MGTA 463 Project 1 Final Report

- ❖ Executive Summary
- ❖ Part I: Description of the data
- ❖ Part II: Data Cleaning
- ❖ Part III: Variable Creation
- ❖ Part IV: Feature Selection
- ❖ Part V: Preliminary Model Explores
- ❖ Part VI: Final Model Performance
- ❖ Part VII: Financial Curves and Recommended Cutoff
- ❖ Summary
- ❖ Appendix - Data Quality Report

Executive Summary

In this report, we aimed to capture credit card fraud transactions by implementing data-driven machine learning techniques. The data of 97,852 credit card transaction data encompasses 95,805 non-fraudulent transactions and 2,047 fraudulent transactions. Each of the records documented 10 fields according to the behavior of the transaction. Our goal is to build a machine learning model that can recognize fraud transactions by differentiating the variation in the fields.

To discover the most suitable machine learning model to detect fraud in the data, we will develop a machine learning pipeline with the processes of data exploration, data cleaning, variable creation, feature selection, model selection. Each of the segments will be defined and explained thoroughly in the following sections. The final part of the report, we will discuss what should be the fraud score cutoff to generate the maximum potential benefit for the enterprise.

To briefly summarize the project, by implementing the LightGBM model, the overall average accuracy is 0.74 for training data, 0.74 for testing data, and 0.53 for out-of-time data. From this approach, we selected FDR@4% as our cutoff, which will be capable of capturing 62% of the fraud transaction that accounts for approximately $40.5 million dollars of savings per year.

Part I - Description of the data

- Data Overview

The dataset is **Credit Card Transaction Data**, which contains **Records of Purchase Information via credit cards.** It includes 97,852 transactions in 2010, 9 fields (2 Numerical, 1 Temporal, and 7 Categorical), and 1 label - Fraud.

- Summary Tables

Numeric Fields Table

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Min	Max	Mean	Standard Deviation	Most Common
0	Amount	numeric	97852	100.0%	0	0.01	3102045.53	425.466438	9949.80	3.62
1	Fraud	numeric	97852	100.0%	95805	0.00	1.00	0.020919	0.14	0.00

Temporal Field Table

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Most Early	Most Late	Most Common
0	Date	datetime	97852	100.0%	0	2010-01-01	2010-12-31	2010-02-28

Categorical Fields Table

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	# Unique Values	Most Common
0	Recnum	categorical	97852	100.0%	0	97852	1
1	Cardnum	categorical	97852	100.0%	0	1645	5142148452
2	Merchnum	categorical	94455	96.5%	0	13091	930090121224
3	Merch description	categorical	97852	100.0%	0	13126	GSA-FSS-ADV
4	Merch state	categorical	96649	98.8%	0	227	TN
5	Merch zip	categorical	93149	95.2%	0	4567	38118.0
6	Transtype	categorical	97852	100.0%	0	4	P

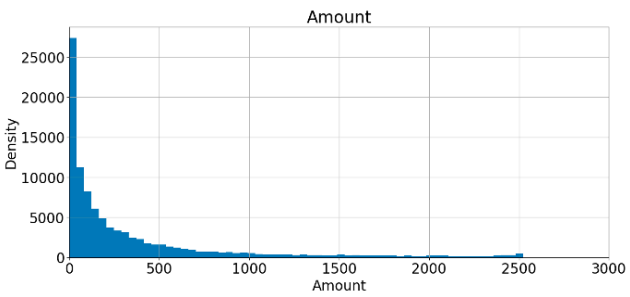
- Critical Fields and Label's Distribution Plot

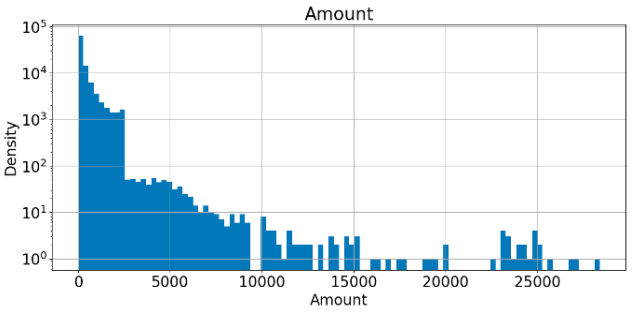
Field Name: Amount

Description: Total Dollars spent in this transaction.

Outliers: 3 records (Threshold at $ 30,275)

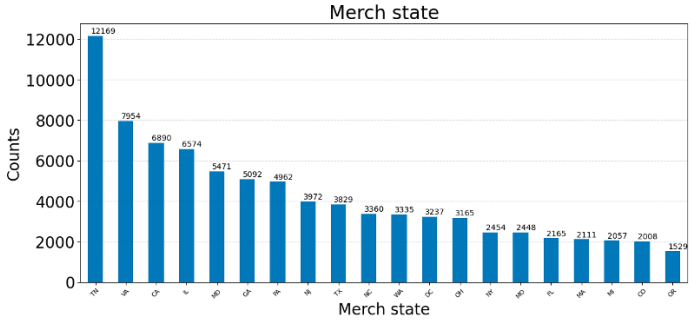
The distribution was highly right-skewed, implying most transactions have low spending. The first distribution has an x-range of 0 to 3000, including 0.99% of the data. The second distribution log-transformed the y-axis, including all data but the outliers.





Field Name: Merch state

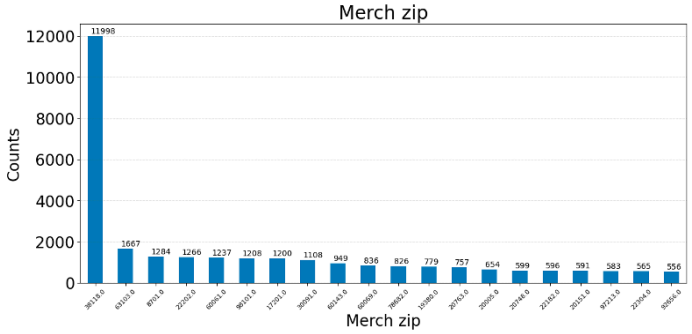
Description: The Merch state indicates where the transaction was made. The distribution shows the top 20 field values of Merch state. The most common Merch state is TN, with a total count of 12,169.



Field Name: Merch zip

Description: The Merch zip indicates the zip code where the transaction was made. The distribution shows the top 20 field values of Merch zip. The most common Merch zip is 38118, with a total count of 11,998.

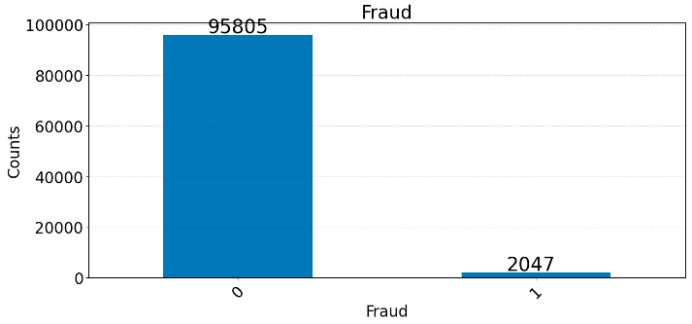
Observation: The zip code 38118 is in TN, aligning with Merch state.



Label Name: Fraud

Description: 1 = Fraud Transaction, 0 = Not Fraud Transaction

This is the label(dependent value) for the dataset, with 95,805 non-fraud transactions and 2,047 fraud transactions.



