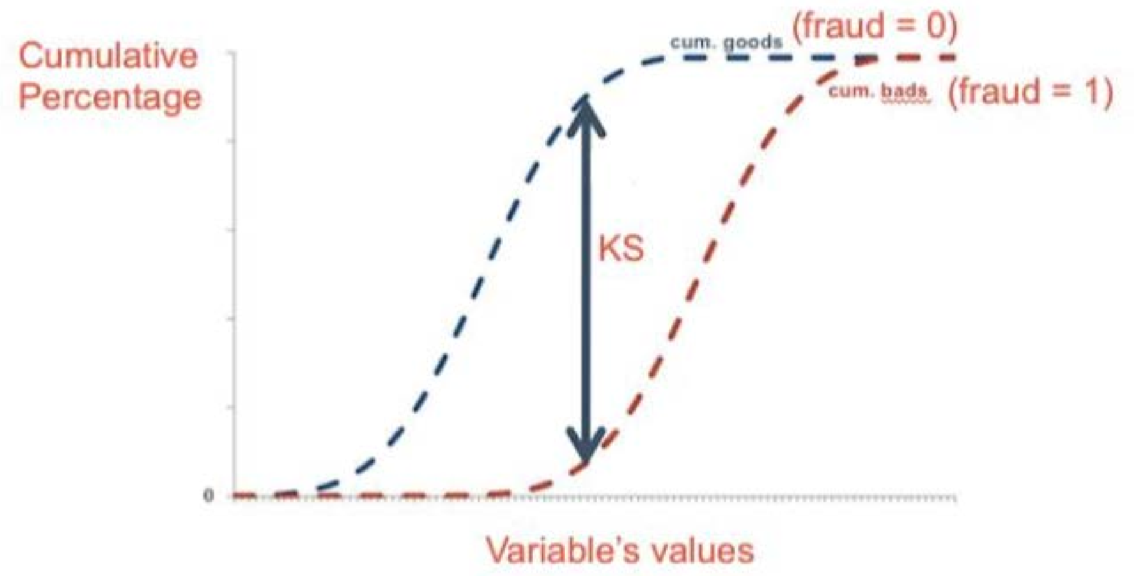
Type I variables are intended to capture unusual amounts of transaction, both at the card level and the merchant level

Type 2 variables are intended to capture unusual transaction frequency during a set period of time, both at the card level and the merchant level

Type 3 are location related variables, which are intended to capture merchants with different zip codes and states in a set period of time.

Type 4 are intended to catch card appearance pattern, either for a merchant or for a card holder.

