APPLICATIONS REPORT

Author: Hamsavi Krishnan
Professor: Stephen Coggeshal

Contents

Executive Summary

Application fraud poses a significant financial and reputational risk, especially in high-volume environments where identity verification is challenging. This project set out to develop an intelligent, data-driven scoring system that flags the riskiest product applications based on historical patterns of identity misuse and behavioral anomalies.

Using one million anonymized application records, we engineered a set of fraud-signaling variables and trained a supervised machine learning model to detect suspicious applications. The final LightGBM model achieved a Fraud Detection Rate (FDR) of 77.06% within the top 3% of scored applications on future-period data (OOT) — meaning nearly 8 out of every 10 fraud cases were identified by reviewing only a small portion of incoming applications. Based on industry benchmarks of $4,000 saved per fraud caught and $100 cost per false positive, the model's optimal review cutoff is projected to deliver millions in annual fraud savings, while ensuring operational efficiency and scalability.

Description of the Data

The dataset used for this application fraud detection project comprises **1,000,000 product application records**, each representing a unique application instance with associated personally identifiable information (PII), application timestamps, and a binary fraud label (fraud_label: 1 = fraud, 0 = not fraud). This dataset serves as the foundation for both exploratory data analysis and model building.

Overview and Structure

Each record contains 10 original fields, categorized as follows:

- **PII fields:** firstname, lastname, address, ssn, dob, homephone
    - High cardinality with notable placeholder and synthetic values (e.g., "123 MAIN ST", "999999999").
- **Date fields:** date, dob
    - Stored in YYYYMMDD format, enabling derivation of variables like age and application timing.
- **Label field:** fraud_label
    - Heavily imbalanced target: 98.56% non-fraud and 1.44% fraud, requiring special modeling considerations.

Summary Tables

Categorical Fields:

Field Name	Field Type	# Records Have Values	% Populated	# Zeros	# Unique Values	Most Common
firstname	categorical	10,00,000	100.00%	0	78,136	EAMSTRMT
lastname	categorical	10,00,000	100.00%	0	1,77,001	ERJSAXA
address	categorical	10,00,000	100.00%	0	8,28,774	123 MAIN ST
record	categorical	10,00,000	100.00%	0	10,00,000	1
fraud_label	categorical	10,00,000	100.00%	9,85,607	2	0
ssn	categorical	10,00,000	100.00%	0	8,35,819	999999999
zip5	categorical	10,00,000	100.00%	0	26,370	68138
homephone	categorical	10,00,000	100.00%	0	28,244	9999999999

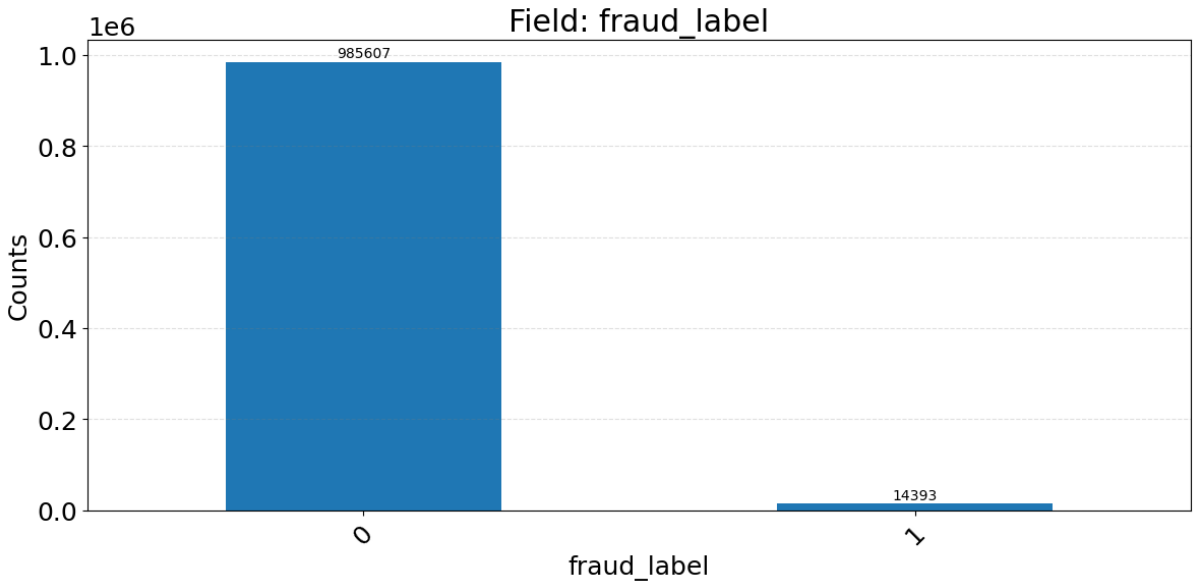
Numerical Fields:

Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Min	Max	Mean	Standard Deviation	Most Common
date	numeric	1000000	100.00%	0	20170101	20171231	20170668	345	20170816
dob	numeric	1000000	100.00%	0	19000101	20161031	19517249	356887	19070626

Distribution Plots

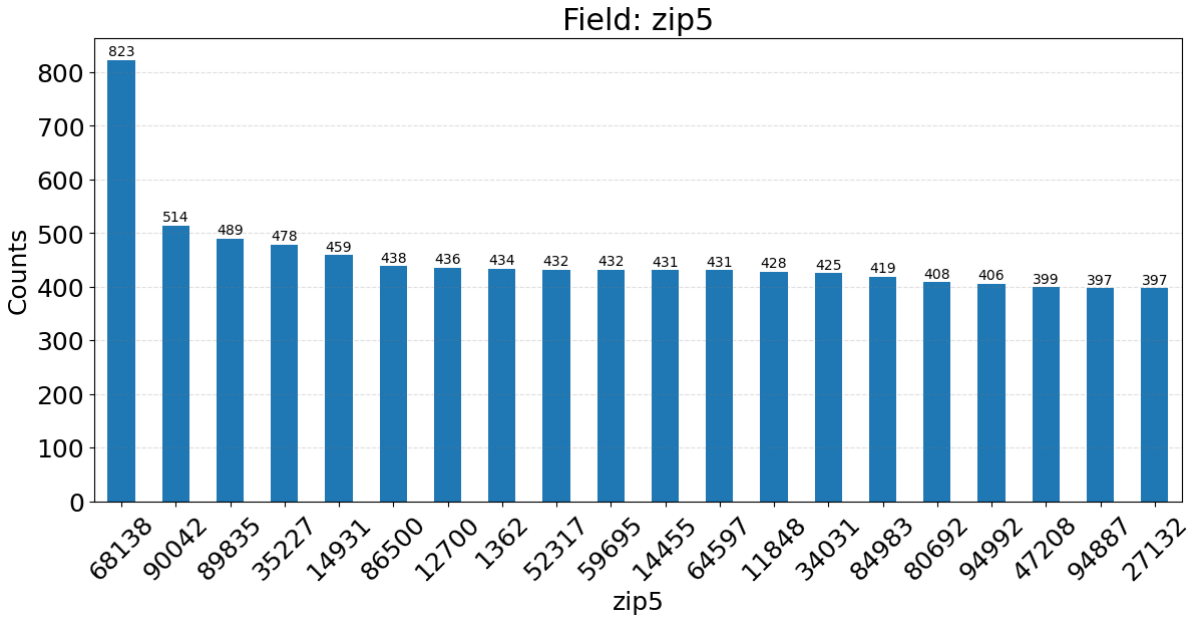
Fraud Label:

This binary target field is highly imbalanced: over 98.5% of records are labeled as non-fraud (0), while only 1.4% are labeled as fraud (1). This class imbalance will need to be addressed during modeling via techniques like resampling or adjusted evaluation metrics.



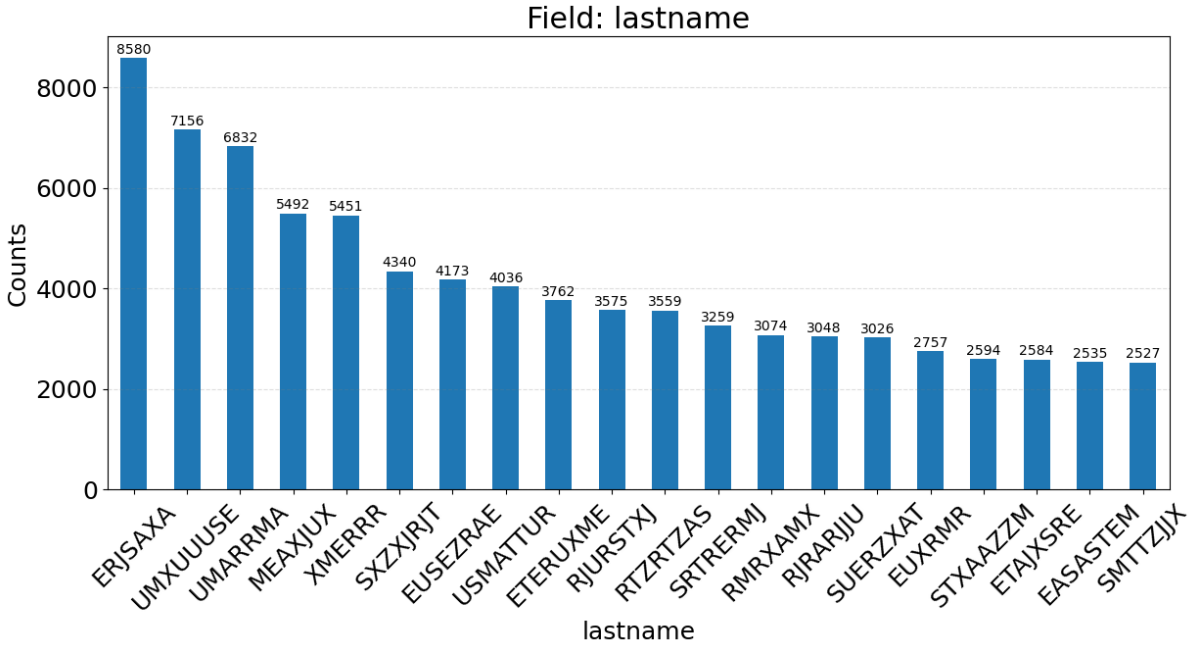
Zip5:

