USC Marshall
School of Business

University of Southern California

Team Presentation:

Credit Card Transaction Fraud

Description of the data

Description of the data

- Year: 2006
- Type: Credit Card Purchase Data
- From: Government Organization
- Number of records: 96,753
- Fraud: 1,059 (1.09%)

Numerical Table:

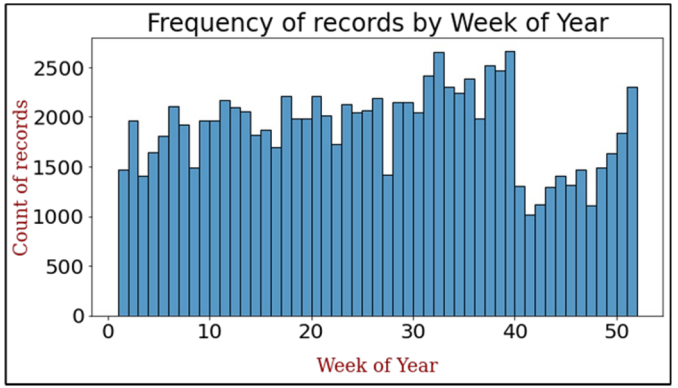
Field	Mean	Std	Max	Min	% Populated	% Zero
Date	2006-06-25 22:21:52	99	2006-12-31 00:00:00	2006-01-01 00:00:00	100.00%	0.00%
Amount	$427.89	$10,006.14	$3,102,045.53	$0.01	100.00%	0.00%

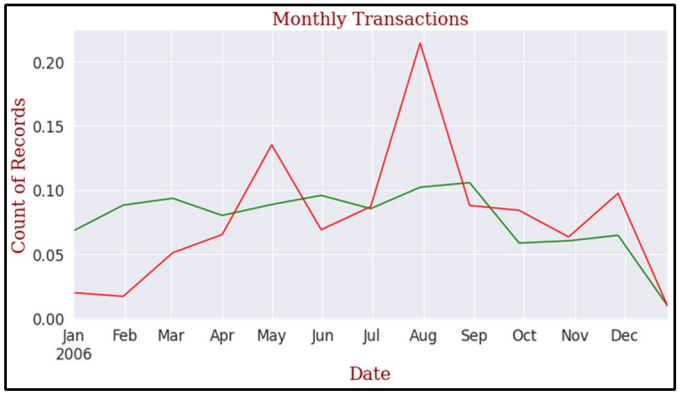
Categorical Table:

Field	# Unique Values	Most Common	Most Common % Total	% Populated
Recnum	96753	All Unique	-	100.00%
Cardnum	1645	5142148452	1.23%	100.00%
Merchnum	13092	930090121224	9.62%	96.51%
Merch description	13126	GSA-FSS-ADV	1.74%	100.00%
Merch state	228	TN	12.44%	100.00%
Merch zip	4568	38118	12.27%	95.19%
Transtype	4	P	99.63%	100.00%
Fraud	2	0	98.91%	100.00%

Fields Of Interest: Date

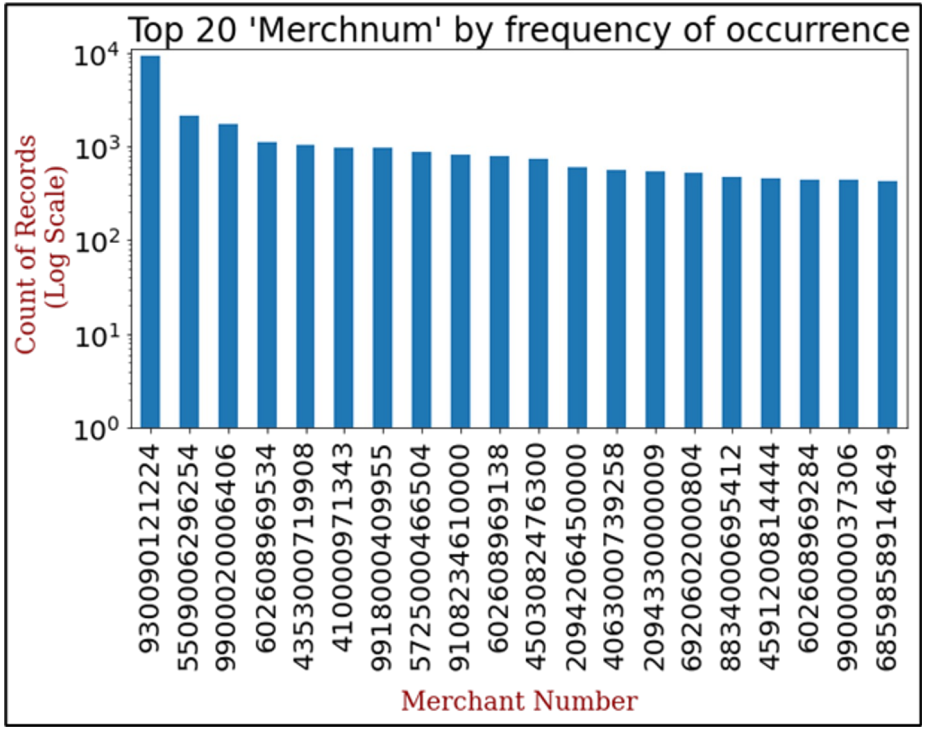
Type	Numeric
Mean	2006-06-25 22:21:52
Standard Deviation	99 Days
Maximum	2006-12-31 00:00:00
Minimum	2006-01-01 00:00:00
% Populated	100%
% Zero	0%





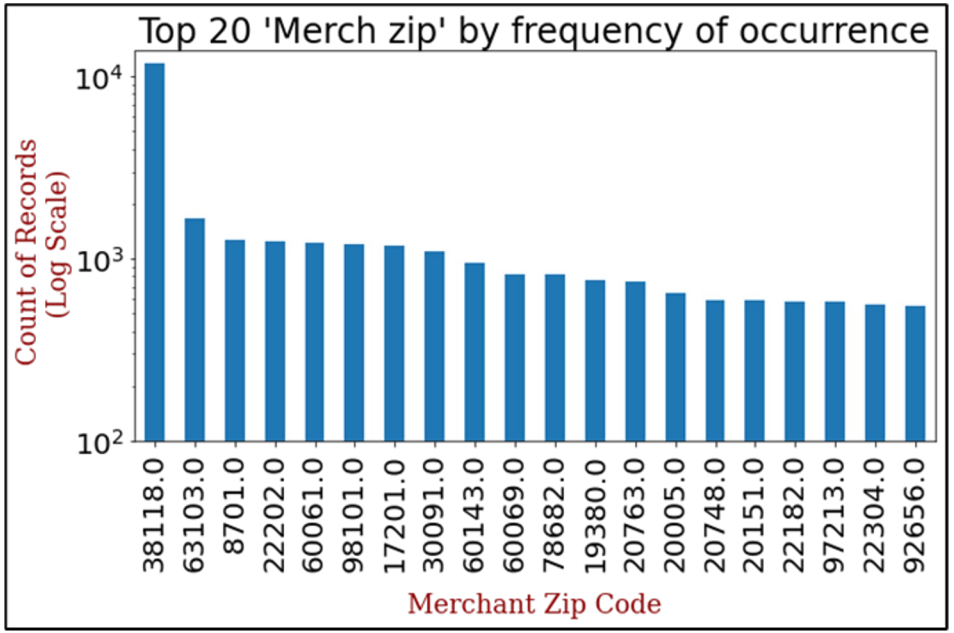
Fields Of Interest: Merchnum

Type	Categorical
# Unique Values	13092
Most Common	930090121224
Most Common %	9.62%
% Populated	96.51%



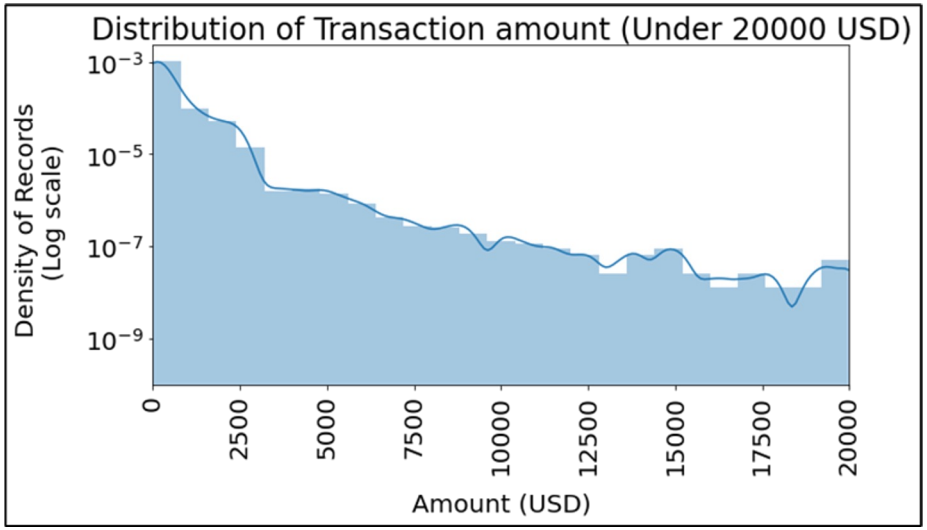
Fields Of Interest: Merch Zip

Type	Categorical
# Unique Values	4568
Most Common	‘38118’
Most Common %	12.27%
% Populated	95.19%



