DSO562 Fraud Analytics

Project 2 Report

Nov 13, 2024

Minjoo Sung

Table of Contents

Section 1. Executive Summary

This project addresses a critical business problem: the need for a reliable fraud detection system to identify and prevent account origination fraud. Fraudulent applications pose a significant financial risk and operational burden, impacting the bank's revenue, reputation, and customer trust. To mitigate these risks, the objective was to develop a model that maximizes fraud detection while minimizing false positives, ensuring legitimate customers experience minimal disruption.

Using advanced machine learning techniques, a model was created to accurately distinguish between legitimate and fraudulent applications. After evaluating several approaches, the LightGBM model was selected for its superior performance and reliability in detecting fraud, achieving a Fraud Detection Rate (FDR) of 61.2% at a 3% threshold for out-of-time (OOT) data. A financial analysis determined the optimal model threshold, resulting in projected annual fraud prevention savings of approximately $18 billion when applied at this 3% cutoff. This threshold represents the best balance between fraud detection and operational efficiency, enabling the bank to reduce fraud losses significantly while maintaining a smooth experience for legitimate applicants. The project's results align with the bank's strategic goals, offering a financially impactful and operationally efficient solution to the fraud detection challenge.

Section 2. Description of the data

The dataset contains 1,000,000 records of synthetic application data for credit card and cell phone applications submitted in 2017. It includes 10 fields: record, date, ssn, firstname, lastname, address, zip5, dob, homephone, and fraud_label. The record field is a unique identifier for each application, while the other fields represent personal identifiable information (PII), such as Social Security Numbers (SSNs), names, addresses, dates of birth (DOBs), and phone numbers. These fields are essential for identifying individuals and are used in fraud detection algorithms. The fraud_label field is a binary variable indicating whether an application was flagged as fraudulent (1) or non-fraudulent (0). This dataset is synthetic but designed to closely mimic the statistical properties of real-world application data. Some fields, like SSNs and addresses, may contain placeholder values or patterns that require further investigation for potential fraud.

The table below provides summary statistics for the numerical and categorical fields.

1. Summary Statistics

1.1 Numerical Table

Field Name	# Records with value	% Populated	% Zeros	Min	Max	Mean	Std	Most Common value
date	1,000,000	100.0%	0	2017-01-01	2017-12-31	-	-	2017-08-16
dob	1,000,000	100.0%	0	1900-01-01	2016-10-31	-	-	1907-06-26

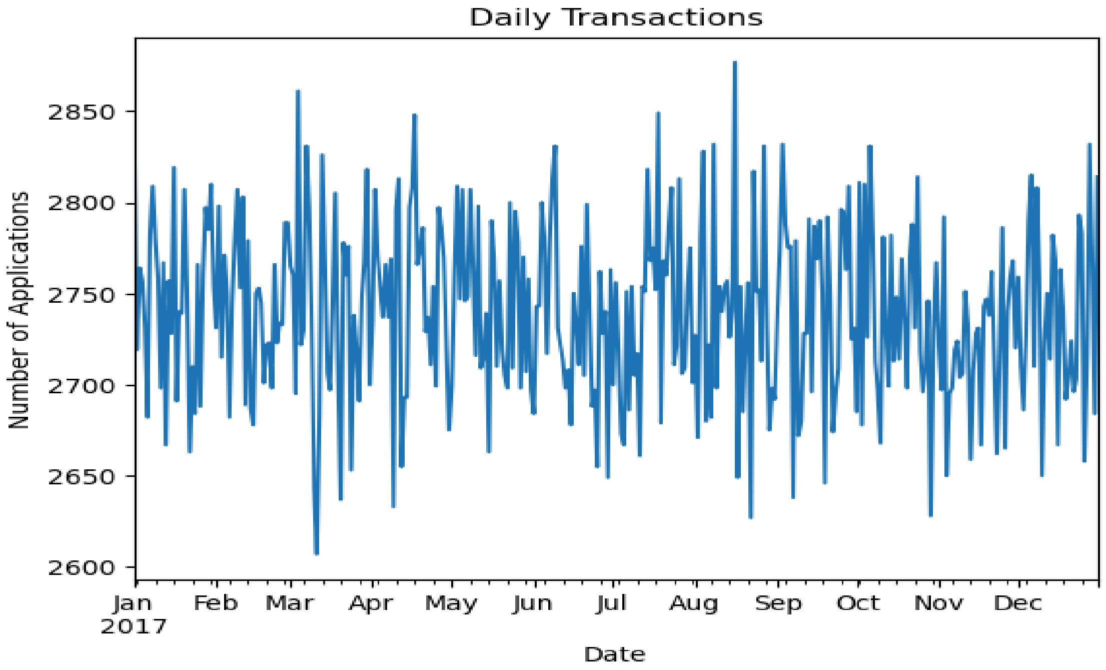
1.2 Categorical Table

Field Name	# Records with Values	% Populated	% Zeros	# Unique Values	Most Common Value
record	1,000,000	100.0%	0	1,000,000	-
ssn	1,000,000	100.0%	0	835,819	999999999
firstname	1,000,000	100.0%	0	78,136	EAMSTRMT
lastname	1,000,000	100.0%	0	177,001	ERJSAXA
address	1,000,000	100.0%	0	828,774	123 MAIN ST

zip5	1,000,000	100.0%	0	26,370	68138
homephone	1,000,000	100.0%	0	28,244	9999999999
fraud_label	1,000,000	100.0%	985,607	2	0

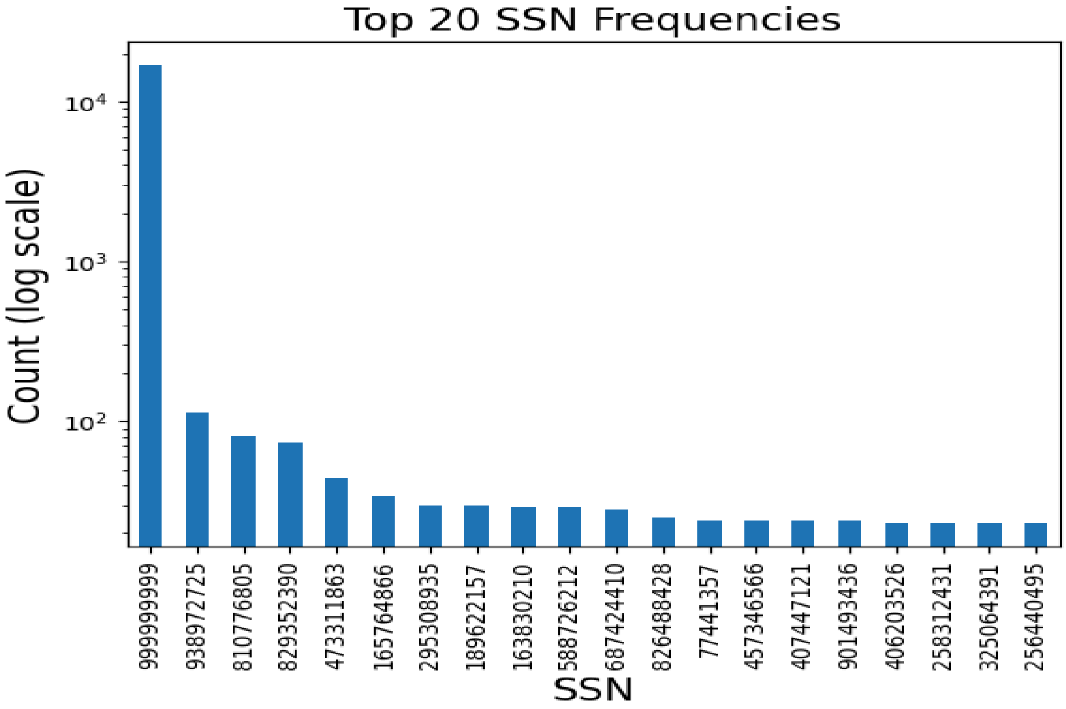
2. Important Field Distributions

2.1 Field Name: date



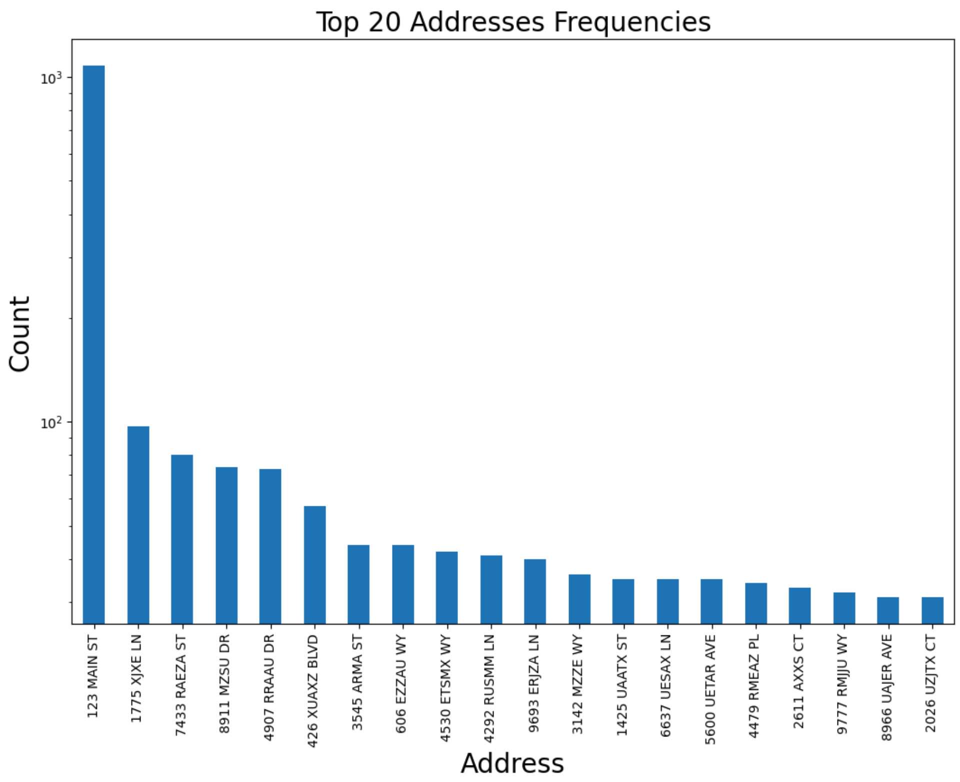
The plot illustrates the daily distribution of submitted applications throughout 2017. Application counts fluctuate between 2,600 and 2,850 per day, without a clear upward or downward trend. The data shows short-term volatility, with some periods experiencing noticeable spikes or drops in applications. This variation could be influenced by external factors like market conditions or promotional events. Overall, the number of daily submissions appears consistent, with no significant anomalies or outliers over the observed period.

2.2 Field Name: ssn



The plot displays the top 20 most frequent Social Security Numbers (SSNs) in the dataset, using a logarithmic scale for the count. The most common SSN, 999999999, appears disproportionately more often than any other, suggesting it is likely a placeholder or default value. Other SSNs, such as 893777275 and 810776805, also appear frequently but at much lower counts. This distribution indicates that a small number of SSNs are being overused, which could be indicative of synthetic identities or data entry issues. The presence of such outliers in the SSN field warrants further investigation to assess the validity of these applications.

2.3 Field Name: address



The plot shows the top 20 most frequent addresses in the dataset. The most common address, 123 MAIN ST, appears significantly more often than any other, suggesting it is a placeholder or a default value, with a count that is orders of magnitude higher than the rest. Other frequent addresses, such as 2175 XYE LM and 7419 RMEAZ ST, may also represent data entry issues or synthetic records. The use of such common placeholder addresses can distort the analysis and should be flagged for further investigation, particularly in identity verification and fraud detection scenarios.

Section 3. Data Cleaning

For this dataset, there were no outliers or exclusions identified. However, we addressed several fields with frivolous values that could create incorrect associations or skew fraud detection.

1. Exclusions

We have no exclusions.

2. Outlier Treatment

We have no outliers.

3. Handling Frivolous Value

Why is this problematic?
Frivolous values in data, such as placeholders or fabricated entries, can distort analysis by falsely linking unrelated records or creating misleading patterns. These values are often used when data is missing, but they must be addressed to ensure accurate results in subsequent analysis.

3.1 Field Name: ssn

In the SSN field, 16,935 records contained the placeholder value "999999999," typically used when the Social Security Number is unavailable. These frivolous values could cause inaccuracies by linking unrelated records or skewing models like fraud detection. To correct this, we replaced each instance of "999999999" with the unique identifier from the 'record' field, ensuring each entry was assigned a distinct value. This prevents false connections between records and maintains data accuracy. After making these replacements, we confirmed that all SSN values were unique and no placeholder values remained, preserving the integrity of the dataset for further analysis.