Feature Selection - KS & Lasso

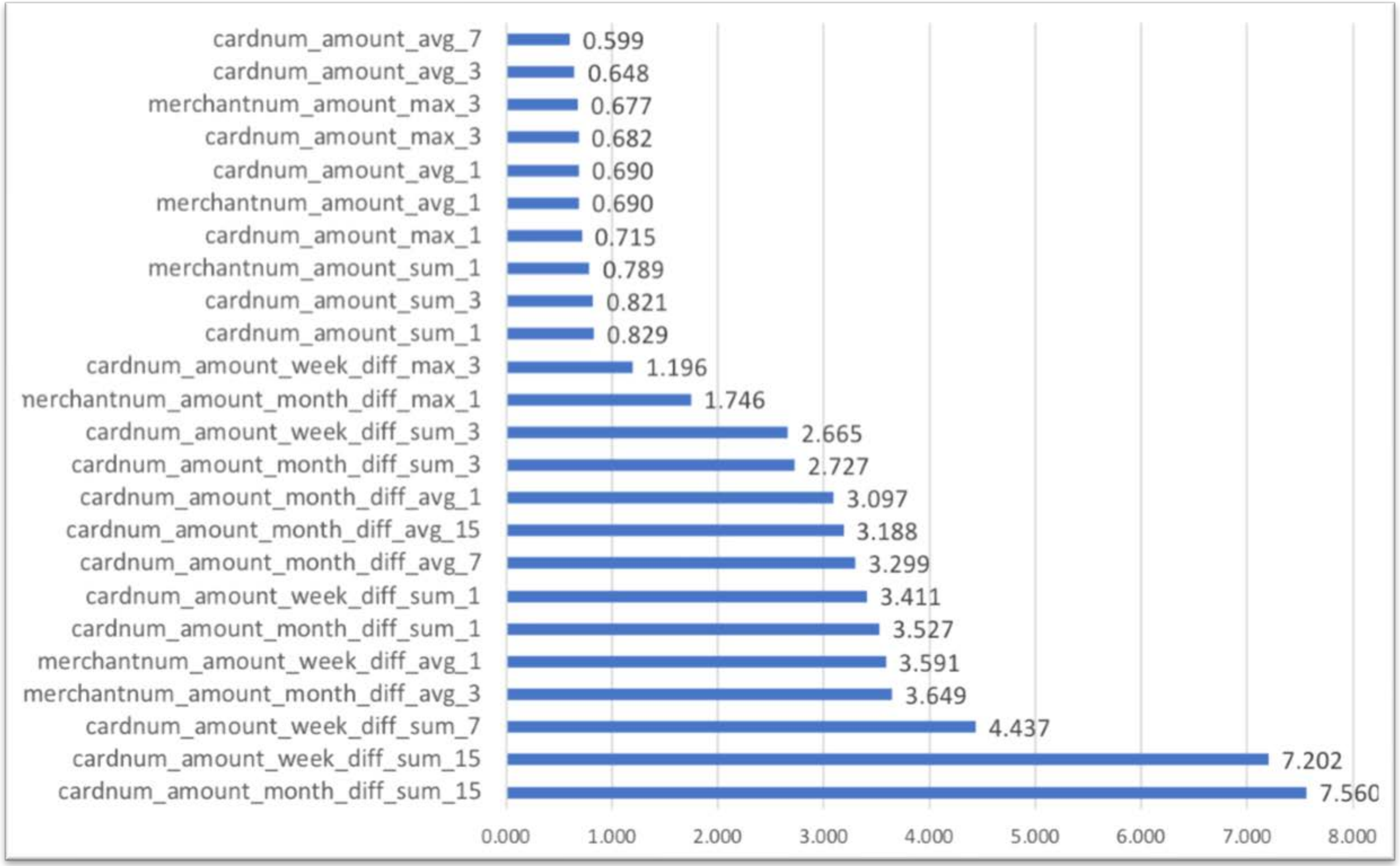


- KS: KS is a robust measure of how well two distributions are separated (goods vs bads)
- Lasso: Lasso can solve the multicollinearity problem between variables

KS **Lasso**

130 variables →→ 40 variables →→ 25 variables

Gini Index – Variable Importance



The Gini coefficient measures the inequality among values of a frequency distribution