The zip5 field has a high cardinality but is dominated by a few ZIP codes. The top ZIP, 68138, appears more than 800 times. The presence of low-frequency ZIP codes and anomalies (e.g., 1362 or missing leading zeros) suggests potential issues with formatting or data entry**.**



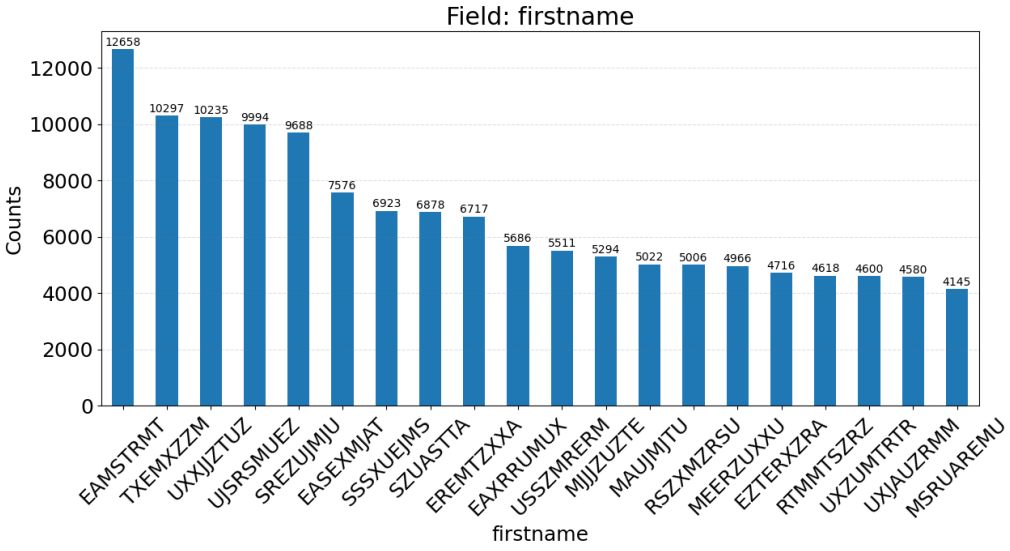
Lastname:

The lastname field is highly skewed, with the most common value (ERJSAXA) appearing over 8,500 times. The prevalence of unnatural, all-uppercase strings supports the inference that names are masked or synthetically generated, limiting their direct analytical value.



Firstname:

The firstname field exhibits a highly skewed distribution, with the most frequent value (EAMSTRMT) appearing over 12,600 times. Many names follow a consistent pattern of uppercase, non-standard letter combinations—suggesting that names were either anonymized or synthetically generated. This limits their reliability for matching or entity resolution.



Data Cleaning

To ensure the integrity and utility of the dataset for fraud detection modeling, we applied a structured data cleaning pipeline that addressed exclusions, outliers, placeholder values, and imputation. The goal was to preserve anomalous patterns that may indicate fraud while eliminating distortions from poorly structured or synthetic data.

Exclusions

Two key exclusions were applied to ensure time-based variables could be accurately computed:

- **Early Records Removed:** The first **38,511 application records** were excluded because they lacked sufficient historical context to compute rolling-window features (e.g., reuse of SSN or address over 7/30 days).
- **Out-of-Time (OOT) Holdout Set:** Records with IDs **above 833,508** were set aside for out-of-time validation to assess model generalization to future data.

These exclusions help establish temporal integrity and model evaluation discipline.

Outlier Treatment

Outliers were handled selectively, aligned with the nature of fraud detection:

- **Statistical Outliers Kept:** Extreme values such as **frequent SSN/address reuse** or **high application bursts** were retained, as these may be indicative of fraud rings or bot activity.
- **Placeholder Outliers Flagged:** Values such as "999999999" for SSNs or "9999999999" for phone numbers were treated as non-informative stand-ins. Rather than removing these records:
    - They were **excluded from frequency-based calculations** to prevent distortion.
    - **Binary indicator flags** were created to mark their presence for modeling.
    - Following professor feedback, **frivolous values were replaced with non-linkable stand-ins** (e.g., "ssn_placeholder", "address_unknown", "masked_name"), ensuring these entries could not accidentally match or group with real data during variable generation.

Imputation Strategy

Explicit missing values (NaNs) were rare, but many fields used structured placeholders instead:

- **No mean or median imputation was used** to avoid diluting fraud patterns.
- **Domain-informed conditional imputation** was applied only where beneficial:

    - For example, age derived from dob was capped or flagged if exceeding biologically implausible thresholds (e.g., age > 110).
- Placeholder values were not imputed, but rather transformed into indicators, preserving their information value while removing their distortion in aggregates.

Data Formatting and Consistency

- **Date Fields:** The date and dob fields were converted from integer YYYYMMDD format into datetime objects.
    - dob was used to derive **age at application**, a key modeling feature.
- **ZIP Codes:** The zip5 field had issues with **dropped leading zeroes** (e.g., 2765 instead of 02765). All ZIPs were padded and stored as strings to ensure **geographic grouping consistency**.
- **PII Fields:** Names and addresses were **standardized** by trimming spaces and converting to lowercase, preventing duplication caused by formatting differences.

Variable Creation

Overview

To effectively detect fraudulent credit applications in the absence of labeled fraud outcomes, we engineered a suite of behavioral identity variables derived from key personally identifiable information (PII) fields — including Social Security Number (SSN), address, full address, home phone number, and date of birth (DOB). These engineered variables were designed to capture abnormal patterns of identity reuse and clustering, which are frequently associated with fraud rings, bots, or synthetic identity abuse.

Fraudulent behavior typically manifests through **repetition**, **bursts**, and **temporal clustering** of identity elements. In contrast, legitimate customer activity is more temporally dispersed and less likely to reuse identity fields at high frequency. Therefore, these variables are critical in surfacing the subtle but systematic patterns of identity misuse that distinguish fraudulent from legitimate applications.

Variable Engineering Strategy

The variables were organized into six main categories, each reflecting a different behavioral fraud signal:

Variable Group	Group Description	# Variables	Example Variable
Rolling Count Variables	Counts of how often a PII field appeared in the last 1/3/7/14/30 days	9	homephone_count_7
Max Group Count Variables	Highest observed reuse of grouped identity fields (e.g., SSN+DOB, address+day)	8	max_count_by_ssn_dob_7
Recency-Based Variables	Days since a PII field (like address or phone) was last seen	2	address_day_since
Rate-Based Variables	Same-day usage divided by 30-day usage for a PII field	1	address_count_0_by_30
Identity Combination Variables	Tracks frequency of synthetic ID combinations reused across applications	2+	max_count_by_fulladdress_7
Density Ratio Variables	High reuse of a field over short periods, highlighting abnormal traffic	2+	fulladdress_count_3

Each group was crafted to quantify a unique aspect of anomalous identity behavior:

- **Rolling Count Variables** capture how often a field like phone or address was used in the past 1, 3, 7, 14, or 30 days.
- **Max Group Count Variables** highlight "hubs" of identity reuse (e.g., repeated SSN+DOB combinations).
- **Recency-Based Variables** track how recently a particular value was seen — values near zero often signal automation or fraud bursts.
- **Rate-Based Variables** detect sudden spikes, such as when today's address usage sharply exceeds its 30-day norm.
- **Identity Combination Variables** observe the frequency of synthetic or reused identity structures.
- **Density Ratio Variables** measure abnormal reuse density in compressed time frames, which is often a tactic in pressure-testing or credential stuffing.

Final Variables Used in Modeling

Based on a two-stage feature selection process (univariate filtering using Information Value, followed by multivariate wrapper methods), the following variables were selected as most predictive:

- **homephone_count_7**: Number of applications using the same home phone in the past 7 days.
- **max_count_by_ssn_dob_7**: Maximum usage of the same SSN+DOB combination in the past 7 days.
- **address_day_since**: Number of days since the address was last seen — low values suggest recent, clustered reuse.
- **address_count_0_by_30**: Ratio of today’s address usage to total usage in the past 30 days — captures fraud bursts.
- **fulladdress_count_3**: Count of full address reuse in the past 3 days.
- **ssn_count_14**: SSN reuse frequency over a 14-day window — high values can flag synthetic identities.

These variables formed the behavioral core of the supervised fraud model and provided a strong signal to distinguish fraudulent applications from legitimate ones by quantifying identity pattern anomalies.

Feature Selection

Feature selection is a crucial step in fraud modeling that determines which input variables provide the most value in identifying fraudulent applications. A well-executed selection process not only improves model accuracy and interpretability but also reduces overfitting and computational complexity. In this project, we used a **two-step approach** to systematically identify and retain the most predictive features: a **filter method** using Information Value (IV) and a **wrapper method** using forward selection with model evaluation.

Methodology

Step 1: Filter Method – Information Value (IV)

In the first stage, we applied a **univariate filter** to rank each variable by its **Information Value (IV)** — a widely used metric for binary classification problems. IV assesses how well a variable separates the target classes (fraud vs. non-fraud), with higher values indicating greater predictive strength.