3.2 Field Name: address

In the address field, "123 MAIN ST" appeared 1,079 times, an unusually high frequency indicating it served as a placeholder or default value. If left untreated, this placeholder could create false linkages between records, leading to incorrect associations or analysis, such as generating false positives in fraud detection. To address this, we replaced each instance of "123 MAIN ST" with a unique value based on the 'record' field, ensuring that each record was distinct. After replacing the placeholder, we confirmed that "123 MAIN ST" no longer appeared in the dataset.

3.3 Field Name: homephone

The homephone field contained 78,512 instances of “9999999999,” a placeholder often used when phone numbers are unavailable. If left untreated, this would result in incorrect associations between unrelated records, as many individuals would appear to share the same phone number. To prevent this, we replaced all occurrences of “9999999999” with a unique identifier derived from the ‘record’ field. This ensured that each record remained distinct, maintaining the accuracy of the dataset without introducing false links due to placeholder phone numbers.

3.4 Field Name: dob

For the date of birth (dob) field, the value “1907-06-26” appeared 126,568 times, an unlikely number that indicates it served as a placeholder. Such a value could lead to incorrect conclusions by falsely linking unrelated records or suggesting artificial patterns in the data. To address this, we identified these overrepresented dates and replaced them with the ‘record’ number to prevent erroneous links between different records. This step ensured that analysis based on birth dates was not influenced by placeholder values and instead reflected legitimate patterns and behaviors in the dataset.

Section 4. Variable Creation

In fraud detection, specific patterns and behaviors in the data can indicate fraudulent activity. Fraudsters often exploit gaps in verification processes by reusing or manipulating identifiable information across multiple applications, such as names, addresses, Social Security Numbers (SSNs), and phone numbers. To capture these behaviors and detect anomalies, we created several new variables, focusing on patterns in frequency, recency, and combinations of key personal attributes.

Fraud typically manifests through repeated attempts to submit fraudulent applications, often using similar or identical information. By analyzing temporal and frequency-based patterns, we can identify unusual spikes in activity or repetitive use of certain identifiers. For example, if an SSN or address appears frequently across applications within a short time frame, it may indicate a synthetic identity or coordinated fraud attempt. Thus, the variables created aim to detect these signs of fraud by analyzing how often, when, and in what combinations certain attributes are used.

4.1 Variable Table

Description of Variables	# Variables Created
dob_dt: Date of birth in datetime format, converted from dob to enable age calculations and other date-based analyses.	1
age_when_apply: Applicant's age at the application date, calculated from date and dob_dt.	1
dow: Day of the week for the application date, extracted from date.	1
dow_risk: Smoothed average fraud risk associated with each day of the week, based on fraud_label. This encoding helps capture the risk pattern across different days.	1
Entity Combination Variables: Unique identifiers created by combining key personal attributes, such as name, address, date of birth, phone number, and ssn.	18
check_date: A duplicate of the date column, potentially used for verification or tracking	1

within df.	
check_record: A duplicate of the record column, likely used for the same purpose as check_date.	1
Days Since Variables: For each attribute in attributes, a new variable calculates the number of days since the last occurrence of that attribute (e.g., address_day_since, ssn_day_since).	14
Velocity Variables: For each attribute and each specified day interval (0, 1, 3, 7, 14, 30), a variable captures the count of occurrences within that time window, reflecting the frequency of activity.	84
Relative Velocity Variables: These variables are ratios of counts across different time intervals for each attribute, providing relative velocity measures that highlight changes in activity over varying periods.	112
Unique Count Variables: These variables represent the count of unique occurrences of a secondary attribute within specified time intervals for each primary attribute, helping identify distinct patterns and relationships over time.	1274
Maximum Indicator Variables: These variables capture the maximum count of occurrences for each attribute within specific time intervals (1, 3, 7, and 30 days), providing insights into peak activity levels for each attribute over time.	56
Age Indicator Variables: These variables represent the maximum, mean, and minimum age of applicants associated with each attribute, providing insights into the age distribution linked to different entity attributes.	42
Additional New Variables	
Recency Indicator Variables: Calculate the number of days since the last activity or occurrence for each attribute. This can help identify recent interactions and might be useful in spotting unusual spikes in activity.	1
Frequency Ratio Variables: Calculate the ratio of occurrences of each attribute over different time periods (e.g., 7 days vs. 30 days). This can show if an attribute's activity is accelerating or decelerating.	1

Change in Activity Variables: Calculate the difference in activity between two time periods (e.g., 1 day and 7 days) to see if there's a sudden change in activity. Sudden increases or decreases may signal unusual behavior.	1
Standard Deviation of Age by Attribute: For each attribute, calculate the standard deviation of applicants' age to understand the diversity in age distribution.	1

Section 5. Feature Selection

Feature selection was a crucial step in the modeling process to identify the most relevant and predictive variables, ultimately enhancing model performance. The goal was to isolate variables that most effectively distinguish between fraudulent and non-fraudulent applications, ensuring that the model focuses on key indicators of fraud without being overburdened by irrelevant features.

What: The feature selection process involved testing multiple filters and models to evaluate the impact of different feature sets on the model's Fraud Detection Rate (FDR) at a 3% threshold. The features selected included frequency-based, recency-based, and entity combination variables designed to capture patterns commonly associated with fraudulent behavior, such as repeated or rapid application submissions using the same identifiers (e.g., SSN, address).

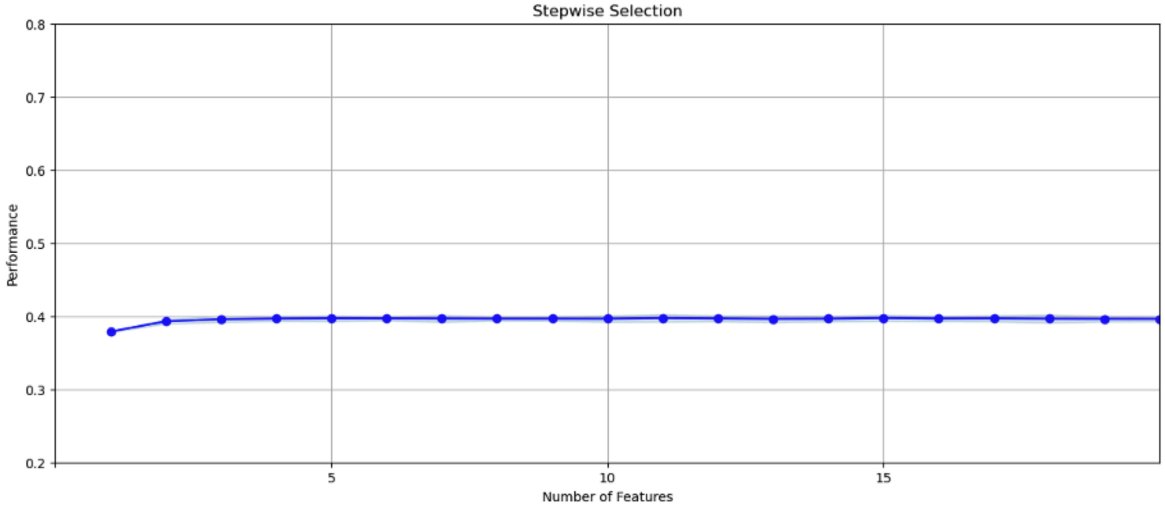
Why: Selecting the right features was essential to improve the model's accuracy, reduce overfitting, and enhance interpretability. By focusing on variables that strongly correlate with fraud, the model can better detect fraudulent patterns while avoiding noise from irrelevant data. This also helps in generalizing the model across datasets, as it can rely on core fraud indicators rather than arbitrary patterns.

How: Several approaches were tested, including forward and backward selection, using Random Forest and LightGBM classifiers with various filter and selection type settings. Through an iterative process:

Model	**num_filter**	**num_wrapper**	**Selection Type**
RandomForestClassifier	50	20	Forward
LGBMClassifier	50	20	Forward
LGBMClassifier	50	20	Backward
RandomForestClassifier	100	20	Forward
(Final) LGBMClassifier	100	20	Forward

5.1 Exploration Plots

5.1.1 Random Forest Classifier: Number of filters: 50, Forward Selection

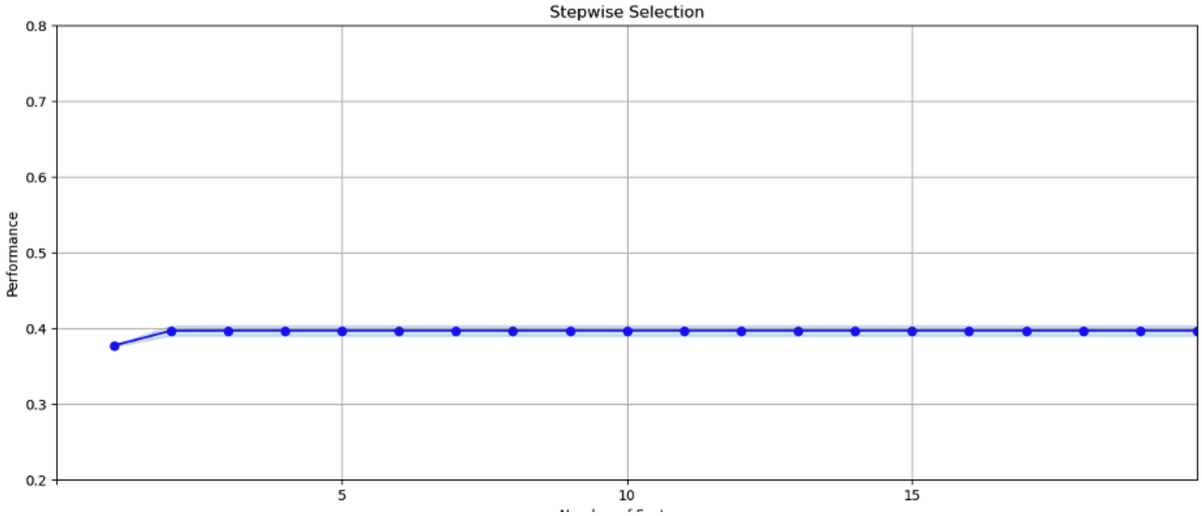


wrapper order	variable	filter score
1	max_count_by_fulladdress_30	0.360
2	address_unique_count_for_name_homephone_60	0.292
3	max_count_by_address_30	0.359
4	fulladdress_count_30	0.332
5	address_unique_count_for_name_homephone_14	0.278
6	address_unique_count_for_dob_14	0.274
7	fulladdress_unique_count_for_dob_14	0.275
8	address_unique_count_for_ssn_30	0.281
9	fulladdress_unique_count_for_name_dob_60	0.285
10	fulladdress_unique_count_for_dob_60	0.283
11	address_unique_count_for_name_homephone_30	0.285
12	address_unique_count_for_name_14	0.276
13	fulladdress_unique_count_for_ssn_14	0.277
14	address_count_30	0.333
15	address_count_14	0.322
16	fulladdress_unique_count_for_name_homephone_30	0.284

17	fulladdress_unique_count_for_dob_30	0.279
18	address_unique_count_for_ssn_14	0.276
19	address_unique_count_for_ssn_60	0.286
20	fulladdress_unique_count_for_ssn_60	0.287

Performance: Initial FDR@3% was below the target threshold of 0.6. Although the model captured some predictive variables, this filter number did not provide the necessary lift in performance. Exploration plots highlighted the importance of a few key variables, but the model's ability to generalize was limited, indicating that more filters were needed to capture sufficient predictive power.