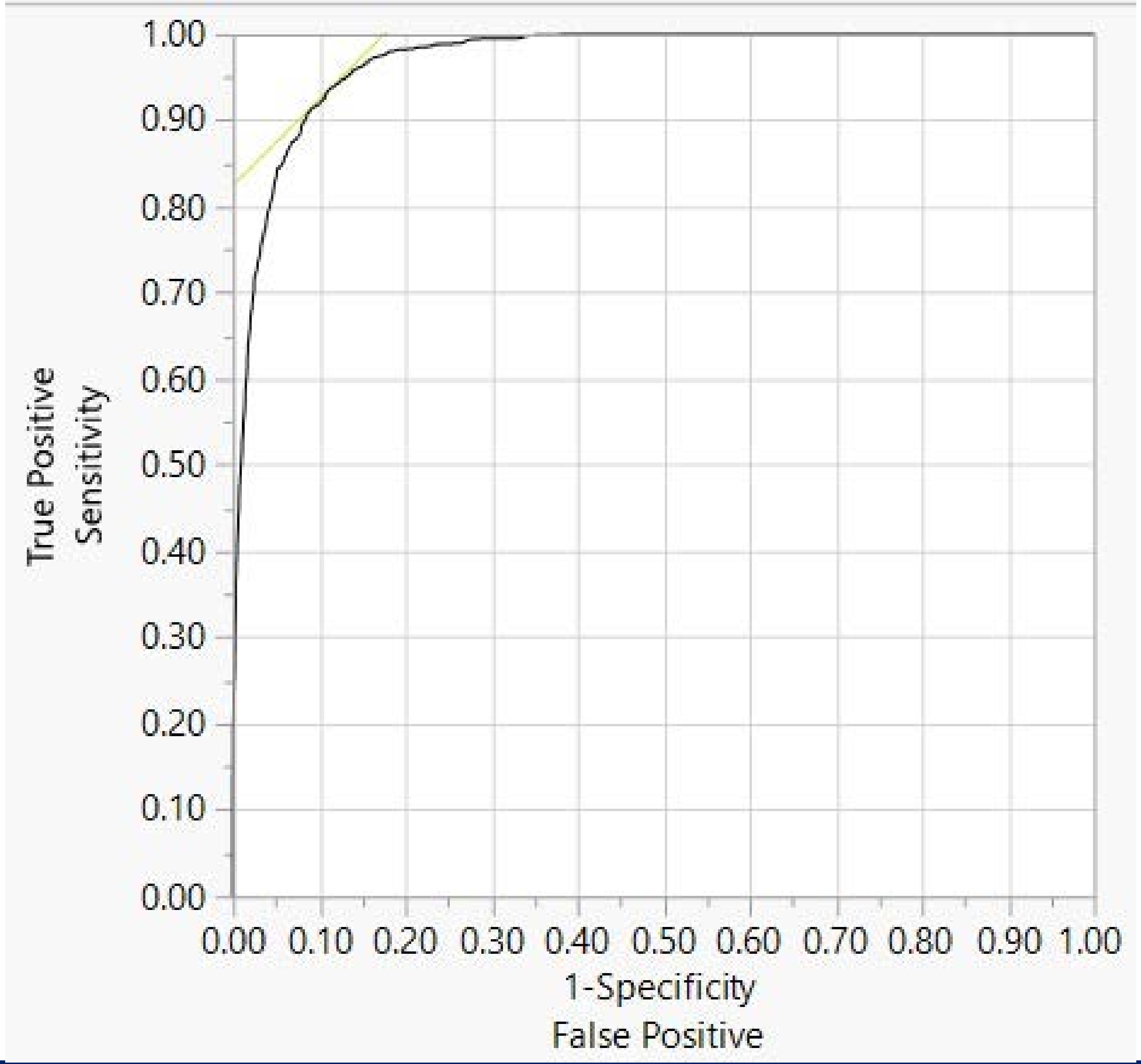
FDR at 2% for Different Models

	Bootstrap Forest	Boosted Trees	Neural Network	Naive Bayes	Logistic Regression
Training	66.51%	64.41%	**70.89%**	59.98%	62.13%
Testing	**58.94%**	53.61%	60.08%	50.00%	49.05%
OOT	**52.37%**	51.48%	42.31%	38.78%	47.63%