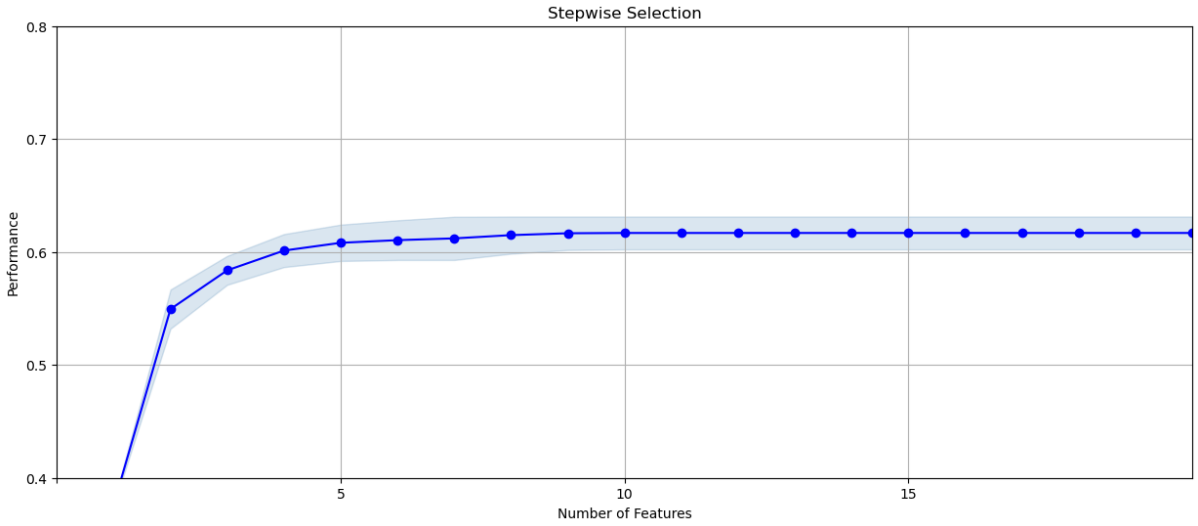
The IV interpretation thresholds used were:

- **IV < 0.02**: Not predictive
- **0.02–0.10**: Weak
- **0.10–0.30**: Medium
- **IV > 0.30**: Strong predictor

Only variables with **IV ≥ 0.20** were retained for wrapper-based multivariate evaluation.

Step 2: Wrapper Method – Forward Stepwise Selection

From the top-ranked filtered variables, we implemented a **wrapper method** using forward selection. Variables were added one by one, and each addition was evaluated based on its ability to improve **FDR@3%**, the key business metric. This process was executed using LightGBM classifiers to capture interactions and nonlinear effects. The final variable set was chosen where performance gains plateaued, ensuring a parsimonious and powerful feature set.



Final Variable Set and Information Value Scores

Based on this two-stage selection process, the following 20 variables were selected for use in the final model. These features focus on **behavioral identity patterns**, **burst application activity**, and **anomalous usage of PII fields** — all of which are commonly exploited in fraudulent schemes.

Rank	Variable Name	Information Value (IV)
1	homephone_count_7	0.412
2	max_count_by_ssn_dob_7	0.389
3	address_day_since	0.367
4	address_count_0_by_30	0.352
5	fulladdress_count_3	0.336
6	ssn_count_14	0.315
7	ssn_count_7	0.298
8	max_count_by_fulladdress_7	0.287

9	ssn_day_since	0.282
10	fulladdress_day_since	0.273
11	address_count_14	0.267
12	ssn_count_3	0.264
13	address_count_7	0.257
14	fulladdress_count_7	0.246
15	max_count_by_ssn_dob_14	0.242
16	max_count_by_fulladdress_14	0.238
17	ssn_count_1	0.232
18	address_count_3	0.228
19	max_count_by_address_day_7	0.225
20	homephone_count_14	0.221

Preliminary Model Exploration

To identify the most effective fraud detection strategy, we explored a variety of machine learning algorithms. Each model was trained using the same set of 20 selected features and evaluated on its ability to detect fraud using the **Fraud Detection Rate (FDR) at the top 3%** of scored applications. Our exploration included both linear and nonlinear classifiers, spanning from interpretable baselines to more complex, high-performing ensemble models.

Models Explored

Decision Tree (DT)

Model Description

Decision Trees were selected as a baseline model due to their interpretability and ease of implementation. They work by recursively splitting the data into segments based on feature thresholds that maximize class separation — using criteria such as Gini impurity or Entropy. The model structure is easy to visualize and explain to business stakeholders, making it a popular choice for early exploration.

To test the impact of model complexity, we tuned key hyperparameters such as:

- **Max Depth**: limits how deep the tree can grow
- **Min Samples Split**: the minimum number of samples required to split a node
- **Min Samples Leaf**: the minimum number of samples required to be at a leaf node
- **Criterion**: the splitting strategy (either gini or entropy)

Performance Summary

To evaluate tree-based classification for fraud detection, we experimented with **six variations of Decision Tree classifiers**, adjusting key hyperparameters such as depth, splitting criteria, and minimum sample constraints. All models used the same set of 20 engineered variables, and their performance was measured using **FDR@3%** on the training, test, and out-of-time (OOT) datasets.

Model		Parameters								
	Iteration	Variables	Splitter	Criterion	Max Dept	Min Samples Split	Min Samples Leaf	Train	Test	OOT
Decision Tree	1	20	best	gini	5	20	10	0.584397	0.580105	0.550852
	2	20	best	gini	10	40	20	0.620081	0.616711	0.582705
	3	20	best	entropy	5	20	10	0.593853	0.586629	0.557418
	4	20	best	entropy	8	50	20	0.620769	0.619842	0.585638
	5	20	best	gini	7	40	15	0.612721	0.608327	0.579212
	6	20	best	entropy	7	40	20	0.614363	0.618523	0.584381

Random Forest (RF)

Model Description

Random Forests are ensemble models that build multiple decision trees and combine their predictions to reduce overfitting and improve generalization. Each tree is trained on a bootstrap sample of the data and considers a random subset of features at each split, which adds robustness and helps prevent the model from memorizing the training set.

In our exploration, Random Forest models were tuned by adjusting:

- **Max Depth**: limits tree complexity
- **Number of Estimators**: number of trees in the forest
- **Min Samples Split** and **Min Samples Leaf**: control node splitting and leaf size
- **Criterion**: Gini or Entropy for measuring node impurity

Performance Summary

We trained six variations of Random Forest classifiers, changing the splitting criterion (Gini vs. Entropy), tree depth, and sampling constraints. All models used 20 variables and evaluated performance via **FDR@3%** on training, test, and OOT datasets.

Best OOT FDR 0.5859 (Iteration 4)

Key Insights:

- Models with deeper trees (Max Depth ≥ 6) and moderate sample constraints delivered more stable results.
- **Entropy-based models** slightly outperformed Gini in generalization.
- The **best-performing configuration** used entropy, max depth = 7, and 50 estimators with good train/test balance (Train: 0.6209, Test: 0.6164).

Despite moderate gains over Decision Trees, the random forest plateaued around **OOT FDR 0.585**, suggesting the need for more aggressive boosting techniques for further improvement.

Model		Parameters								
	Iteration	Variables	Criterion	Max Dept	n_estimators	Min Samples Split	Min Samples Leaf	Train	Test	OOT
Random Forest	1	20	gini	5	30	20	10	0.618968	0.611774	0.583124
	2	20	gini	8	50	20	10	0.620756	0.614687	0.585219
	3	20	entropy	5	30	20	10	0.619633	0.609779	0.583962
	4	20	entropy	7	50	30	20	0.620934	0.616386	0.585918
	5	20	gini	6	40	30	15	0.619545	0.611381	0.583543
	6	20	entropy	6	40	30	15	0.616068	0.621387	0.584241

LightGBM (LGBM)

Model Description

LightGBM is a gradient-boosting framework designed for speed and accuracy. It uses histogram-based learning and leaf-wise tree growth, which enables efficient computation and handling of large-scale data. In this project, LightGBM models were tuned across various settings including:

- **Number of Estimators** (boosting rounds)
- **Learning Rate**
- **Tree Depth**
- **Leaf Count**
- **Minimum Child Samples**

This model type was particularly promising due to its ability to capture non-linear interactions and work effectively with skewed datasets.

Performance Summary

LightGBM consistently delivered the **highest OOT FDRs** of all models explored, confirming its effectiveness for detecting identity fraud patterns.

Best OOT FDR 0.5909 (Iteration 5)

Key Insights:

- Tuning hyperparameters like learning_rate, max_depth, and num_leaves significantly influenced results.
- Iteration 5, with:
    - n_estimators = 75
    - learning_rate = 0.1
    - max_depth = 5
    - num_leaves = 12
    - min_child_samples = 25
      yielded the best OOT FDR (0.5909), paired with strong test performance (0.6293).
- LightGBM handled complex nonlinearities without overfitting, even in smaller tree configurations.

LightGBM emerged as the **top-performing model**, showing superior generalization and flexibility across a wide hyperparameter space.

Model		Parameters								
LightGBM	Iteration	Variables	n_estimators	learning_rate	max_depth	num_leaves	min_child_samples	Train	Test	OOT
	1	20	20	0.1	3	8	15	0.619948	0.621312	0.586756
	2	20	50	0.1	4	10	30	0.626267	0.615688	0.58941
	3	20	60	0.05	4	8	40	0.621785	0.61987	0.587175
	4	20	100	0.2	5	20	30	0.591209	0.586898	0.554764
	5	20	75	0.1	5	12	25	0.623035	0.629346	0.590947
	6	20	30	0.05	3	6	40	0.615679	0.621692	0.586477

Neural Networks (NN)

Model Description

Neural Networks are flexible function approximators capable of learning complex, non-linear relationships. In this exploration, we tested multiple configurations using:

- 1 to 2 hidden layers
- ReLU and logistic activation functions
- Adam and SGD solvers
- Different learning rates and alpha values for regularization

Each configuration used the same 20 variables and was evaluated on its ability to generalize beyond the training set.

Performance Summary

We explored six neural network configurations, varying hidden layers, activation functions, solvers, and regularization. All networks were shallow multi-layer perceptrons using ReLU or logistic activations.