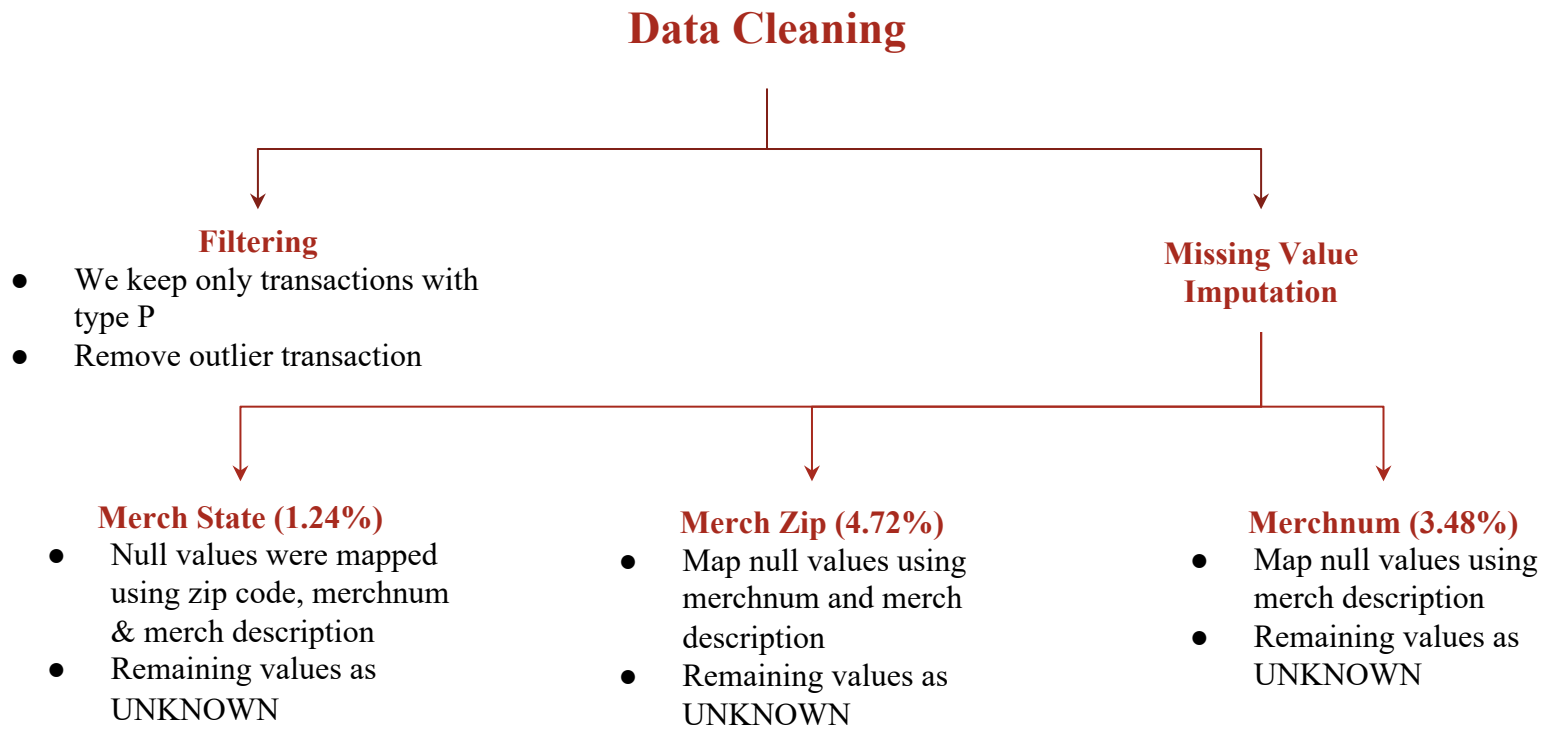
Fields Of Interest: Amount

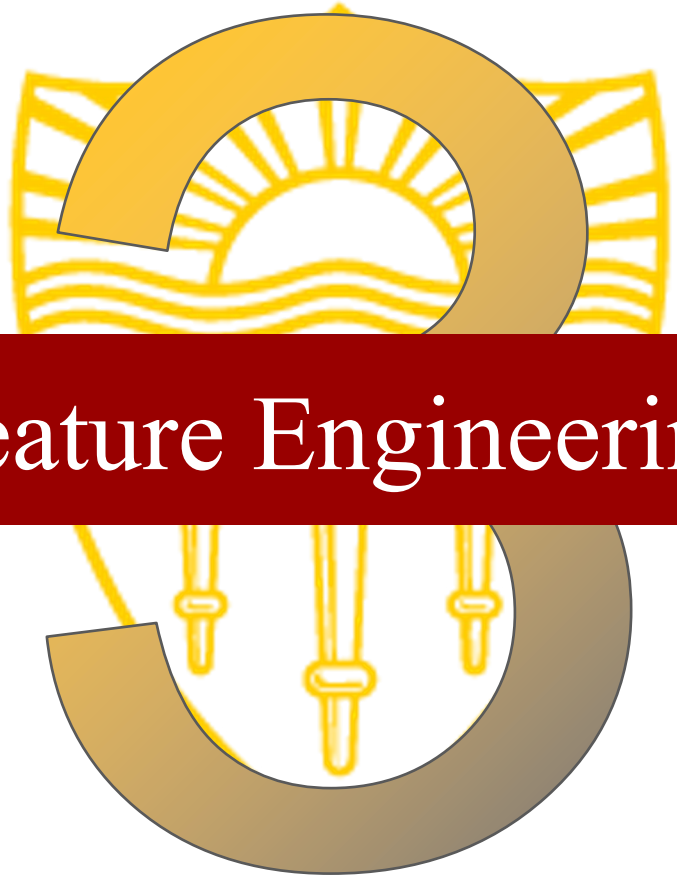
Type	Numeric
Mean	$427.89
Standard Deviation	$10,006.14
Maximum	$3,102,045.53
Minimum	$0.01
% Populated	100%
% Zero	0%



Data Cleaning

Data Cleaning



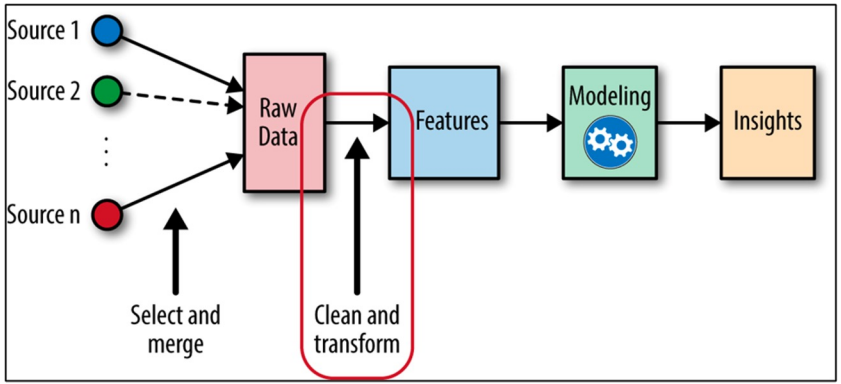


Feature Engineering

Feature Engineering

- A feature is any measurable input that can be used in a predictive model
- Feature engineering is the process of leveraging domain knowledge to extract new features from the raw data
- Why? - It can produce new features for both supervised and unsupervised learning
  - Simplifies & speeds up data transformations
  - Enhances model accuracy

- Feature engineering consists of
  - Creation
  - Transformation
  - Extraction, and
  - Selection of features



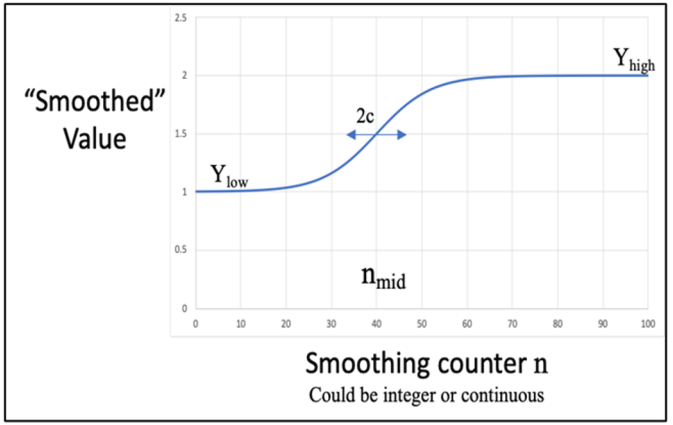
Feature Engineering: Target Encoding

- Several methods: One hot-encoding, binary encoding, label encoding, frequency encoding
- **Target encoding** as cardinality is more than 2
- This method ensures no dimensionality expansion & direct encoding of the prediction variable
- Drawbacks - Interaction Information Loss & Overfitting
- 2 variables:
  - Day of week to create the **dow** variable
  - Merchant State to create the **merchstate** variable
- Prior to target encoding
  - Train-Test dataset: Jan 1 – Oct 31, 2006
  - Validation dataset: Nov 1 – Dec 31, 2006

$$xx \Rightarrow xx_{new} = \begin{cases} v_{aa} \text{ when } xx=aa \\ v_{bb} \text{ when } xx=bb \\ v_{cc} \text{ when } xx=cc \\ v_{dd} \text{ when } xx=dd \end{cases}$$