Part II - Data Cleaning

- Fields Count

Column Name	Non-Null Counts
Recnum	97,852
Cardnum	97,852
Date	97,852
Merchnum	**94,455**
Merch description	97,852
Merch state	**96,649**
Merch zip	**93,149**
Transtype	97,852
Amount	97,852
Fraud (Label)	97,852

Shape: (97852, 10)

Merchnum, Merch description, Merch zip are the 3 fields containing null values.

- Outlier Removal

Field: Amount

The outlier threshold of the dataset is $ 30,275 (3 std away from the mean). Recnum #53179 has transaction amount $3,102,045. After discussing with our business manager, we decided to remove this anomaly from the model.

Total transaction amount: 97,852 - 1 = **97,851**

- Data Exclusions

Field Name: Transtype

There are 4 categories in Transtype: P(97,497), A(181), D(173), Y(1). Since the business manager can't identify the meaning of A, D, Y, we decided to remove data that are Transtype A, D, Y.

Total transaction amount: 97,851 - 355 = **97,496**

- Discovering Potential Null Values

Field: Merchnum

In the field Merchnum, there are 59 transactions that have a 0 as the Merchandise number. We should consider them null values as well.

Fields with null values to be filled:

Column Name	Null Counts
Merchnum	3,279
Merch state	1,028
Merch zip	4,347

- Null Values Imputation for the Fields

Field: Merchnum

1) There are **3,279** null values for Merchnum. The following steps set forth the imputation process for Merchnum.
2) Recover Merchnums that have Merch description with corresponding Merchnums that are displayed on other non-null data. This action took care of 1,164 records, leaving **2,115** null values remaining.
3) Merch description 'RETAIL CREDIT ADJUSTMENT' and 'RETAIL DEBIT ADJUSTMENT' do not have corresponding Merchnums. Replace these null values to 'unknown'. This action took care of 694 records, leaving **1,421** null values remaining.
4) For the remaining 1,421 records without Merchnum, there are 515 unique Merch descriptions, indicating that the remaining transactions are mostly evenly distributed. We decided to give each of the 515 unique Merch descriptions a new Merch num, winding down the number of missing values in Merchnum to **0**.

Field: Merch state

1) There are **1,028** null values for Merch state. The following steps set forth the imputation process for Merch state.
2) Recover Merch states that have Merch zip information. We can align back the Merch state information if the zip codes are provided. There're 22 zip codes that are not in the not-null data but can be manually searched. With 4,567 unique Merch zip that have Merch state information. We took care of 74 records, leaving **954** null values remaining.
3) We replicated the same strategy again with Merchnum and Merch description. We took care of 2 records, leaving **952** null values remaining.
4) Same as the third step in the imputation process for Merchnum, there are several transactions involving 'RETAIL CREDIT ADJUSTMENT' and 'RETAIL DEBIT ADJUSTMENT'. We decided to set their Merch state as 'unknown'. We took care of 655 records, leaving **297** null values remaining.
5) For the remaining 297 null values, we fill the Merch state value with 'unknown'.
6) (Additional verification) Since determining whether it is a foreign transaction can be pivotal to identify fraud, we replaced the value in Merch state with 'foreign' if it is not a US State, null value, nor 'unknown'.

Field: Merch zip

1) There are **4,347** null values for Merch zip. The following steps set forth the imputation process for Merch zip.
2) Recover Merch zips that have Merch num or Merch description information, tracing back the Merch zips from non-null values. We took care of **1,722** records, leaving **2,625** null values remaining.
3) For records with Merch state but not Merch zip, we placed the most populated zip codes in each state for null values. We took care of **1,409** records, leaving **1,216** null values remaining.
4) For the remaining null values in Merch zip, we decided to place 'unknown'.

Part III - Variable creation

- How Fraud Occurs

The financial fraud we're mainly discovering in this project is CNP(Card-not-Present) Fraud. CNP Fraud is mainly caused by skimming devices(skimmer) that are installed on top of credit card readers. The skimmer can read the magnetic strips on the back of credit cards to gain enough information to make a counterfeit credit card. The fraudsters typically placed the skimmer in a place without a cashier's physical presence, such as a gas station, convenient store. Hence, highly suspicious data can be a transaction with the same card but transactions made from different locations(state, zip code).

- Variables Creation Table

We tried to create as many fields as possible based on our original data to observe the possible pattern of behavior when CNP fraud occurred. We created the fields according to four main principles:

1. Amount Variable: For each entity, the statistical measures within a certain amount of days.
2. Frequency Variable: For each entity, the occurrence within a certain amount of days.
3. Velocity Change Variable: The ratio in between two entities' statistical measures over the past N days.
4. Days-since Variable: For each entity, the statistical measures within a certain time span.

Please refer to the table for detailed explanation for each field.

Description	# of Field Created	Cumulative Fields
Original Fields The primitive fields in the dataset.	10	10
Dow/Dow Risk Display the day of the week and its corresponding target encoding value for each transaction record.	2	12
Linkings Logical combination of all entities.	21	33
Target Encoding Numeric representation of categorical variables: Merch state, Merch zip, dow	3	36
Day-Since Variable Number of days apart since the last transaction.	23	59
Frequency Variable For each entity, the number of transactions recorded within {0, 1, 3, 7, 14, 30, 60} days.	161	220

Amount Variable For each entity, the {Average, Max, Median, Sum, Amount/Average, Amount/Max, Amount/Median, Amount/Sum} of the transaction's Amount within {0, 1, 3, 7, 14, 30, 60} days.	1,288	1,508
Velocity Change Variable 1. For each entity, the transaction's count and sum over the last {0, 1} days divided by the count and sum of transactions over the last {7, 14, 30, 60} days. 2. For each entity, each transaction's count over the last {0, 1} days is divided by the day since the last transaction over the last {7, 14, 30, 60} days.	552	2,060
Amount Variable For each entity, the {Mean, Max, Median} of the amount differences between transactions over the past {0, 1, 3, 7, 14, 30} days.	414	2,474
Frequency Variable Splitting the 23 entities into 4 distinct lists. In each list, identify the unique counts for one entity while grouping another entity over the past {1, 3, 7, 14, 30, 60} days.	696	3,170
Velocity Change Variable By grouping each entity, the count of the transaction in the past {0, 1} days divided by the count of the transaction in the past {7, 14, 30, 60) days, then divided by the square root of {7, 14, 30, 60} days.	184	3,354
Binning the Amount Field Binning the Amount field into 5 bins.	1	3,355
Foreign Zip Code Identification Identify if the transaction is made in the U.S.	1	3,356
[New Variable] **Cardnum unfamiliar entities recognition** Creating new fields 'cardnum_{entity}_first' to recognize whether the record has a new entity information for this cardnum. The formula neglects the first 50 records for each cardnum for data collection purposes.	22	3,378
[New Variable] **Holiday purchase detection** Determine whether the record was made on a holiday since government facilities don't usually work on holidays.	1	3,379

Notes:

1) Entities:
   'Cardnum', 'Merchnum', 'card_merch', 'card_zip','card_state','merch_state', 'state_des', 'Card_Merchdesc', 'Card_dow', 'Merchnum_desc', 'Merchnum_dow', 'Merchdesc_dow', 'Card_Merchnum_desc', 'Card_Merchnum_Zip', 'Card_Merchdesc_Zip', 'Merchnum_desc_State', 'Merchnum_desc_Zip', 'merchnum_zip', 'Merchdesc_State', 'Merchdesc_Zip', 'Card_Merchnum_State', 'Card_Merchdesc_State'

   Total: 23 entities

2) New Variables building methodologies:
   1. Cardnum unfamiliar entities recognition:
      Concept
      Fraudulent transactions are more likely to occur in a new location and involve the purchase of unfamiliar products.
      Methodology
      We exclude the first 50 transactions for each Cardnum to better understand the behavior of the card owner. Subsequently, we have introduced new fields for each Cardnum-entity pair, named new_{entity}_cardnum. Whenever a transaction includes new data from that entity, the corresponding _{entity}_cardnum field is set to 1.

