BDA 610 Advanced Business Statistics

Project Title: A Statistical Approach for Detecting Legit and Fraudulent Bank Users

Prepared By:
Jairo Onate – MUID: 11062465
John Pole Madhu – MUID: 11062542
Ajay Katta – MUID: 11055284

Submitted To:
Rui Sun, Ph.D.
Assistant Professor of Economics
Stetson-Hatcher School of Business
Mercer University

October 11, 2024

Table of Contents

1. Introduction

The technology advances lead to the resizing of computers by the 1980's, making them accessible to the community and user friendly. By 1989, in California it was reported that almost 36% of the population owned or had access to them at work or home [8]. This technology developments and software introductions like Prodigy owned by Sears to access customer bank accounts through a private network, opened an opportunity window to companies like Wells Fargo to offer online bank account services. During 1989 and 1990's out of their 3.5 million customers, 10,000 used their service through Prodigy. The first online banking platform went live in 1995 and was created by Wells Fargo.

The evolution in the banking systems had led to different strategies to perform fraud in the system on behalf of the customers or external entities. In the United States in 2023, consumers losses total $10 billions due to fraud according to the Federal Trade Commission. The most common fraud modalities are led by Investment scams with $4.6 billion losses reported with a 21% increase related to the amount reported in 2022, followed by Imposter Scams with totaling $2.7 billion losses [2]. The purpose of the study is to determine how can classification algorithms determine whether a customer is legit or fraudulent?

To conduct this research, the data that will be used is Bank Account Fraud Dataset Suite (NeurIPS 2022) from the Bank Account Fraud (BAF). The purpose is to analyze the relation of a customer's characteristics such as their annual income, the similarity between their email and legal name, age, banking account transfer amounts and number of months lived in the previous address, in the likelihood of being classified as fraudulent or legit customer.

2. Literature Review

The following papers have been considered as part of the literature review for the understanding of bank fraud in the United States and the tools available in the industry, to face and reduce the problem.

Source 1 - Effective Strategies For Protecting Your Bank Account From Fraud

To protect your bank account from fraud, take advantage of your bank's security features. Enable two-factor authentication for added login security. Turn on transaction alerts to track account activity in real-time [9]. Keep your contact information up to date for notifications about unusual activity. Stay vigilant and pay attention to fraud detection tips provided by your bank. Following these precautions can help secure your account and prevent unauthorized access.

Steps to Take If Your Bank Account Numbers Are Compromised:

Report the Fraud: Immediately inform your bank to halt further transactions and prevent additional loss, as advised by Sean Dyon, director of strategic alliances for HID Global.

Report to CFPB: If a scammer has your bank account numbers, promptly report the fraud to the Consumer Financial Protection Bureau. According to Dyon, this entity provides essential services to help navigate the situation.

Source 2 – Examining Check Fraud Banks And Community Impact

Understanding Check Fraud: Check fraud involves using fraudulent checks or deceptive practices to exploit the banking system, including forgery, counterfeit checks, and altering details for illegal gains [6].

Financial Losses for Banks: Banks face significant financial losses when handling fraudulent checks, impacting their profitability and increasing the costs of fraud investigation and prevention.

Negative Impact on Customers: Customers may suffer financial losses, damaged credit, or legal issues if they unknowingly deposit fraudulent checks, eroding trust in their banks.

Fraud detection software automatically stops online fraud by analyzing user actions, payments, and signups for potential risks, often using AI-driven insights. Modern solutions offer a consolidated platform with real-time intelligence, customizable risk rules, and seamless integration. These systems help prevent payment fraud, identity theft, and account takeovers, improving efficiency with minimal human oversight.

Source 3 – Fraud Detection Softwares And Tools In 2024

Here are the top 8 fraud detection software companies: SEON offers a flexible, AI-driven platform with real-time insights and seamless integration. TruValidate focuses on identity and transaction analysis but lacks machine learning and AML solutions. Emailage provides email risk scoring but has limited customization. ThreatMetrix uses vast data for authentication, though lacks real-time accuracy [10]. Sift specializes in digital trust but lacks comprehensive AML tools. Feedzai excels in AI-powered AML, but its interface is fragmented. ArkOwl provides email and phone verification, while Trustfull offers onboarding solutions but lacks full fraud journey coverage.