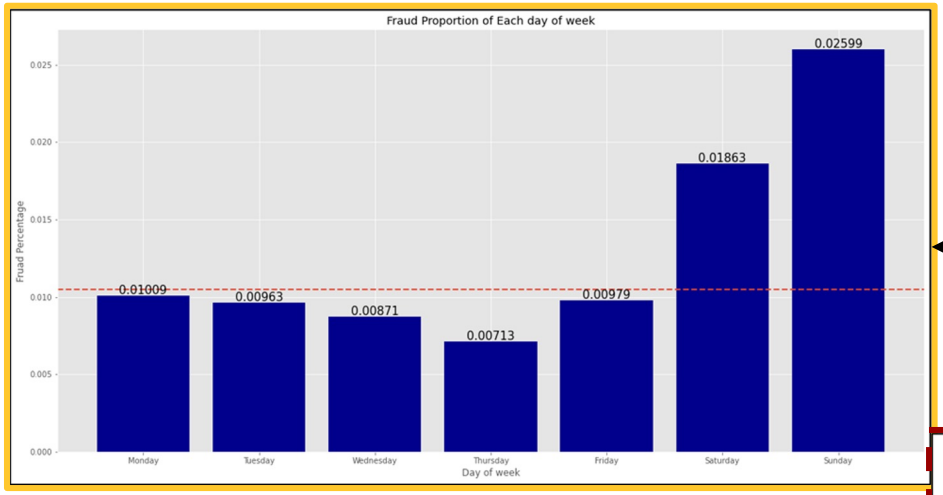
Feature Engineering: Statistical Smoothing

- A smoothing formula is used to smoothly transition a value from one number to another
- It is used in target encoding to overcome the problem of overfitting
- where:
  - Y-low is one number
  - Y-high is the other number
  - Nmid (15) is the value of n where the smoothed value is halfway between Y-low and Y-high
  - c (3) is a measure of how quickly it transitions
  - n is the smoothing counter (integer/continuous)



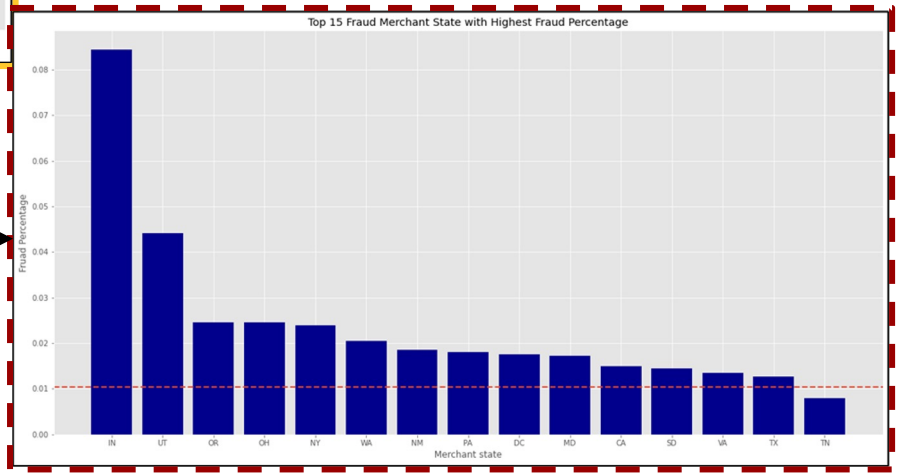
$$\text{Value} = Y_{\text{low}} + \frac{Y_{\text{high}} - Y_{\text{low}}}{1 + e^{-(n - n_{mid})/c}}$$

Feature Engineering: Statistical Smoothing

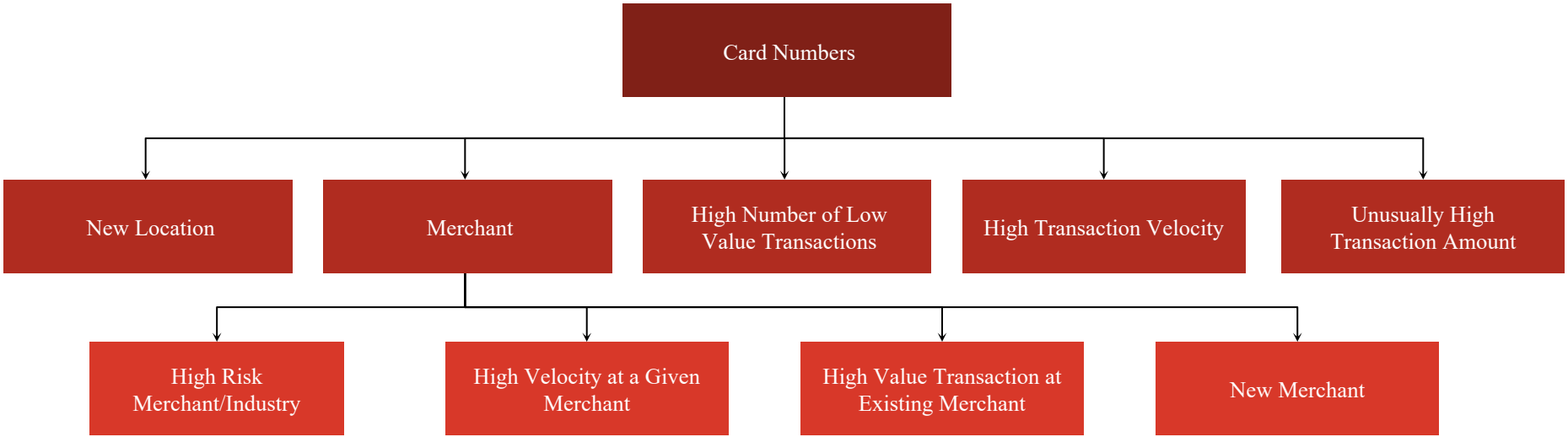


DOW Variable

Merchstate Variable



Feature Engineering: Rationale behind creation of variables



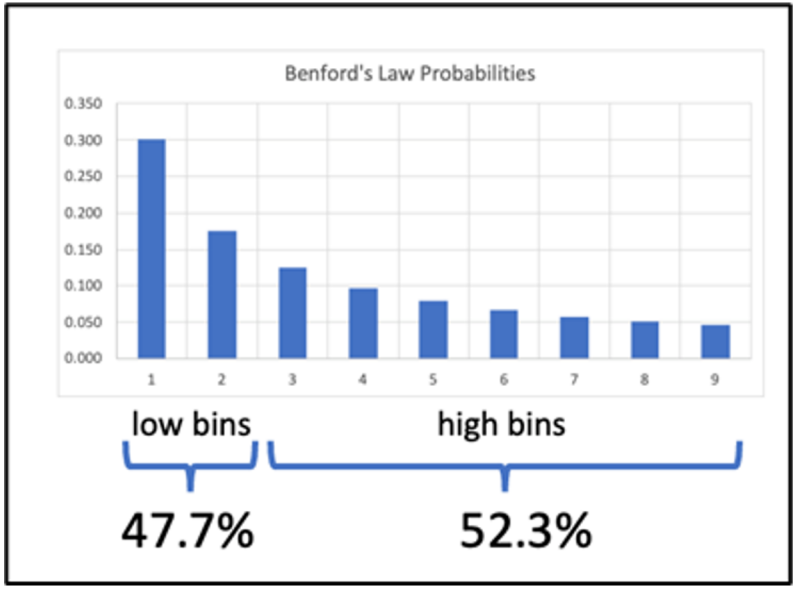
Feature Engineering: Variable Creation

1806 Variables

Type of Variables	Number of Variables
Risk table variables	2
Benford's law variables	2
Days-since variables	13
Frequency and amount variables	1053
Velocity related variables	156
Velocity days-since Entity variables	26
Unique variables	156
Acceleration variables	156
Variability variables	234
Interesting variables	8

Feature Engineering: Variable Creation - Benford's Law Variables

- The idea behind the creation of these variables is to identify scenarios where this law is not followed
- We combine the probability of occurrence of first digit '1' or '2' and evaluate if it follows those dictated by Benford's law



$$U^* = 1 + \left(\frac{U - 1}{1 + \exp^{-t}} \right)$$

Feature Engineering: Creation of combination groups

- A list of initial combination groups that we created out of combining entities which are further used to create variables such as velocity variables, acceleration variables, etc.,

