Credit Card Transaction Fraud Detection

DSO 562 Fraud Analytics Project 3

GU,XINXUE(SUE) SUN,YANG WANG,BUFAN

XIE,BINGHONG YANG,SIQIN ZHAO,ZIHAO

USC Marshall School of Business

USC Marshall
School of Business

Team Member

Sue Gu
Business Analyst

Peter Sun
Business Analyst

Bufan Wang
Business Analyst

Binghong Xie
Business Analyst

Siqin Yang
Business Analyst

Zihao Zhao
Business Analyst

Objective and Executive Summary

Objective:

- Predicting credit card transaction fraud label and maximize Fraud Detection Rate (FDR). Choosing an FDR cutoff to maximize money savings.

Dataset:

- Source: real credit transaction data purchased from a US government organization.
- Shape: 96753 rows, 10 columns (1059 fraud records)
- Time: 2010-01-01 ~ 2010-12-31

Approach:

- Built 516 variables and selected 30 variables that have the highest predictive power.
- Attempted 6 machine learning algorithms and rigorously tuned the hyperparameters.
- Used k-fold cross-validation to address issues caused by small data size.

Result:

Fraud Detection Rate at 5% :

- 94.9% on training
- 92.69% on testing
- 59.22% on OOT
- Financial Savings $185000 in 2 months

USC Marshall
School of Business

Agenda

01 DQR & EDA

02 Data Cleaning

03 Feature Engineering

04 Feature Selection

05 Model Exploration

06 Final Model

USC Marshall
School of Business

Data Quality Report & Exploratory Data Analysis

01

Data Quality Summary

Datetime:

	Name	dtypes	#of Records	% populated	# NA	# Unique Values	Maximum	Minimum	% Most Common Field	Most Common Field	Entropy
0	Date	datetime64[ns]	96753	100.0	0	365	2010-12-31	2010-01-01	0.71	2010-02-28	8.21

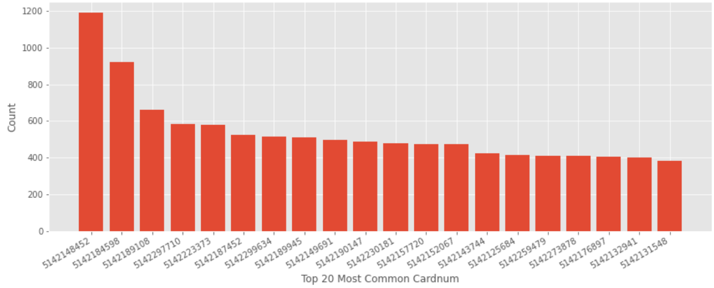
Categorical:

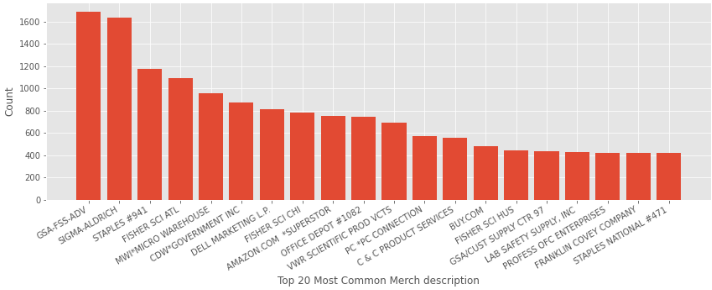
	Name	dtypes	#of Records	% populated	# NA	# Unique Values	First_Value	Second_Value	Third_Value	% Most Common Field	Most Common Field	Entropy
1	Cardnum	object	96753	100.0	0	1645	5142190439	5142183973	5142131721	1.23	[5142148452]	9.73
2	Merchnum	object	93378	96.51	3375	13091	5509006296254	61003026333	4503082993600	9.97	[930090121224]	10.41
3	Merch description	object	96753	100.0	0	13126	FEDEX SHP 12/23/09 AB#	SERVICE MERCHANDISE #81	OFFICE DEPOT #191	1.74	['GSA-FSS-ADV']	11.19
4	Merch state	object	95558	98.76	1195	227	TN	MA	MD	12.59	[TN]	4.7
5	Merch zip	object	92097	95.19	4656	4567	38118	1803	20706	12.27	[38118]	8.86
6	Transtype	object	96753	100.0	0	4	P	P	P	99.63	[P]	0.04
7	Fraud	object	96753	100.0	0	2	0	0	0	98.91	[0]	0.09

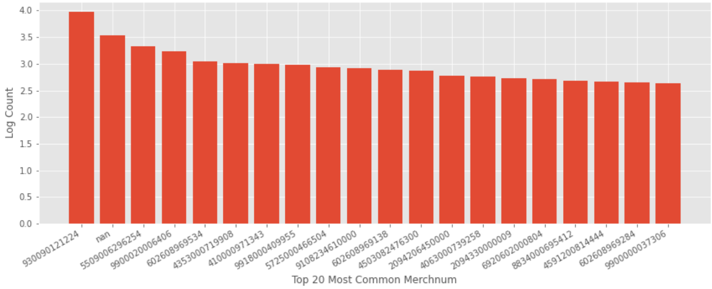
Numeric:

	Name	dtypes	# of Records	% populated	# NA	# Zeros	Missing (NA+Zero)	% Missing	Uniques	Mean	Maximum	Minimum	Standard Deviation	Entropy
0	Amount	float64	96753	100.0	0	0	0	0.0	34909	427.89	3102045.53	0.01	10006.14	13.28

USC Marshall School of Business



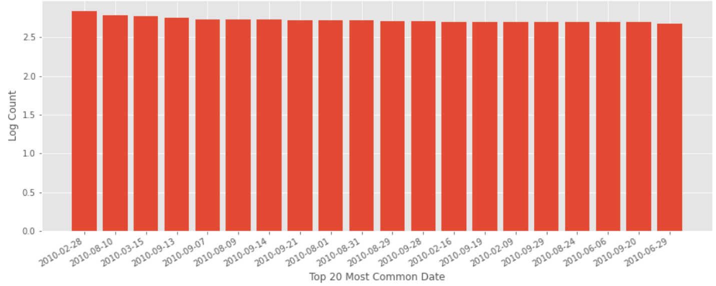


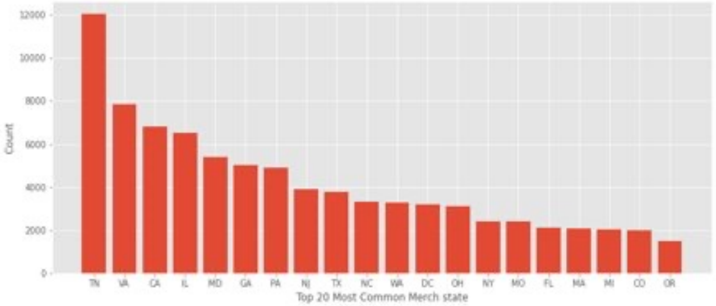


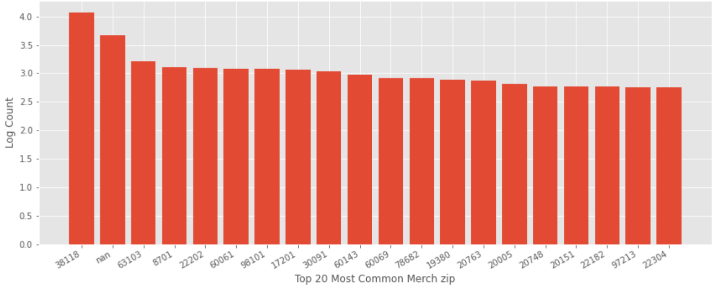
Univariate:

Distibution of top 20 most common value in:

- Cardnum
- Merch Description
- Merchnum





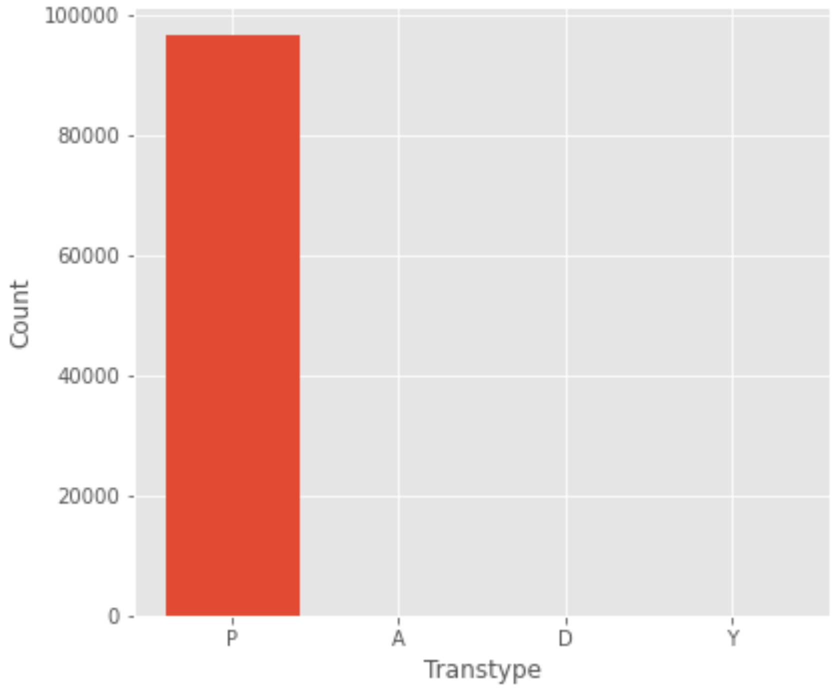


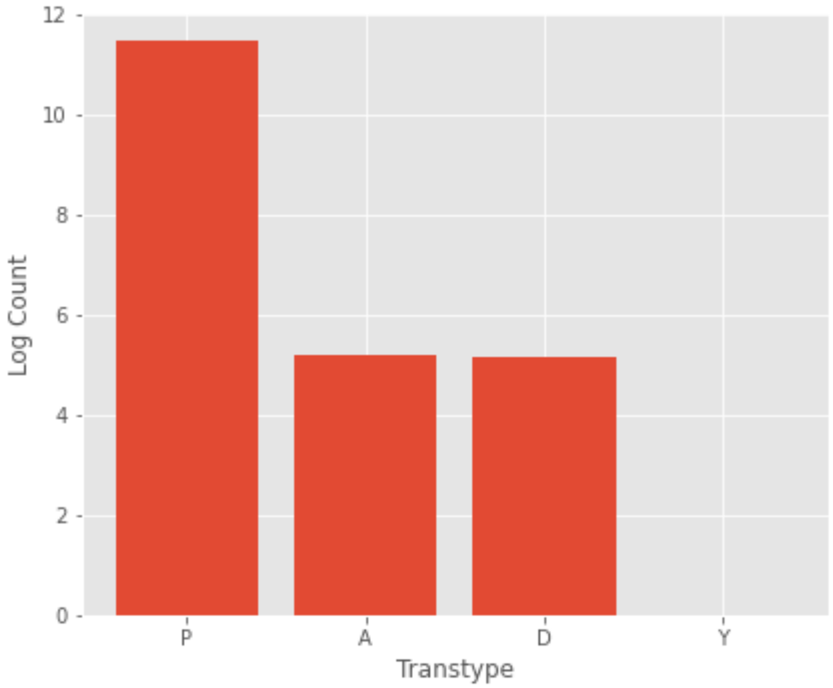
Univariate:

Distribution of top 20 most common value in:

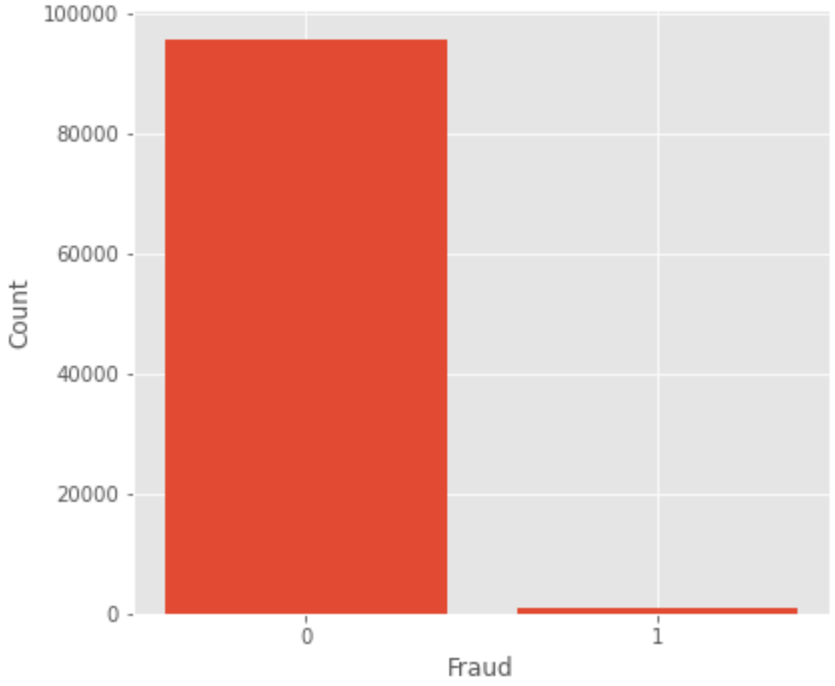
- Date
- Merch state
- Merch zip

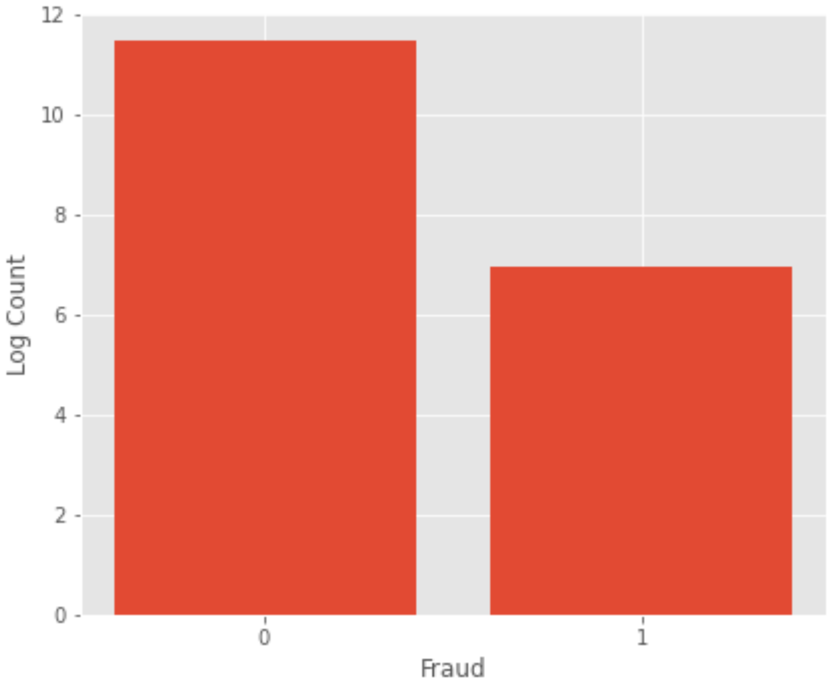
Univariate: Transtype



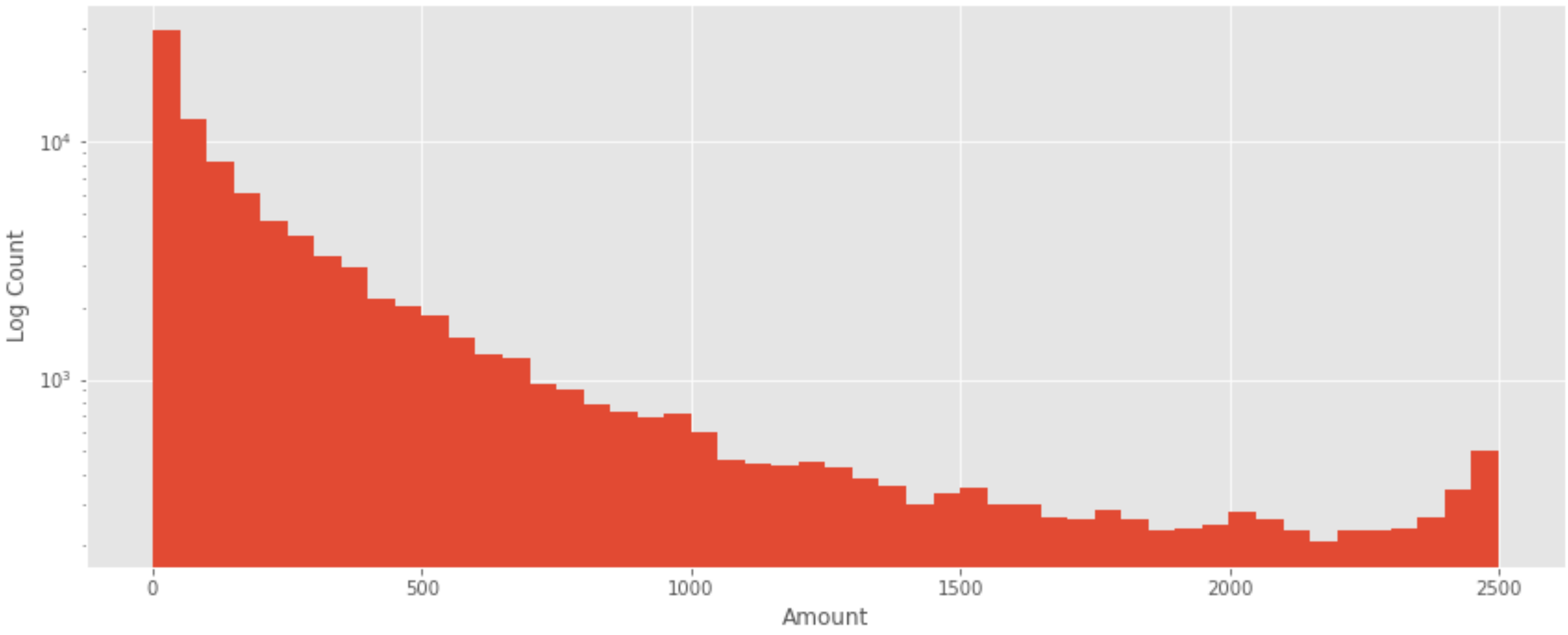


Univariate: Fraud

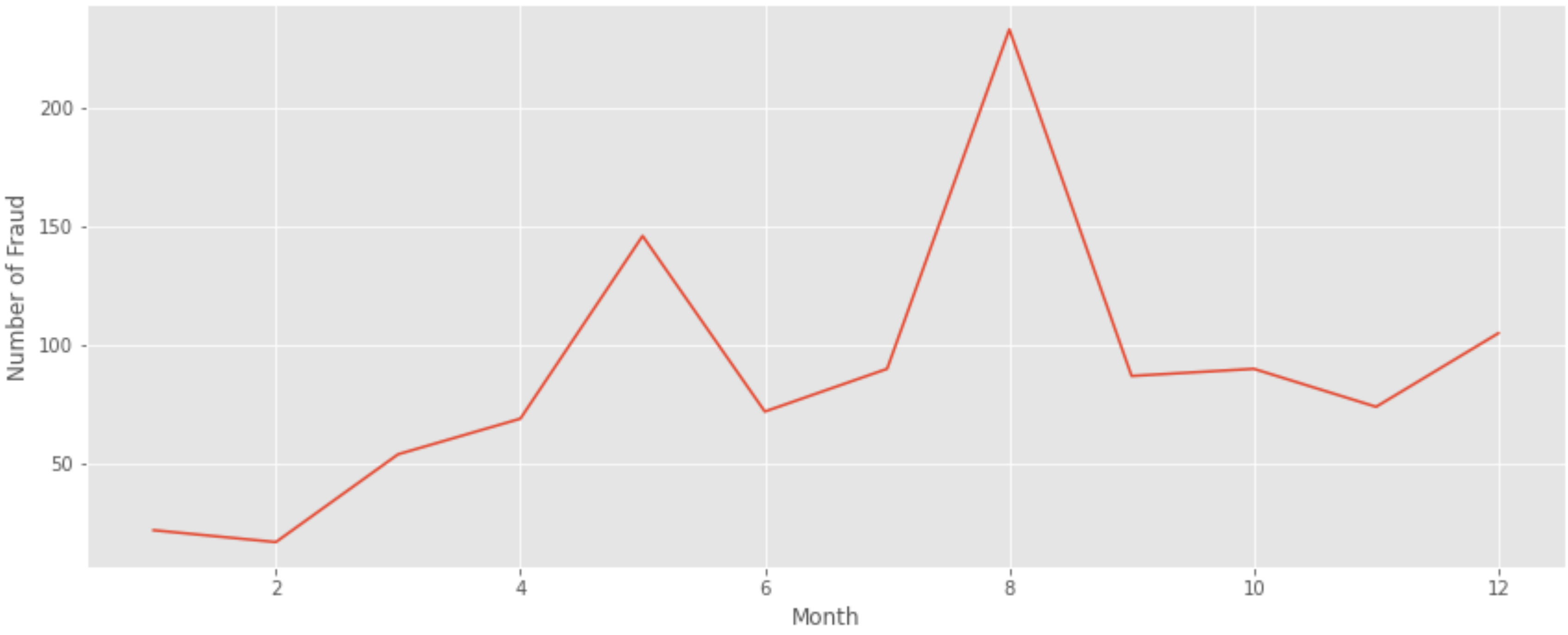




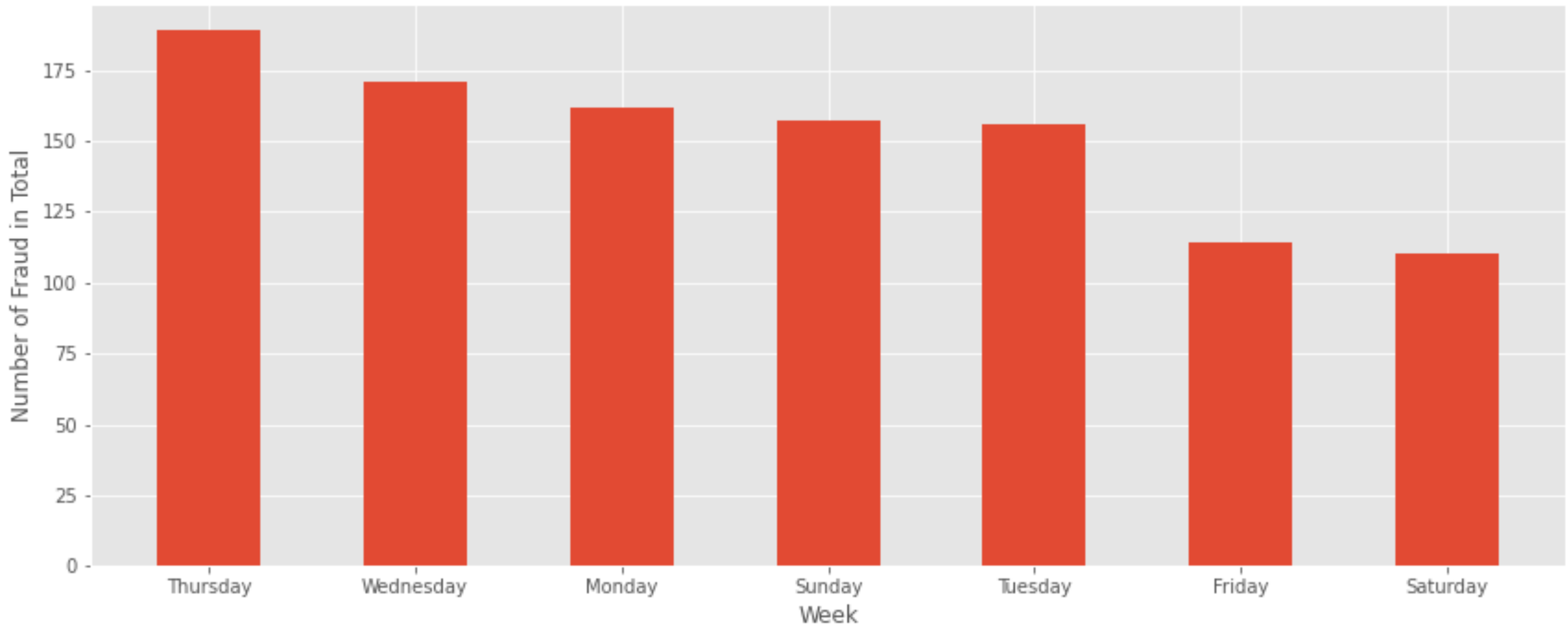
Univariate: Amount



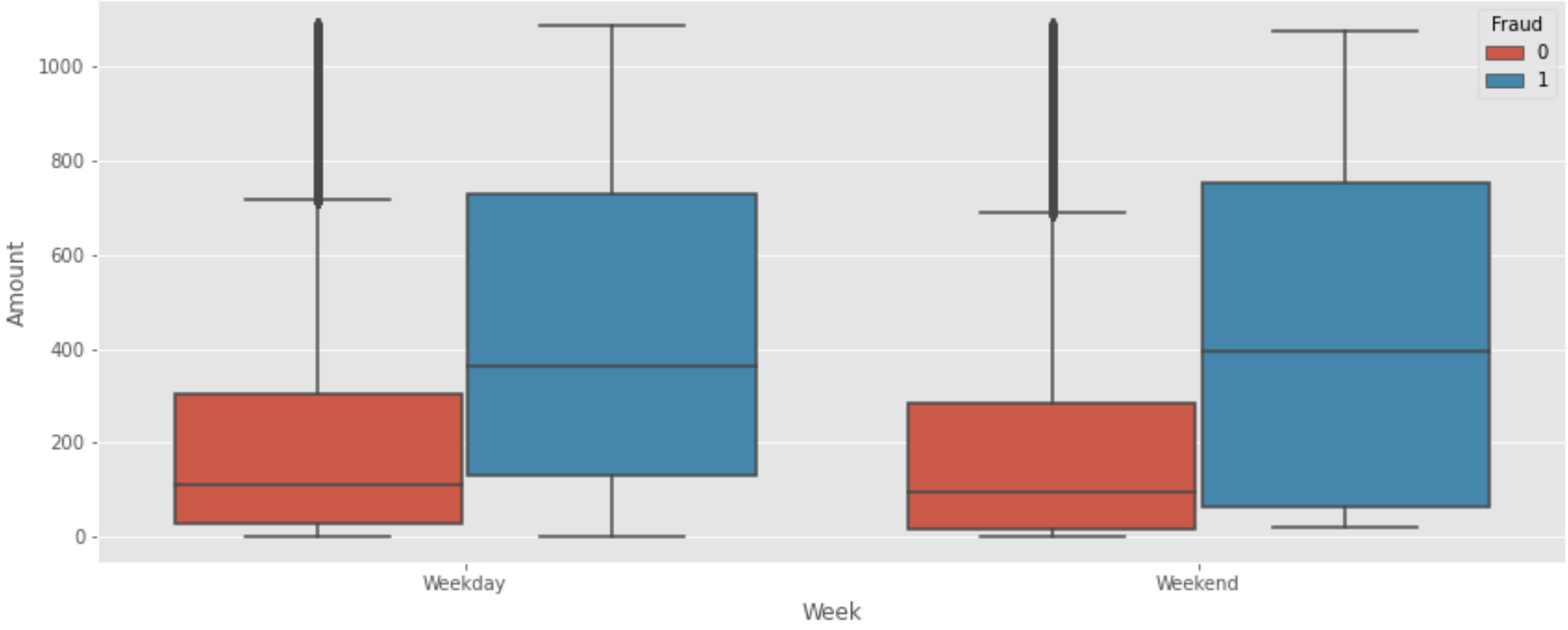
Bivariate: Fraud by Month



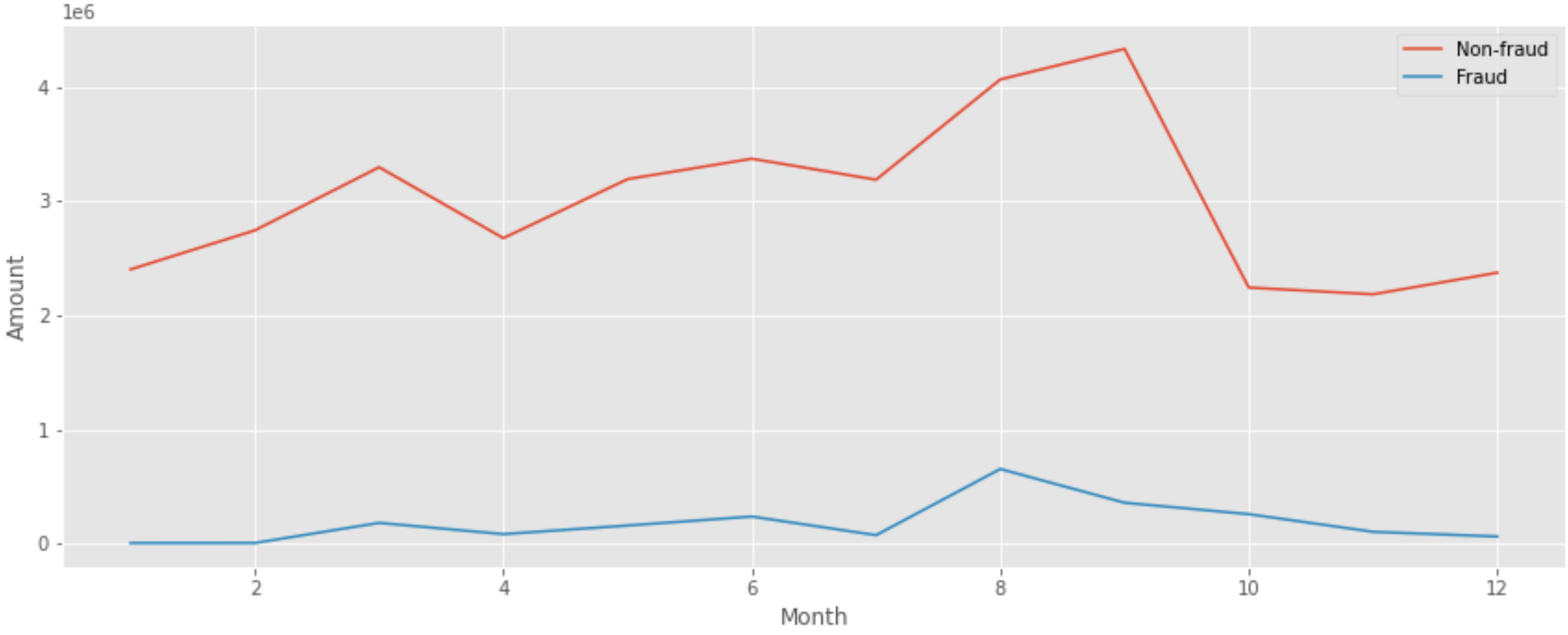
Bivariate: Fraud by Week



Multivariate: Amount by Week & Fraud



Multivariate: Amount by Month & Fraud



USC Marshall School of Business

Data Cleaning

02

Removing Outliers

Only one outlier:

	Recnum	Cardnum	Date	Merchnum	Merch description	Merch state	Merch zip	Transtype	Amount	Fraud
52714	52715	5142189135	2010-07-13	NaN	INTERMEXICO	NaN	NaN	P	3102045.53	0

Solved by directly removing it from the dataset

Filling in Missing Values

Filling Logic: Use the average or most common value of that field over a relevant subset of records

- Most relevant field
- Second most relevant field

Group into categories, replace the missing field with the average or most common value for its appropriate group

- Values from previous row (Because values in those fields have a pattern)

Most relevant field > Second most relevant field > Values from previous row

Filling in Merchnum

First, use Merch description to fill in missing values:

`2038 missing Merchnum values left`

Then, use Cardnum to fill in missing values:

`57 missing Merchnum values left`

Last, use values from the previous row:

`No missing value`

Filling in Merch state

First, use Merch description to fill in missing values:

```
363 missing Merchnum values left
```

Then, use values from the previous row:

```
No missing value
```

Filling in Merch zip

First, use Merch description to fill in missing values:

```
2043 missing Merchnum values left
```

Then, use Cardnum to fill in missing values:

```
42 missing Merchnum values left
```

Last, use values from the previous row:

```
No missing value
```

USC Marshall
School of Business

Feature Engineering

03

Basic Ideas

- Looking for unusual entity's repetitive pattern.
- Looking for unusual transaction amount.
- Looking for unusual entity's frequency.