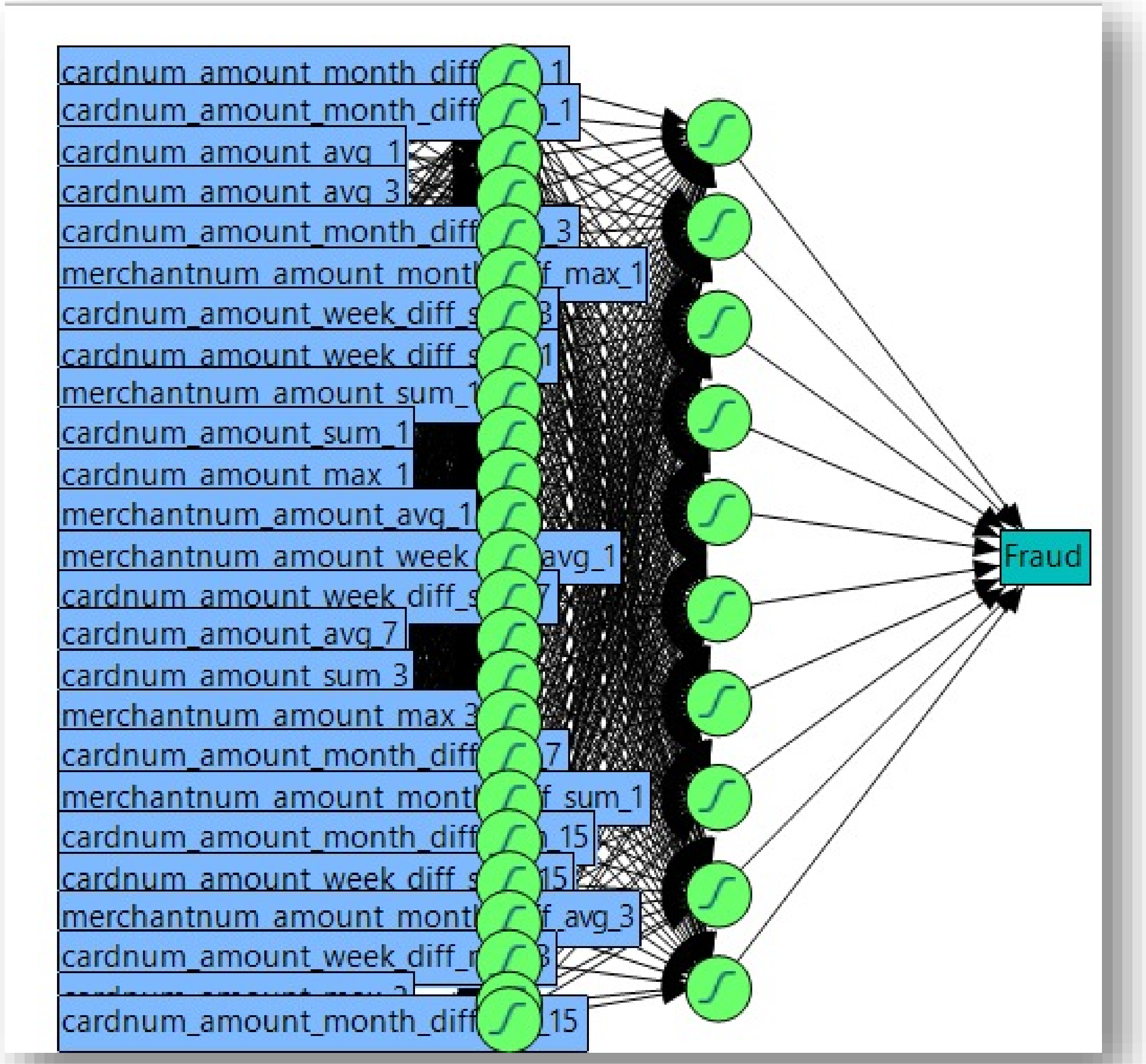
Variable Importance based on Bootstrap Forest