Combination groups		
Card_merch	Zip3	Merchnum_zip
Card_zip	Card_zip3	Merchnum_zip3
Card_state	Merchnum_state	Card_merch_zip3

Feature Engineering: Variable Creation - Other Variables

- Risk table variables
  - Used to convert categorical variables to numeric values
- Days-since variables
  - Tracks the number of days since the entity was last observed
- Frequency variables
  - Tracks the number of times the entity was encountered in the past n days
- Amount variables
  - Measures statistical summary metrics such as average, maximum, median, total and ratios
- Velocity related variables
  - Relative velocity variables
    - Determines the ratio of short-term velocity to a longer-term averaged velocity.
  - Velocity change variables
    - measures the change in number of card transactions in the past 0 or 1 days over the average daily number of transactions in the past 7, 14, 30, 45, 60 & 90 days.
  - Velocity days-since variables
    - measures the velocity change variables for an entity over the days-since variables for the same entity.
- Unique entity variables
  - Calculates the occurrences of unique entities/combination groups for a particular entity/combination group
- Acceleration variables
  - calculating the ratio of velocity change variables for given entities over the square of average daily number of card transactions with the same entities over the past n days

Feature Engineering: Variable Creation - Interesting Variables

- ‘card_merch’
  - We created a variable ‘card_merch’ merging card number and merchant number to identify unique occurrences of these combinations, and further use these in creating velocity, days-since and acceleration variables
- ‘zip3’
  - Also, we created a ‘zip3’ variable by considering only the first three digits of the traditional 5-digit zip code variable
  - We also used this as an entity to further link it with other entities/combination groups and hence the above-mentioned variables.
- ‘card_merch_zip3’
  - In addition, we have created a ‘card_merch_zip3’ variable which includes the zip3 variable, card number and merchant number to zoom in further on specific merchant-card transactions and investigate in our analysis to detect fraud

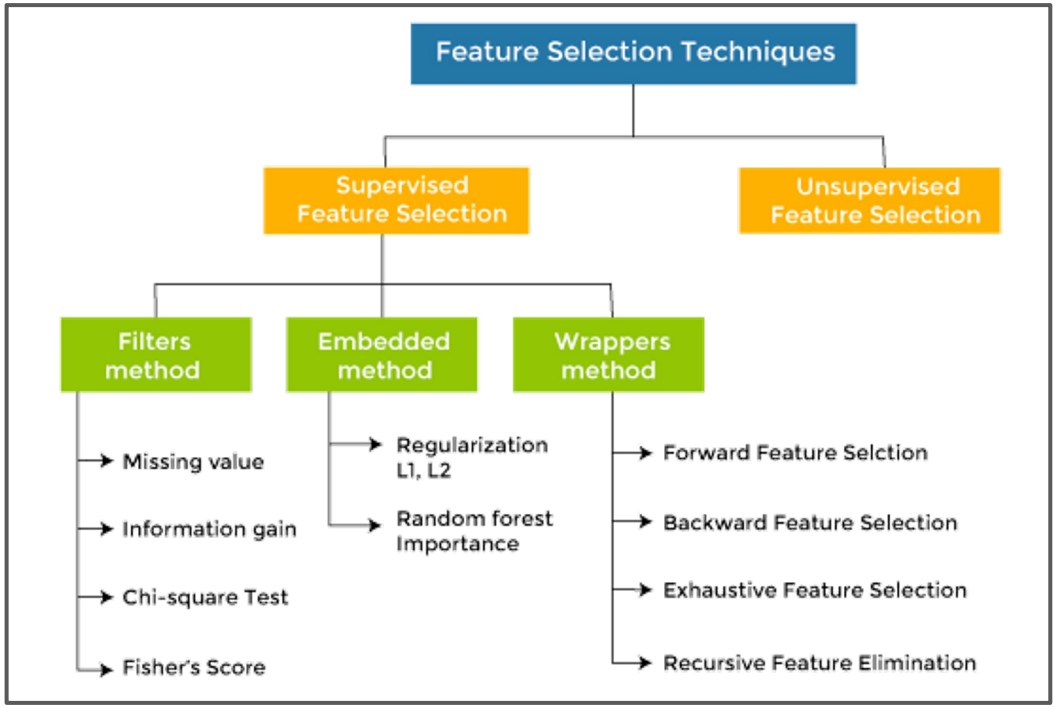
Feature Selection

Feature Selection

- It is the process of selecting a subset of relevant variables for use in models construction.
- It is one of the most important steps.
- Two-step approach:
  i. Filtering
  ii. Wrapping

Objectives:

- Avoid curse of dimensionality
- Reduce overfitting
- Shorten training time
- Simplify models



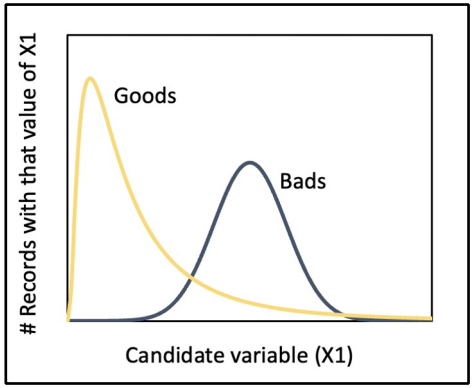
Filtering

- Independent of any ML model.
- Less computational time.
- Variables are selected based on their statistical scores for their correlation with the target variable.
- Since our target variable is binary discrete, we have used univariate KS (Kolmogorov-Smirnov) filtering method.

Variable / Response	Continuous/Metric Ordering	Categorical
Continuous/Metric Ordering	Pearson's Coefficient or KS	LDA
Categorical	Anova	Chi-Squared

Kolmogorov-Smirnov (KS) Filtering

- Normal distributions for Goods and Bads.
- The more different the curves are, the better is the variable for separating. Thus, more important the variable is.
- KS is a simple and an efficient measure of separate these curves are.
- Candidate variables with low KS scores have been discarded.
- We chose top 80 variables with high KS scores after the filtering process.



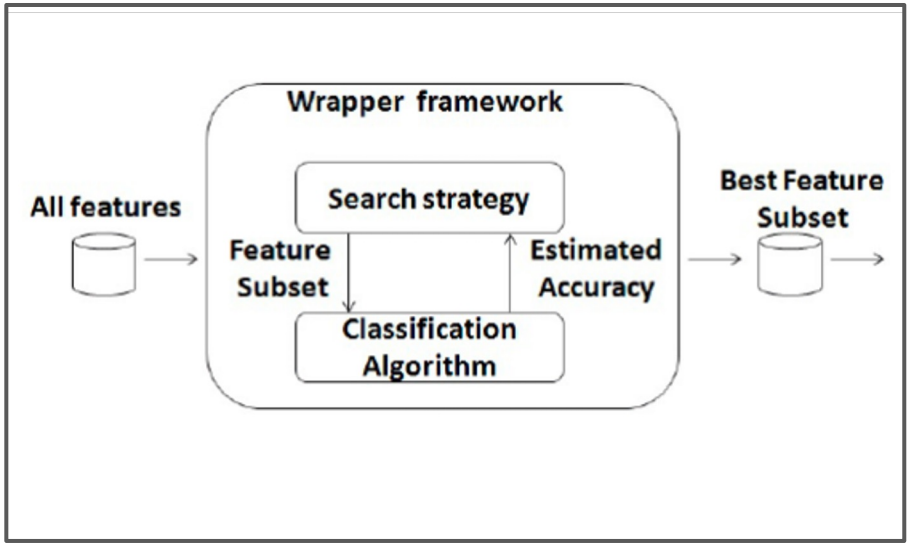
Goods: Records with fraud label = 0
Bads: Records with fraud label = 1

KS is given by:

$$KS = \max_{x} \int_{x_{min}}^{x} [P_{\text{goods}} - P_{\text{bads}}]\, dx$$

Wrapping

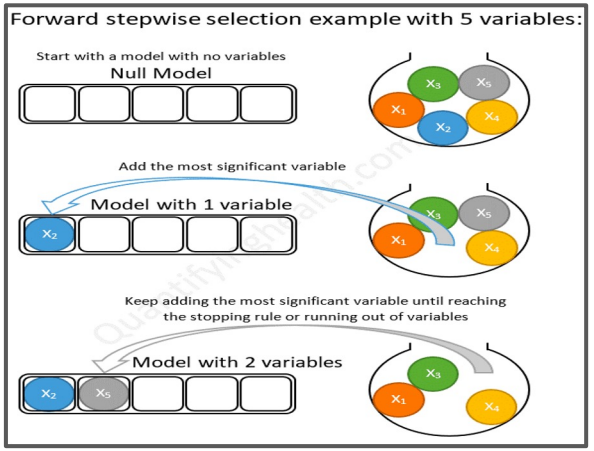
- Take a subset of variables and train the model around them.
- Since a model is “wrapped” around the process, it is called a wrapper.
- It is an iterative process, variables are added/removed based on the model performance.
- Stepwise selection methods:
  - Forward Stepwise Selection
  - Backward Stepwise Selection
  - General Stepwise Selection

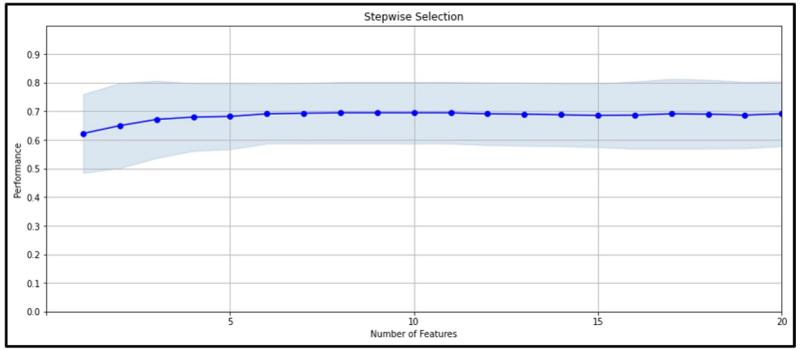


Note: We have used Boosted Tree (LGBM Classifier) model in the wrapping process.