Create Entity

- Total 6 entities:
  - Cardnum
  - Merchnum
  - Cardnum + merch_description
  - Cardnum + merchnum
  - Cardnum + zip
  - Cardnum + State

USC Marshall
School of Business

Create Variables

Amount Variables	Average Maximum Median Minimum Total Range (max – min) Range Percent (max-min)/min Actual/average Actual/maximum Actual/median Actual/total	amount by/at	6 different entities	over the past	0 days 1 day 3 days 7 days 14 days 30 days			11 x 6 x 6= 396
Frequency Variables	Number of transactions with this	6 different entities	over the past	0 days 1 day 3 days 7 days 14 days 30 days				6 x 6=36
Days-since Variables	Current date minus date of most recent transaction with same	6 different entities						6
Velocity change Variables	Number Amount	of transactions with same	6 different entities	over the past	0 days 1 day			6 x 2 x 3 x 2= 72
	Average daily	number amount	of transactions with same	6 different entities	over the past	7 days 14 days 30 days		Use same entity in both numerator and denominator
Unique # of Merchnum for each Cardnum	over the past	0 days 1 day 3 days 7 days 14 days 30 days						6

Total 516 variables created

Feature Selection

04

Basic Ideas

Use univariate KS score, and FDR@3% to select top 80 variables out of 516 variables.

Use backward feature selection to select 30 variables out of 80 variables.

Backward Feature Selection

- Use backward selection method.
- Model: Logistic regression.
- Scoring: FDR at 3%.
- Selected 30 final variables.

Model Exploration

- Logistic Regression
- Neural Network
- Random Forest
- Gradient Boosting Classifier
- Extreme Gradient Boosting
- Light Gradient Boosting Machine

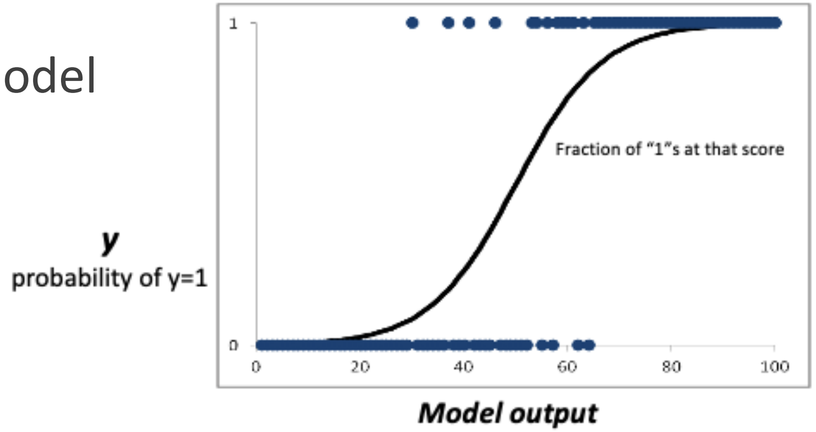
05

Deal with Small Data Size --
15-time Random Split

- Dataset: 96,753 records.
- Conduct 15 times random split to improve model robustness.
  - One random split may be biased
- Implementation:
  - For each set of hyperparameters of each model
  - Randomly split the dataset into training and testing data 15 times
  - Take the average FDR at 3% as the evaluation metric

Logistic Regression

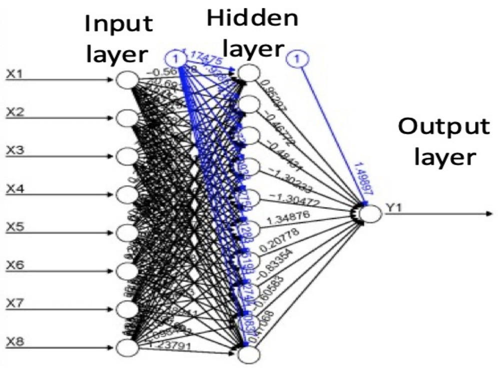
- A statistical model that uses a logistic function to model a binary dependent variable.
- Serves as a baseline.



Logistic Regression	penalty	Train	Test	OOT
1	l2	64.0%	64.0%	36.0%
2	None	64.0%	64.0%	36.0%

Neural Network

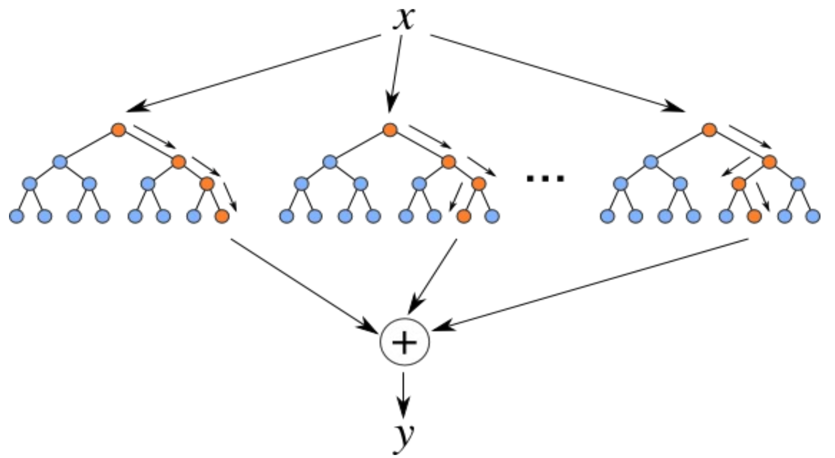
- Set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns.
- Use a simple multilayer perceptron (MLP) to identify fraud.



Neural Network	hidden_layer _sizes	activation	learning _rate	learning _rate_init	alpha	Train	Test	OOT
1	(20,)	relu	constant	0.0001	1e-5	69.8%	69.8%	50.3%
2	(5,10)	tanh	invscaling	0.001	8e-4	74.7%	74.4%	58.7%
3	(10,5)	tanh	invscaling	0.001	8e-4	76.5%	74.9%	58.1%
4	(10,10)	tanh	invscaling	0.001	8e-4	76.4%	75.2%	58.7%
5	(10, 10)	relu	constant	0.001	1e-4	78.4%	77.3%	59.8%
6	(10,20)	relu	constant	0.001	1e-4	78.4%	76.3%	57.5%

Random Forest

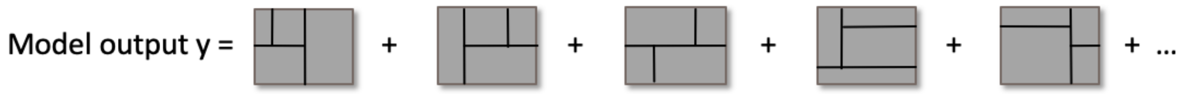
- An ensemble learning method for classification that operate by constructing multitude of decision trees.
- Pros:
  - Flexible, easy to use
  - Higher accuracy than a single decision tree



Random Forest	n_estimators	max_leaf_nodes	criterion	Train	Test	OOT
1	100	5	gini	68.7%	67.3%	44.5%
2	150	10	entropy	69.4%	68.3%	45.1%
3	200	5	entropy	67.8%	67.2%	44.2%
4	250	10	gini	70.2%	68.8%	47.6%
5	300	5	gini	67.8%	67.3%	44.4%
6	350	10	entropy	70.2%	68.8%	48.8%
7	400	5	entropy	67.9%	67.3%	44.2%

Gradient Boosting Classifier

- Machine learning technique that produces predictions via ensemble of weak prediction models.
- Generally outperforms random forest.