Source 4 - Fraud Prevention And Detection: A Macro Perspective

"Fraud Prevention and Detection in the United States: A Macro Perspective" by Sara Aliabadi, Alireza Dorestani, and Mohammed Qadri (2011), the authors examine the limitations of the auditing profession in detecting fraud [1]. Key types of fraud include fraudulent financial reporting, involving overstated revenues or understated expenses, and misappropriation of assets. On average, companies lose about 7% of their revenues to fraud.

Historical cases like Charles Ponzi (1920), Bernie Madoff (2009), and Satyam (2009) illustrate how corporate fraud repeats old patterns. Madoff's Ponzi scheme led to over $50 billion in losses by falsely promising 8-12% annual returns. The Satyam scandal involved inflating assets by $1 billion, causing its value to drop from $2.6 billion to $600 million, with auditors failing to detect the fraud.

A study of 216 fraud cases (1996-2006) showed that auditors detected only 11% of frauds. Other parties, such as employees (17%), media (13%), and analysts (14%), played a larger role in uncovering fraud (Hyatt, 2010). This raises concerns about auditor training.

The Fraud Triangle (SAS 99) identifies incentive/pressure, opportunity, and rationalization as conditions for fraud. However, the authors find significant gaps in fraud-related education, only 3.5% of universities offer undergraduate courses, 4% offer graduate courses, and none offer standalone ethics course. Their survey of 201 U.S. universities showed that over 95% do not offer fraud prevention courses.

To address these shortcomings, the authors propose the Fraud Deterrence Triangle, which includes:
1. Fraud Prevention/Detection Education
2. Ethics Training
3. Corporate Governance.

Recommendations include requiring continuous training in fraud prevention and ethics for auditors by SEC, PCAOB, and AICPA. Public accounting firms should develop in-house programs to educate employees on long-term fraud detection, and universities should expand offerings in this area.

Source 5 - Combating Banking Fraud With It: Integrating Machine Learning And Data Analytics

Through the years with the evolution of technology, this has facilitated new mechanisms to the fraudsters to threaten and vulnerate the system (Buchanan, 2019). The integration of Machine Learning (ML) and Data Analytics has helped against the fight of detecting activities leading to fraud by identifying patterns from the data [4].

The implementation of detection techniques such as logistic regression models allows to predict the likelihood of a fraud transaction based on historic data (Moreira et al., 2022), while unsupervised learning algorithms such as clustering and anomaly detection will help identifying outliers that are often cataloged as fraudulent activities. In the other hand the K-means and isolation forest techniques have been used to detect anomalies in transactions (Sambrow & Iqbal, 2023; Zhuang et al., 2006). Deep Learning involves neuronal networks are effective to capture complex patterns in large transactional data and can be applied with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to perform the detection (Sambrow &Iqbal, 2023; Jurgovsky et al., 2018). The future trends in fraud detection in the banking industry lies in the constant evolution of AI and developments such as XAI (Explanatory Artificial Intelligence), enriching the interpretation of Machine Learning (ML) models (Adadi & Berrada,2018). Also, Blockchain is being studied due to its potential to enhance the security and transparence of transactions in the financial industry.

Despite of the efforts of implementing ML models for fraud detection, there are limitations associate to this as its needed large and high-quality labeled datasets, risks of over fitting and difficulties when explaining and interpreting complex models (Adadi & Berrada, 2018; Jurgovsky et al., 2018). It is important to integrate other areas like finance, criminology and psychology to combine insights and understand how the behaviors studied by other fields can lead to an early detection in fraud.

3. Identify data, data sources and data characteristics

3.1 Dataset collection

The dataset used for this analysis was sourced from **Kaggle** [3], a well-known platform that provides a wide range of datasets for data analysis, machine learning, and research.

The dataset, titled ***Bank Account Fraud Dataset (NeurIPS 2022)***, is part of the **NeurIPS 2022 competition** and is focused on the detection of fraudulent activities associated with bank accounts.

Kaggle has public data collections in their website, where the bank fraud dataset was manually downloaded in CSV format from this source and subsequently imported into R Studio for further exploration and analysis. The dataset is structured to help detect fraudulent activities in bank accounts, making it a suitable candidate for machine learning models focused on fraud detection.

A more detailed explanation of the data collection process, including the file structure and preliminary data exploration, can be found in the sections below.