	Fraud			
Predictor	Contribution	Portion		Rank
cardnum_amount_month_diff_sum_3	47.1125	0.1676		1
cardnum_amount_sum_3	30.3670	0.1080		2
cardnum_amount_sum_1	30.1621	0.1073		3
cardnum_amount_week_diff_sum_1	28.3481	0.1009		4
cardnum_amount_week_diff_sum_3	28.0321	0.0997		5
cardnum_amount_month_diff_sum_1	16.7704	0.0597		6
merchantnum_amount_sum_1	13.6179	0.0484		7
merchantnum_amount_month_diff_sum_1	13.0390	0.0464		8
cardnum_amount_max_3	10.8681	0.0387		9
cardnum_amount_week_diff_sum_7	7.1813	0.0255		10
cardnum_amount_week_diff_max_3	5.7419	0.0204		11
cardnum_amount_month_diff_sum_15	4.9791	0.0177		12
merchantnum_amount_month_diff_max_1	4.8036	0.0171		13
merchantnum_amount_avg_1	4.6146	0.0164		14
cardnum_amount_max_1	4.3300	0.0154		15
cardnum_amount_avg_3	4.1590	0.0148		16
cardnum_amount_avg_1	4.1219	0.0147		17
merchantnum_amount_max_3	3.8105	0.0136		18
merchantnum_amount_month_diff_avg_3	3.7181	0.0132		19
merchantnum_amount_week_diff_avg_1	3.6167	0.0129		20
cardnum_amount_week_diff_sum_15	3.1844	0.0113		21
cardnum_amount_month_diff_avg_15	2.8551	0.0102		22
cardnum_amount_month_diff_avg_1	2.6148	0.0093		23
cardnum_amount_avg_7	1.9606	0.0070		24
cardnum_amount_month_diff_avg_7	1.0637	0.0038		25

ROC Curve – Bootstrap forest



Neural Network (Nodes: 25, 10)