Boosted tree
Each additional model makes the overall model slightly better

Gradient Boost	n_estimators	learning_rate	max_depth	min_samples_split	Train	Test	OOT
1	150	0.01	5	2	71%	63%	16.8%
2	150	0.025	5	2	83%	65%	17.9%
3	150	0.05	5	2	90%	57%	16.8%
4	200	0.025	5	2	87%	65%	17.3%
5	250	0.025	5	2	88%	67%	17.3%
6	250	0.025	5	4	88%	66%	17.3%
7	300	0.025	5	2	90%	66%	16.8%

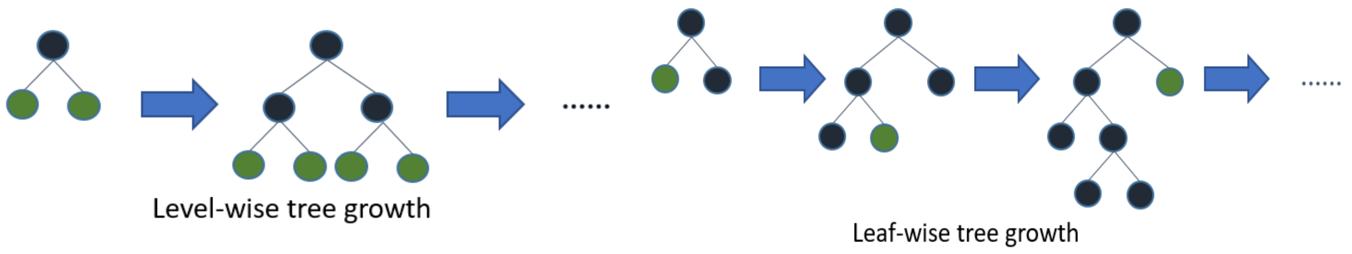
Extreme Gradient Boosting

- Implementation of the Gradient Boosting method which uses more accurate approximations to find the best tree model.
- Compute second order gradients of loss function that provides more information about the direction of gradients and how to get to the minimum of loss function.
- More regularized form of Gradient Boosting. Uses advanced regularization , which improves model generalization capabilities.

XGBoost	n_estimators	min_child_weight	learning_rate	Train	Test	OOT
1	100	20	0.1	85.6%	76.6%	44.7%
2	150	50	0.3	86.8%	76.8%	46.4%
3	200	40	0.4	89.6%	77.6%	45.8%
4	300	60	0.4	87.9%	75.5%	49.7%
5	300	80	0.3	82.8%	74.3%	49.2%
6	400	50	0.5	92.1%	75.8%	48.6%

Light Gradient Boosting Machine

- Leaf-wise tree instead of level-wise tree. Chooses the number of leaves that yield the largest decrease in loss.



Light GBM	n_estimators	learning_rate	max_depth	num_leaves	Train	Test	OOT
1	200	0.05	3	6	75.4%	50.2%	32.5%
2	300	0.01	4	12	76.5%	65.2%	39.5%
3	400	0.01	4	12	80.7%	73.5%	39.8%
4	500	0.01	5	8	87.0%	82.3%	42.6%
5	600	0.01	5	30	93.5%	89.3%	39.3%
6	700	0.01	6	30	95.8%	87.7%	34.2%
7	800	0.05	4	18	96.5%	88.5%	37.1%

Final Model

06

Final Model

- Choose Light GBM as final choice of model; further tuning hyperparameters
- Train model on all modeling data (training + testing)

Final hyperparameters

Model	n_estimators	max_depth	num_leaves	learning_rate
Light GBM	900	3	12	0.01

Result Table

Final Model Result

Data	FDR @ 3%
Training	91.12%
Testing	88.85%
OOT	50.28%

- 50.28% FDR@3% on OOT
- Low performance on OOT compared to Training and Testing

OOT	# Records		# Goods		# Bads		Fraud Rate					
	12427		12248		179		0.01440412					
	Bins Statistics					Cumulative Statistics						
Population bin %	# Record	# Goods	# Bads	% Goods	% Bads	Total # of records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	125	61	64	48.80%	51.20%	125	61	64	0.50%	35.75%	35.26	0.95
2	125	101	24	80.80%	19.20%	250	162	88	1.32%	49.16%	47.84	1.84
3	125	123	2	98.40%	1.60%	375	285	90	2.33%	50.28%	47.95	3.17
4	125	117	8	93.60%	6.40%	500	402	98	3.28%	54.75%	51.47	4.10
5	125	117	8	93.60%	6.40%	625	519	106	4.24%	59.22%	54.98	4.90
6	125	121	4	96.80%	3.20%	750	640	110	5.23%	61.45%	56.23	5.82
7	125	121	4	96.80%	3.20%	875	761	114	6.21%	63.69%	57.47	6.68
8	125	123	2	98.40%	1.60%	1000	884	116	7.22%	64.80%	57.59	7.62
9	125	123	2	98.40%	1.60%	1125	1007	118	8.22%	65.92%	57.70	8.53
10	125	122	3	97.60%	2.40%	1250	1129	121	9.22%	67.60%	58.38	9.33
11	125	122	3	97.60%	2.40%	1375	1251	124	10.21%	69.27%	59.06	10.09
12	125	124	1	99.20%	0.80%	1500	1375	125	11.23%	69.83%	58.61	11.00
13	125	121	4	96.80%	3.20%	1625	1496	129	12.21%	72.07%	59.85	11.60
14	125	124	1	99.20%	0.80%	1750	1620	130	13.23%	72.63%	59.40	12.46
15	125	121	4	96.80%	3.20%	1875	1741	134	14.21%	74.86%	60.65	12.99
16	125	123	2	98.40%	1.60%	2000	1864	136	15.22%	75.98%	60.76	13.71
17	125	124	1	99.20%	0.80%	2125	1988	137	16.23%	76.54%	60.31	14.51
18	125	124	1	99.20%	0.80%	2250	2112	138	17.24%	77.09%	59.85	15.30
19	125	120	5	96.00%	4.00%	2375	2232	143	18.22%	79.89%	61.66	15.61
20	125	124	1	99.20%	0.80%	2500	2356	144	19.24%	80.45%	61.21	16.36
21	125	125	0	100.00%	0.00%	2625	2481	144	20.26%	80.45%	60.19	17.23
22	125	124	1	99.20%	0.80%	2750	2605	145	21.27%	81.01%	59.74	17.97
23	125	125	0	100.00%	0.00%	2875	2730	145	22.29%	81.01%	58.72	18.83
24	125	125	0	100.00%	0.00%	3000	2855	145	23.31%	81.01%	57.70	19.69
25	125	125	0	100.00%	0.00%	3125	2980	145	24.33%	81.01%	56.68	20.55