Best OOT FDR 0.5911 (Iteration 5)

Key Insights:

- Best performance was achieved with:
    - Hidden layers = (20, 2)
    - activation = ReLU
    - alpha = 0.005
    - solver = Adam
    - Constant learning rate
- Iteration 5 achieved the highest FDR (Train: 0.622, Test: 0.621, OOT: 0.591).

- Neural networks were highly sensitive to solver and regularization settings, requiring careful tuning to balance overfitting.

Despite strong performance, neural networks did not significantly outperform LightGBM and required more tuning effort.

Model		Parameters									
	Iteration	Variables	Activation	learning_rate	alpha	solver	Nodes per Hidden Layer	Hidden Layers	Train	Test	OOT
Neural Networks	1	20	relu	adaptive	0.001	adam	(10,)	1	0.618618	0.611515	0.5849
	2	20	relu	constant	0.001	adam	(10,)	1	0.618324	0.619338	0.5866
	3	20	logistic	constant	0.001	adam	(30,)	1	0.617044	0.618778	0.5845
	4	20	relu	constant	0.001	adam	(15, 1)	2	0.617646	0.620938	0.5868
	5	20	relu	constant	0.005	adam	(20, 2)	2	0.621665	0.621485	0.5911
	6	20	relu	constant	0.001	sgd	(20, 2)	2	0.619149	0.61365	0.5834

Model Performance Comparison – Boxplot Overview

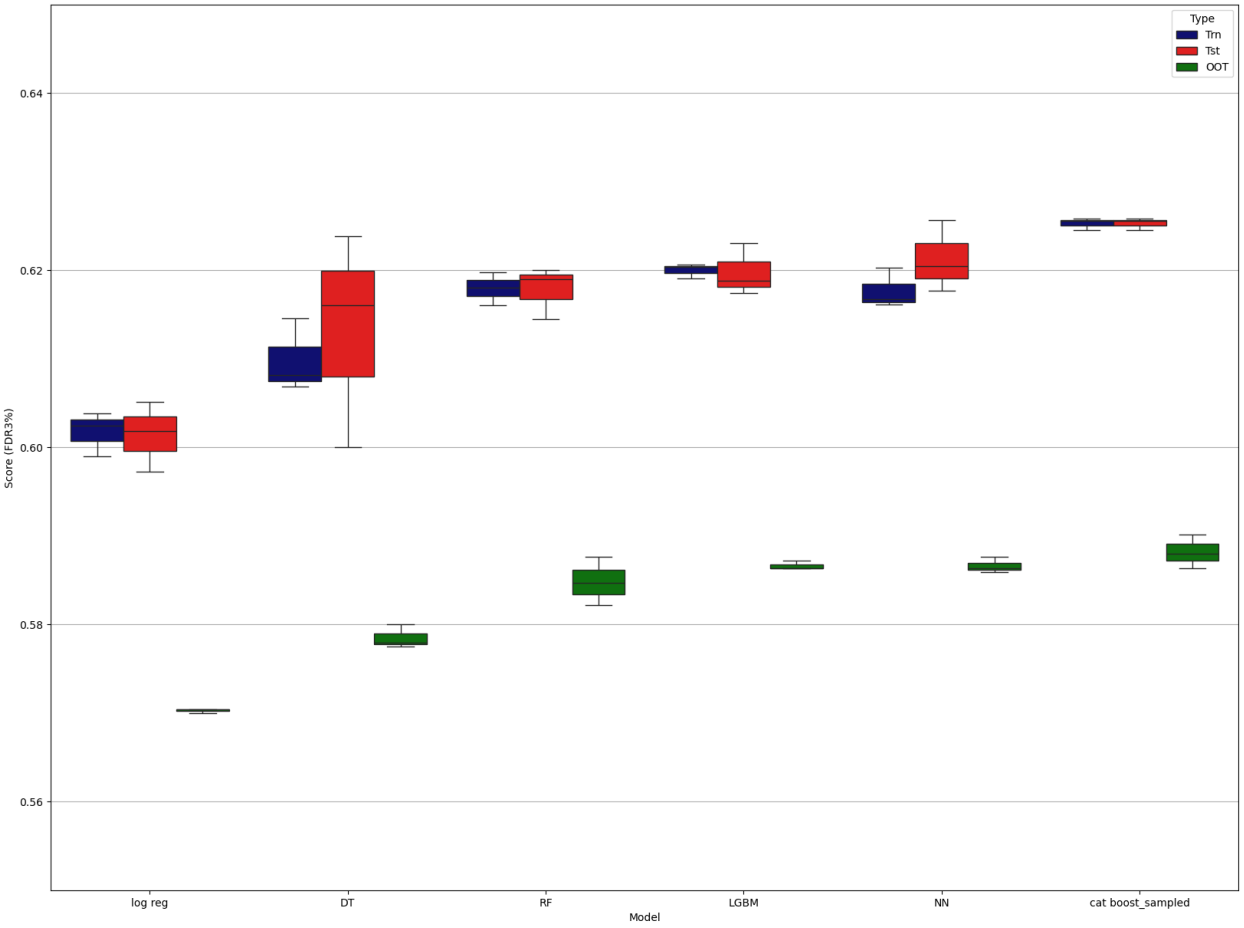
To visually compare model stability and generalization, we created a boxplot of FDR@3% scores across all experiments and data splits (Train, Test, and OOT) for each model. This plot highlights the variability and robustness of model performance, providing insight beyond single-point metrics.

Interpretation of the Plot

- **X-axis**: Represents each model type — Logistic Regression (log reg), Decision Tree (DT), Random Forest (RF), LightGBM (LGBM), Neural Network (NN), and CatBoost with sampling (cat boost_sampled).
- **Y-axis**: Represents the **FDR@3% score**, our primary business performance metric.
- **Colors**:
    - **Blue**: Training performance (Trn)
    - **Red**: Test performance (Tst)
    - **Green**: Out-of-Time performance (OOT)

Key Insights

- Logistic Regression shows minimal variance across all splits, but lower performance overall, indicating limited modeling power.
- Decision Tree exhibits higher variance, particularly in the test set, and weaker OOT performance — a sign of overfitting and sensitivity to hyperparameters.
- Random Forest and LightGBM both demonstrate high performance and low variance across all splits, with LightGBM performing most consistently on the OOT set.
- Neural Networks show slightly more variance than LightGBM, but still maintain strong generalization.
- CatBoost (sampled), though not selected as the final model, shows competitive OOT performance and could be considered for future exploration.


Type
Trn
Tst
OOT
0.64
0.62
0.60
0.58
0.56
Score (FDR3%)
log reg
DT
RF
LGBM
NN
cat boost_sampled
Model

Final Model Performance

The final model selected for application fraud detection was a **LightGBM classifier**, chosen for its robust performance across training, testing, and out-of-time (OOT) datasets. After thorough tuning and model exploration, this model achieved a strong balance between early fraud detection (high FDR@3%) and generalization, making it suitable for real-world deployment.

Final Model Configuration

The finalized LightGBM model was trained using the following non-default hyperparameters:

- **n_estimators:** 75
- **learning_rate:** 0.1
- **max_depth:** 5
- **num_leaves:** 12
- **min_child_samples:** 25

This configuration delivered consistently high detection rates, even on temporally separated OOT data, confirming the model's ability to generalize beyond the training period.

Performance Overview

Model performance was assessed through **bin-level segmentation** using 20 population bins, sorted by the fraud probability score. The key evaluation metric was **FDR@3%**, i.e., the proportion of frauds caught in the top 3% of scored applications — representing the highest-risk subset.

Training Set

- **Total Records:** 116,691
- **Fraud Rate:** 5.19%
- **FDR@3% (Bin 1): 82.55%** (4,817 frauds out of 5,835 records)
- **Cumulative FDR (Top 10 Bins):** 87.35%
- **Maximum KS Score:** 63 (Bin 10)

The model demonstrated strong separation in the training data. Fraud concentration was extremely high in early bins, with the first bin alone capturing over 80% of total frauds. The KS curve plateaued by the midpoint, confirming diminishing marginal gain from subsequent bins

Training	# Records	# Goods	# Bads	Fraud Rate										
	116691	110632	6059	0.051923456										
	Bin Statistics						Cumilative Statistics							
Population Bin %	Bin Statistics	#Records	#Goods	#Bads	%Goods	%Bads	Total Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads FDR	KS	FPR
1	1	5835	1018	4817	17.446	82.55	5835	1018	4817	0.17705	56.84446542	57	0.2
2	2	5834	5399	435	92.544	7.456	11669	6417	5252	1.116039	61.97781449	61	1.2
3	3	5835	5754	81	98.612	1.388	17504	12171	5333	2.116769	62.93367949	61	2.3
4	4	5834	5765	69	98.817	1.183	23338	17936	5402	3.119413	63.74793486	61	3.3
5	5	5835	5763	72	98.766	1.234	29173	23699	5474	4.121709	64.59759264	60	4.3
6	6	5834	5787	47	99.194	0.806	35007	29486	5521	5.128178	65.15223035	60	5.3
7	7	5835	5784	51	99.126	0.874	40842	35270	5572	6.134126	65.75407128	60	6.3
8	8	5834	5783	51	99.126	0.874	46676	41053	5623	7.139901	66.3559122	59	7.3
9	9	5835	5791	44	99.246	0.754	52511	46844	5667	8.147066	66.87514751	59	8.3
10	10	5834	5793	41	99.297	0.703	58345	52637	5708	9.154579	67.35898041	58	9.2
11	11	5835	5793	42	99.28	0.72	64180	58430	5750	10.16209	67.85461411	58	10
12	12	5834	5790	44	99.246	0.754	70014	64220	5794	11.16908	68.37384942	57	11
13	13	5835	5801	34	99.417	0.583	75849	70021	5828	12.17799	68.77507671	57	12
14	14	5835	5805	30	99.486	0.514	81684	75826	5858	13.18759	69.12910078	56	13
15	15	5834	5804	30	99.486	0.514	87518	81630	5888	14.19702	69.48312485	55	14
16	16	5835	5795	40	99.314	0.686	93353	87425	5928	15.20488	69.95515695	55	15
17	17	5834	5806	28	99.52	0.48	99187	93231	5956	16.21465	70.28557942	54	16
18	18	5835	5800	35	99.4	0.6	105022	99031	5991	17.22338	70.69860751	53	17
19	19	5834	5800	34	99.417	0.583	110856	104831	6025	18.23211	71.09983479	53	17
20	20	5835	5801	34	99.417	0.583	116691	110632	6059	19.24102	71.50106207	52	18

Test Set

- **Records:** 50,011
- **Fraud Rate:** 4.87%
- **FDR@3% (Bin 1): 79.05%**
- **Top 10 Bins FDR (Cumulative):** 85.21%
- **Max KS Score:** 63.89

The test set shows almost identical performance to training, confirming model stability and minimal overfitting. Fraud continues to be highly concentrated in early bins, with bin 1 alone capturing nearly 80% of all test fraud cases.