3.1 Dictionary

- **income (numeric):** Annual income of the applicant (in decile form). Ranges between [0.1, 0.9].
- **name_email_similarity (numeric):** Metric of similarity between email and applicant’s name. Higher values represent higher similarity. Ranges between [0, 1].
- **prev_address_months_count (numeric):** Number of months in previous registered address of the applicant, i.e. the applicant’s previous residence, if applicable. Ranges between [−1, 380] months (-1 is a missing value).
- **current_address_months_count (numeric):** Months in currently registered address of the applicant. Ranges between [−1, 429] months (-1 is a missing value).
- **customer_age (numeric):** Applicant’s age in years, rounded to the decade. Ranges between [10, 90] years.
- **days_since_request (numeric):** Number of days passed since application was done. Ranges between [0, 79] days.
- **intended_balcon_amount (numeric):** Initial transferred amount for application. Ranges between [−16, 114] (negatives are missing values).
- **payment_type (categorical):** Credit payment plan type. 5 possible (anonymized) values.
- **zip_count_4w (numeric):** Number of applications within same zip code in last 4 weeks. Ranges between [1, 6830].
- **velocity_6h (numeric):** Velocity of total applications made in last 6 hours i.e., average number of applications per hour in the last 6 hours. Ranges between [−175, 16818].
- **velocity_24h(numeric):** Velocity of total applications made in last 24hours, i.e. average number of applications per hour in the last 24 hours. Ranges between [1297, 9586]

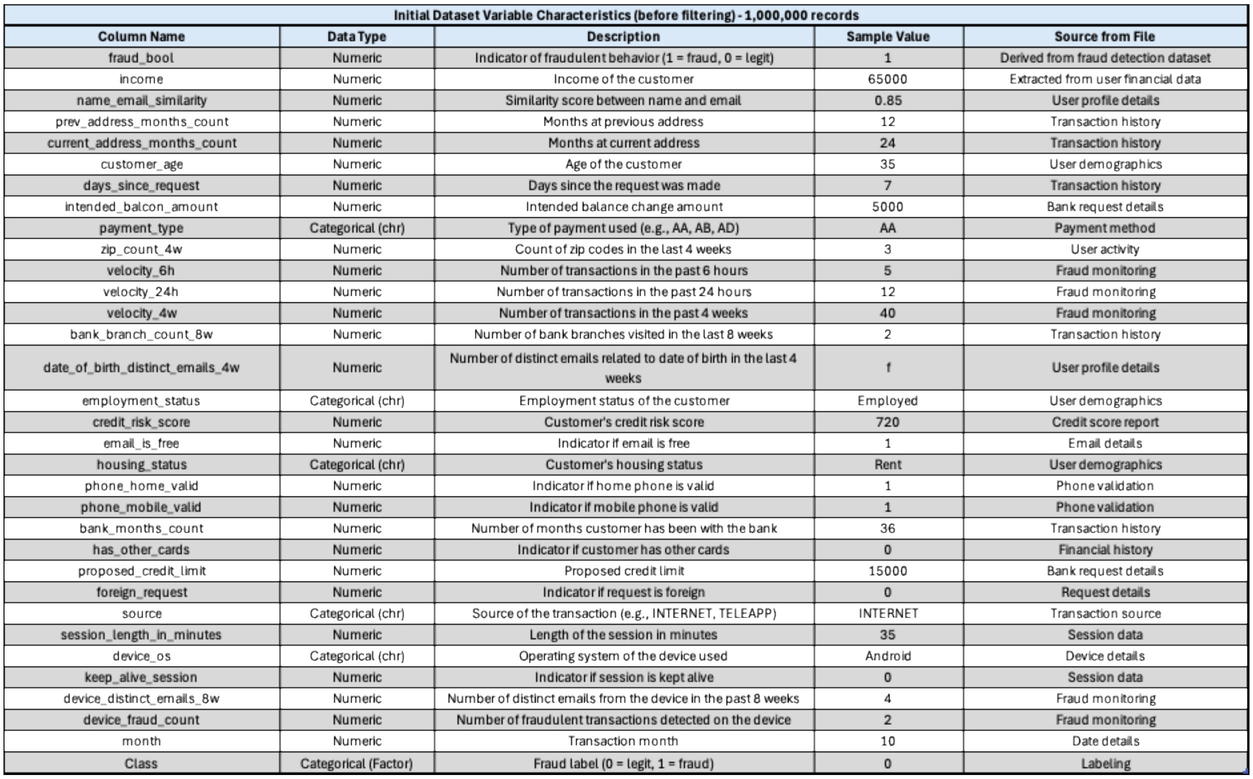
- **velocity_4w (numeric):** Velocity of total applications made in last 4 weeks, i.e., average number of applications per hour in the last 4 weeks. Ranges between [2825, 7020].
- **bank_branch_count_8w (numeric):** Number of total applications in the selected bank branch in last 8 weeks. Ranges between [0, 2404].
- **date_of_birth_distinct_emails_4w (numeric):** Number of emails for applicants with same date of birth in last 4 weeks. Ranges between [0, 39].
- **employment_status (categorical):** Employment status of the applicant. 7 possible (annonymized) values.
- **credit_risk_score (numeric):** Internal score of application risk. Ranges between [−191, 389].
- **email_is_free (binary):** Domain of application email (either free or paid).
- **housing_status (categorical):** Current residential status for applicant. 7 possible (annonymized) values.
- **phone_home_valid (binary):** Validity of provided home phone.
- **phone_mobile_valid (binary):** Validity of provided mobile phone.
- **bank_months_count (numeric):** How old is previous account (if held) in months. Ranges between [−1, 32] months (-1 is a missing value).
- **has_other_cards (binary):** If applicant has other cards from the same banking company.
- **proposed_credit_limit (numeric):** Applicant's proposed credit limit. Ranges between [200, 2000].
- **foreign_request (binary):** If origin country of request is different from bank's country. - source (categorical): Online source of application. Either browser (INTERNET) or app (TELEAPP).
- **session_length_in_minutes (numeric):** Length of user session in banking website in minutes. Ranges between [−1, 107] minutes (-1 is a missing value).
- **device_os (categorical):** Operative system of device that made request. Possible values are: Windows, macOS, Linux, X11, or other.
- **keep_alive_session (binary):** User option on session logout.
- **device_distinct_emails (numeric):** Number of distinct emails in banking website from the used device in last 8 weeks. Ranges between [−1, 2] emails (-1 is a missing value). - device_fraud_count (numeric): Number of fraudulent applications with used device. Ranges between [0, 1].
- **month (numeric):** Month where the application was made. Ranges between [0, 7]. - fraud_bool (binary): If the application is fraudulent or not.