Forward Stepwise Selection

- Train the model with a single variable, the one which gives the better result based on the evaluation metric.
- Select a second variable which in combination with the first gives the best performance.
- Continue the above steps.
- Stop when no significant improvement is observed or the limit of required variables is reached.
- We chose top 20 variables from the wrapper to finally feed into the models.





High level summary of Feature Selection

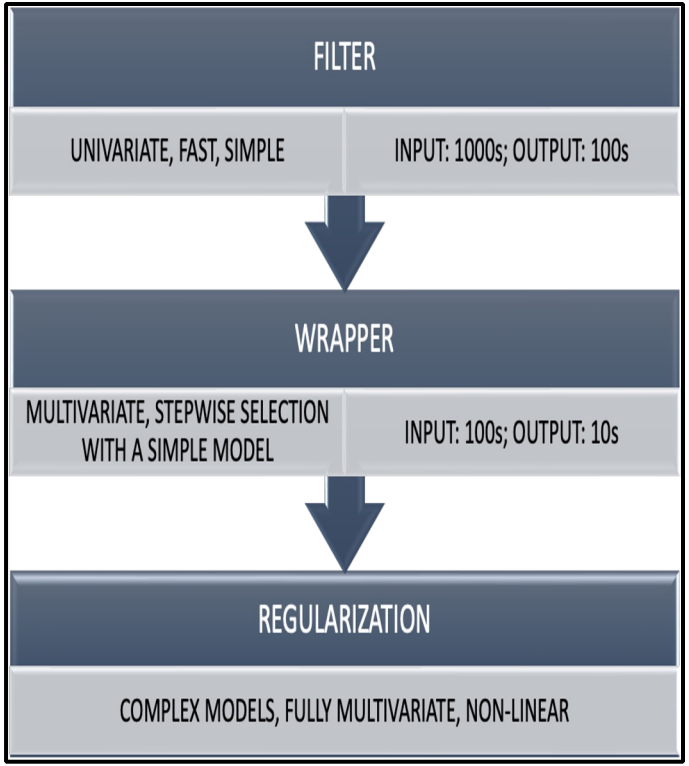
Top 7 variables with KS scores

	variable	score
9	Fraud	1.000000
793	card_zip3_total_7	0.695635
784	card_zip3_total_3	0.687836
547	card_zip_total_7	0.684517
802	card_zip3_total_14	0.681686
465	card_merch_total_7	0.681080
538	card_zip_total_3	0.677563
456	card_merch_total_3	0.675092

Bottom 7 variables with KS scores

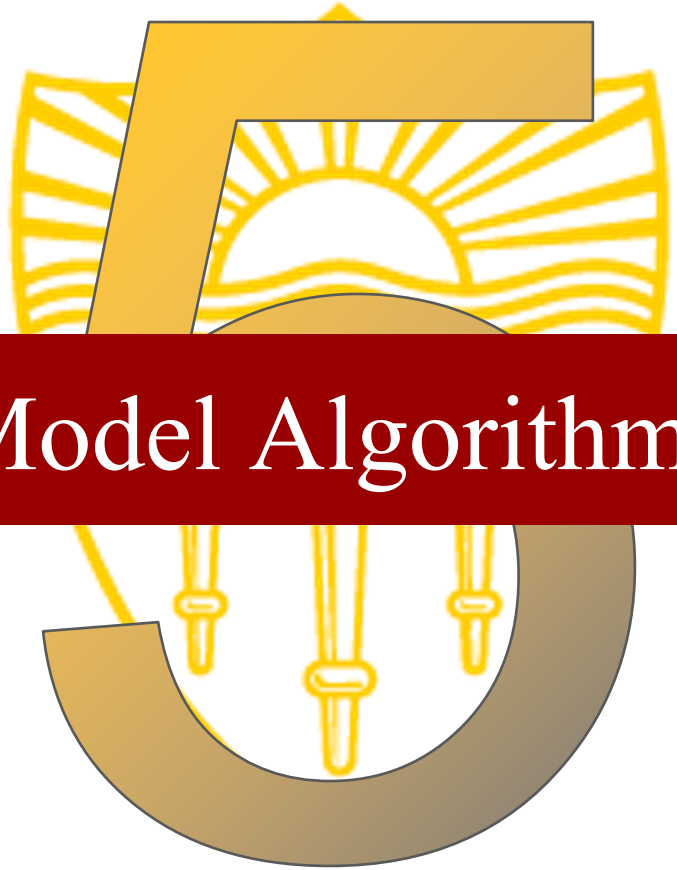
1416	merchnum_zip3_Merchnum_nunique	0.0
1332	card_merch_Merchnum_nunique	0.0
1423	merchnum_zip3_zip3_nunique	0.0
1331	card_merch_Cardnum_nunique	0.0
1326	Merch zip_zip3_nunique	0.0
1347	card_zip_Merch zip_nunique	0.0
1350	card_zip_zip3_nunique	0.0

Feature Selection



Top 20 variables after Forward Selection

Top 20 Variables after Forward Selection (Sorted by Importance)			
feature_idx	Index of Most Important Variables	Variable name	Order of Most important variables
(0,)	0	card_zip3_total_7	1
(0, 73)	73	merchnum_state_total_1	2
(0, 73, 78)	78	Cardnum_total_3	3
(0, 35, 73, 78)	35	card_merch_max_3	4
(0, 23, 35, 73, 78)	23	card_zip_total_30	5
(0, 23, 35, 38, 73, 78)	38	card_zip_max_45	6
(0, 23, 35, 38, 64, 73, 78)	64	card_zip_total_0	7
(0, 23, 28, 35, 38, 64, 73, 78)	28	card_merch_max_30	8
(0, 23, 28, 35, 38, 56, 64, 73, 78)	56	card_zip3_total_0	9
(0, 23, 28, 35, 38, 56, 59, 64, 73, 78)	59	card_merch_total_0	10
(0, 23, 28, 35, 38, 56, 59, 61, 64, 73, 78)	61	card_state_total_0	11
(0, 23, 28, 35, 38, 45, 56, 59, 61, 64, 73, 78)	45	card_state_total_30	12
(0, 11, 23, 28, 35, 38, 45, 56, 59, 61, 64, 73, 78)	11	card_state_total_14	13
(0, 11, 23, 27, 28, 35, 38, 45, 56, 59, 61, 64, 73, 78)	27	card_zip_max_3	14
(0, 11, 23, 27, 28, 35, 38, 45, 56, 59, 61, 64, 73, 77, 78)	77	merchnum_zip3_total_1	15
(0, 10, 11, 23, 27, 28, 35, 38, 45, 56, 59, 61, 64, 73, 77, 78)	10	card_state_total_7	16
(0, 10, 11, 14, 23, 27, 28, 35, 38, 45, 56, 59, 61, 64, 73, 77, 78)	14	card_zip3_total_30	17
(0, 10, 11, 14, 23, 27, 28, 35, 38, 45, 56, 59, 61, 64, 68, 73, 77, 78)	68	card_state_total_45	18
(0, 10, 11, 14, 23, 27, 28, 31, 35, 38, 45, 56, 59, 61, 64, 68, 73, 77, 78)	31	card_state_max_3	19
(0, 10, 11, 14, 23, 27, 28, 31, 35, 38, 45, 51, 56, 59, 61, 64, 68, 73, 77, 78)	51	card_state_max_1	20



Model Algorithms

Model Algorithms

We have explored following machine learning algorithms:

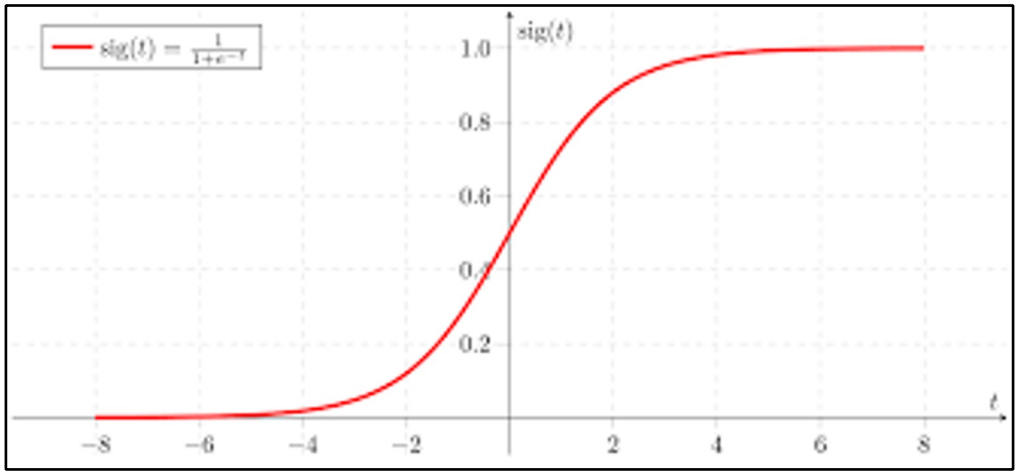
1. Logistic Regression
2. Decision Tree
3. Random forest
4. Gradient boosting
5. Neural Network

Logistic Regression

- This method is used for classification and models the probability of response variable given independent variables using a logistic function. It can be summarized mathematically as follows:

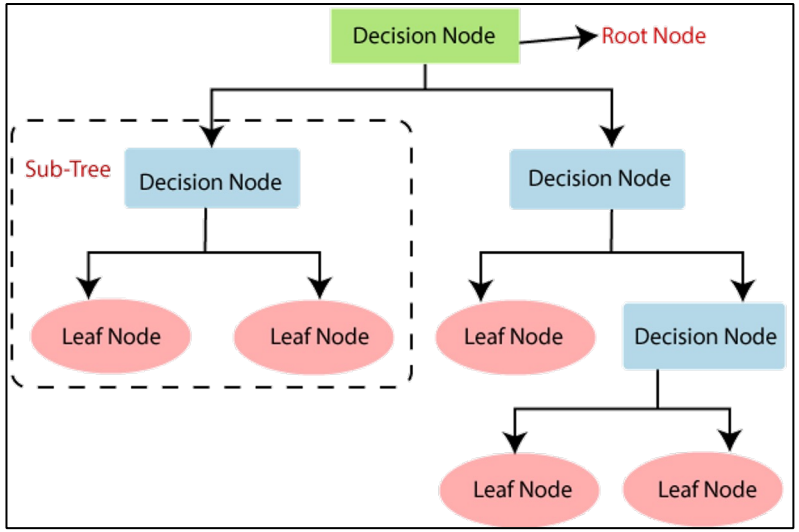
$$P = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Where P is P(Y = 1 | X = x)

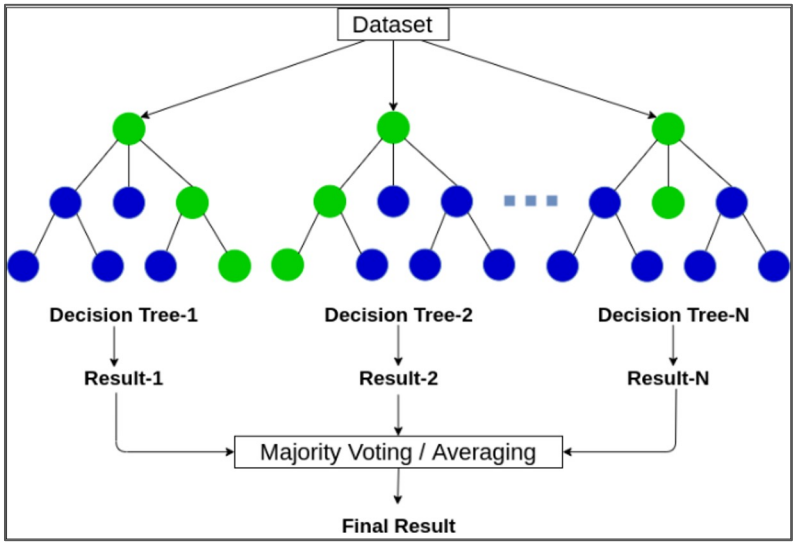


- Hyperparameters:
  - Penalty
  - C
  - Solver

Decision trees and random forests



- Criterion
- Max_depth
- Min samples split
- Min samples leaf

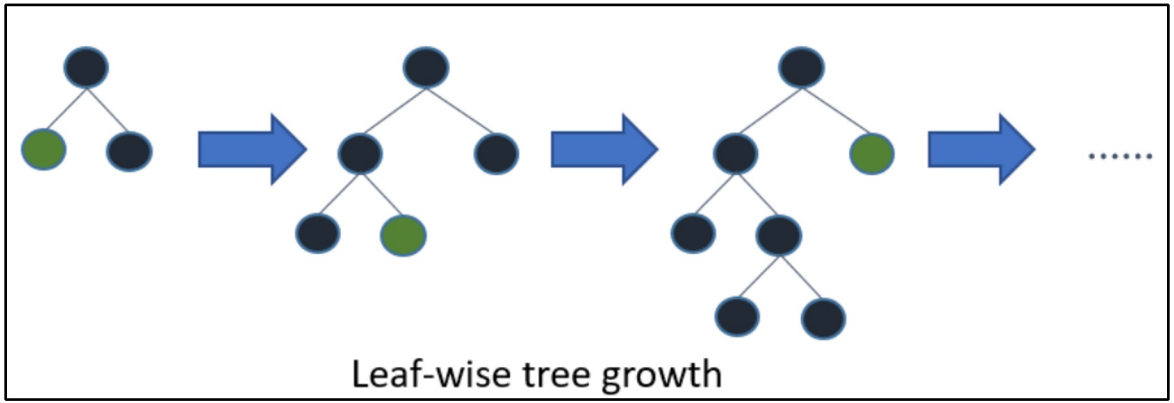


- Decision tree hyperparameters
- N_estimators or number of trees

Light gradient boosting

- These are different from random forests because construction of trees is sequential and not random.
- Unlike other boosting algorithms, light GBM grows leaf wise. It also uses less memory and is faster than other boosting algorithms.

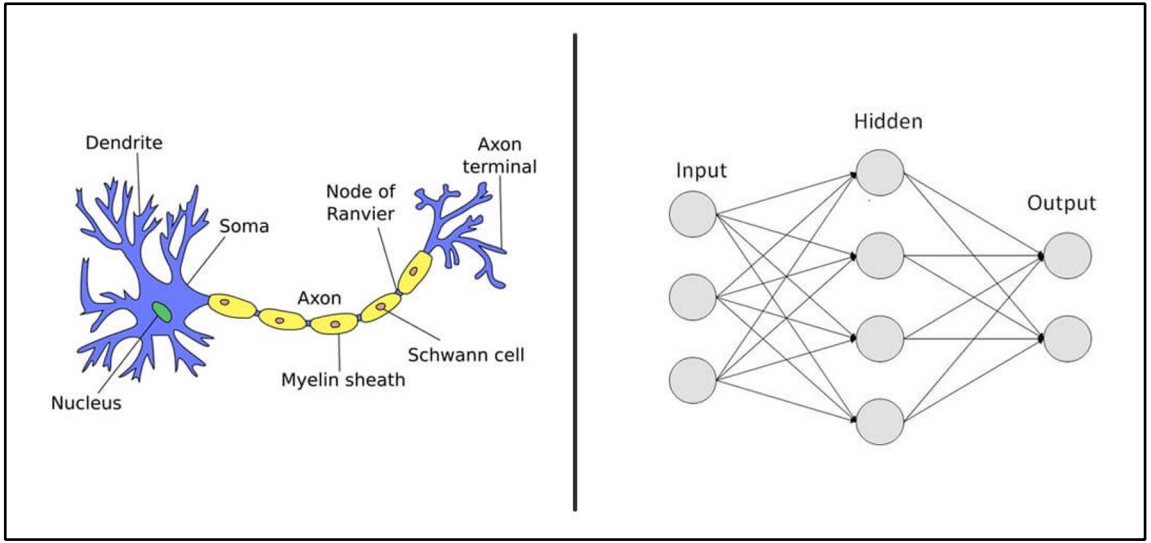
- Hyperparameters:
  - N_estimators
  - Max_depth
  - Num_leaves
  - learning_rate



Neural Networks

- Mimics how the human brain works.
- Composed of a layers: input layer, one or more hidden layers, and an output layer.
- Each node connects to another and has an associated weight and threshold.
- If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

- Hyperparameters:
  - N_hidden
  - Hidden_layer_sizes
  - Activation
  - solver
  - Alpha
  - Learning_rate
  - Learning_rate_init