5.1.2 LGBM Classifier: Number of filters: 50, Forward Selection

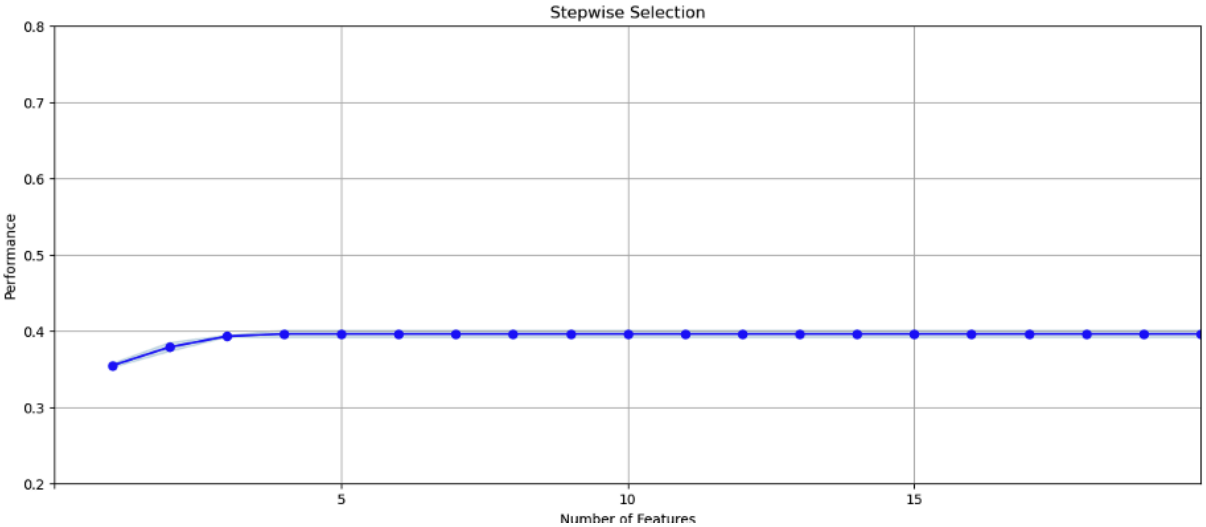


wrapper order	variable	filter score
1	max_count_by_address_30	0.359
2	address_unique_count_for_name_homephone_60	0.292
3	fulladdress_unique_count_for_ssn_60	0.287
4	max_count_by_fulladdress_3	0.330
5	max_count_by_address_7	0.343
6	max_count_by_address_3	0.329
7	max_count_by_address_1	0.315
8	max_count_by_fulladdress_1	0.315
9	address_unique_count_for_name_60	0.287
10	address_unique_count_for_ssn_60	0.286

11	address_unique_count_for_name_homephone_30	0.286
12	fulladdress_unique_count_for_name_homephone_30	0.284
13	fulladdress_unique_count_for_dob_60	0.283
14	address_unique_count_for_dob_60	0.282
15	fulladdress_unique_count_for_ssn_30	0.282
16	address_unique_count_for_ssn_30	0.281
17	address_unique_count_for_name_30	0.281
18	fulladdress_unique_count_for_name_dob_30	0.280
19	fulladdress_unique_count_for_dob_30	0.279
20	address_count_3	0.279

Performance: Similar to Random Forest, the LightGBM model with 50 filters showed minimal improvement in FDR@3%, with performance leveling off around 0.4–0.5 across iterations. Although LightGBM generally provided better separation between fraudulent and non-fraudulent classes than Random Forest, this configuration still fell short of the target, indicating that 50 filters were insufficient to capture essential features for reliable fraud detection.

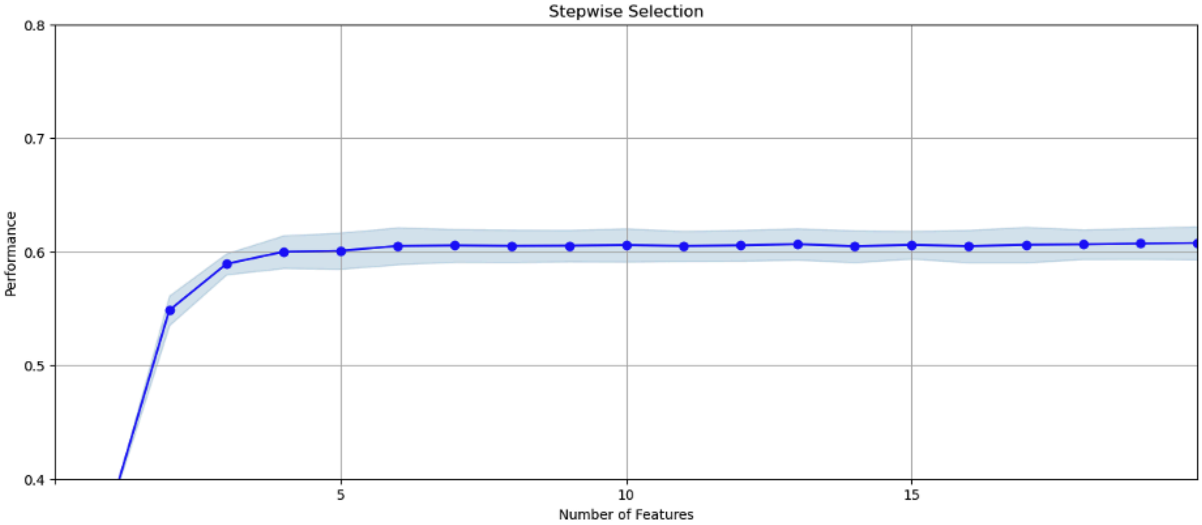
5.1.3 LGBM Classifier: Number of filters: 50, Backward Selection



wrapper order	variable	filter score
1	max_count_by_fulladdress_30	0.360
2	address_unique_count_for_ssn_7	0.273
3	fulladdress_unique_count_for_ssn_7	0.273
4	fulladdress_unique_count_for_name_homephone_7	0.273
5	address_unique_count_for_name_homephone_7	0.274
6	address_unique_count_for_dob_14	0.274
7	fulladdress_unique_count_for_dob_14	0.275
8	fulladdress_unique_count_for_name_dob_14	0.275
9	address_unique_count_for_name_14	0.276
10	address_unique_count_for_ssn_14	0.276
11	fulladdress_unique_count_for_ssn_14	0.277
12	address_unique_count_for_dob_30	0.278
13	fulladdress_unique_count_for_name_homephone_14	0.278
14	address_unique_count_for_name_homephone_14	0.278
15	address_count_3	0.279
16	fulladdress_count_3	0.279
17	fulladdress_unique_count_for_name_dob_30	0.280
18	address_unique_count_for_name_30	0.281
19	address_unique_count_for_ssn_30	0.281
20	fulladdress_count_0_by_14	0.281

Performance: The backward selection approach with LightGBM and 50 filters showed similar limitations. The plot demonstrates only a slight improvement in performance with additional features, ultimately stabilizing well below the desired FDR@3% threshold. This configuration did not enhance model performance meaningfully compared to forward selection, suggesting that backward selection did not provide additional benefit with this filter number.

5.1.4 Random Forest Classifier: Number of filters: 100, Forward Selection

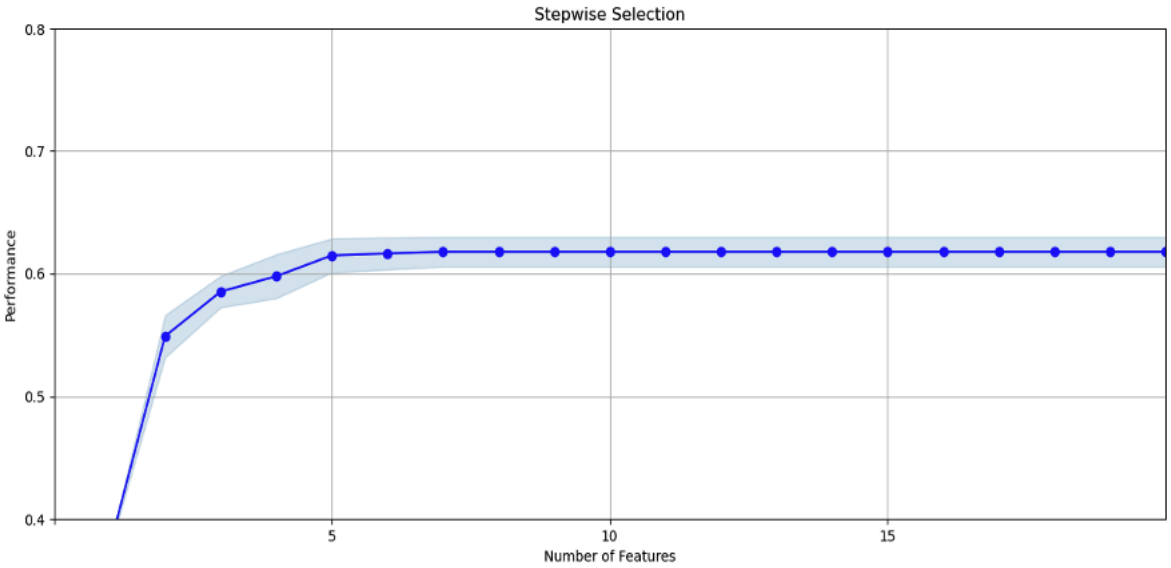


wrapper order	variable	filter score
1	max_count_by_fulladdress_30	0.360
2	max_count_by_ssn_30	0.241
3	max_count_by_homephone_7	0.232
4	zip5_count_1	0.221
5	fulladdress_homephone_count_30	0.231
6	fulladdress_count_3	0.279
7	max_count_by_fulladdress_homephone_7	0.234
8	max_count_by_fulladdress_7	0.343
9	fulladdress_count_0_by_14	0.281
10	max_count_by_fulladdress_homephone_3	0.220
11	address_unique_count_for_name_3	0.263
12	fulladdress_unique_count_for_name_homephone_7	0.273
13	fulladdress_unique_count_for_name_homephone_14	0.278
14	fulladdress_count_0_by_3	0.231
15	fulladdress_count_0_by_30	0.291
16	fulladdress_unique_count_for_ssn_60	0.287
17	max_count_by_fulladdress_homephone_30	0.250
18	address_unique_count_for_dob_7	0.272
19	address_count_7	0.302

20	address_unique_count_for_ssn_1	0.244

Performance: With an increase to 100 filters, the Random Forest model demonstrated a noticeable improvement in FDR@3%, initially rising to around 0.6 before plateauing. The performance plot shows that adding more features resulted in higher predictive accuracy in the initial stages, though it stabilized afterward. This configuration achieved better results than previous iterations, but further improvements might still be needed to ensure consistency across datasets.

Final Model Choice: LGBM Classifier: Number of filters: 100, Forward Selection



Final Top 20 Variables with KS Scores

wrapper order	variable	filter score
1	max_count_by_fulladdress_30	0.360
2	max_count_by_address_30	0.359
3	max_count_by_address_7	0.343
4	max_count_by_fulladdress_7	0.343
5	address_day_since	0.334
6	fulladdress_day_since	0.333
7	address_count_30	0.333
8	fulladdress_count_30	0.332

9	max_count_by_fulladdress_3	0.330
10	max_count_by_address_3	0.329
11	address_count_14	0.322
12	fulladdress_count_14	0.322
13	max_count_by_address_1	0.315
14	max_count_by_fulladdress_1	0.315
15	address_count_7	0.302
16	fulladdress_count_7	0.302
17	address_unique_count_for_name_homephone_60	0.292
18	address_count_0_by_30	0.292
19	fulladdress_count_0_by_30	0.291
20	fulladdress_unique_count_for_name_homephone_60	0.290

Performance: The LightGBM model with 100 filters and forward selection demonstrated the best results among all tested configurations. The performance plot shows a sharp increase in FDR@3% as the number of features grows, reaching and surpassing the 0.6 target. This model maintained a stable performance level across different feature counts, confirming that 100 filters struck an optimal balance. The forward selection process was efficient, achieving the highest FDR@3% on both test and OOT datasets, and provided a reliable separation between classes, making it the best choice as the final model.

Section 6. Preliminary Models Exploration

During model exploration, five machine learning algorithms—one linear and four non-linear—were evaluated to effectively identify fraudulent transactions. Each model was tuned to optimize performance, balancing model complexity to prevent overfitting and maintain generalization across datasets.

Logistic Regression: Serving as a baseline model, Logistic Regression is a linear classifier known for its simplicity. Different regularization strengths (C) were tested with both L1 and L2 penalties, and various solvers like liblinear, lbfgs, and saga were experimented with to control overfitting. The model showed moderate performance, with the highest FDR@3% on the out-of-time (OOT) dataset reaching 0.589 in iteration 4 (penalty: L2, C: 0.2, solver: lbfgs, max_iter: 300).

Decision Tree: As a non-linear model, Decision Trees can capture complex relationships in data. Parameters such as max_depth, min_samples_split, and min_samples_leaf were adjusted to find an optimal balance. The best performance occurred in iteration 5 (max_depth: 5, min_samples_split: 12, min_samples_leaf: 5), achieving an FDR@3% of 0.593 on the OOT dataset. This model offered flexibility but showed limitations in generalization, with FDR@3% scores fluctuating across training, test, and OOT datasets.