   2. Holiday purchase detection:
      Concept
      The data source is from a government facility. Normally, government facilities don't work during holiday seasons. Any transaction that occurs within this period should be highly suspicious.
      Methodology
      We first listed out all the holidays in a year. With the list, we can determine whether the 'Date' variable occurred during the holiday. 'Is_hoilday' is the new field name. It is set to 1 when it's a holiday, otherwise 0.

Part IV - Feature Selection

F/B	num of filters	num of wrappers	Model	Stochastic	Avg Performance
Forward	130	10	LBGM	No	0.71
Forward	130	10	RF	Yes	0.66
Forward	130	20	LBGM	No	0.71
Forward	130	20	RF	Yes	0.66
Forward	260	10	LBGM	No	0.72
Forward	260	10	RF	Yes	0.66
Forward	260	20	LBGM	No	0.72
Forward	260	20	RF	Yes	0.68

The feature selection process encompasses 2 phases: **Filtering** and **Wrapping**.

- Filtering

To filter out the top candidate field, we sorted each field independently adhering **KS** as the scoring metric. For this initial phase, we take into account **130 and 260** (10%/20% of the total number of fields) as the number of candidate fields moving on to the next phase.

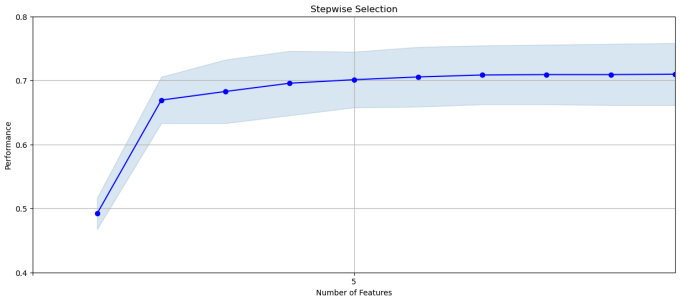
- Wrapping

We implement two different models to **forward select** the best combination of features. The models are **LBGM**, a non-stochastic model that has the same result, and **Random Forest**, a stochastic model that has different results with different trials. To fix this issue, for the Random Forest classifier, we repeat the training process 5 times to find the most common combination. We selected the combination of 10 and 20 wrappers to identify whether the performance will have a significant increase with more features. The wrapper uses detection rate (FDR@3%) as the measure of wrapper model performance.

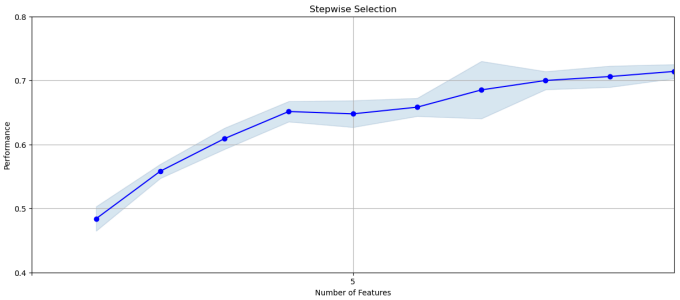
- Key Factor

We can observe that LBGM is a better model and the more filters that we include the better the performance will be. This phenomenon indicates that the model can consider more combinations that are beneficial to the performance with a larger pool of candidate fields. Hence, we decided to add more filters to improve the performance.

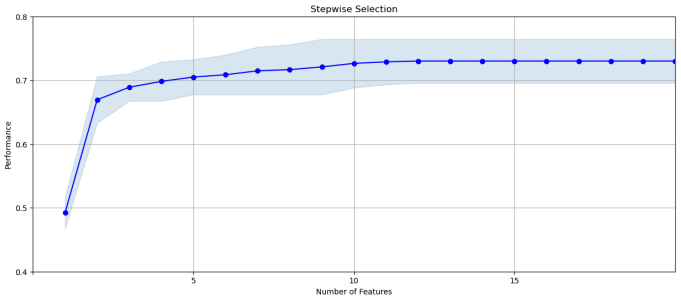
F/B	Num of filters	num of wrappers	Model	Stochastic	Avg Performance
Forward	260	10	LBGM	No	0.72
Forward	260	10	RF	Yes	0.66
Forward	260	20	LBGM	No	0.72
Forward	260	20	RF	Yes	0.68
Forward	520	10	LBGM	No	0.72
Forward	520	10	RF	Yes	0.71
Forward	520	20	LBGM	No	**0.73**
Forward	520	20	RF	Yes	0.73



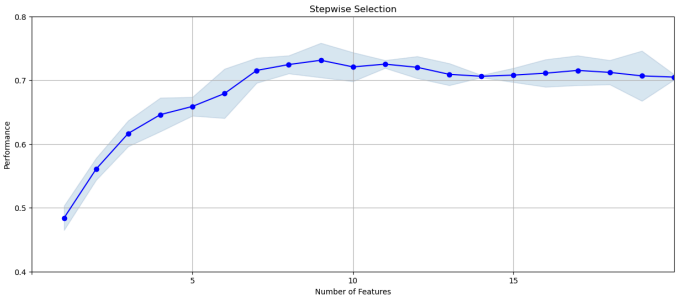
Filter=520, Wrapper=10 LGBM



Filter=520, Wrapper=10 RF



Filter=520, Wrapper=20 LGBM



Filter=520, Wrapper=20 RF

- Conclusion

From the exploration plot, we can observe that with an additional number of filters, we can enhance the average performance. To conclude, although with 520 filters and 20 wrappers, Random Forest and LGBM have the same performance of 0.73. We still select the LGBM model since it is non-stochastic and will provide stable candidate fields. The below table is the sorted list of variables from forward selection. We can see that there's no increase in performance after the 12th trial. Yet, we will still consider 2 times the number of the saturation point, taking into account all 20 fields as variables to be implemented in the model.

- The importance sequence of the 20 fields:

Field Name	**cul score**
Cardnum_unique_count_for_card_state_1	**0.492025**
Card_Merchdesc_State_total_7	**0.669325**
Cardnum_count_1_by_30	**0.688957**
Cardnum_max_14	**0.69816**
Card_dow_vdratio_0by60	**0.704908**
Card_dow_vdratio_0by14	**0.708589**
Merchnum_desc_State_total_3	**0.714724**
Card_Merchdesc_total_7	**0.716564**
Card_dow_unique_count_for_merch_zip_7	**0.720859**
Cardnum_actual/toal_0	**0.72638**
Card_dow_vdratio_0by7	**0.728834**
Cardnum_vdratio_1by7	**0.730061**
Cardnum_unique_count_for_card_state_3	**0.730061**
Cardnum_unique_count_for_card_zip_3	**0.730061**
Merchnum_desc_Zip_total_3	**0.730061**
Cardnum_unique_count_for_Merchnum_3	**0.730061**
Cardnum_actual/toal_1	**0.730061**
Cardnum_unique_count_for_card_state_7	**0.730061**
Cardnum_actual/max_0	**0.730061**
Card_dow_unique_count_for_merch_state_1	**0.730061**