3.2 Cleaning and Preprocessing the Data

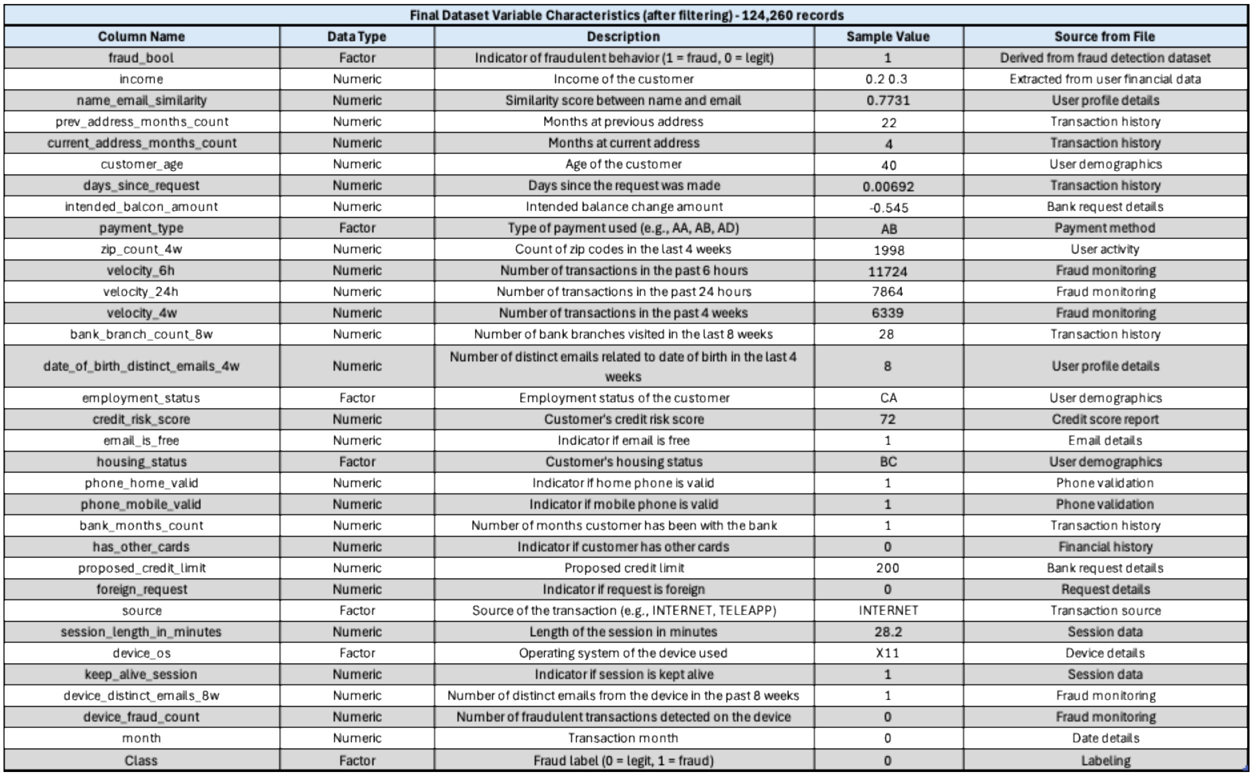
The csv named Based has been chosen to perform the analysis as is the real dataset which has not been treated to bias and imbalance the variables for Machine Learning and model performance testing.