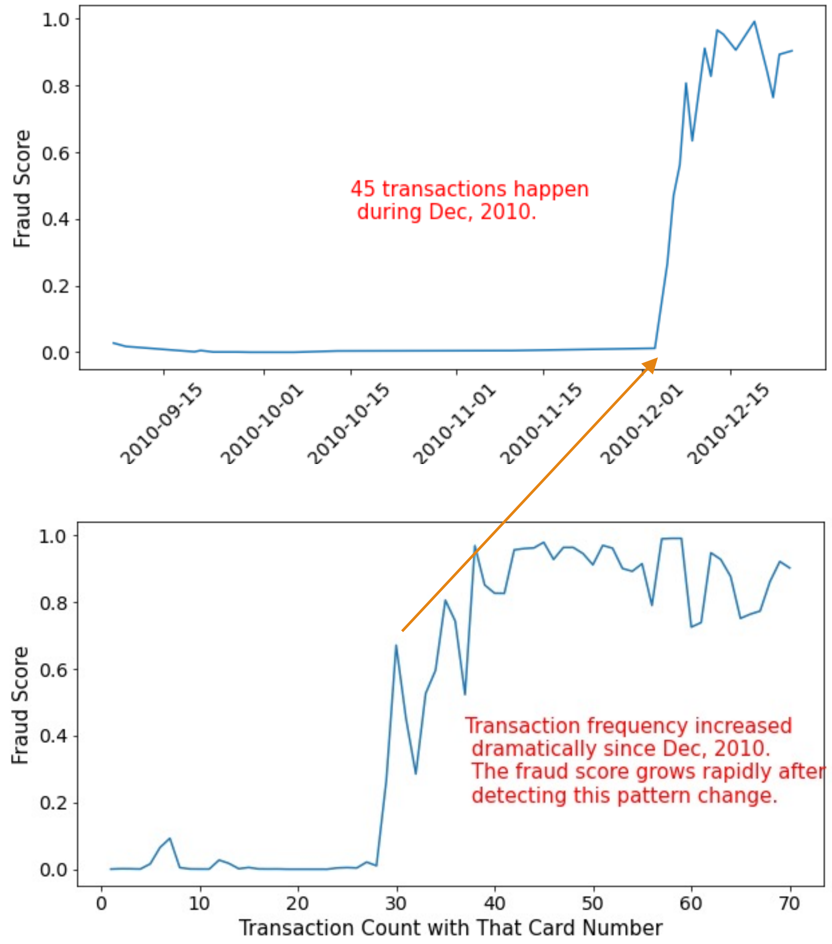
USC Marshall School of Business

Investigate OOT dataset

total_amount_over_3_for_cardnum_zip	total_amount_over_14_for_cardnum_merchnum	total_amount_over_7_for_cardnum_merchdescription	Fraud	Probability
13.617107	12.371058	13.043785	1	0.985911
12.877993	11.696042	12.542589	1	0.980776
11.063131	10.038568	10.490091	1	0.978815
13.496833	12.261215	12.907763	1	0.978815

total_amount_over_3_for_cardnum_zip	total_amount_over_14_for_cardnum_merchnum	total_amount_over_7_for_cardnum_merchdescription	Fraud	Probability
5.967984	5.385283	6.741888	0	0.985821
4.600999	4.136846	5.195911	0	0.983673
5.723749	5.162229	6.465673	0	0.979979
7.243376	6.550071	8.184277	0	0.979722
8.763003	7.937912	9.902882	0	0.978192

Fraud Score and Card Activities

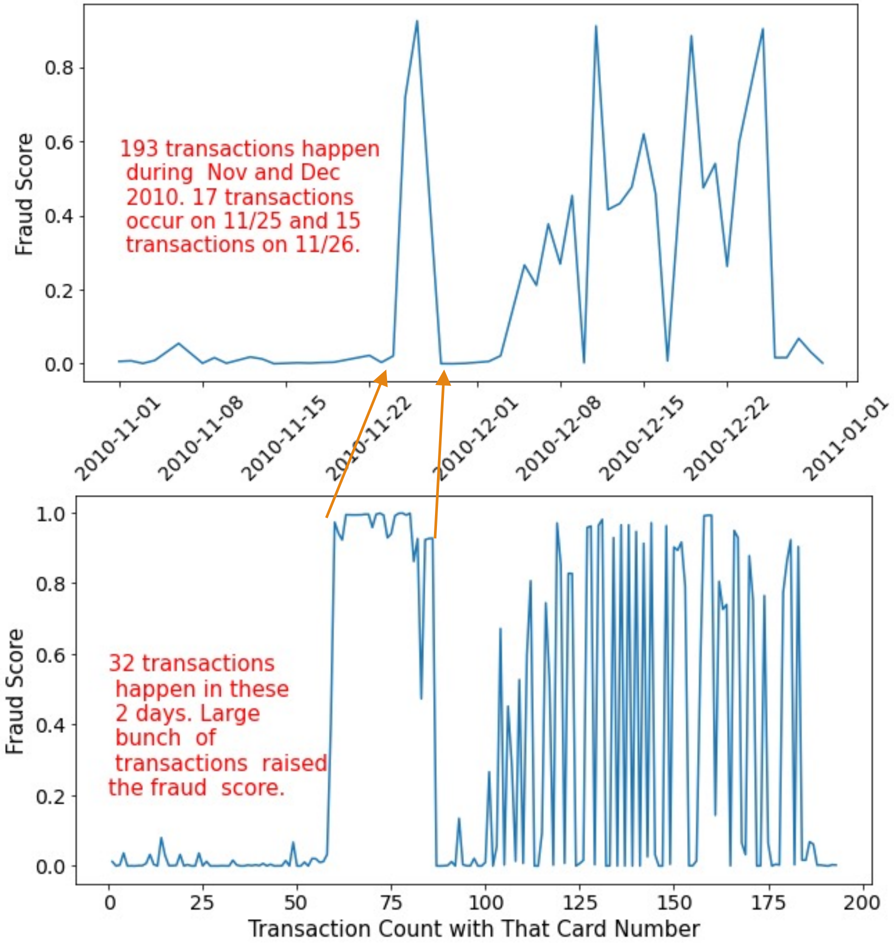


Card Number = 5142199009

Before Dec 2020: Average 2 transactions per month

After Dec 2020: 45 transactions over the month

Fraud Score and Merchant Activities



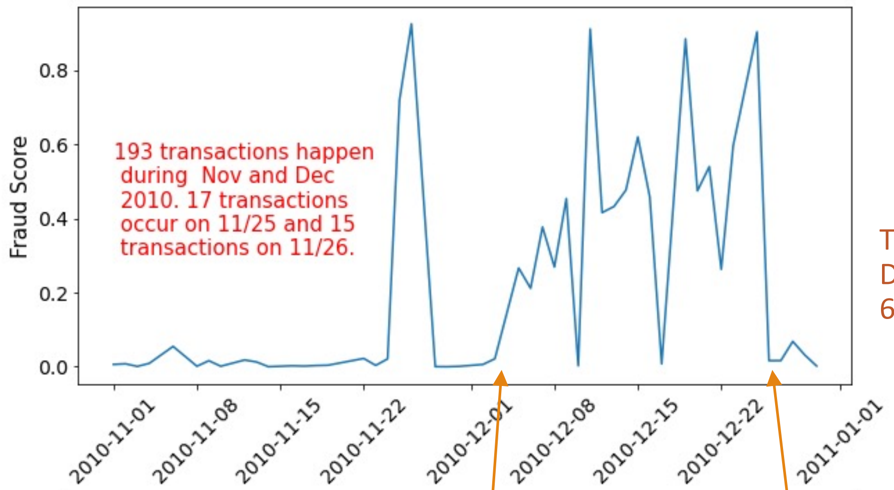
Merchant Number = 4353000719908

NOV 2010 :

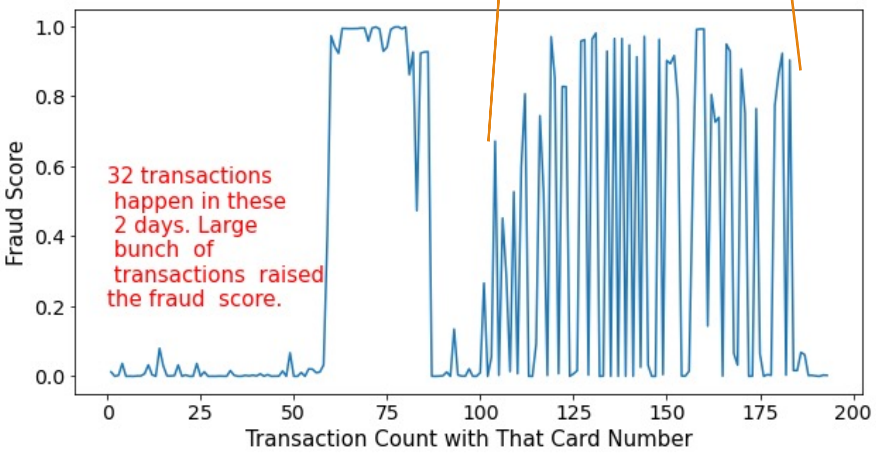
- 90 transactions in total
- 32 transactions on 11/25 and 11/26