Part V - Preliminary Model Explores

Baseline Model: Logistic Regression

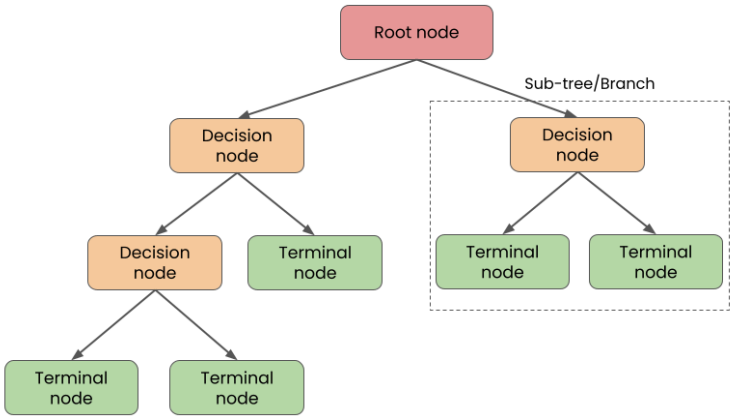
Non-Linear Models: Decision Tree, Random Forest, LightGBM, Neural Net

- ## Non-linear models Introduction

  We selected the below four non-linear machine learning models to compare with the baseline linear model for better performance.The following are high-level descriptions of each model.

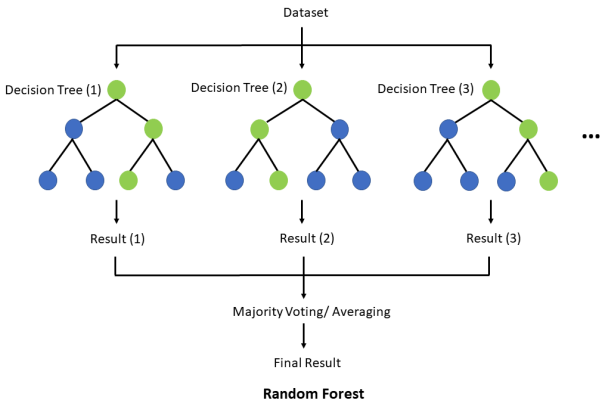
1) Decision Tree

   Decision tree is like a flowchart that splits a dataset into smaller subsets. The final result is a tree with decision nodes and leaf nodes, where each decision node denotes a test on an attribute and each leaf node represents a class label. The paths from root to leaf represent classification rules or paths of decision.



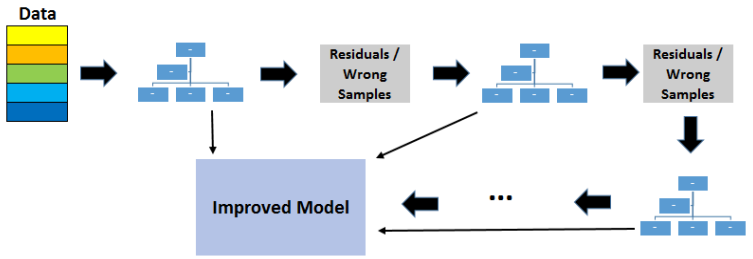
2) Random Forest (Ensemble)

   Random Forest is an ensemble learning method, primarily used for classification and regression, that operates by constructing a large number of decision trees at training time and outputting the class that is the mode of the classes for classification or mean prediction on regression of the individual trees. They are robust against overfitting and accurate on large datasets.



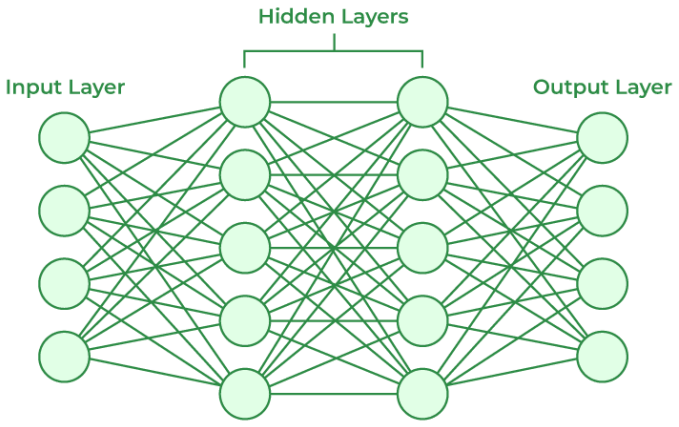
3) Light LBM (Boosting)

LightGBM (Light Gradient Boosting Machine) is a gradient boosting model that uses tree-based learning algorithms. It is designed for distributed and efficient training, particularly on large datasets. LightGBM improves on the efficiency of model building by using a technique known as Gradient-based One-Side Sampling (GOSS) to filter out the data instances to find a split value, while also using Exclusive Feature Bundling (EFB) which bundles mutually exclusive features to reduce the number of features, enhancing the efficiency significantly. It's popular for its training speed and efficiency, especially in cases where the amount of data and number of features are very large.



4) Neural Net

Neural Networks are models inspired by the human brain, designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling, or clustering of raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text, or time series, must be translated. Neural networks help us cluster and classify information. They can be trained to learn features and tasks by considering examples, generally without being programmed with any task-specific rules.