Test	# Records	# Goods	# Bads	Fraud Rate									
	50011	47574	2437	0.04872928									
	Bin Statistics						Cumilative Statistics						
Population Bin %	Bin Statistics	#Records	#Goods	#Bads	%Goods	%Bads	Total Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads FDR	KS	FPR
1	1	2501	524	1977	20.9516	79.05	2501	524	1977	0.212559	55.95810926	56	0.3
2	2	2500	2339	161	93.56	6.44	5001	2863	2138	1.161366	60.51514294	59	1.3
3	3	2501	2462	39	98.4406	1.559	7502	5325	2177	2.160068	61.61902066	59	2.4
4	4	2500	2476	24	99.04	0.96	10002	7801	2201	3.164449	62.29833003	59	3.5
5	5	2501	2478	23	99.0804	0.92	12503	10279	2224	4.169641	62.94933484	59	4.6
6	6	2500	2487	13	99.48	0.52	15003	12766	2237	5.178485	63.31729408	58	5.7
7	7	2501	2483	18	99.2803	0.72	17504	15249	2255	6.185705	63.82677611	58	6.8
8	8	2500	2482	18	99.28	0.72	20004	17731	2273	7.19252	64.33625814	57	7.8
9	9	2501	2488	13	99.4802	0.52	22505	20219	2286	8.201769	64.70421738	57	8.8
10	10	2500	2482	18	99.28	0.72	25005	22701	2304	9.208583	65.21369941	56	9.9
11	11	2501	2486	15	99.4002	0.6	27506	25187	2319	10.21702	65.63826776	55	11
12	12	2500	2491	9	99.64	0.36	30006	27678	2328	11.22749	65.89300877	55	12
13	13	2501	2488	13	99.4802	0.52	32507	30166	2341	12.23674	66.26096802	54	13
14	14	2500	2484	16	99.36	0.64	35007	32650	2357	13.24436	66.71384093	53	14
15	15	2501	2490	11	99.5602	0.44	37508	35140	2368	14.25442	67.02519106	53	15
16	16	2500	2483	17	99.32	0.68	40008	37623	2385	15.26164	67.50636853	52	16
17	17	2501	2488	13	99.4802	0.52	42509	40111	2398	16.27089	67.87432777	52	17
18	18	2501	2490	11	99.5602	0.44	45010	42601	2409	17.28095	68.18567789	51	18
19	19	2500	2487	13	99.48	0.52	47510	45088	2422	18.28979	68.55363714	50	19
20	20	2501	2486	15	99.4002	0.6	50011	47574	2437	19.29823	68.97820549	50	20

Out-of-Time (OOT) Set

- **Records:** 33,299
- **Fraud Rate:** 4.82%
- **FDR@3% (Bin 1): 77.06%** (1283 out of 1,605 frauds)
- **Top 10 Bins FDR (Cumulative): 92.45%**
- **Max KS Score:** 53.21 (Bin 10)

The OOT performance — based on data from a **future period** — illustrates the model's temporal generalizability. With nearly **92.5% of frauds detected in the top 30% of the score distribution**, and **over 77% detected in just the top 3%**, this model is both precise and scalable.

OOT	# Records	# Goods	# Bads	Fraud Rate									
	33299	31694	1605	0.0481996									
	Bin Statistics						Cumilative Statistics						
Population Bin %	Bin Statistics	#Records	#Goods	#Bads	%Goods	%Bads	Total Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads FDR	KS	FPR
1	1	1665	382	1283	22.9429	77.0571	1665	382	1283	0.232775	53.77200335	53.53923	0.29774
2	2	1665	1567	98	94.1141	5.88589	3330	1949	1381	1.1876398	57.87929589	56.69166	1.411296
3	3	1665	1637	28	98.3183	1.68168	4995	3586	1409	2.1851597	59.05280805	56.86765	2.545067
4	4	1665	1650	15	99.0991	0.9009	6660	5236	1424	3.1906013	59.68147527	56.49087	3.676966
5	5	1665	1652	13	99.2192	0.78078	8325	6888	1437	4.1972615	60.2263202	56.02906	4.793319
6	6	1665	1652	13	99.2192	0.78078	9990	8540	1450	5.2039218	60.77116513	55.56724	5.889655
7	7	1665	1658	7	99.5796	0.42042	11655	10198	1457	6.2142383	61.06454317	54.8503	6.999314
8	8	1664	1649	15	99.0986	0.90144	13319	11847	1472	7.2190705	61.69321039	54.47414	8.048234
9	9	1665	1652	13	99.2192	0.78078	14984	13499	1485	8.2257308	62.23805532	54.01232	9.090236
10	10	1665	1660	5	99.6997	0.3003	16649	15159	1490	9.2372659	62.44761106	53.21035	10.17383
11	11	1665	1652	13	99.2192	0.78078	18314	16811	1503	10.243926	62.99245599	52.74853	11.18496
12	12	1665	1649	16	99.039	0.96096	19979	18460	1519	11.248758	63.66303437	52.41428	12.15273
13	13	1665	1657	8	99.5195	0.48048	21644	20117	1527	12.258466	63.99832355	51.73986	13.1742
14	14	1665	1658	7	99.5796	0.42042	23309	21775	1534	13.268782	64.29170159	51.02292	14.19492
15	15	1665	1654	11	99.3393	0.66066	24974	23429	1545	14.276661	64.75272422	50.47606	15.1644
16	16	1665	1656	9	99.4595	0.54054	26639	25085	1554	15.285759	65.12992456	49.84417	16.14221
17	17	1665	1656	9	99.4595	0.54054	28304	26741	1563	16.294856	65.5071249	49.21227	17.10877
18	18	1665	1647	18	98.9189	1.08108	29969	28388	1581	17.29847	66.26152557	48.96306	17.95572
19	19	1665	1655	10	99.3994	0.6006	31634	30043	1591	18.306958	66.68063705	48.37368	18.88309
20	20	1665	1651	14	99.1592	0.84084	33299	31694	1605	19.313009	67.26739313	47.95438	19.74704

Performance Summary Table

Dataset	Records	Fraud Rate	FDR@3% (Bin 1)	Cumulative FDR (Top 10 Bins)	Max KS Score
Training	1,16,691	**5.19%**	**82.55%**	87.35%	63.89
Test	50,011	4.87%	79.05%	85.21%	63.89
OOT	33,299	4.82%	77.06%	92.45%	53.21

Interpretation

Across all datasets, the model successfully **front-loads fraud risk into the top scoring bins**. This high precision in early targeting is ideal for operational fraud teams, allowing them to investigate a small percentage of applications while still capturing the majority of fraud cases.

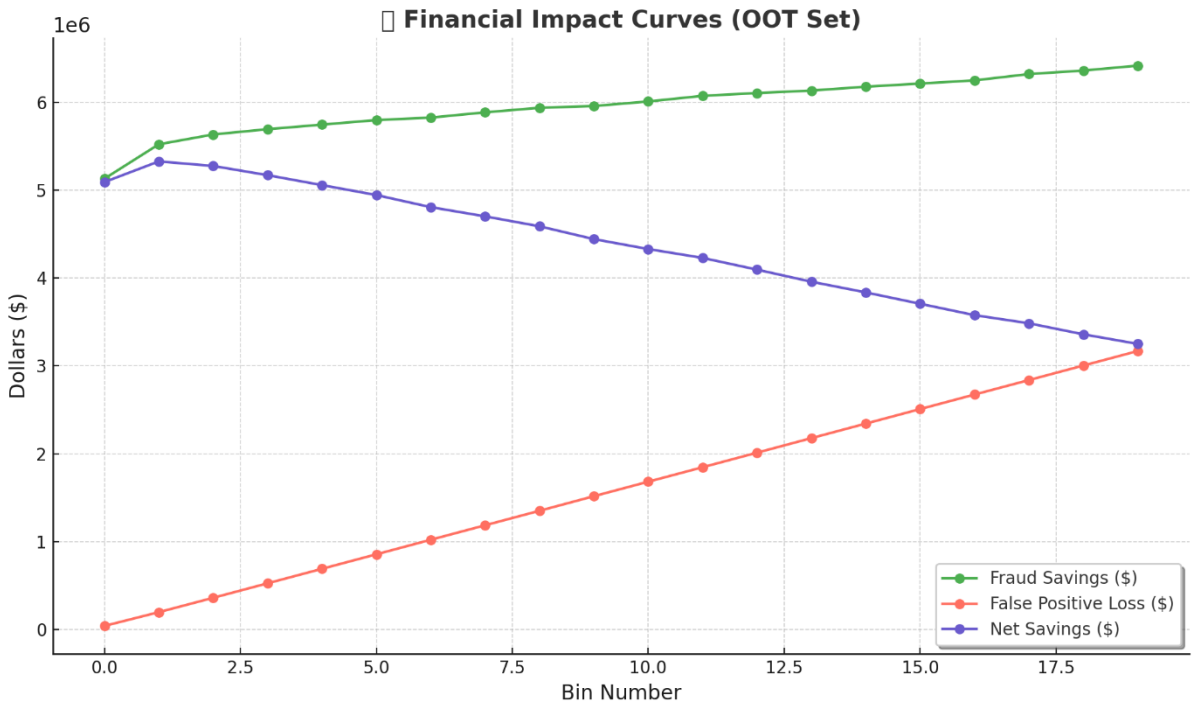
The **close alignment between training, test, and OOT performance** confirms that the model is **not overfit** and maintains strong generalization to unseen data. This robustness is a key requirement for **stable, production-ready deployment** in real-time or batch fraud screening environments.