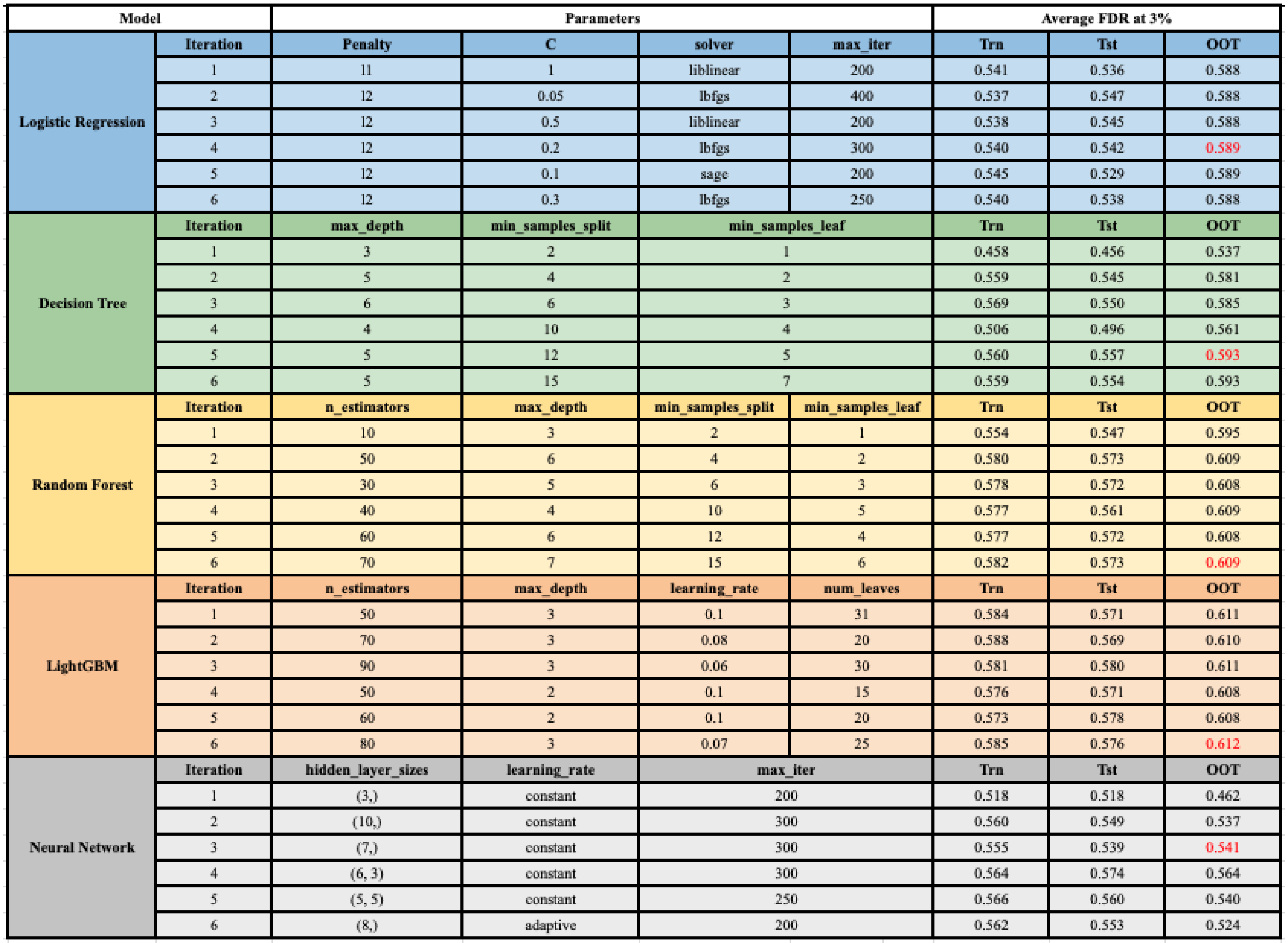
Random Forest: Random Forests combine predictions from multiple decision trees, reducing variance and improving predictive stability. Using a range of n_estimators from 10 to 70, max_depth, min_samples_split, and min_samples_leaf were optimized. The highest FDR@3% on the OOT dataset was 0.609 in iteration 2 (n_estimators: 50, max_depth: 6, min_samples_split: 4, min_samples_leaf: 2). Random Forest demonstrated strong performance, particularly in avoiding overfitting, but required higher computational resources.

LightGBM: A gradient-boosting algorithm, LightGBM is known for its efficiency with large datasets. Different combinations of n_estimators, max_depth, learning_rate, and num_leaves were experimented with. The best-performing iteration, iteration 6 (n_estimators: 80, max_depth: 3, learning_rate: 0.07, num_leaves: 25), achieved an FDR@3% of 0.612 on the OOT dataset, making it the top-performing model in this exploration. LightGBM was highly effective, capturing complex patterns while maintaining stability across datasets.

Neural Network (MLPClassifier): Finally, a Multi-Layer Perceptron (MLP) neural network was tested, with adjustments to hidden_layer_sizes, learning_rate, and max_iter to balance complexity and training stability. Although the neural network showed some potential, its performance was inconsistent. The highest FDR@3% on the OOT dataset was 0.564 in iteration 4 (hidden_layer_sizes: (6, 3), learning_rate: constant, max_iter: 300), but it struggled with stability, as shown by fluctuating scores in other iterations.

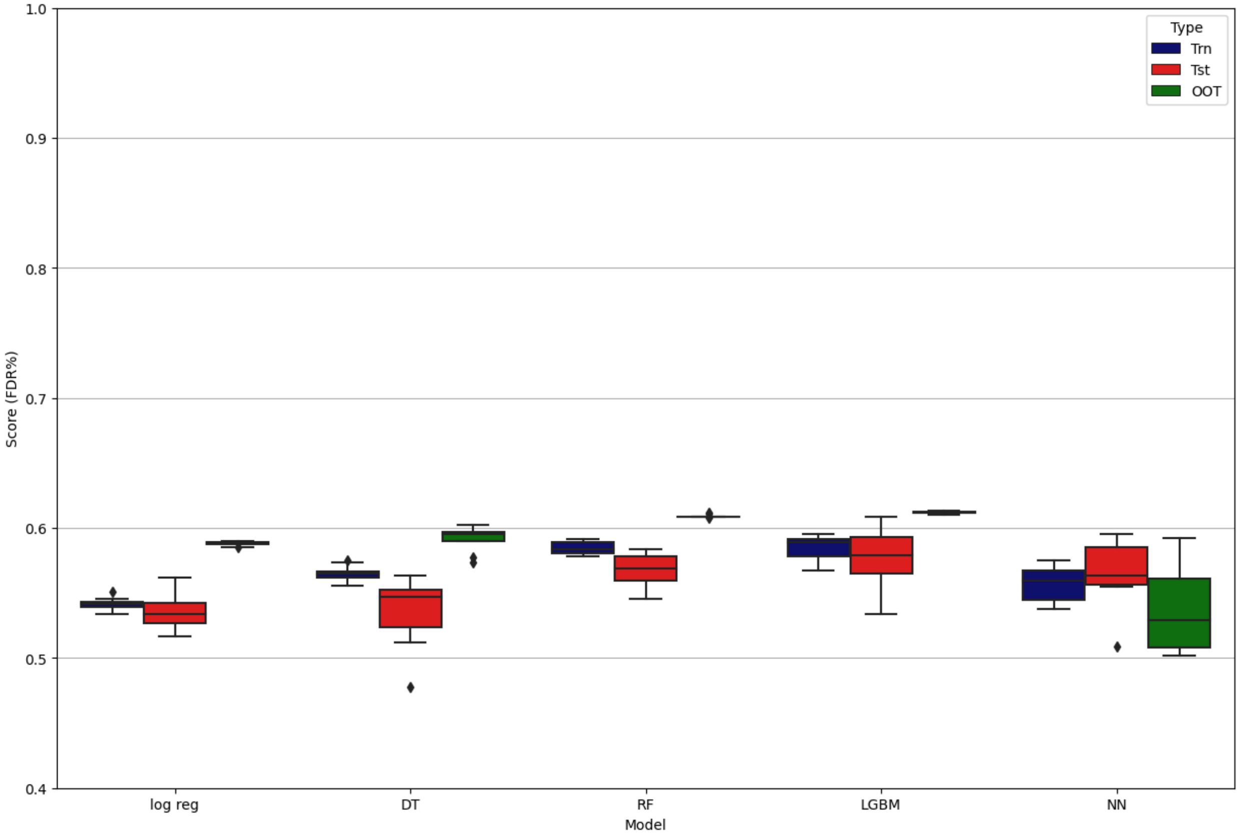
Each model was evaluated on training, test, and OOT datasets to assess performance consistency and overfitting risks. Performance metrics, particularly FDR@3%, provided insights into each algorithm's effectiveness in capturing fraud patterns. Through this exploration, LightGBM emerged as the best choice due to its superior performance, efficient learning, and consistent results across datasets, achieving the highest FDR@3% of 0.612 on the OOT dataset.z

6.1 Model Exploration Table



Model		Parameters				Average FDR at 3%		
Logistic Regression	Iteration	Penalty	C	solver	max_iter	Trn	Tst	OOT
	1	l1	1	liblinear	200	0.541	0.536	0.588
	2	l2	0.05	lbfgs	400	0.537	0.547	0.588
	3	l2	0.5	liblinear	200	0.538	0.545	0.588
	4	l2	0.2	lbfgs	300	0.540	0.542	0.589
	5	l2	0.1	sage	200	0.545	0.529	0.589
	6	l2	0.3	lbfgs	250	0.540	0.538	0.588
Decision Tree	Iteration	max_depth	min_samples_split	min_samples_leaf		Trn	Tst	OOT
	1	3	2	1		0.458	0.456	0.537
	2	5	4	2		0.559	0.545	0.581
	3	6	6	3		0.569	0.550	0.585
	4	4	10	4		0.506	0.496	0.561
	5	5	12	5		0.560	0.557	0.593
	6	5	15	7		0.559	0.554	0.593
Random Forest	Iteration	n_estimators	max_depth	min_samples_split	min_samples_leaf	Trn	Tst	OOT
	1	10	3	2	1	0.554	0.547	0.595
	2	50	6	4	2	0.580	0.573	0.609
	3	30	5	6	3	0.578	0.572	0.608
	4	40	4	10	5	0.577	0.561	0.609
	5	60	6	12	4	0.577	0.572	0.608
	6	70	7	15	6	0.582	0.573	0.609
LightGBM	Iteration	n_estimators	max_depth	learning_rate	num_leaves	Trn	Tst	OOT
	1	50	3	0.1	31	0.584	0.571	0.611
	2	70	3	0.08	20	0.588	0.569	0.610
	3	90	3	0.06	30	0.581	0.580	0.611
	4	50	2	0.1	15	0.576	0.571	0.608
	5	60	2	0.1	20	0.573	0.578	0.608
	6	80	3	0.07	25	0.585	0.576	0.612
Neural Network	Iteration	hidden_layer_sizes	learning_rate	max_iter		Trn	Tst	OOT
	1	(3,)	constant	200		0.518	0.518	0.462
	2	(10,)	constant	300		0.560	0.549	0.537
	3	(7,)	constant	300		0.555	0.539	0.541
	4	(6, 3)	constant	300		0.564	0.574	0.564
	5	(5, 5)	constant	250		0.566	0.560	0.540
	6	(8,)	adaptive	200		0.562	0.553	0.524

6.2 Performance Plot



The **LightGBM model** was chosen as the final model due to its strong and consistent performance across the training (Trn), test (Tst), and out-of-time (OOT) datasets. As illustrated in the performance plot, LightGBM maintained a close balance between train and test FDR@3% scores, which indicates that the model generalizes well and is not overfitting to the training data. With a well-tuned set of hyperparameters—such as n_estimators set to 80, max_depth at 3, learning_rate at 0.07, and num_leaves at 25—LightGBM captured complex fraud patterns effectively. Notably, it achieved an FDR@3% of 0.612 on the OOT dataset, the highest among all models explored, demonstrating its robustness on unseen data. This balance across train and test datasets, along with the high OOT performance, confirms LightGBM's suitability for fraud detection, as it reliably identifies fraudulent patterns across diverse data samples.

Section 7: Final Model Performance

The final model selected for fraud detection was LightGBM, chosen for its superior performance, consistency across datasets, and ability to generalize well without overfitting. The LightGBM model effectively captured complex fraud patterns, resulting in the highest FDR@3% on the out-of-time (OOT) dataset, which is critical for evaluating real-world performance on new, unseen data.

Best Model Choice - Non-default Hyperparameter Choices

Final Model	n_estimators	max_depth	learning_rate	num_leaves
LightGBM	80	3	0.07	25

The following hyperparameters were optimized to achieve this performance:

n_estimators: Set to 80, this parameter controls the number of boosting rounds. A moderate value of 80 provided a balance between performance and computational efficiency, allowing the model to learn sufficient patterns without excessive rounds that could lead to overfitting.

max_depth: Limited to 3, this parameter controls the maximum depth of each tree in the ensemble. Setting a low depth prevented the model from becoming too complex, reducing the risk of overfitting while still capturing meaningful interactions between features.

learning_rate: Set to 0.07, this parameter determines the rate at which the model learns from each boosting round. A learning rate of 0.07 allowed the model to make gradual improvements, stabilizing its performance across the train, test, and OOT datasets without abrupt jumps in prediction accuracy.