- Table of Tests

Baseline Model: Logistic Regression

Non-Linear Models: Decision Tree, Random Forest, LightGBM, Neural Net

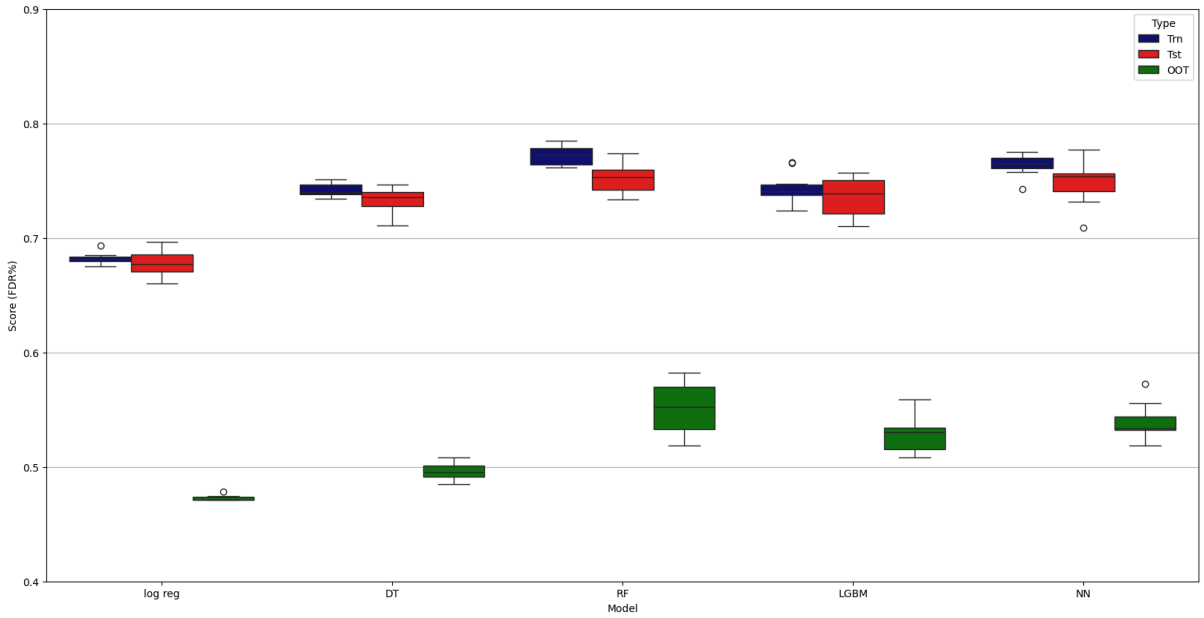
X	Not applicable
O	Mild
V	Applicable

	Hyperparameters					Train Perf	Test Perf	OOT Perf	Overfitting	Local Minimum	
Logistic Regression	penalty	C	Solver								
	l2	1	lbfgs			0.677859	0.686162	0.46835	X	X	Default
	l2	0.1	lbfgs			0.679901	0.683103	0.46734	X	X	
	l2	0.01	lbfgs			0.686833	0.674281	0.466667	X	X	
	l2	1	liblinear			0.680677	0.682871	0.46532	X	X	
	l2	0.1	liblinear			0.685453	0.677437	0.464646	X	X	
	l2	0.01	liblinear			0.685549	0.678474	0.472391	X	X	
	l1	1	liblinear			0.681501	0.678251	0.467003	X	X	
	l1	0.1	liblinear			0.68197	0.680595	0.470034	X	X	
	l1	0.01	liblinear			0.680881	0.680983	0.472727	X	X	Selected Params
DT	criterion	max_depth	min_samples_split	min_samples_leaf							
	gini	None	2	1		1	0.663425	0.386195	V	X	Default
	gini	2	2	1		0.570883	0.561743	0.413468	X	X	
	gini	5	2	1		0.705939	0.685659	0.483502	X	X	
	gini	2	8	4		0.572511	0.568386	0.412458	X	X	
	gini	8	8	4		0.763939	0.723064	0.489226	X	X	
	gini	6	30	15		0.727909	0.724169	0.503704	X	X	
	gini	8	20	10		0.763924	0.731189	0.512795	O	X	
	gini	8	120	60		0.741635	0.731934	0.495286	X	X	Selected Params
	gini	10	60	30		0.778956	0.736103	0.555556	O	X	
	gini	20	60	30		0.810582	0.748751	0.508081	V	X	
	gini	15	30	15		0.857681	0.75956	0.472391	V	X	
	log_loss	8	8	4		0.823136	0.744417	0.500337	V	X	
	log_loss	10	60	30		0.817475	0.763708	0.489899	V	O	
	log_loss	20	60	30		0.824591	0.751852	0.50101	V	O	
	log_loss	15	30	15		0.876559	0.762074	0.486532	V	X	
	log_loss	8	6	3		0.82467	0.750315	0.498653	V	O	
RF	criterion	n_estimators	max_depth	min_samples_split	min_samples_leaf						
	gini	100	None	2	1	1	0.81761	0.552862	V	X	Default
	gini	3	3	2	1	0.652212	0.642684	0.450168	X	X	
	gini	30	8	120	60	0.772628	0.752414	0.551178	X	X	Selected Params
	gini	50	5	2	1	0.73347	0.717327	0.482155	X	X	
	gini	100	5	4	2	0.73454	0.709888	0.484175	O	X	
	gini	200	5	4	2	0.735161	0.710539	0.484512	X	X	
	gini	200	10	4	2	0.853282	0.773194	0.543771	V	X	
	gini	100	10	4	2	0.852065	0.77512	0.532323	V	X	
	gini	50	20	4	2	0.99911	0.80543	0.557576	V	X	
	gini	50	10	4	2	0.85031	0.773108	0.545118	V	X	
	log_loss	50	10	4	2	0.907203	0.793312	0.557239	V	X	
	log_loss	25	10	4	2	0.906323	0.791059	0.557912	V	X	
	log_loss	25	10	2	1	0.918878	0.78433	0.551178	V	X	
	log_loss	25	7	4	2	0.805871	0.750952	0.530976	O	X	
	log_loss	25	5	2	1	0.742162	0.731897	0.488215	X	X	
	log_loss	25	6	6	3	0.767165	0.742946	0.504714	O	X	
LBGM	n_estimators	max_depth	num_leaves	learning_rate							
	100	Max	31	0.1		0.982967	0.808876	0.509764	V	X	Default
	10	2	2	0.1		0.656775	0.663652	0.452525	X	X	
	30	Max	4	0.1		0.743776	0.735795	0.52963	X	X	Selected Params
	50	5	2	0.1		0.701361	0.705437	0.481145	X	X	
	100	5	2	0.1		0.714037	0.701381	0.490236	X	X	
	100	5	5	0.1		0.792034	0.773034	0.538721	X	X	
	100	5	5	0.2		0.80795	0.774382	0.498316	X	X	
	100	5	8	0.1		0.834321	0.778079	0.519865	O	X	
NN	activation	hidden_layer_sizes	solver	alpha	learning_rate_init						
	relu	(1, 1)	adam	0.0001	0.001	0.360718	0.355162	0.270707	X	X	Default
	relu	(3,)	adam	0.0001	0.001	0.698898	0.685023	0.476431	X	X	
	relu	(15,)	adam	0.0001	0.001	0.728341	0.711803	0.491919	X	X	
	relu	(15, 15)	adam	0.0001	0.001	0.766349	0.733312	0.512795	O	X	
	relu	(20, 20)	adam	0.0001	0.002	0.794064	0.767312	0.49798	O	X	
	relu	(20, 20)	adam	0.005	0.01	0.763686	0.748999	0.538047	X	X	Selected Params
	relu	(30, 30)	adam	0.0001	0.001	0.808854	0.761213	0.501347	O	X	
	relu	(30, 30)	adam	0.0001	0.005	0.860764	0.756799	0.502694	V	X	
	tanh	(15, 15)	adam	0.0001	0.001	0.768012	0.733907	0.514815	X	X	
	tanh	(20, 15)	adam	0.0001	0.001	0.784481	0.743747	0.527946	X	X	
	tanh	(30, 30)	adam	0.0001	0.001	0.848838	0.767271	0.509091	V	X	
	tanh	(30, 30)	adam	0.0005	0.001	0.830039	0.768479	0.503367	V	X	
	tanh	(30, 30)	sgd	0.0001	0.01	0.752437	0.734854	0.49798	X	X	
	tanh	(50, 50)	sgd	0.00001	0.01	0.76282	0.738253	0.502694	X	X	
	tanh	(50, 50)	sgd	0.00001	0.025	0.827425	0.761814	0.515825	V	X	
	tanh	(65, 65)	sgd	0.000025	0.025	0.831406	0.763936	0.519192	V	X	

- **Hyperparameters selection for each model:**

  Logistic Regression: {penalty='l1', C='0.01', Solver='liblinear'}
  Decision Tree: {criterion='gini', max_depth=8, min_samples_split=120, min_samples_leaf=60}
  Random Forest: {criterion='gini', n_estimators=30, max_depth=8, min_samples_split=120, min_samples_leaf=60}
  LGBM: {n_estimators=30, max_depth=default, num_leaves=4, learning_rate=0.1}
  NN: {activation='relu', hidden_layer_size=(20, 20), solver='adam', alpha=0.005, learning_rate_init=0.01}

- **Box Plot of trn/test/oot data with all models**



Part VI - Final Model Performance

- Final Model Description

We selected **LGBM** as the best model since it doesn't suffer the issue of overfitting yet with relatively higher OOT performance than the other models.