USC Marshall School of Business

Fraud Score and Merchant Activities



There are 6 days in Dec 2010 with over 6 transactions



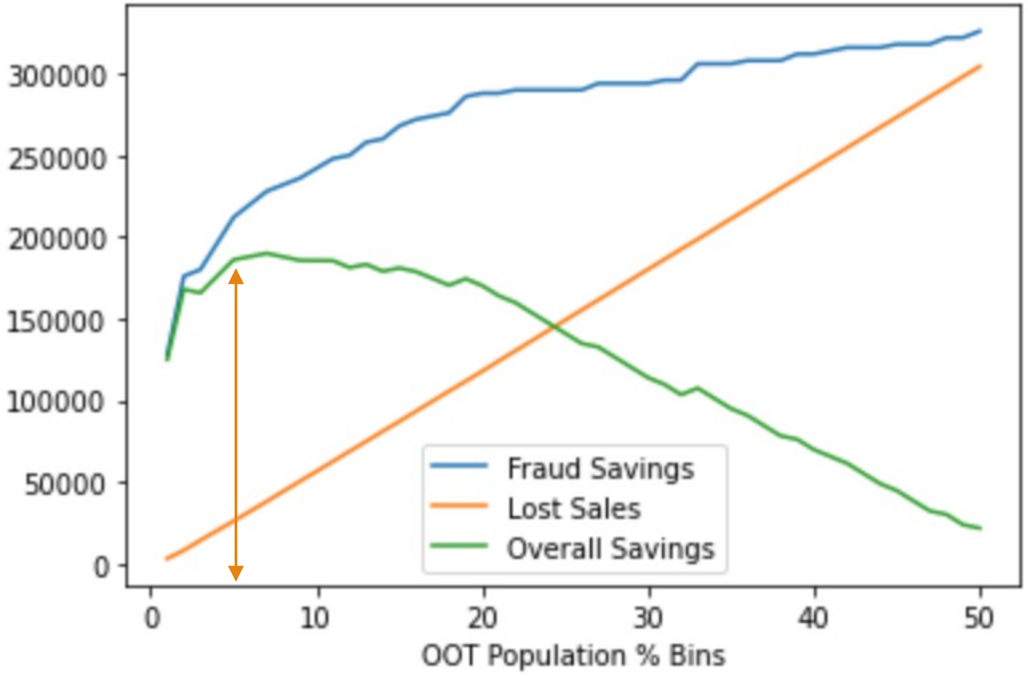
High variation in daily merchant activities leads to high variation in fraud score.

Merchant Number = 4353000719908

DEC 2010 :

- 103 transactions in total
- 3.3 transactions per day on average
- Daily merchant activities increase every few days

Fraud savings and score cutoff



We recommend a score cutoff at 5%

Assumptions:

- Saving on each fraud caught : $2,000
- Loss for each false positive : $50

Maximum overall savings > $185,000 (5% - 8%)

Conclusion

Conclusion

Steps

1. **Data Quality Check and Exploration Analysis**
   Detect missing values, outliers and fraud label imbalance
2. **Data Cleaning**
   Fill missing values using the most common value of that field over a relevant subset of records; *remove extreme outliers*
3. **Feature Engineering**
   Create 5 categories, in total 516 variables
4. **Feature Selection**
   Use filter, average FDR and KS rank to select 80 variables
   Use backward stepwise selection to select the top 30 variables
5. **Model Exploration and Selection**
   Try 6 algorithms with different hyperparameters
   Select the one with the highest FDR@3% on testing data
6. **Implement the Final Model**
   Fit the final model with all the data available
   Select FDR cut-off of 5% to maximize the financial benefits

Final Model result

FDR@5 %	Train	Test	OOT	Financial Saving *
Result	94.90%	92.69%	59.22%	$185000

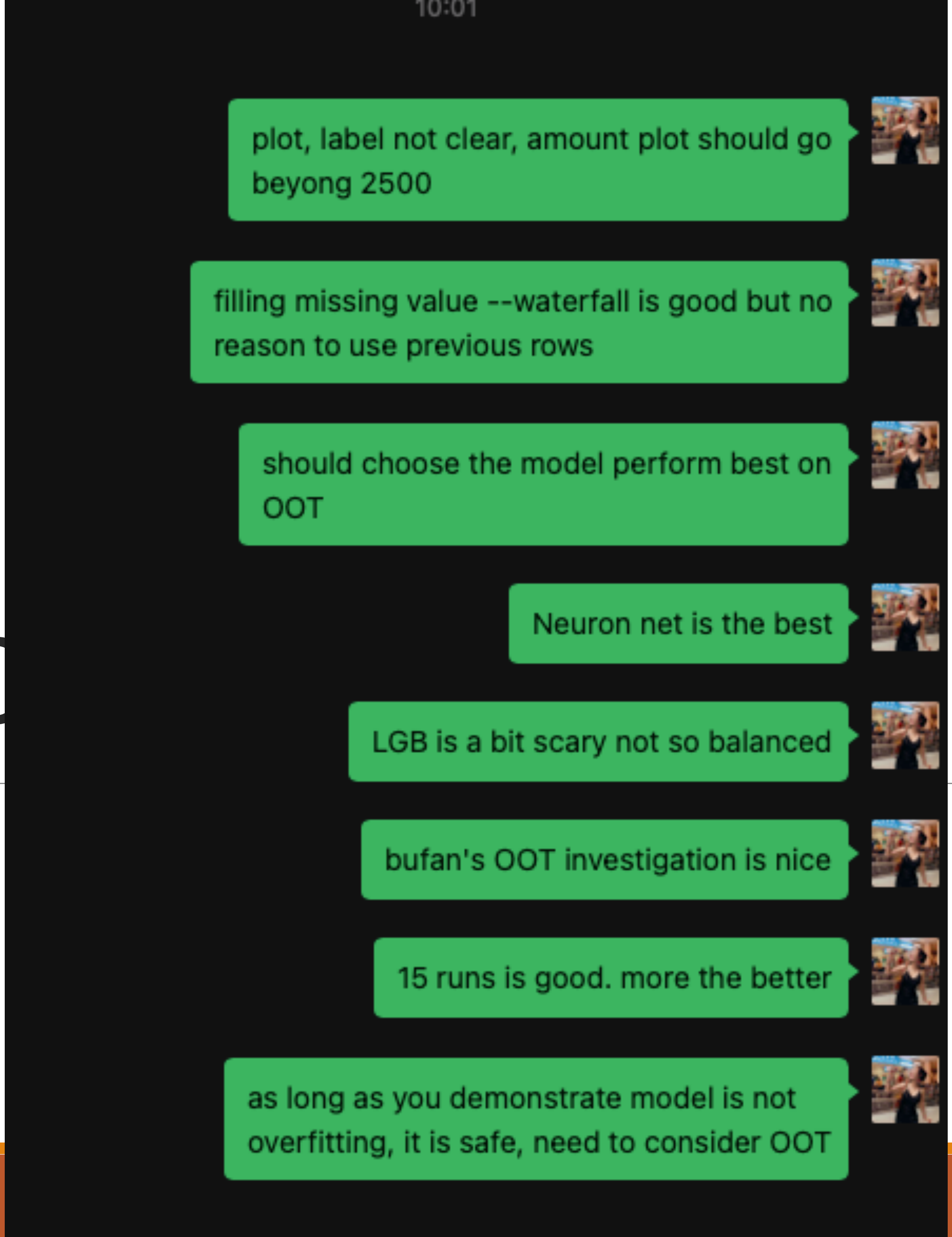
** Financial saving estimate based on the last two months (OOT)*

Future Study – Given More Data

- Implement subsampling on the training data to have higher bad/good ratio (1/10)
- Build a model and then rebuild one after removing the goods with low fraud scores from the training data. Repeat for a few times.

Thanks !

Feedb



10:01
plot, label not clear, amount plot should go beyong 2500
filling missing value --waterfall is good but no reason to use previous rows
should choose the model perform best on OOT
Neuron net is the best
LGB is a bit scary not so balanced
bufan's OOT investigation is nice
15 runs is good. more the better
as long as you demonstrate model is not overfitting, it is safe, need to consider OOT