Hyperparameter tuning

Sl No.	Model	Variables	Hyperparameters									Average FDR at 3%		
	Model Name	Iteration	#Total Variables	penalty			C			solver		Train	Test	OOT
1	*Logistic Regression*	1	15	None (default)			1.0 (default)			saga		68.42%	67.32%	31.79%
		2	15	l2			0.1			sag		68.45%	68.19%	32.23%
		3	15	l1			0.001			saga		62.04%	61.33%	43.24%
		4	15	l1			0.01			saga		62.73%	61.97%	42.30%
		5	15	l1			0.001			liblinear		63.47%	64.15%	41.73%
		6	15	l2			50			lbfgs		68.67%	67.45%	32.85%
		7	15	l2			0.001			newton-cg		66.28%	67.36%	32.79%
		8	15	l2			10			lbfgs		68.62%	67.43%	32.79%
		9	15	l2			1			newton-cg		68.23%	69.28%	32.68%
	Model Name	Iteration	# Total Variables	max_features	criterion	max_depth	min_samples_split		min_samples_leaf	splitter		Train	Test	OOT
2	*Decision Trees*	1	15	None (default)	gini (default)	None (default)	2 (default)		1 (default)	best		100.00%	62.73%	25.14%
		2	15	None	gini	3	10		10	best		56.31%	52.98%	30.73%
		3	15	None	gini	15	100		20	best		86.48%	77.57%	47.21%
		4	15	None	gini	50	500		50	random		69.69%	69.10%	44.25%
		5	15	None	entropy	100	1100		70	random		66.50%	66.11%	42.29%
		6	15	None	gini	15	1000		1	best		75.54%	72.82%	50.56%
		7	15	None	gini	100	500		2	random		73.20%	71.23%	50.00%
		8	15	None	gini	15	500		2	random		72.25%	71.30%	49.61%
		9	15	None	gini	100	500		1	random		72.54%	72.77%	49.11%
		10	15	None	gini	15	500		1	random		72.72%	70.34%	48.72%
		11	15	None	gini	100	500		5	random		72.25%	69.73%	48.44%
	Model Name	Iteration	# Total Variables	n_estimators	max_features	max_depth	min_samples_split		min_samples_leaf	bootstrap		Train	Test	OOT
3	*Random Forest*	1	15	100	5	None (default)	2 (default)		1(default)	TRUE		100.00%	85.22%	56.59%
		2	15	150	10	5	10		10	TRUE		74.53%	72.45%	52.68%
		3	15	50	15	8	100		20	TRUE		81.07%	76.74%	57.21%
		4	15	20	10	10	500		30	FALSE		80.29%	77.66%	57.49%
		5	15	10	5	10	500		20	FALSE		79.16%	76.21%	55.70%
		6	15	30	10	8	500		30	TRUE		76.98%	74.02%	55.20%
		7	15	20	10	8	500		30	FALSE		76.83%	74.07%	55.14%
		8	15	20	10	10	100		20	FALSE		76.00%	73.28%	54.86%
		9	15	30	5	8	100		20	FALSE		75.69%	74.79%	54.80%
		10	15	10	10	8	100		20	TRUE		76.71%	75.75%	54.75%
		11	15	10	3	20	1000		50	TRUE		76.87%	73.60%	46.82%
	Model Name	Iteration	# Total Variables	n_estimators	num_leaves	max_depth		learning rate		subsample		Train	Test	OOT
4	*Boosted Trees*	1	15	100 (default)	31 (default)	default		0.1 (default)		1.0 (default)		99.95%	84.86%	53.30%
		2	15	10	100	4		0.1		0.9		82.10%	76.06%	55.03%
		3	15	50	200	5		0.01		0.8		83.09%	79.58%	55.59%
		4	15	50	50	4		0.01		0.9		79.12%	76.24%	53.91%
		5	15	100	50	4		0.01		0.9		81.16%	78.64%	53.18%
		6	15	100	100	4		0.01		0.9		82.16%	79.12%	53.07%
		7	15	50	100	4		0.01		0.9		78.27%	75.52%	52.85%
		8	15	10	200	4		0.01		0.9		73.80%	72.88%	50.00%
		9	15	1000	50	8		0.02		0.6		100.00%	83.45%	57.82%
		10	15	100	50	-1		0.01		0.9		92.96%	82.27%	59.55%
	Model Name	Iteration	# Total Variables	max_iter	hidden_layer_sizes	activation	solver	alpha	learning_rate	learning rate init	momentum	nesterovs_momentum	Train	Test	OOT
5	*Neural Network*	1	15	200 (default)	100 (default)	relu (default)	adam (default)	0.0001 (default)	N/A	0.001 (default)	N/A	N/A	82.80%	79.19%	52.46%
		2	15	100	50	relu	adam	0.0001	N/A	0.005	N/A	N/A	84.53%	78.98%	54.69%
		3	15	50	1	logistic	sgd	0.001	adaptive	0.02	0.9 (default)	TRUE (default)	69.10%	69.88%	35.92%
		4	15	50	10	relu	lbfgs	0.0001	N/A	0.0001	N/A	N/A	74.93%	73.52%	50.00%
		5	15	300	200	identity	lbfgs	0.01	N/A	0.01	N/A	N/A	68.55%	67.19%	32.07%
		6	15	200	200	tanh	adam	0.0001	N/A	0.001	N/A	N/A	81.84%	79.16%	55.53%
		7	15	100	100	tanh	adam	0.0001	N/A	0.001	N/A	N/A	80.46%	78.71%	55.14%
		8	15	200	200	tanh	adam	0.001	N/A	0.001	N/A	N/A	81.06%	77.82%	54.97%
		9	15	100	100	relu	lbfgs	0.001	N/A	0.001	N/A	N/A	80.51%	77.51%	54.75%
		10	15	100	200	relu	lbfgs	0.0001	N/A	0.001	N/A	N/A	80.97%	76.50%	55.81%
		11	15	500	400	logistic	sgd	0.001	adaptive	0.02	0.9 (default)	TRUE (default)	68.32%	68.06%	32.01%

Overfitting

Underfitting

Overfitting

Overfitting

Overfitting

Underfitting

Results

Results

- Final model: Random forest with 20 trees, 10 max_features, 10 max_depth, 100 min_samples_split, 20 min_samples_leaf and bootstrap as false with 58.10% FDR @ 3%
- Top 5 variables:

Feature	Importance
Cardnum_total_3	**0.43**
card_zip3_total_7	**0.2899**
card_merch_max_3	**0.0585**
merchnum_zip3_total_1	**0.0552**
card_zip_max_45	**0.0528**

FDR across population bins

Training	#Records			#Goods		#Bads			Fraud Rate		
	59008			58401		607			0.010286741		
	Bin Statistics					Cumulative Statistics					
Population Bin%	#Records	#Goods	#Bads	%Goods	%Bads	Total	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	590	250	340	42.37	57.63	590	250	340	0.43	56.01	55.59	0.74
2	590	491	99	83.22	16.78	1180	741	439	1.27	72.32	71.05	1.69
3	590	539	51	91.36	8.64	1770	1280	490	2.19	80.72	78.53	2.61
4	590	566	24	95.93	4.07	2360	1846	514	3.16	84.68	81.52	3.59
5	590	584	6	98.98	1.02	2950	2430	520	4.16	85.67	81.51	4.67
6	590	584	6	98.98	1.02	3540	3014	526	5.16	86.66	81.49	5.73
7	591	578	13	97.80	2.20	4131	3592	539	6.15	88.80	82.65	6.66
8	590	582	8	98.64	1.36	4721	4174	547	7.15	90.12	82.97	7.63
9	590	586	4	99.32	0.68	5311	4760	551	8.15	90.77	82.62	8.64
10	590	588	2	99.66	0.34	5901	5348	553	9.16	91.10	81.95	9.67
11	590	588	2	99.66	0.34	6491	5936	555	10.16	91.43	81.27	10.70
12	590	589	1	99.83	0.17	7081	6525	556	11.17	91.60	80.43	11.74
13	590	585	5	99.15	0.85	7671	7110	561	12.17	92.42	80.25	12.67
14	590	587	3	99.49	0.51	8261	7697	564	13.18	92.92	79.74	13.65
15	590	588	2	99.66	0.34	8851	8285	566	14.19	93.25	79.06	14.64
16	590	588	2	99.66	0.34	9441	8873	568	15.19	93.57	78.38	15.62
17	590	589	1	99.83	0.17	10031	9462	569	16.20	93.74	77.54	16.63
18	590	588	2	99.66	0.34	10621	10050	571	17.21	94.07	76.86	17.60
19	591	589	2	99.66	0.34	11212	10639	573	18.22	94.40	76.18	18.57
20	590	590	0	100.00	0.00	11802	11229	573	19.23	94.40	75.17	19.60