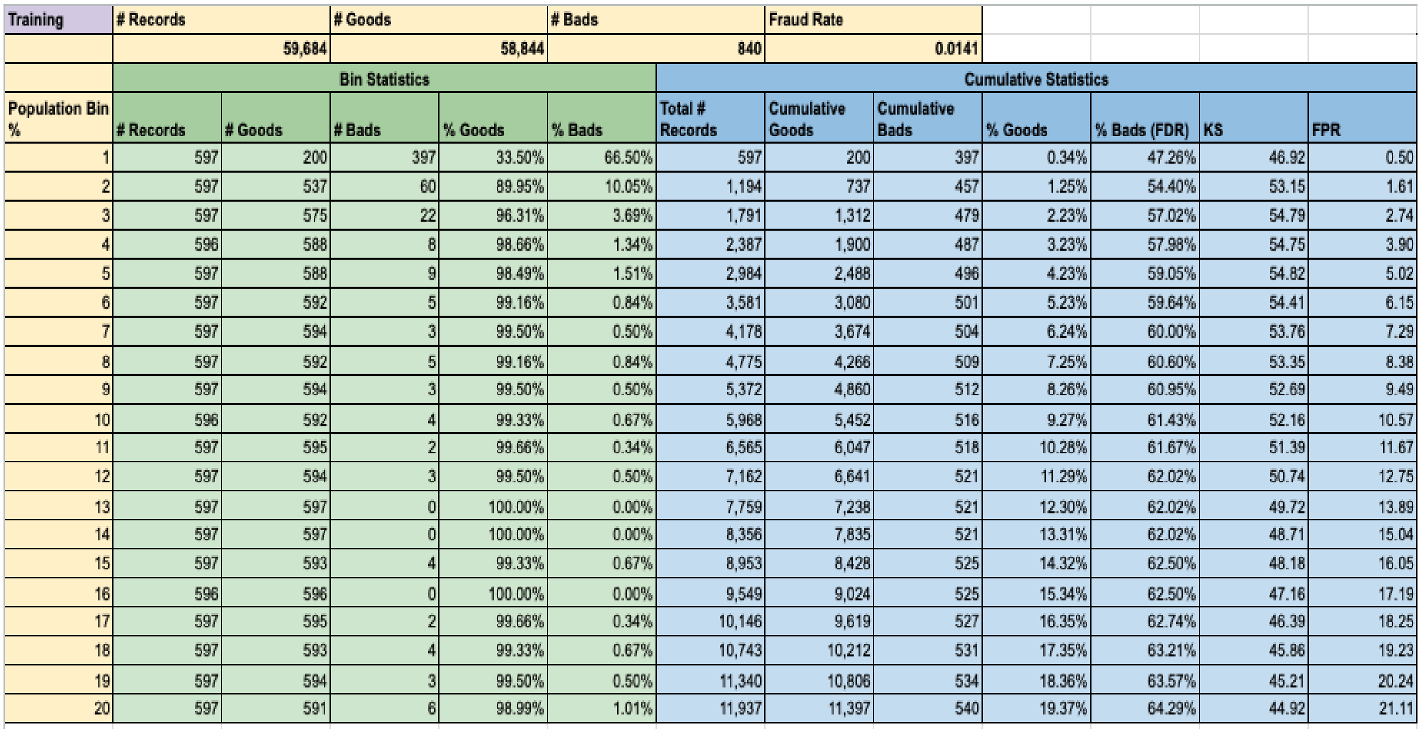
num_leaves: Set to 25, this parameter controls the maximum number of leaves per tree. With 25 leaves, the model maintained sufficient complexity to capture subtle patterns in the data without making the trees overly intricate, which could introduce noise.

The model’s performance metrics show its effectiveness in detecting fraudulent cases with minimal risk of overfitting. On the training dataset, the model achieved an FDR@3% of 0.585, and on the test dataset, the FDR@3% was 0.576. These similar scores across training and test sets indicate that the model is well-balanced and has not overfitted to the training data, maintaining stability and generalization across datasets.

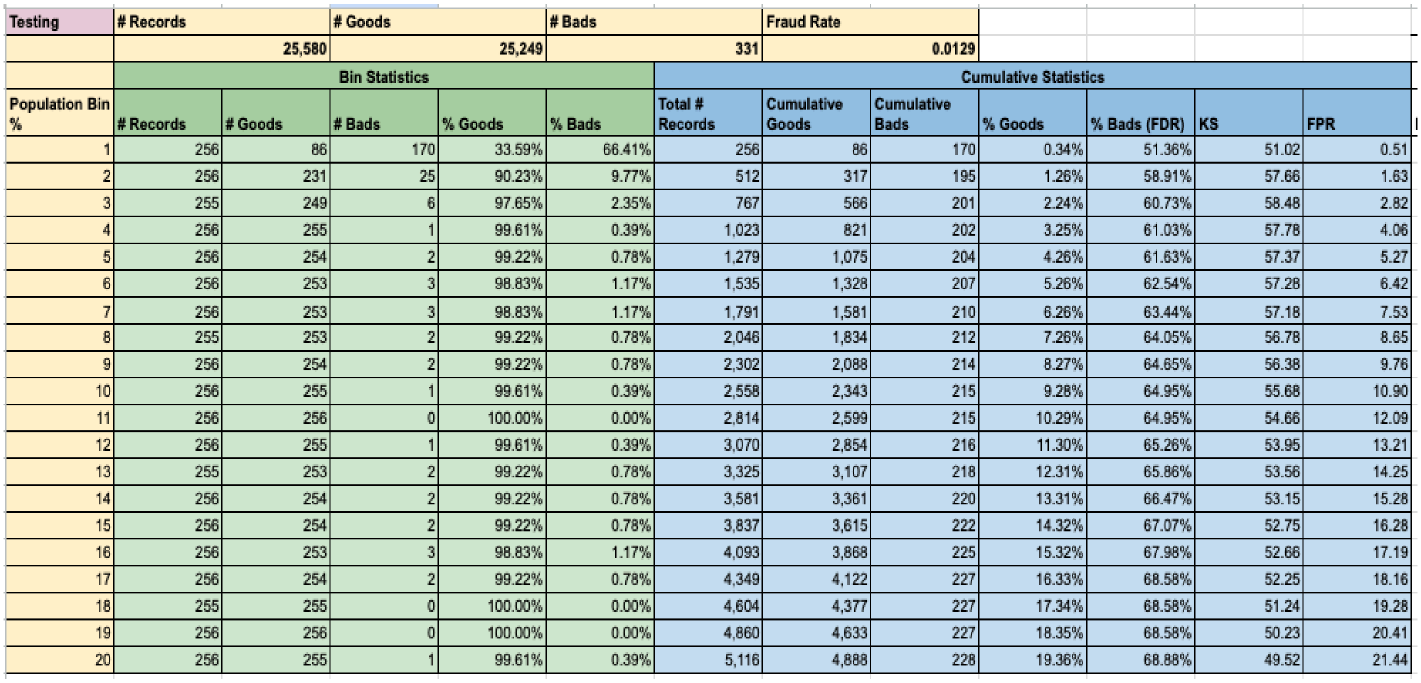
The highest FDR@3% score of 0.612 was observed on the OOT dataset, highlighting the model's robustness and its ability to generalize effectively to new, unseen data. This result underscores the model's reliability in real-world applications, where it can effectively detect fraud in datasets it hasn't encountered before.

By configuring the model with 80 estimators, a learning rate of 0.07, a maximum depth of 3, and 25 leaves, the LightGBM model achieved a strong balance between complexity and predictive power. This final model configuration allows for accurate fraud detection while controlling false positives, making it the optimal choice for deployment in the bank's fraud detection system.

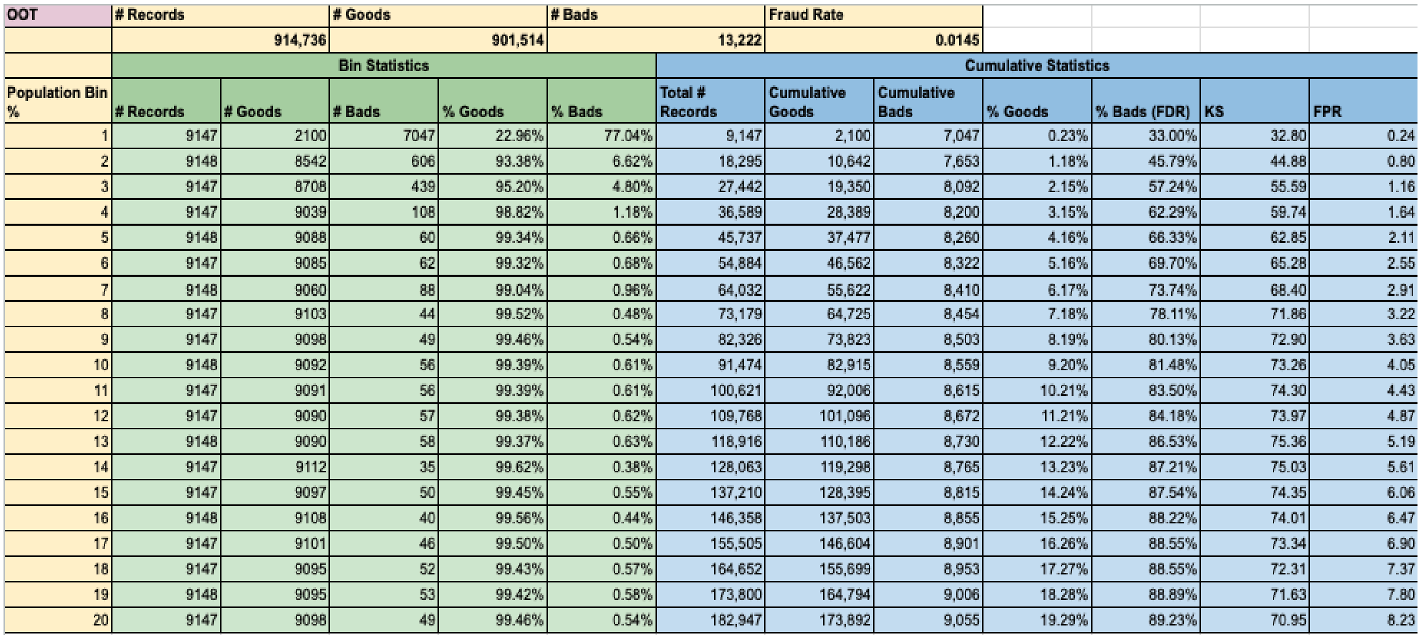
7.1 Three Summary Tables



Training	# Records		# Goods		# Bads		Fraud Rate					
		59,684		58,844		840		0.0141				
	Bin Statistics						Cumulative Statistics					
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	597	200	397	33.50%	66.50%	597	200	397	0.34%	47.26%	46.92	0.50
2	597	537	60	89.95%	10.05%	1,194	737	457	1.25%	54.40%	53.15	1.61
3	597	575	22	96.31%	3.69%	1,791	1,312	479	2.23%	57.02%	54.79	2.74
4	596	588	8	98.66%	1.34%	2,387	1,900	487	3.23%	57.98%	54.75	3.90
5	597	588	9	98.49%	1.51%	2,984	2,488	496	4.23%	59.05%	54.82	5.02
6	597	592	5	99.16%	0.84%	3,581	3,080	501	5.23%	59.64%	54.41	6.15
7	597	594	3	99.50%	0.50%	4,178	3,674	504	6.24%	60.00%	53.76	7.29
8	597	592	5	99.16%	0.84%	4,775	4,266	509	7.25%	60.60%	53.35	8.38
9	597	594	3	99.50%	0.50%	5,372	4,860	512	8.26%	60.95%	52.69	9.49
10	596	592	4	99.33%	0.67%	5,968	5,452	516	9.27%	61.43%	52.16	10.57
11	597	595	2	99.66%	0.34%	6,565	6,047	518	10.28%	61.67%	51.39	11.67
12	597	594	3	99.50%	0.50%	7,162	6,641	521	11.29%	62.02%	50.74	12.75
13	597	597	0	100.00%	0.00%	7,759	7,238	521	12.30%	62.02%	49.72	13.89
14	597	597	0	100.00%	0.00%	8,356	7,835	521	13.31%	62.02%	48.71	15.04
15	597	593	4	99.33%	0.67%	8,953	8,428	525	14.32%	62.50%	48.18	16.05
16	596	596	0	100.00%	0.00%	9,549	9,024	525	15.34%	62.50%	47.16	17.19
17	597	595	2	99.66%	0.34%	10,146	9,619	527	16.35%	62.74%	46.39	18.25
18	597	593	4	99.33%	0.67%	10,743	10,212	531	17.35%	63.21%	45.86	19.23
19	597	594	3	99.50%	0.50%	11,340	10,806	534	18.36%	63.57%	45.21	20.24
20	597	591	6	98.99%	1.01%	11,937	11,397	540	19.37%	64.29%	44.92	21.11