Model	train	test	oot
LGBM	0.743776	0.748999	0.52963

Hyperparameters Selection:
n_estimators=30: This parameter specifies the number of boosting stages the model has to go through. In other words, LightGBM will build 30 trees sequentially, where each tree learns from the errors of the previous ones, making the model more robust with each step.

max_depth=default: Typically, max_depth controls the maximum depth of the trees. By leaving it at default, you're allowing LightGBM to control this aspect automatically based on the data, which can help in handling overfitting or underfitting.

num_leaves=4: This parameter dictates the maximum number of leaves in each tree. Having more leaves will make the model more complex and can lead to overfitting. In your case, setting it to 4 helps to keep the model simple and prevents it from becoming overly complex, which is beneficial for generalization.

learning_rate=0.1: The learning rate controls how much the model is adjusted during each boosting round. A value of 0.1 strikes a balance between speed of learning and the risk of overfitting by not allowing the model to make very large adjustments in any single step, which helps in achieving a more generalized model.

- Training Results Table

Training	# Records		# Goods		# Bads	Fraud Rate						
	59684		58461		1223	0.02049125394						
	Bin Statistics					Cumulative Statistics						
Population Bin%	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	597	60	537	10.050	89.950	597	60	537	0.103	43.908	43.806	0.112
2	597	279	318	46.734	53.266	1194	339	855	0.580	69.910	69.330	0.396
3	597	503	94	84.255	15.745	1791	842	949	1.441	77.596	76.155	0.887
4	596	537	59	90.101	9.899	2387	1379	1008	2.361	82.420	80.059	1.368
5	597	565	32	94.640	5.360	2984	1944	1040	3.329	85.037	81.707	1.869
6	597	567	30	94.975	5.025	3581	2511	1070	4.301	87.490	83.189	2.347
7	597	581	16	97.320	2.680	4178	3092	1086	5.296	88.798	83.502	2.847
8	597	578	19	96.817	3.183	4775	3670	1105	6.287	90.352	84.065	3.321
9	597	589	8	98.660	1.340	5372	4259	1113	7.296	91.006	83.710	3.827
10	596	587	9	98.490	1.510	5968	4846	1122	8.302	91.742	83.440	4.319
11	597	588	9	98.492	1.508	6565	5434	1131	9.309	92.478	83.168	4.805
12	597	591	6	98.995	1.005	7162	6025	1137	10.322	92.968	82.646	5.299
13	597	589	8	98.660	1.340	7759	6614	1145	11.331	93.622	82.291	5.776
14	597	592	5	99.162	0.838	8356	7206	1150	12.346	94.031	81.686	6.266
15	597	594	3	99.497	0.503	8953	7800	1153	13.363	94.276	80.913	6.765
16	596	589	7	98.826	1.174	9549	8389	1160	14.373	94.849	80.476	7.232
17	597	594	3	99.497	0.503	10146	8983	1163	15.390	95.094	79.704	7.724
18	597	595	2	99.665	0.335	10743	9578	1165	16.410	95.258	78.848	8.221
19	597	596	1	99.832	0.168	11340	10174	1166	17.431	95.339	77.908	8.726
20	597	595	2	99.665	0.335	11937	10769	1168	18.451	95.503	77.052	9.220

- Testing Results Table

Testing	# Records	# Goods	# Bads	Fraud Rate
	25580	25053	527	0.02060203284

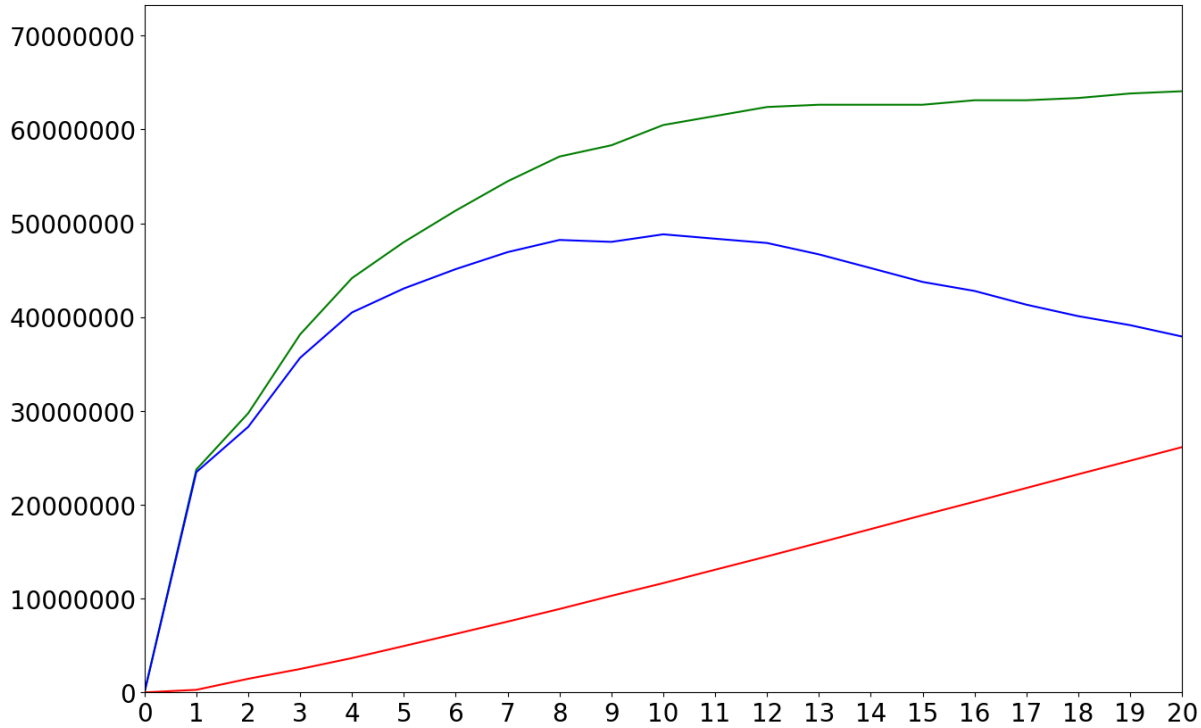
	Bin Statistics					Cumulative Statistics						
Population Bin%	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	256	28	228	10.938	89.063	256	28	228	0.002	43.264	43.261	0.123
2	256	122	134	47.656	52.344	512	150	362	0.599	68.691	68.092	0.414
3	255	214	41	83.922	16.078	767	364	403	1.455	76.471	75.015	0.903
4	256	233	23	91.016	8.984	1023	597	426	2.389	80.835	78.446	1.401
5	256	240	16	93.750	6.250	1279	837	442	3.350	83.871	80.521	1.894
6	256	250	6	97.656	2.344	1535	1087	448	4.352	85.009	80.657	2.426
7	256	251	5	98.047	1.953	1791	1338	453	5.358	85.958	80.600	2.954
8	255	248	7	97.255	2.745	2046	1586	460	6.352	87.287	80.935	3.448
9	256	251	5	98.047	1.953	2302	1837	465	7.358	88.235	80.877	3.951
10	256	251	5	98.047	1.953	2558	2088	470	8.364	89.184	80.820	4.443
11	256	252	4	98.438	1.563	2814	2340	474	9.374	89.943	80.569	4.937
12	256	252	4	98.438	1.563	3070	2592	478	10.383	90.702	80.319	5.423
13	255	248	7	97.255	2.745	3325	2840	485	11.377	92.030	80.653	5.856
14	256	255	1	99.609	0.391	3581	3095	486	12.399	92.220	79.821	6.368
15	256	252	4	98.438	1.563	3837	3347	490	13.409	92.979	79.570	6.831
16	256	254	2	99.219	0.781	4093	3601	492	14.427	93.359	78.932	7.319
17	256	253	3	98.828	1.172	4349	3854	495	15.441	93.928	78.487	7.786
18	255	252	3	98.824	1.176	4604	4106	498	16.451	94.497	78.046	8.245
19	256	255	1	99.609	0.391	4860	4361	499	17.473	94.687	77.214	8.739
20	256	256	0	100.000	0.000	5116	4617	499	18.499	94.687	76.188	9.253