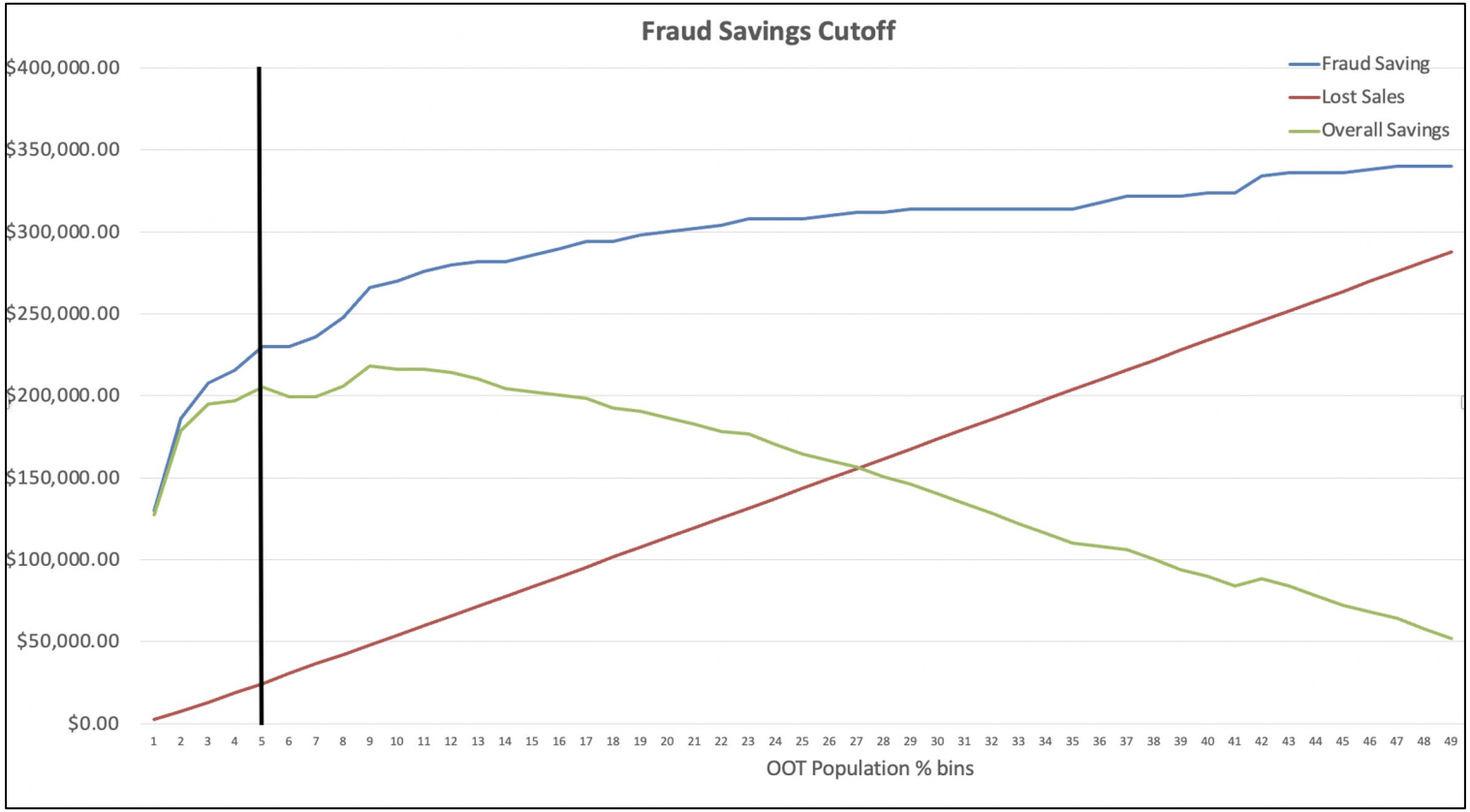
Test	#Records			#Goods		#Bads			Fraud Rate		
	25290			25017		273			0.010794781		
	Bin Statistics					Cumulative Statistics					
Population Bin%	#Records	#Goods	#Bads	%Goods	%Bads	Total	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	253	94	159	37.15	62.85	253	94	159	0.38	58.24	57.87	0.59
2	253	213	40	84.19	15.81	506	307	199	1.23	72.89	71.67	1.54
3	253	237	16	93.68	6.32	759	544	215	2.17	78.75	76.58	2.53
4	253	244	9	96.44	3.56	1012	788	224	3.15	82.05	78.90	3.52
5	252	246	6	97.62	2.38	1264	1034	230	4.13	84.25	80.12	4.50
6	253	248	5	98.02	1.98	1517	1282	235	5.12	86.08	80.96	5.46
7	253	249	4	98.42	1.58	1770	1531	239	6.12	87.55	81.43	6.41
8	253	246	7	97.23	2.77	2023	1777	246	7.10	90.11	83.01	7.22
9	253	251	2	99.21	0.79	2276	2028	248	8.11	90.84	82.74	8.18
10	253	253	0	100.00	0.00	2529	2281	248	9.12	90.84	81.72	9.20
11	253	253	0	100.00	0.00	2782	2534	248	10.13	90.84	80.71	10.22
12	253	252	1	99.60	0.40	3035	2786	249	11.14	91.21	80.07	11.19
13	253	252	1	99.60	0.40	3288	3038	250	12.14	91.58	79.43	12.15
14	253	250	3	98.81	1.19	3541	3288	253	13.14	92.67	79.53	13.00
15	253	253	0	100.00	0.00	3794	3541	253	14.15	92.67	78.52	14.00
16	252	251	1	99.60	0.40	4046	3792	254	15.16	93.04	77.88	14.93
17	253	252	1	99.60	0.40	4299	4044	255	16.17	93.41	77.24	15.86
18	253	252	1	99.60	0.40	4552	4296	256	17.17	93.77	76.60	16.78
19	253	253	0	100.00	0.00	4805	4549	256	18.18	93.77	75.59	17.77
20	253	253	0	100.00	0.00	5058	4802	256	19.19	93.77	74.58	18.76

Validation	#Records			#Goods		#Bads			Fraud Rate		
	12099			11920		179			0.014794611		
	Bin Statistics					Cumulative Statistics					
Population Bin%	#Records	#Goods	#Bads	%Goods	%Bads	Total	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	121	56	65	46.28	53.72	121	56	65	0.47	36.31	35.84	0.86
2	121	93	28	76.86	23.14	242	149	93	1.25	51.96	50.71	1.60
3	121	110	11	90.91	9.09	363	259	104	2.17	58.10	55.93	2.49
4	121	117	4	96.69	3.31	484	376	108	3.15	60.34	57.18	3.48
5	121	114	7	94.21	5.79	605	490	115	4.11	64.25	60.14	4.26
6	121	121	0	100.00	0.00	726	611	115	5.13	64.25	59.12	5.31
7	121	118	3	97.52	2.48	847	729	118	6.12	65.92	59.81	6.18
8	121	115	6	95.04	4.96	968	844	124	7.08	69.27	62.19	6.81
9	121	112	9	92.56	7.44	1089	956	133	8.02	74.30	66.28	7.19
10	121	119	2	98.35	1.65	1210	1075	135	9.02	75.42	66.40	7.96
11	121	118	3	97.52	2.48	1331	1193	138	10.01	77.09	67.09	8.64
12	121	119	2	98.35	1.65	1452	1312	140	11.01	78.21	67.21	9.37
13	121	120	1	99.17	0.83	1573	1432	141	12.01	78.77	66.76	10.16
14	121	121	0	100.00	0.00	1694	1553	141	13.03	78.77	65.74	11.01
15	121	119	2	98.35	1.65	1815	1672	143	14.03	79.89	65.86	11.69
16	121	119	2	98.35	1.65	1936	1791	145	15.03	81.01	65.98	12.35
17	121	119	2	98.35	1.65	2057	1910	147	16.02	82.12	66.10	12.99
18	121	121	0	100.00	0.00	2178	2031	147	17.04	82.12	65.08	13.82
19	121	119	2	98.35	1.65	2299	2150	149	18.04	83.24	65.20	14.43
20	121	120	1	99.17	0.83	2420	2270	150	19.04	83.80	64.76	15.13

- **Train-Test split 70:30**

Train	Test	OOT
61%	26%	13%

Suggested Score Cutoff



7 Conclusion

Conclusion - Give me 5, I'll give you 65!

1. Exploratory data analysis & building Data Quality Report which outlines the overall distribution of data

1. Data Cleaning & imputation

1. Feature engineering - the most important step, because with right set of features we can get great results even with linear models

1. Created 1806 variables -> Filter for top 100 variables -> Wrapper (Forward Stepwise Selection, Accuracy) with 20 variables -> 15 variables with best model

1. We ran several linear and non-linear models on our data, and chose Random forest as our final method
   - ➢ FDR 58.1% at 3% of population **or**
   - ➢ FDR 64.25% at 5% of the population

1. Improvements: Explore other model algorithms (KNN) and explore cross-validation techniques, handle data imbalance in labels, reach out to experts

END

Thank You!