Testing	# Records		# Goods		# Bads		Fraud Rate					
		25,580		25,249		331		0.0129				
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	256	86	170	33.59%	66.41%	256	86	170	0.34%	51.36%	51.02	0.51
2	256	231	25	90.23%	9.77%	512	317	195	1.26%	58.91%	57.66	1.63
3	255	249	6	97.65%	2.35%	767	566	201	2.24%	60.73%	58.48	2.82
4	256	255	1	99.61%	0.39%	1,023	821	202	3.25%	61.03%	57.78	4.06
5	256	254	2	99.22%	0.78%	1,279	1,075	204	4.26%	61.63%	57.37	5.27
6	256	253	3	98.83%	1.17%	1,535	1,328	207	5.26%	62.54%	57.28	6.42
7	256	253	3	98.83%	1.17%	1,791	1,581	210	6.26%	63.44%	57.18	7.53
8	255	253	2	99.22%	0.78%	2,046	1,834	212	7.26%	64.05%	56.78	8.65
9	256	254	2	99.22%	0.78%	2,302	2,088	214	8.27%	64.65%	56.38	9.76
10	256	255	1	99.61%	0.39%	2,558	2,343	215	9.28%	64.95%	55.68	10.90
11	256	256	0	100.00%	0.00%	2,814	2,599	215	10.29%	64.95%	54.66	12.09
12	256	255	1	99.61%	0.39%	3,070	2,854	216	11.30%	65.26%	53.95	13.21
13	255	253	2	99.22%	0.78%	3,325	3,107	218	12.31%	65.86%	53.56	14.25
14	256	254	2	99.22%	0.78%	3,581	3,361	220	13.31%	66.47%	53.15	15.28
15	256	254	2	99.22%	0.78%	3,837	3,615	222	14.32%	67.07%	52.75	16.28
16	256	253	3	98.83%	1.17%	4,093	3,868	225	15.32%	67.98%	52.66	17.19
17	256	254	2	99.22%	0.78%	4,349	4,122	227	16.33%	68.58%	52.25	18.16
18	255	255	0	100.00%	0.00%	4,604	4,377	227	17.34%	68.58%	51.24	19.28
19	256	256	0	100.00%	0.00%	4,860	4,633	227	18.35%	68.58%	50.23	20.41
20	256	255	1	99.61%	0.39%	5,116	4,888	228	19.36%	68.88%	49.52	21.44



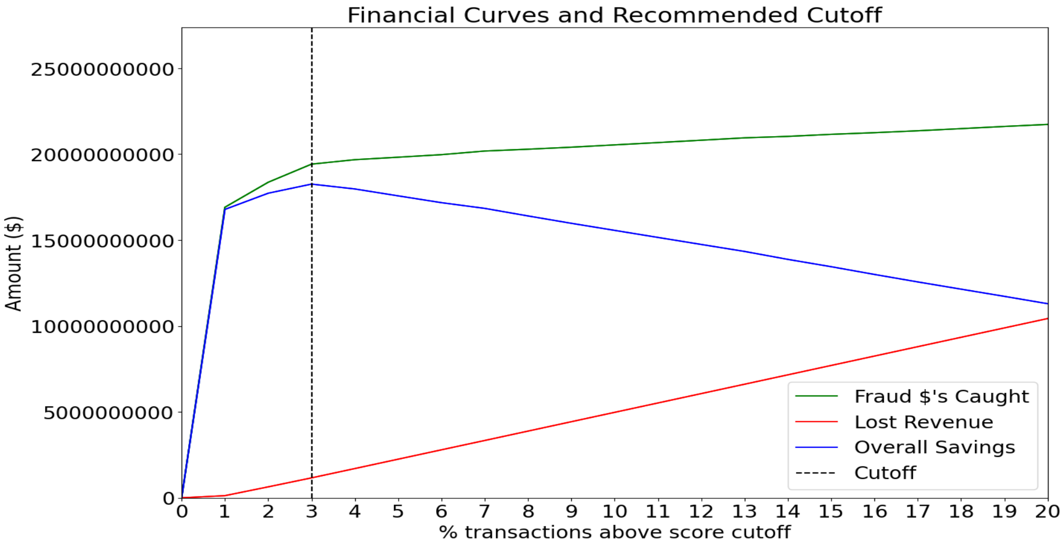
OOT	# Records		# Goods		# Bads		Fraud Rate					
		914,736		901,514		13,222		0.0145				
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	9147	2100	7047	22.96%	77.04%	9,147	2,100	7,047	0.23%	33.00%	32.80	0.24
2	9148	8542	606	93.38%	6.62%	18,295	10,642	7,653	1.18%	45.79%	44.88	0.80
3	9147	8708	439	95.20%	4.80%	27,442	19,350	8,092	2.15%	57.24%	55.59	1.16
4	9147	9039	108	98.82%	1.18%	36,589	28,389	8,200	3.15%	62.29%	59.74	1.64
5	9148	9088	60	99.34%	0.66%	45,737	37,477	8,260	4.16%	66.33%	62.85	2.11
6	9147	9085	62	99.32%	0.68%	54,884	46,562	8,322	5.16%	69.70%	65.28	2.55
7	9148	9060	88	99.04%	0.96%	64,032	55,622	8,410	6.17%	73.74%	68.40	2.91
8	9147	9103	44	99.52%	0.48%	73,179	64,725	8,454	7.18%	78.11%	71.86	3.22
9	9147	9098	49	99.46%	0.54%	82,326	73,823	8,503	8.19%	80.13%	72.90	3.63
10	9148	9092	56	99.39%	0.61%	91,474	82,915	8,559	9.20%	81.48%	73.26	4.05
11	9147	9091	56	99.39%	0.61%	100,621	92,006	8,615	10.21%	83.50%	74.30	4.43
12	9147	9090	57	99.38%	0.62%	109,768	101,096	8,672	11.21%	84.18%	73.97	4.87
13	9148	9090	58	99.37%	0.63%	118,916	110,186	8,730	12.22%	86.53%	75.36	5.19
14	9147	9112	35	99.62%	0.38%	128,063	119,298	8,765	13.23%	87.21%	75.03	5.61
15	9147	9097	50	99.45%	0.55%	137,210	128,395	8,815	14.24%	87.54%	74.35	6.06
16	9148	9108	40	99.56%	0.44%	146,358	137,503	8,855	15.25%	88.22%	74.01	6.47
17	9147	9101	46	99.50%	0.50%	155,505	146,604	8,901	16.26%	88.55%	73.34	6.90
18	9147	9095	52	99.43%	0.57%	164,652	155,699	8,953	17.27%	88.55%	72.31	7.37
19	9148	9095	53	99.42%	0.58%	173,800	164,794	9,006	18.28%	88.89%	71.63	7.80
20	9147	9098	49	99.46%	0.54%	182,947	173,892	9,055	19.29%	89.23%	70.95	8.23

Section 8. Financial Curves and Recommended Cutoff

To evaluate the financial impact of the fraud detection model and determine an optimal cutoff threshold, financial curves were generated based on the costs associated with undetected fraud and false positives. In this analysis, two primary cost factors were considered: an average fraud loss of $4,000 per undetected case and a cost of $100 for each false positive. The fraud loss represents the potential financial damage incurred when a fraudulent transaction is not detected, while the false positive cost reflects operational expenses associated with investigating and handling mistakenly flagged transactions, which can impact customer experience and add to processing overhead.

The recommended cutoff threshold is set at 3%, where overall savings reach their highest point, achieving a net benefit of approximately $18 billion annually. This threshold strikes an optimal balance between maximizing fraud detection and controlling false positive costs. Lowering the cutoff further would increase fraud detection, but the resulting rise in false positives would offset the gains, reducing net savings. By setting the cutoff at 3%, the model captures the majority of fraud cases, as indicated by the plateau in the fraud savings curve, while keeping false positive costs within acceptable limits. This threshold aligns with the bank's objectives, allowing most legitimate transactions to be processed smoothly while maintaining robust financial protection against fraud. The model's performance at the 3% cutoff provides a practical, cost-effective solution for fraud detection, ensuring the highest possible savings with minimal disruption to customers.

8.1 Financial Curves and Recommended Cutoff, *showing the Fraud Savings, Lost Revenue, and Overall Savings curves with the recommended 3% cutoff line.*



Section 9. Summary

This project aimed to develop a robust fraud detection model capable of identifying fraudulent applications in a dataset containing 1,000,000 synthetic credit card and cell phone application records. The data preparation process involved extensive cleaning, including the replacement of placeholder values in fields such as SSN, address, phone number, and date of birth to prevent false connections and ensure accuracy. After cleaning, I created new variables to capture patterns commonly associated with fraud, focusing on temporal, frequency, and entity combination variables to identify potential fraudulent behaviors.

Feature selection was then conducted to isolate the most predictive variables for fraud detection, ensuring the model's focus on relevant indicators. Through a series of iterations, I tested various machine learning models, including Logistic Regression, Decision Tree, Random Forest, LightGBM, and Neural Network. Each model was tuned and evaluated based on its Fraud Detection Rate (FDR) at a 3% threshold. LightGBM emerged as the best-performing model, achieving an FDR@3% of 0.612 on out-of-time (OOT) data, indicating strong generalization capabilities and suitability for fraud detection.

The final model configuration for LightGBM included 80 estimators, a maximum depth of 3, a learning rate of 0.07, and 25 leaves. This setup effectively balanced model complexity with predictive power, resulting in FDR@3% scores of 0.585 for training data, 0.576 for test data, and 0.612 for OOT data. These consistent results across datasets demonstrate that the model is well-balanced and avoids overfitting, making it a reliable choice for real-world application.

A financial analysis was conducted to determine the optimal score cutoff for the model, balancing fraud savings with false positive costs. At a 3% cutoff, the model achieves maximum overall savings, projected to result in an annual fraud prevention savings of approximately $18 billion. This threshold ensures high fraud detection rates while minimizing operational disruptions from false positives, aligning with the bank's strategic objectives.

Looking forward, additional enhancements could involve testing alternative model architectures, such as ensemble methods that combine multiple algorithms for potentially higher accuracy, and exploring cost-sensitive learning techniques to further optimize the balance between fraud detection and operational efficiency. Overall, this project provides a comprehensive, financially impactful solution to the bank's fraud detection challenge, effectively addressing both technical and business requirements.

Appendix: DQR

Data Quality Report

1. Data Description

The dataset contains 1,000,000 records of synthetic application data for credit card and cell phone applications submitted in 2017. It includes 10 fields: record, date, ssn, firstname, lastname, address, zip5, dob, homephone, and fraud_label. The record field is a unique identifier for each application, while the other fields represent personal identifiable information (PII), such as Social Security Numbers (SSNs), names, addresses, dates of birth (DOBs), and phone numbers. These fields are essential for identifying individuals and are used in fraud detection algorithms. The fraud_label field is a binary variable indicating whether an application was flagged as fraudulent (1) or non-fraudulent (0). This dataset is synthetic but designed to closely mimic the statistical properties of real-world application data. Some fields, like SSNs and addresses, may contain placeholder values or patterns that require further investigation for potential fraud.

2. Summary Table

2.1 Numeric Table

Field Name	# Records with value	% Populated	% Zeros	Min	Max	Mean	Std	Most Common value
date	1,000,000	100.0%	0	2017-01-01	2017-12-31	-	-	2017-08-16
dob	1,000,000	100.0%	0	1900-01-01	2016-10-31	-	-	1907-06-26

2.2 Categorical Table

Field Name	# Records with Values	% Populated	% Zeros	# Unique Values	Most Common Value
record	1,000,000	100.0%	0	1,000,000	1
ssn	1,000,000	100.0%	0	835,819	999999999
firstname	1,000,000	100.0%	0	78,136	EAMSTRMT
lastname	1,000,000	100.0%	0	177,001	ERJSAXA

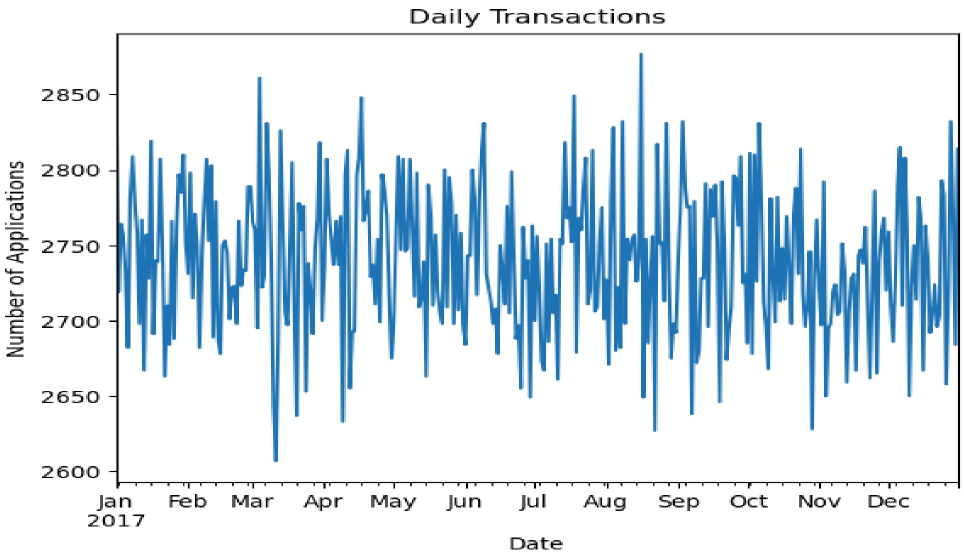
address	1,000,000	100.0%	0	828,774	123 MAIN ST
zip5	1,000,000	100.0%	0	26,370	68138
homephone	1,000,000	100.0%	0	28,244	9999999999
fraud_label	1,000,000	100.0%	985,607	2	0