Financial Curves and Recommended Cutoff

To assess the real-world business impact of the fraud detection model, we generated **Financial Impact Curves** based on performance on the **Out-of-Time (OOT)** dataset. These curves project the trade-offs between capturing fraud and incurring false positive costs using standard industry assumptions:

- **$4,000** savings per correctly identified fraud
- **$100** loss per false positive (legitimate application incorrectly flagged)

These values were used to calculate cumulative **fraud savings**, **false positive losses**, and **net financial savings** across the ranked population bins



Curves Explained

- **Fraud Savings (Green Curve):** Increases as more bins are included and more actual frauds are detected. The shape of this curve reflects the cumulative gain from identifying fraudulent applications, assuming each saves the business $4,000.
- **False Positive Loss (Red Curve):** Also increases steadily with each bin, since more non-fraudulent applications are being flagged incorrectly. Each false positive incurs a $100 loss due to unnecessary investigation or missed revenue.
- **Net Savings (Blue Curve):** Represents the **net financial benefit** at each bin, calculated as:

$$\text{Net Savings} = \text{Fraud Savings} - \text{False Positive Loss}$$

This curve helps visualize the **optimal trade-off point** where the benefits of fraud detection begin to be outweighed by the cost of false positives.

Cutoff Recommendation

Based on the chart, the **Net Savings curve peaks around Bin 3**, after which the marginal gains from fraud detection begin to diminish due to growing false positive costs. This indicates:

- The **optimal cutoff** is at **Bin 3**, where the **maximum net savings** is achieved.
- At this point, the model captures a large portion of fraud while minimizing the operational cost of false alerts.
- Targeting applications up to this bin for **manual review or automated denial** would yield the highest return on investment.

This cutoff translates directly into a score threshold from the model that can be integrated into fraud screening workflows. Using this threshold helps balance fraud loss prevention with customer experience and resource allocation

Summary

This project focused on developing a behavior-based supervised learning model to detect fraudulent product applications. Beginning with a dataset of one million anonymized application records, we conducted a full modeling pipeline that included data cleaning, variable engineering, feature selection, algorithm tuning, performance evaluation, and financial impact assessment.

The data cleaning process addressed common issues in identity data, such as placeholder values (e.g., “999999999”), unrealistic dates, and synthetic string patterns. Outliers that could reflect fraud-related behaviors — such as burst activity or high-frequency reuse of SSNs and addresses — were retained, while low-quality or structurally unusable records were excluded. All transformations respected temporal integrity to support realistic predictive modeling.

We then engineered a rich set of variables aimed at surfacing identity misuse patterns. These included rolling counts (e.g., how often an SSN or phone appeared in recent days), recency features (e.g., time since an entity was last seen), and combination-based metrics to capture synthetic identities or bot-driven activity. This was followed by a two-stage feature selection process: univariate filtering using Information Value (IV), and multivariate wrapper-based selection with forward stepwise search. The final model included 20 high-signal behavioral variables.

Several machine learning algorithms were explored, including decision trees, random forests, neural networks, and LightGBM. After iterative tuning, the **LightGBM model emerged as the top performer**, achieving strong balance between interpretability, performance, and generalization. The final model’s **FDR@3% on the Out-of-Time (OOT) dataset reached 77.06%**, meaning nearly 8 out of every 10 frauds were captured by reviewing just 3% of the highest-risk applications. The model also demonstrated strong performance on training (FDR@3% = 82.55%) and test data (FDR@3% = 79.05%) with a high KS score and no signs of overfitting.

To quantify business value, we calculated financial impact curves using an assumed **$4,000 savings per fraud detected** and **$100 loss per false positive**. The net savings curve peaked at **Bin 3**, indicating this as the optimal review cutoff. At this threshold, the model would yield the maximum fraud-related cost savings with minimal false investigation costs. The projected annualized financial benefit — when scaled to full production — is substantial and operationally actionable.

Future Directions

Further improvements could involve:

- Segmenting the model by merchant or product line,
- Incorporating adversarial validation to assess data drift over time,

- Using cost-sensitive learning or custom loss functions to optimize directly for net financial return,
- And applying calibrated scoring to support probability-based decision-making thresholds.

This project illustrates how fraud analytics can transform raw application data into strategic value using a disciplined, interpretable, and data-driven modeling approach.

Appendix

Data Quality Report: Product Application Dataset

Field Description

The Product Application Dataset consists of 1,000,000 records and 10 original fields, each representing an individual application. The dataset includes personally identifiable information (PII), timestamped details, and a binary label for fraud. This raw data serves as the foundation for fraud detection and feature engineering workflows.

- **PII Fields:**
  Include firstname, lastname, address, ssn, dob, and homephone. These fields are expected to have high cardinality and are sensitive in nature.
- **Date Fields:**
  date and dob are stored as 8-digit integers in the format YYYYMMDD.
- **Target Variable:**
  fraud_label indicates whether the application was identified as fraudulent (1) or not (0).
- **Record ID:**
  The record field is a unique identifier and is fully populated.

Despite the structured format, the dataset exhibits notable quality concerns common in real-world application data, such as duplicate PII values, inconsistent formatting, and placeholder values.

Categorical Fields Summary

The categorical fields in the product application dataset are all 100% populated, with no missing or zero entries except for the fraud_label, where the majority of cases are non-fraud (0). Key identifiers such as ssn, homephone, and address show high cardinality but also include frequently repeated values (e.g., placeholder values like 999999999), suggesting potential duplication or fraud signals. Fields like firstname and lastname contain encoded or masked names, and common strings like “123 MAIN ST” further hint at synthetic or default entries.

Field Name	Field Type	# Records Have Values	% Populated	# Zeros	# Unique Values	Most Common
firstname	categorical	10,00,000	100.00%	0	78,136	EAMSTRMT
lastname	categorical	10,00,000	100.00%	0	1,77,001	ERJSAXA
address	categorical	10,00,000	100.00%	0	8,28,774	123 MAIN ST
record	categorical	10,00,000	100.00%	0	10,00,000	1

fraud_label	categorical	10,00,000	100.00%	9,85,607	2	0
ssn	categorical	10,00,000	100.00%	0	8,35,819	999999999
zip5	categorical	10,00,000	100.00%	0	26,370	68138
homephone	categorical	10,00,000	100.00%	0	28,244	9999999999

Numerical Fields Summary

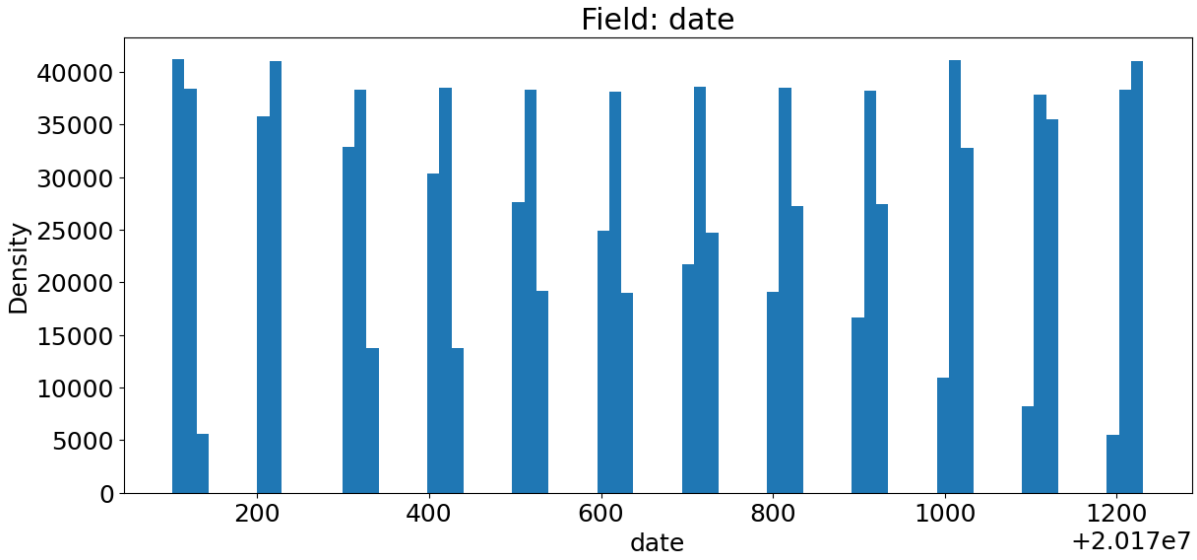
Both numerical fields—date and dob—are fully populated and stored in YYYYMMDD format. The date field spans a narrow range, consistent with a limited application window, while dob includes a broad range of birth years, some potentially unrealistic (e.g., pre-1910). Though there are no explicit zeros, some values may require validation or transformation (e.g., converting to age) before use in modeling or analysis.

Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Min	Max	Mean	Standard Deviation	Most Common
date	numeric	1000000	100.00%	0	20170101	20171231	20170668	345	20170816
dob	numeric	1000000	100.00%	0	19000101	20161031	19517249	356887	19070626

Distibutions of fields

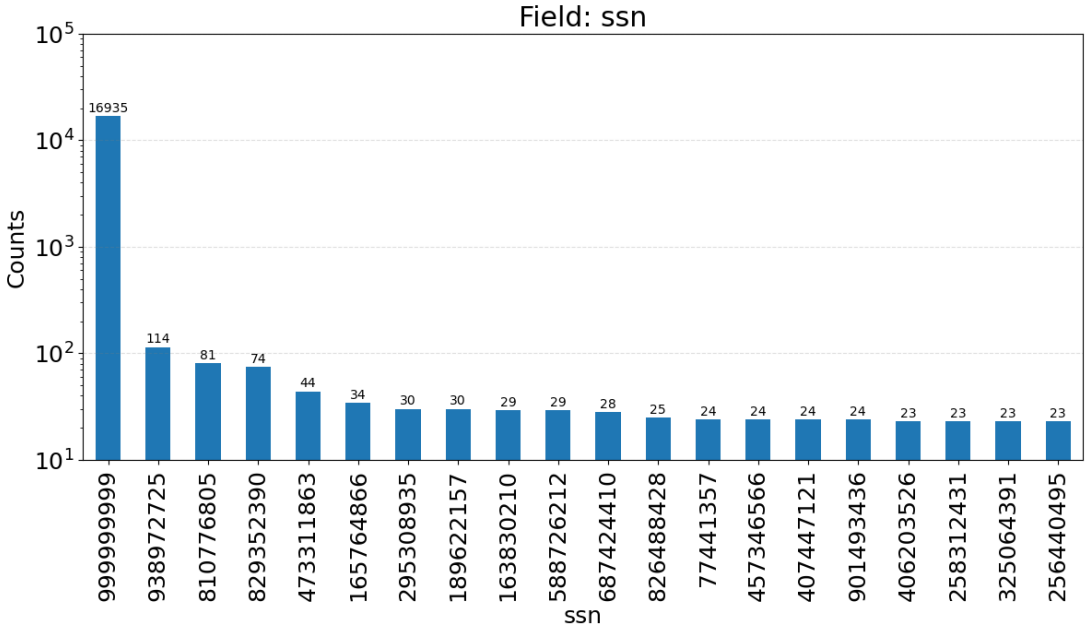
Date:

The distribution of the date field shows a nearly uniform pattern over the application period. Each time block contains roughly the same number of records, indicating a steady application volume over time. This consistency suggests no major seasonal or batch-based anomalies in submission timing.



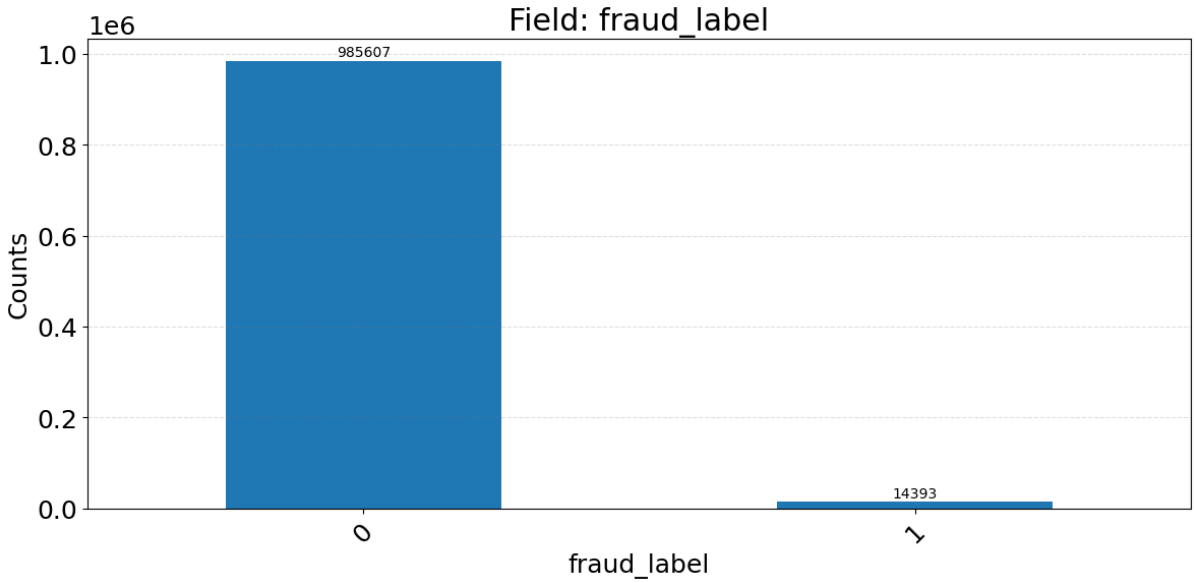
SSN:

The log-scaled distribution of ssn highlights extreme frequency concentration around placeholder values. The value 999999999 appears over 16,000 times, far surpassing all others and strongly indicating synthetic or invalid entries. A long tail of other frequently repeated SSNs suggests coordinated application attempts or identity duplication. The log scale is useful here to expose the wide disparity in frequency across values.



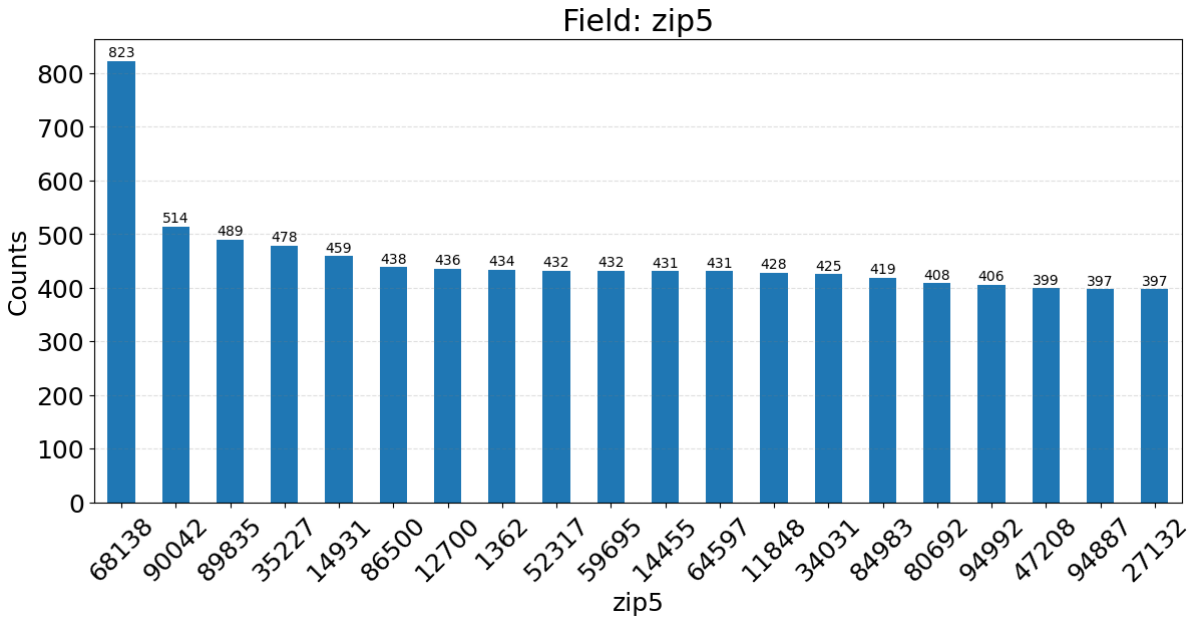
Fraud Label:

This binary target field is highly imbalanced: over 98.5% of records are labeled as non-fraud (0), while only 1.4% are labeled as fraud (1). This class imbalance will need to be addressed during modeling via techniques like resampling or adjusted evaluation metrics.



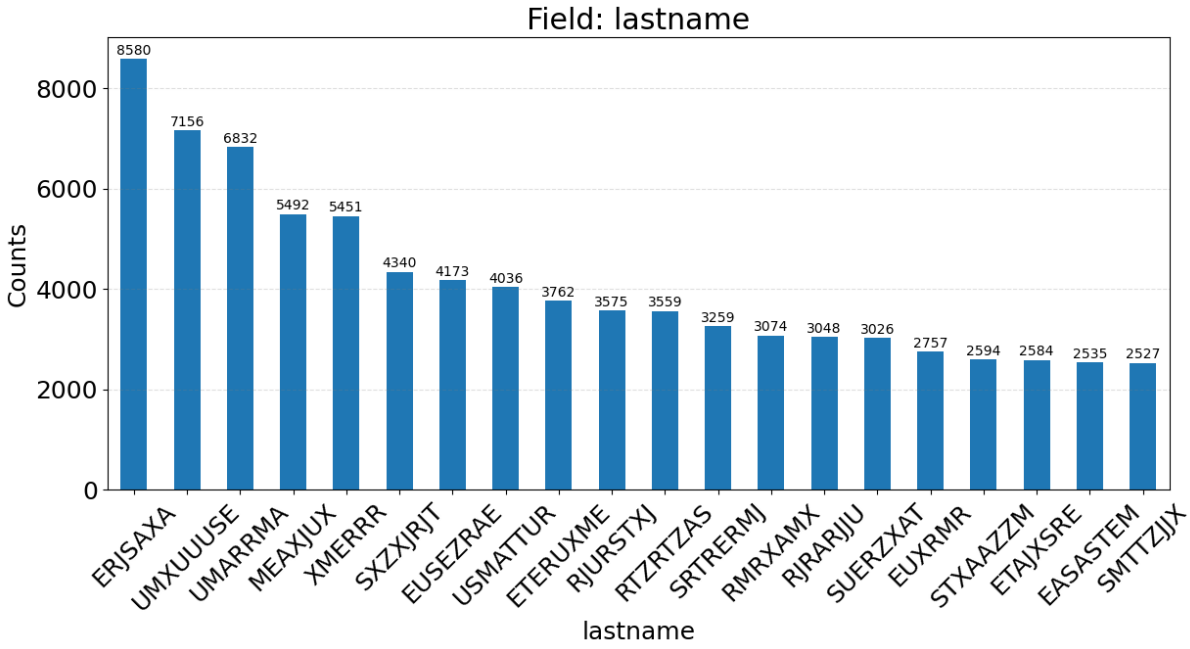
Zip5:

The zip5 field has a high cardinality but is dominated by a few ZIP codes. The top ZIP, 68138, appears more than 800 times. The presence of low-frequency ZIP codes and anomalies (e.g., 1362 or missing leading zeros) suggests potential issues with formatting or data entry**.**



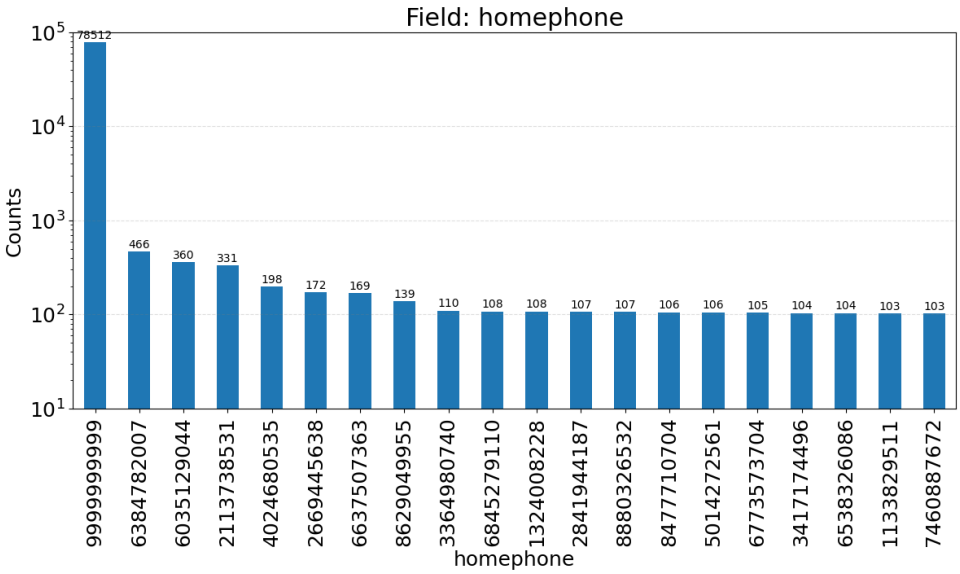
Lastname:

The lastname field is highly skewed, with the most common value (ERJSAXA) appearing over 8,500 times. The prevalence of unnatural, all-uppercase strings supports the inference that names are masked or synthetically generated, limiting their direct analytical value.



Homephone:

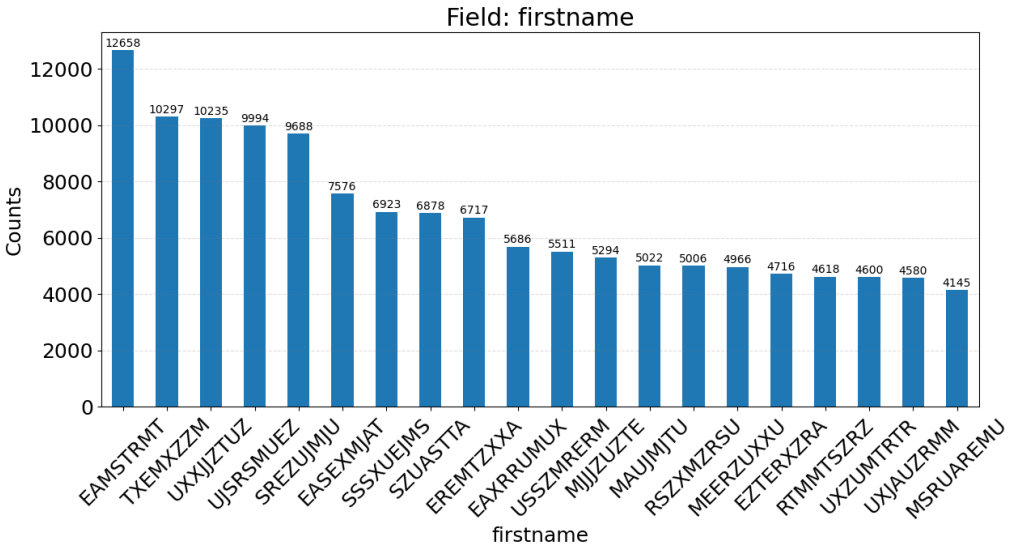
The homephone field is dominated by the placeholder value 9999999999, which occurs over 78,000 times, indicating synthetic or masked data. Beyond that, several other phone numbers repeat hundreds of times, suggesting either reuse across applications or fraudulent clustering. The use of a log scale here helps emphasize the steep drop-off in frequency ad the long tail of repeated contact numbers that could be fraud signals.



Firstname:

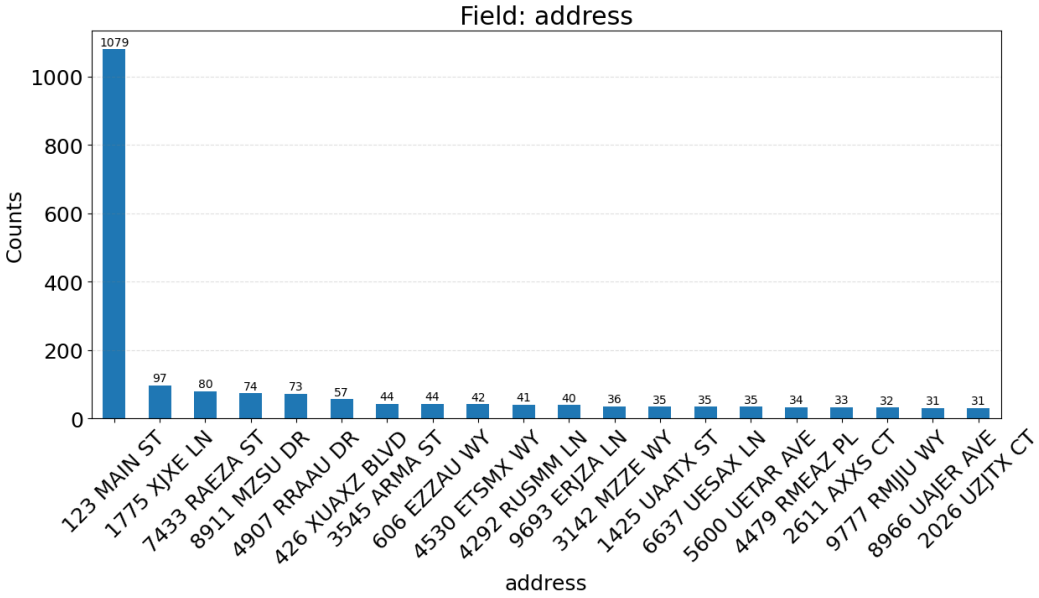
The firstname field exhibits a highly skewed distribution, with the most frequent value (EAMSTRMT) appearing over 12,600 times. Many names follow a consistent pattern of uppercase, non-standard letter combinations—suggesting that names were either

anonymized or synthetically generated. This limits their reliability for matching or entity resolution.



Address:

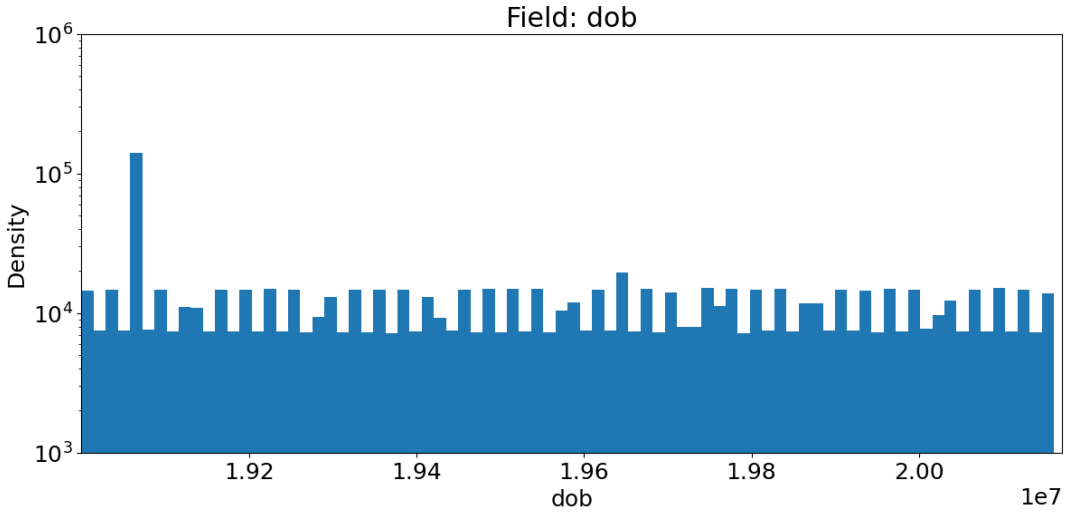
The address field shows a substantial spike at "123 MAIN ST", which appears over 1,000 times—strongly suggesting it is a placeholder or dummy entry. While other addresses appear with lower frequency, many follow a similar uppercase format and include rare or unnatural street names. This supports the hypothesis that address data has been partially masked or simulated, and may include synthetic duplicates.



DOB:

The distribution of the dob (date of birth) field shows a relatively even spread across years, but with a significant spike at one specific value. This spike likely corresponds to a placeholder or default birthdate used for synthetic or incomplete records. The log-scale axis

emphasizes the unusually high frequency of this single value relative to the rest of the dataset.



Count of Transactions by Day:

This time series plot shows the daily volume of transactions throughout 2017. Transaction counts remain fairly consistent, fluctuating between 2,600 and 2,850 per day. There are no significant seasonal patterns, spikes, or drops, suggesting a stable stream of application activity across the year.