- Out-Of-Time Results Table

OOT	# Records	# Goods	# Bads	Fraud Rate
	12232	11935	297	0.02428057554

	Bin Statistics					Cumulative Statistics						
Population Bin%	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	122	23	99	18.852	81.148	122	23	99	0.193	33.333	33.141	0.232
2	123	98	25	79.675	20.325	245	121	124	1.015	41.751	40.735	0.976
3	122	87	35	71.311	28.689	367	208	159	1.751	53.535	51.784	1.308
4	122	97	25	79.508	20.492	489	305	184	2.570	61.953	59.383	1.658
5	123	107	16	86.992	13.008	612	412	200	3.474	67.340	63.867	2.060
6	122	108	14	88.525	11.475	734	520	214	4.386	72.054	67.668	2.430
7	122	109	13	89.344	10.656	856	629	227	5.307	76.431	71.124	2.771
8	123	112	11	91.057	8.943	979	741	238	6.253	80.135	73.882	3.113
9	122	117	5	95.902	4.098	1101	858	243	7.241	81.818	74.577	3.531
10	122	113	9	92.623	7.377	1223	971	252	8.196	84.848	76.652	3.853
11	123	119	4	96.748	3.252	1346	1090	256	9.201	86.195	76.994	4.258
12	122	118	4	96.721	3.279	1468	1208	260	10.199	87.542	77.343	4.646
13	122	121	1	99.180	0.820	1590	1329	261	11.221	87.879	76.658	5.092
14	122	122	0	100.000	0.000	1712	1451	261	12.252	87.879	75.627	5.559
15	123	123	0	100.000	0.000	1835	1574	261	13.291	87.879	74.588	6.031
16	122	120	2	98.361	1.639	1957	1694	263	14.305	88.552	74.247	6.441
17	122	122	0	100.000	0.000	2079	1816	263	15.336	88.552	73.217	6.905
18	123	122	1	99.187	0.813	2202	1938	264	16.366	88.889	72.522	7.341
19	122	120	2	98.361	1.639	2324	2058	266	17.381	89.562	72.182	7.737
20	122	121	1	99.180	0.820	2446	2179	267	18.403	89.899	71.496	8.161

Part VII - Financial Curves and Recommended Cutoff



Green: Dollar of fraud caught
Red: Lost Revenue due to False Positive
Blue: Overall Savings
Assumptions:
1) $400 gain per fraud caught
2) $20 loss per false positive
3) The current data includes approximately 100,000 records from a portfolio of 10 million transactions/year. We upscaled the data by 10,000,000/100,000.
4) The current data includes 2 months of transactions. To acquire yearly savings, we upscaled by 12/2.

- Recommended Cutoff Percentage

The guiding principle to decide the cutoff is to discover as many fraud transactions while avoiding misidentifying authentic transactions as fraud Although disclosing frauds can save an average of $400/transaction and is the objective in this project, misidentifying transaction as fraud can not only cause an average of $20/transaction but also implicit cost, such as customer satisfaction, reputation, and additional service cost. We decided to set the cutoff at 4% to identify 62% of fraudulent transactions with 1.658% of probability to falsely recognize an authentic transaction as fraud, saving 40.5 million dollars per year.

Cutoff %	Overall Savings	FDR	FPR
1	$ 23,484,000	33.3%	0.232%
2	$ 28,308,000	41.8%	0.976%
3	$ 35,664,000	53.5%	1.308%
4	$ 40,500,000	62.0%	1.658%
5	$ 43,056,000	67.3%	2.060%
6	$ 45,120,000	72.1%	2.430%

Part VII - Summary

This project aims to focus on credit card fraud detection implementing data analysis and machine learning techniques. We identify fraudulent transactions within a dataset of 97,852 credit card transactions, which included a small fraction of fraudulent cases.

The machine learning pipeline began with a thorough data cleaning phase where outliers were identified and removed, and missing data points were imputed through various methods, ensuring robustness in the dataset. Then, we generate new features that could potentially indicate fraudulent behavior based on our domain knowledge on CNP fraud. Feature selection was crucial to refining the model, employing both filtering and wrapper methods to determine the top 20 most crucial features. This was followed by testing several models including Logistic Regression, Decision Trees, Random Forests, LightGBM, and Neural Networks. LightGBM was ultimately selected due to its high performance, particularly OOT validation, which is critical for assessing how the model will perform in real-world scenarios. The model was tuned with parameters like number of trees (n_estimators=30), maximum depth (default), number of leaves (num_leaves=4), and learning rate (0.1) to balance complexity and generalization capabilities.

The final outcomes included a detailed performance evaluation of the LightGBM model and financial analysis to determine the optimal fraud detection cutoff, balancing the detection of fraud and minimizing false positives. At a 4% cutoff, the model achieved a Fraud Detection Rate (FDR) of 62% in OOT data, equating to substantial annual savings estimated at around $40.5 million, considering both the detection benefits and costs associated with false positives. Further improvements could include exploring additional data sources, refining feature engineering, or experimenting with alternative machine learning models to enhance predictive accuracy and reduce false positives.

Appendix - Data Quality Report

- Summary Tables

Numeric Fields Table

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Min	Max	Mean	Standard Deviation	Most Common
0	Amount	numeric	97852	100.0%	0	0.01	3102045.53	425.466438	9949.80	3.62
1	Fraud	numeric	97852	100.0%	95805	0.00	1.00	0.020919	0.14	0.00

Temporal Field Table

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Most Early	Most Late	Most Common
0	Date	datetime	97852	100.0%	0	2010-01-01	2010-12-31	2010-02-28

Categorical Fields Table

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	# Unique Values	Most Common
0	Recnum	categorical	97852	100.0%	0	97852	1
1	Cardnum	categorical	97852	100.0%	0	1645	5142148452
2	Merchnum	categorical	94455	96.5%	0	13091	930090121224
3	Merch description	categorical	97852	100.0%	0	13126	GSA-FSS-ADV
4	Merch state	categorical	96649	98.8%	0	227	TN
5	Merch zip	categorical	93149	95.2%	0	4567	38118.0
6	Transtype	categorical	97852	100.0%	0	4	P

- Visualization of Each Field

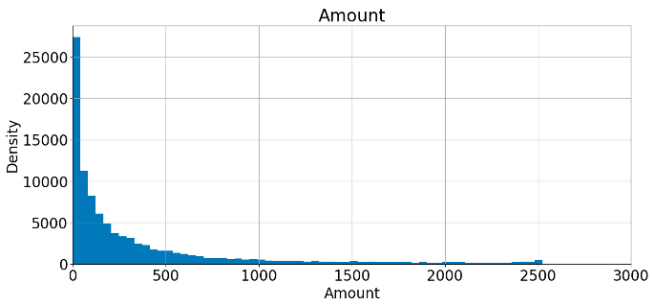
Numeric:

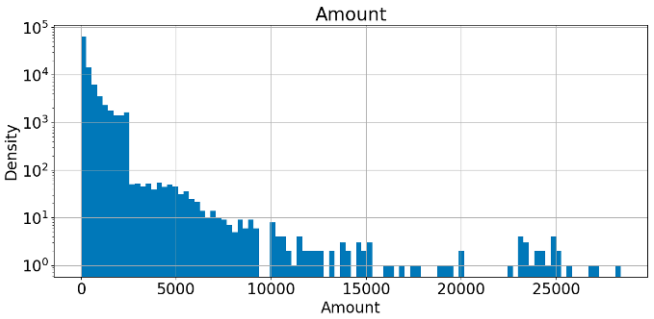
1) Field Name: Amount

Description: Total Dollars spent in this transaction.