2. Visualization of Each Field

2.1 Field Name: record

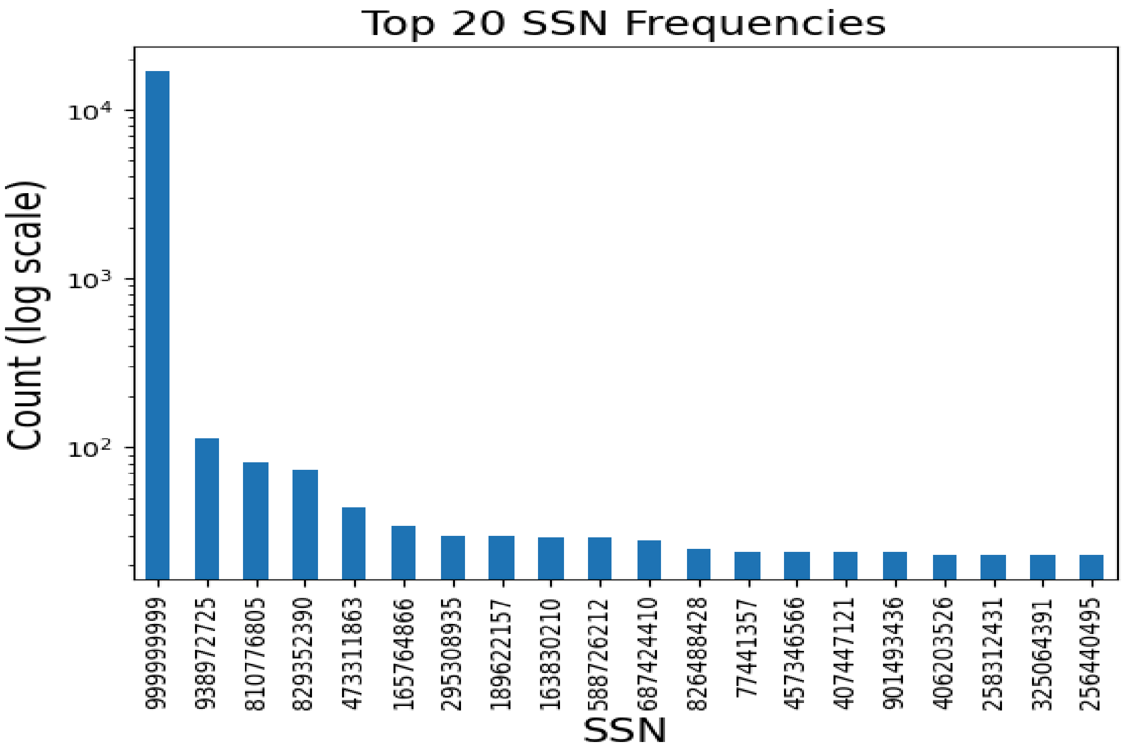
This is an ordinal unique identifier for each record in the dataset, ranging from 1 to 1,000,000.

2.2 Field Name: date



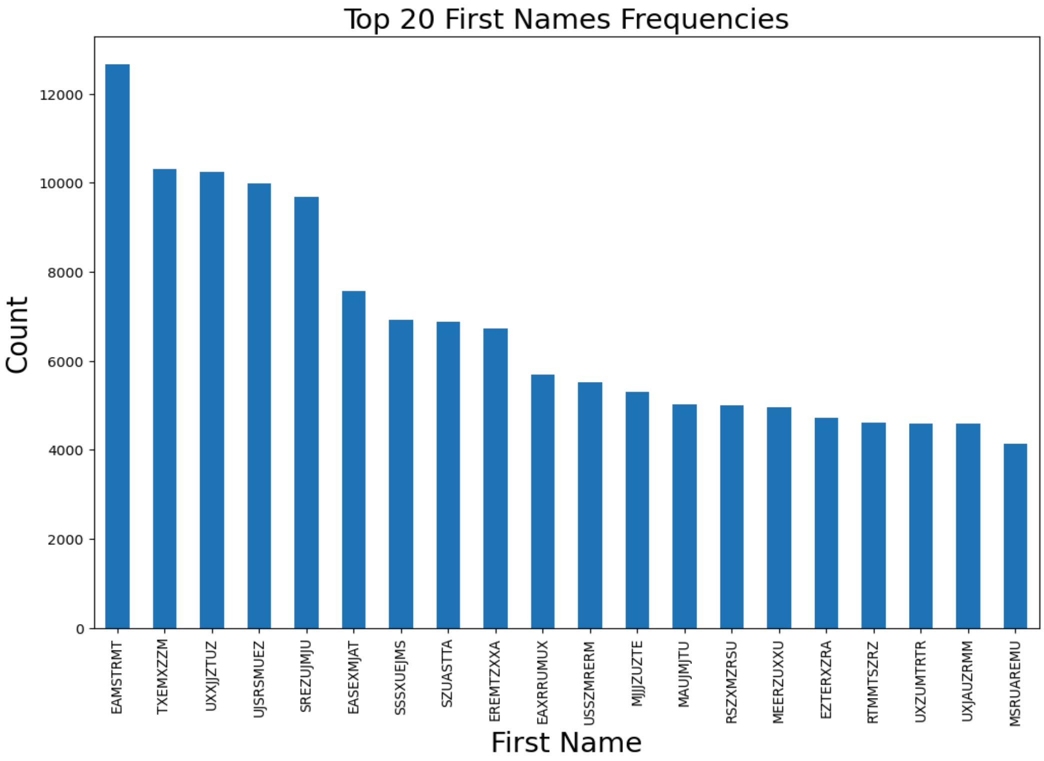
The plot illustrates the daily distribution of submitted applications throughout 2017. Application counts fluctuate between 2,600 and 2,850 per day, without a clear upward or downward trend. The data shows short-term volatility, with some periods experiencing noticeable spikes or drops in applications. This variation could be influenced by external factors like market conditions or promotional events. Overall, the number of daily submissions appears consistent, with no significant anomalies or outliers over the observed period.

2.3 Field Name: ssn



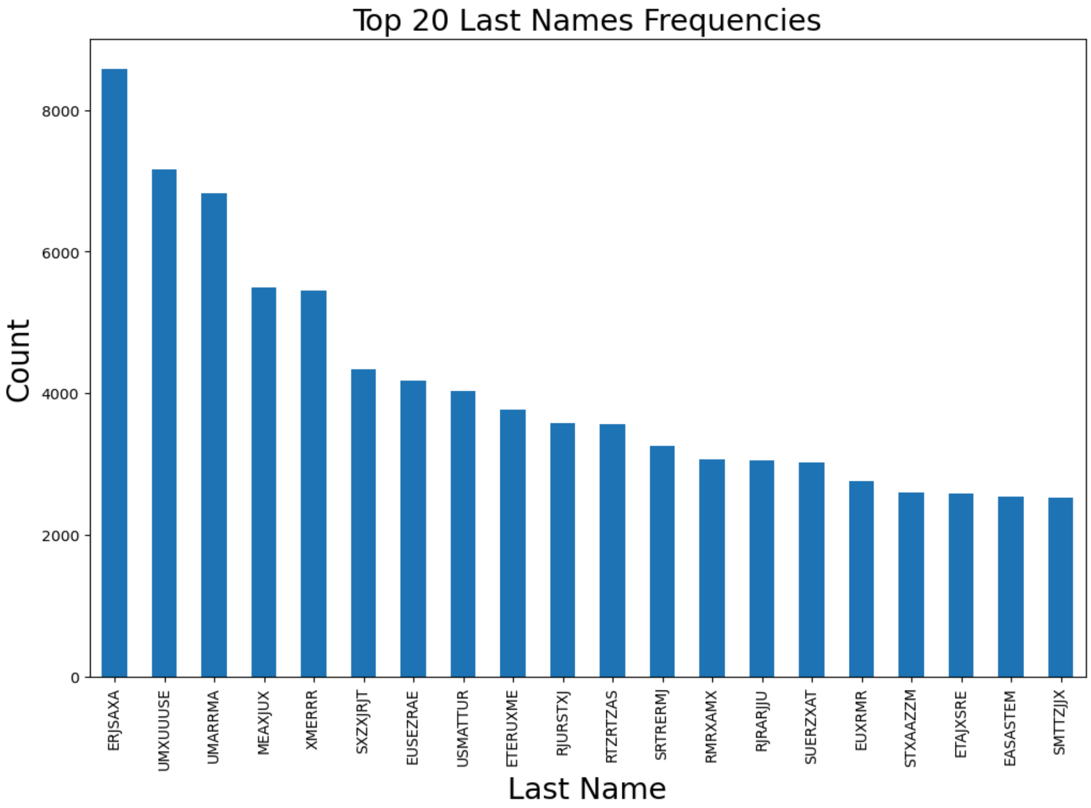
The plot displays the top 20 most frequent Social Security Numbers (SSNs) in the dataset, using a logarithmic scale for the count. The most common SSN, 999999999, appears disproportionately more often than any other, suggesting it is likely a placeholder or default value. Other SSNs, such as 893777275 and 810776805, also appear frequently but at much lower counts. This distribution indicates that a small number of SSNs are being overused, which could be indicative of synthetic identities or data entry issues. The presence of such outliers in the SSN field warrants further investigation to assess the validity of these applications.

2.4 Field Name: firstname



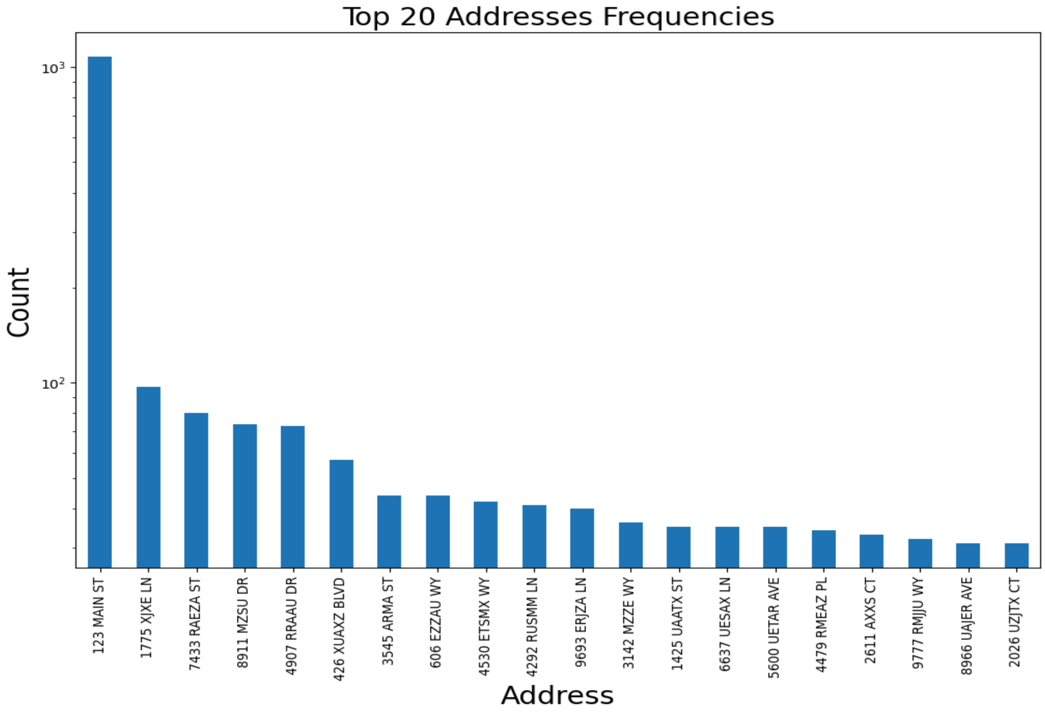
The plot shows the top 20 most frequent first names in the dataset. The most common name, EAMSTRMT, appears significantly more often than others, followed by TXEMZRXI and UXXJZNJZ. These names, which do not resemble typical first names, suggest the presence of placeholder values or potential data quality issues. Such frequent occurrences of similar and unusual names could indicate synthetic identities or misentries in the dataset. These patterns may need further investigation to determine if they are associated with fraudulent behavior or data generation artifacts.