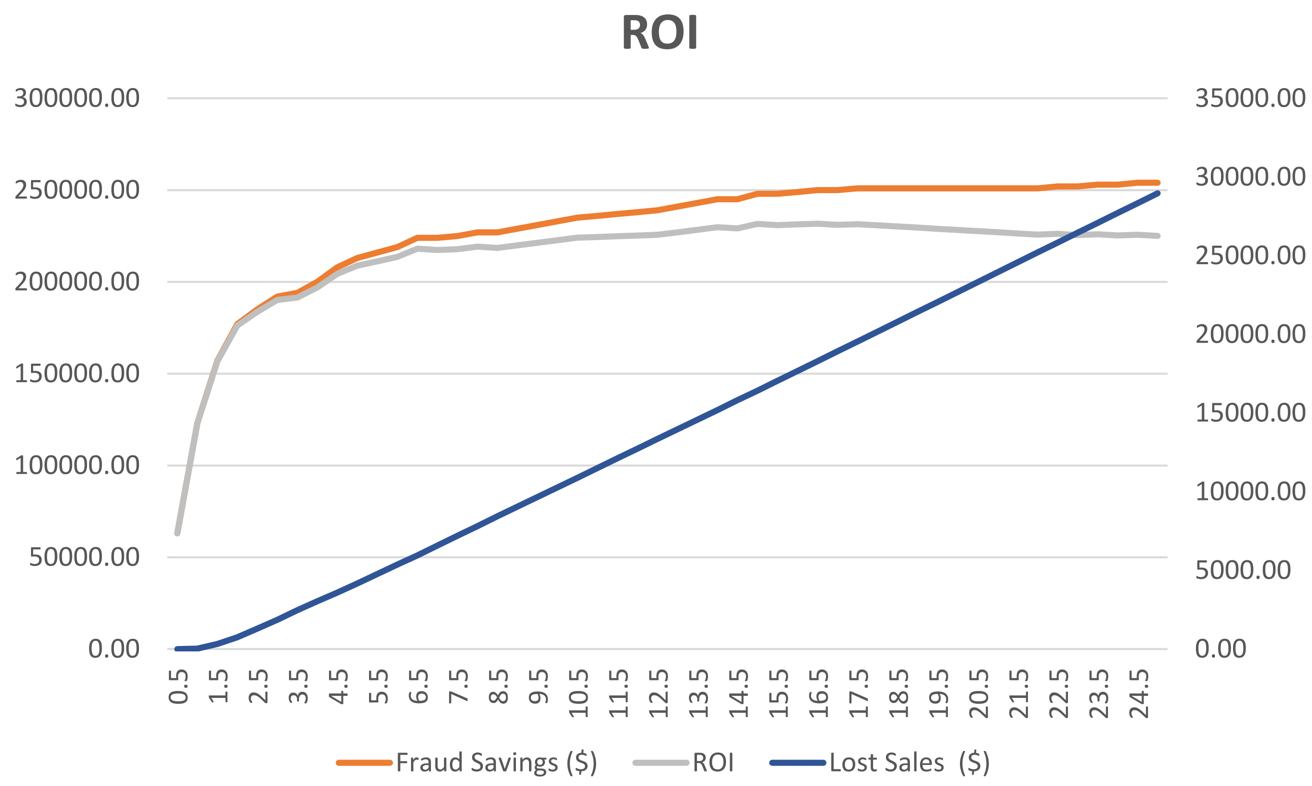


Bin Table

Overall Bad Rate is 2.69%	Bin Statistics					Cumulative Statistics					
Population Bin %	Total # records	# Bad	# Good	% Bad	% Good	Cumulative Bad	Cumulative Good	% Bad (FDR)	% Good	KS	False Pos. Ratio
0.5	63.000	63.000	-	100.000	-	63.000	-	18.639	-	18.639	-
1	63.000	60.000	3.000	95.238	4.762	123.000	3.000	36.391	0.024	36.366	0.024
1.5	63.000	34.000	29.000	53.968	46.032	157.000	32.000	46.450	0.261	46.188	0.204
2	63.000	20.000	43.000	31.746	68.254	177.000	75.000	52.367	0.612	51.755	0.424
2.5	63.000	8.000	55.000	12.698	87.302	185.000	130.000	54.734	1.061	53.672	0.703
3	63.000	7.000	56.000	11.111	88.889	192.000	186.000	56.805	1.519	55.286	0.969
3.5	63.000	2.000	61.000	3.175	96.825	194.000	247.000	57.396	2.017	55.380	1.273
4	63.000	6.000	57.000	9.524	90.476	200.000	304.000	59.172	2.482	56.690	1.520
4.5	63.000	8.000	55.000	12.698	87.302	208.000	359.000	61.538	2.931	58.607	1.726
5	63.000	5.000	58.000	7.937	92.063	213.000	417.000	63.018	3.405	59.613	1.958
5.5	63.000	3.000	60.000	4.762	95.238	216.000	477.000	63.905	3.895	60.011	2.208
6	63.000	3.000	60.000	4.762	95.238	219.000	537.000	64.793	4.384	60.409	2.452
6.5	63.000	5.000	58.000	7.937	92.063	224.000	595.000	66.272	4.858	61.414	2.656
7	63.000	-	63.000	-	100.000	224.000	658.000	66.272	5.372	60.900	2.938
7.5	63.000	1.000	62.000	1.587	98.413	225.000	720.000	66.568	5.879	60.690	3.200
8	63.000	2.000	61.000	3.175	96.825	227.000	781.000	67.160	6.377	60.783	3.441
8.5	63.000	-	63.000	-	100.000	227.000	844.000	67.160	6.891	60.269	3.718
9	63.000	2.000	61.000	3.175	96.825	229.000	905.000	67.751	7.389	60.363	3.952
9.5	63.000	2.000	61.000	3.175	96.825	231.000	966.000	68.343	7.887	60.456	4.182
10	63.000	2.000	61.000	3.175	96.825	233.000	1,027.000	68.935	8.385	60.550	4.408
10.5	63.000	2.000	61.000	3.175	96.825	235.000	1,088.000	69.527	8.883	60.644	4.630
11	63.000	1.000	62.000	1.587	98.413	236.000	1,150.000	69.822	9.389	60.433	4.873
11.5	63.000	1.000	62.000	1.587	98.413	237.000	1,212.000	70.118	9.895	60.223	5.114
12	63.000	1.000	62.000	1.587	98.413	238.000	1,274.000	70.414	10.402	60.013	5.353
12.5	63.000	1.000	62.000	1.587	98.413	239.000	1,336.000	70.710	10.908	59.802	5.590

Conclusion

- Assume $1000 loss for every fraud that's not caught
- Assume $10 loss for every false positive (good that's flagged as a bad)