Outliers: 3 records (Threshold $30,275)

The distribution was highly right-skewed, implying most transactions have low spending. The first distribution has an x-range of 0 to 3,000, including 1% of the data. The second distribution log-transformed the y-axis, including all data but the outliers.

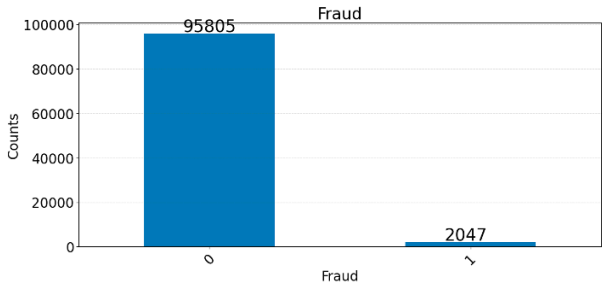




2) Field Name: Fraud (Label)

Description: 1 = Fraud Transaction, 0 = Not Fraud Transaction

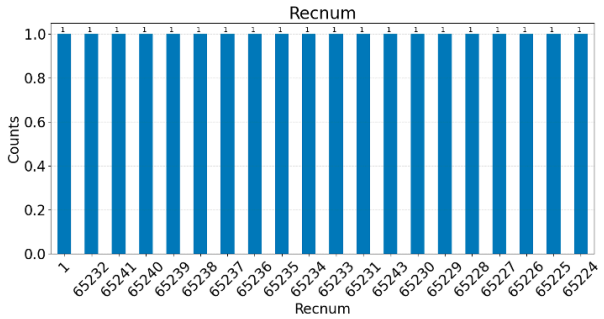
This is the label/dependent value for the dataset, with 95805 non-fraud transactions and 2047 fraud transactions.



Categorical:

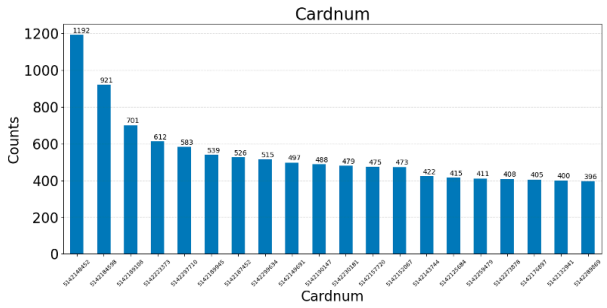
3) Field Name: Recnum

Description: The record number of each transaction. One unique value for each record.



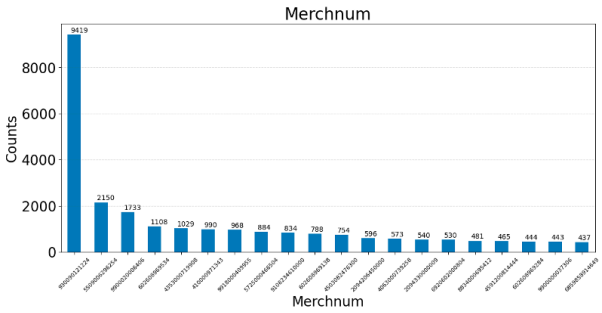
4) Field Name: Cardnum

Description: The credit card numbers used in the transactions. The distribution shows the top 20 field values of Cardnum. The most common Cardnum used is 5142148452, with a total count of 1,192.



5) Field Name: Merchnum

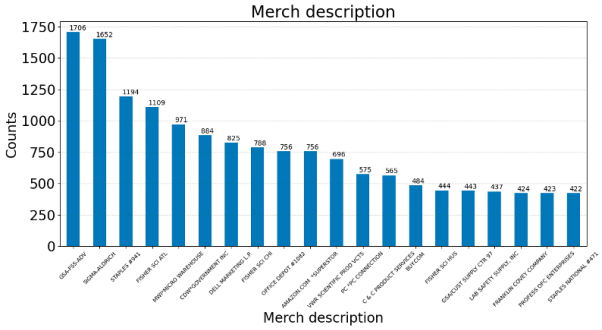
Description: The merchandise number purchased in the transaction. The distribution shows the top 20 field values of Merchnum. The most common Merchnum used is 930090121224, with a total count of 9,419.



6) Field Name: Merch description

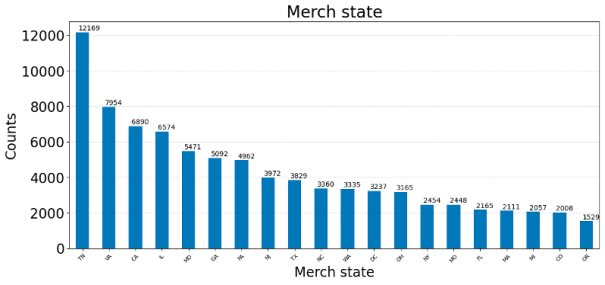
Description: The description of the merchandise in the transaction. The distribution shows the top 20 field values of Merch description. The most common Merch description used is GSA-FSS-ADV, with a total count of 1,706.

Observation: The highest Merchnum 930090121224 doesn't align with the highest Merch description since the highest Merchnum description includes shipping dates that alter through time. On the other hand, the highest Merch description GSA-FSS-ADV's Merchnum is 9900020006406, which is the third highest Merch num.



7) Field Name: Merch state

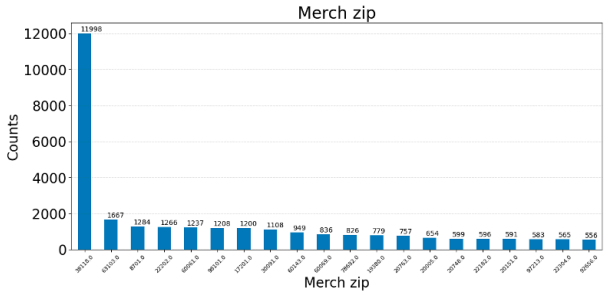
Description: The Merch state indicates where the transaction was made. The distribution shows the top 20 field values of Merch state. The most common Merch state is TN, with a total count of 12,169.



8) Field Name: Merch zip

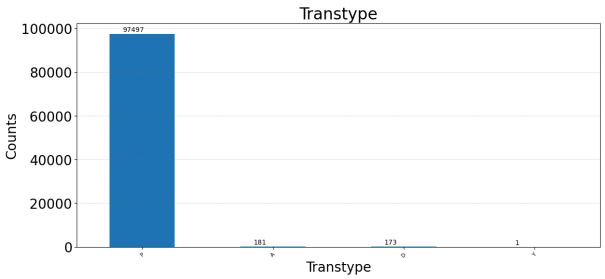
Description The Merch zip indicates the zip code where the transaction was made. The distribution shows the top 20 field values of Merch zip. The most common Merch zip is 38118, with a total count of 11,998.

Observation: The zip code 38118 is in TN, aligning with Merch state.



9) Field Name: Transtype

Description The **Transtype** indicates the transaction type. The distribution shows the most transactions are type P with 97,497, 181 for type A, 173 for type D, and 1 for type Y.



10) Field Name: Date

Description: Transaction Date. The first distribution shows the number of daily transactions across time. The second distribution shows the number of weekly transactions across time.