2.5 Field Name: lastname



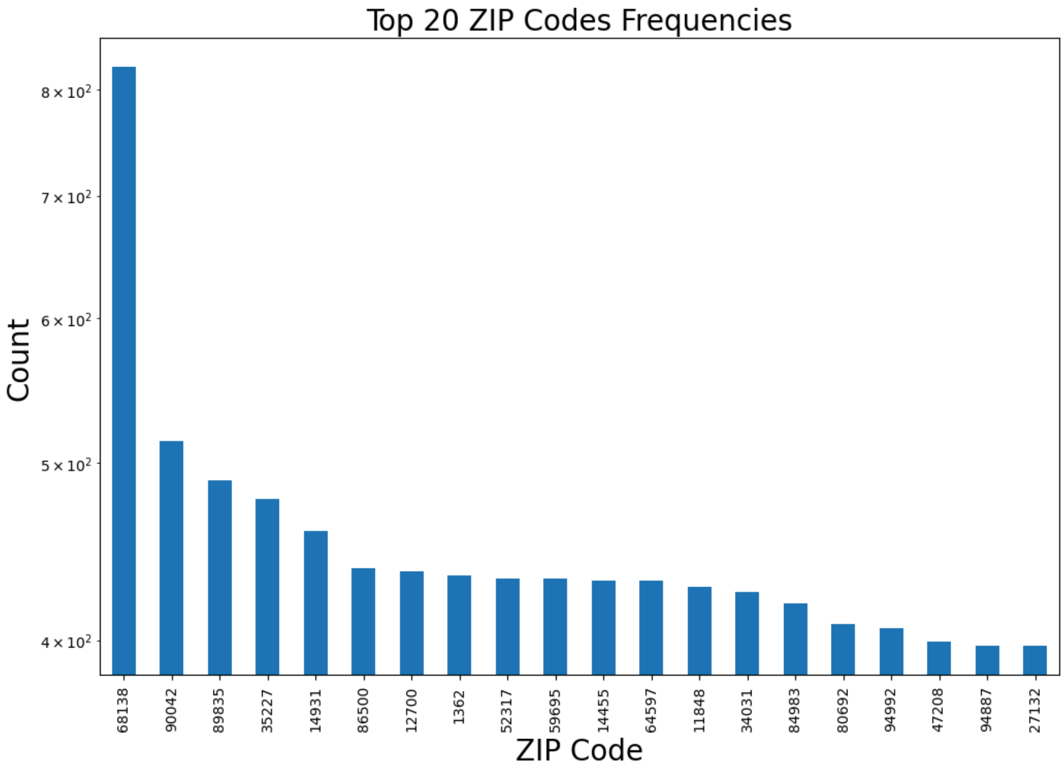
The plot shows the top 20 most frequent last names in the dataset. ERJSAXA appears most frequently, followed by UMVAQLUSE and VHRAMMA. Similar to the first names, many of these last names do not resemble typical real-world surnames, suggesting potential data generation artifacts or placeholder values. The frequency of such names could indicate synthetic identities or data quality issues. These anomalies in the last name field should be further analyzed to ensure they do not skew fraud detection models or lead to inaccurate conclusions in identity verification processes.

2.6 Field Name: address



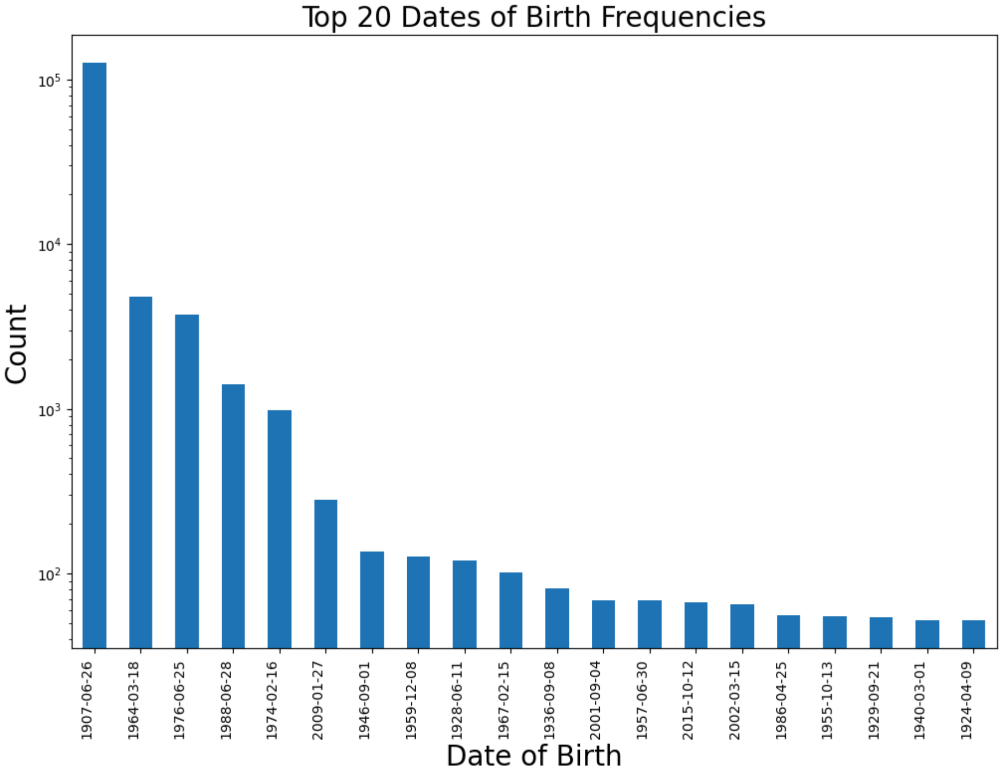
The plot shows the top 20 most frequent addresses in the dataset. The most common address, 123 MAIN ST, appears significantly more often than any other, suggesting it is a placeholder or a default value, with a count that is orders of magnitude higher than the rest. Other frequent addresses, such as 2175 XYE LM and 7419 RMEAZ ST, may also represent data entry issues or synthetic records. The use of such common placeholder addresses can distort the analysis and should be flagged for further investigation, particularly in identity verification and fraud detection scenarios.

2.7 Field Name: zip5



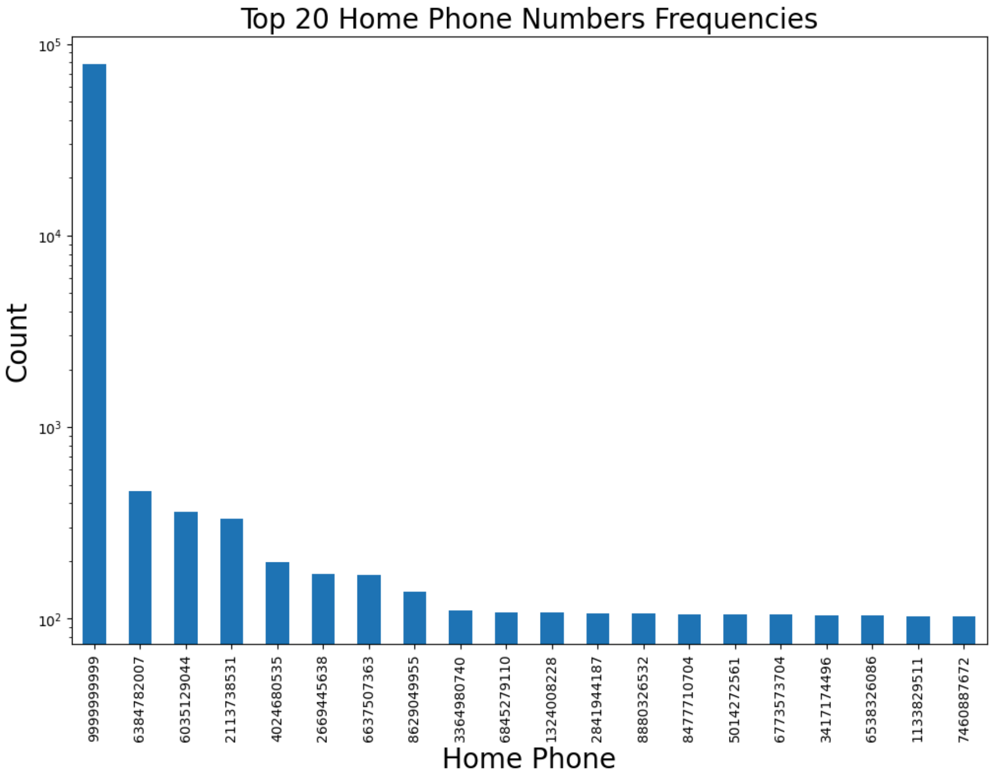
The plot shows the top 20 most frequent ZIP codes in the dataset, with 68138 being the most common, followed by 90042 and 89835. The disproportionate frequency of certain ZIP codes, particularly 68138, could indicate the use of default or placeholder values. This trend is similar to what is observed in other fields, such as addresses, and may signal synthetic records or data entry issues. The presence of these frequently used ZIP codes should be flagged for further investigation, as they could distort analysis in fraud detection algorithms.

2.8 Field Name: dob



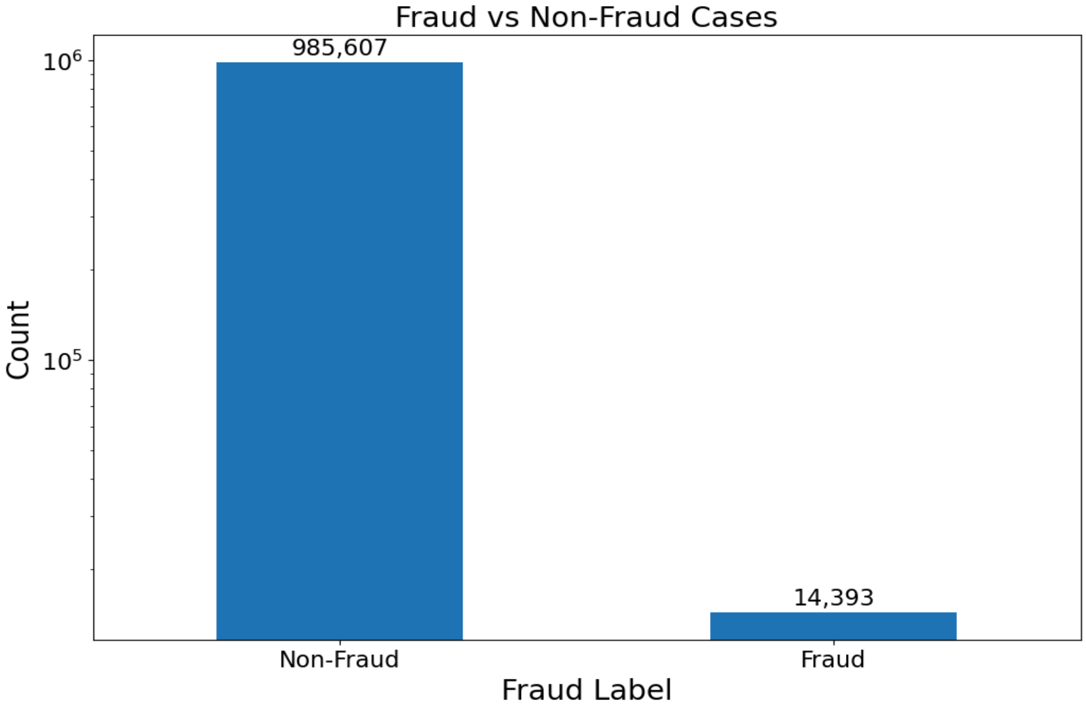
The plot illustrates the top 20 most frequent dates of birth (DOB) in the dataset. The most common DOB, 1907-06-26, appears significantly more often than others, likely representing a placeholder or default value. Other frequent dates, such as 1964-03-18 and 1976-06-25, may also suggest synthetic or incorrect entries. These unusually frequent dates of birth could indicate data entry issues or fraud-related patterns. The high concentration of specific dates should be further investigated, as they might be indicative of suspicious behavior or systematic errors in the data.

2.9 Field Name: homephone



The plot shows the top 20 most frequent home phone numbers in the dataset. The number 9999999999 appears overwhelmingly more frequently than any other, likely serving as a placeholder or invalid value. Other numbers, such as 6318720607 and 6011239044, are also repeated but at much lower counts. The high occurrence of placeholder or potentially fake phone numbers suggests potential data quality issues or synthetic records. These frequently used numbers should be flagged for further investigation, especially in the context of fraud detection.

2.10 Field Name: fraud_label



The plot shows the distribution of fraud (1) versus non-fraud (0) cases in the dataset. The majority of applications, **985,607** cases, are non-fraudulent, while **14,393** applications are flagged as fraudulent. This highly imbalanced dataset, with fraud accounting for less than 2% of the total, is typical in real-world fraud detection problems. The imbalance underscores the importance of handling the data carefully, as models may be biased toward predicting non-fraud unless specific techniques are employed to account for this disparity.