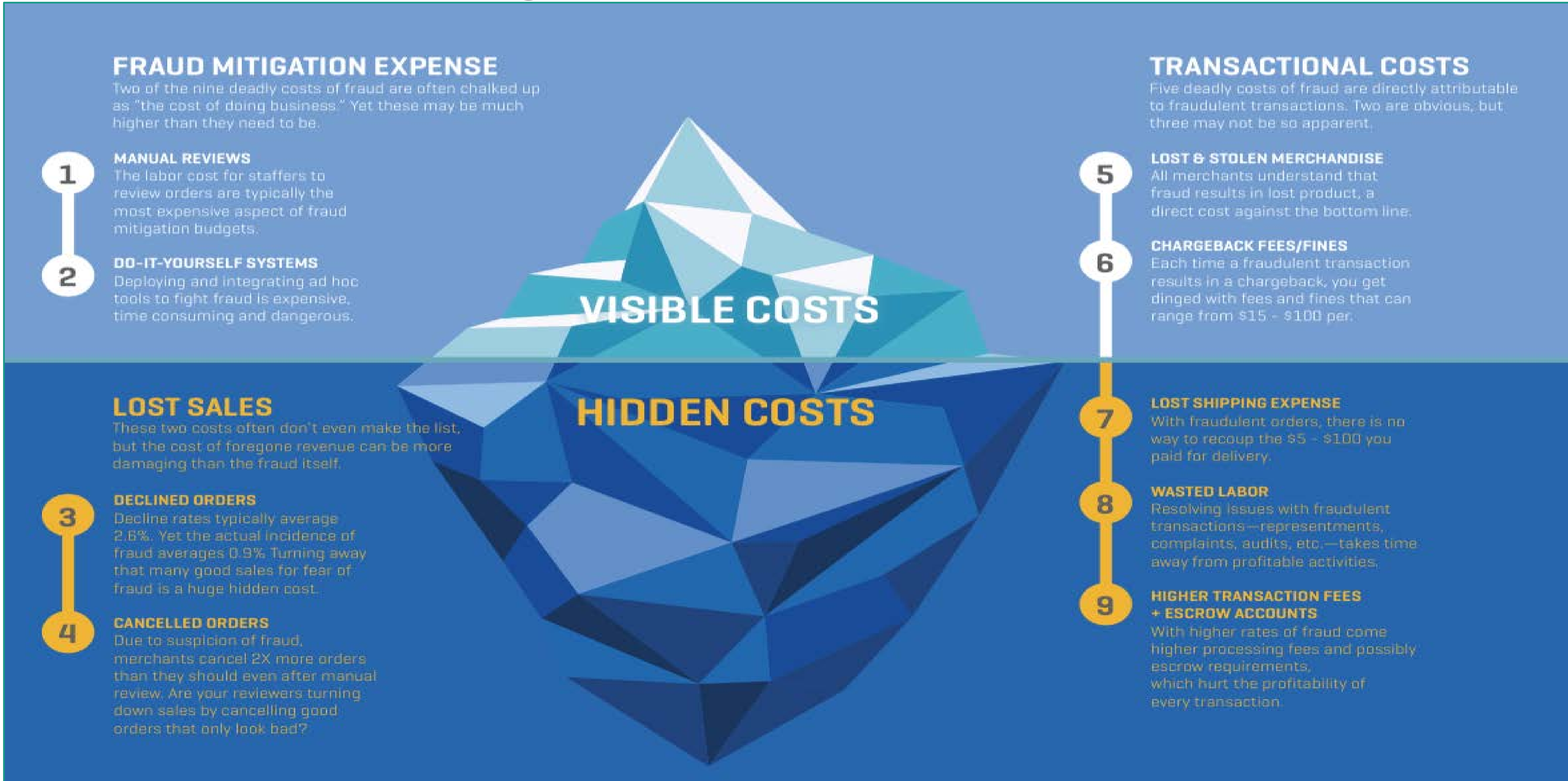
$237,000

12,586

Cut off: 16.5%

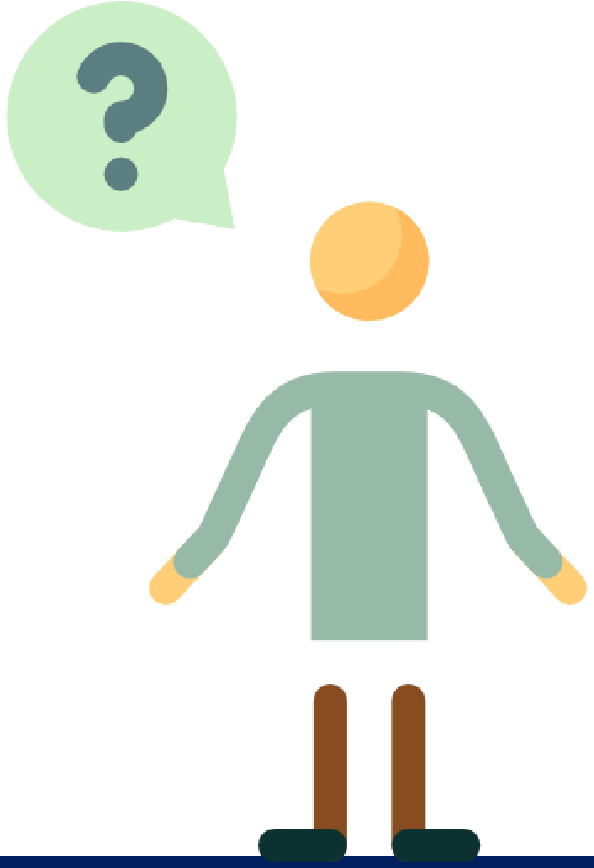
Business Insights

Business Insight

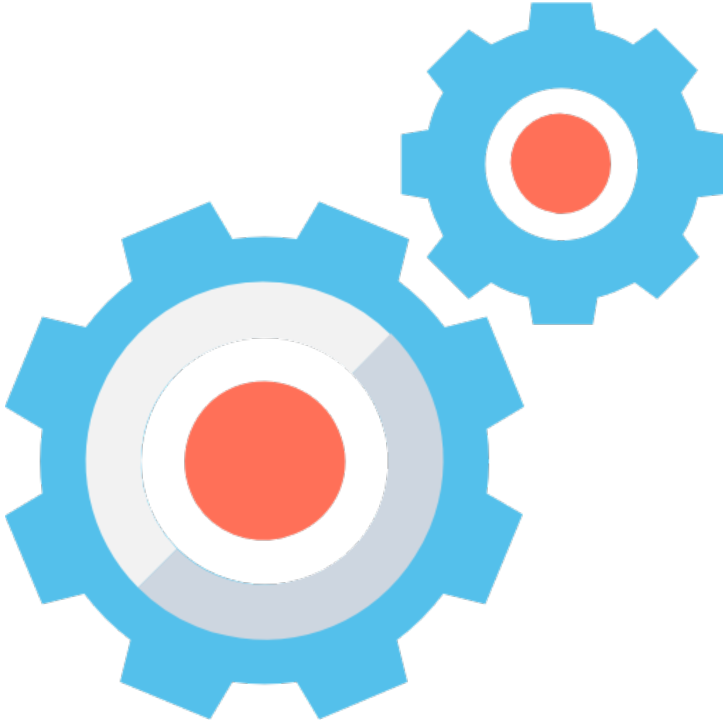


Thank You!

Q&A



Expert Variable Creation



We created 130 expert variables and we kept 40 variables after KS.

Since this analysis involves time, with limited data, we chose four different time windows 1, 3, 7, 15 and 30 days. The rationale is to capture more (and different types of) fraudulent records that might be detected in those time windows.

Furthermore, we kept 25 variables after Lasso, we use these 25 to train our models.