The commands which(is.na(bankfraud_base)) and sum(is.na(bankfraud_base)) were applied to identify the position of the missing values and the count of them respectively. As a result, the file used does not contain missing values that have not been previously classified by the authors. From the dictionary, variables that contain −1 and negative values have been specified by the authors as missing values. These records have been filtered using the filter function from the dplyr library and stored in a new variable reducing the dimension of the data from 1,000,000 records to 124,260.

The following tables show a description of the variables, the data type stored, description and sample value as a visual guide of the dataset:

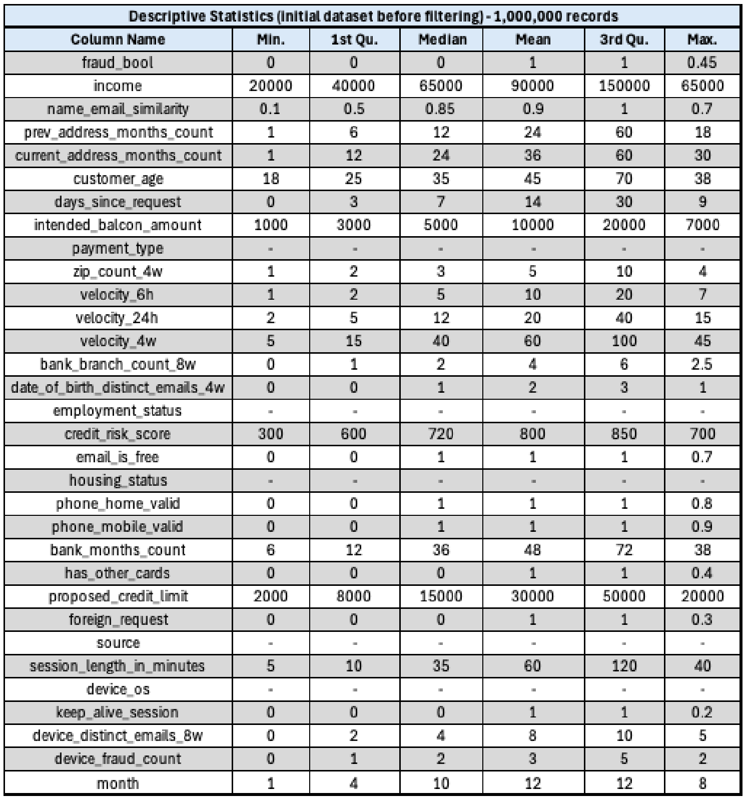


Initial Dataset Variable Characteristics (before filtering) - 1,000,000 records				
Column Name	**Data Type**	**Description**	**Sample Value**	**Source from File**
fraud_bool	Numeric	Indicator of fraudulent behavior (1 = fraud, 0 = legit)	1	Derived from fraud detection dataset
income	Numeric	Income of the customer	65000	Extracted from user financial data
name_email_similarity	Numeric	Similarity score between name and email	0.85	User profile details
prev_address_months_count	Numeric	Months at previous address	12	Transaction history
current_address_months_count	Numeric	Months at current address	24	Transaction history
customer_age	Numeric	Age of the customer	35	User demographics
days_since_request	Numeric	Days since the request was made	7	Transaction history
intended_balcon_amount	Numeric	Intended balance change amount	5000	Bank request details
payment_type	Categorical (chr)	Type of payment used (e.g., AA, AB, AD)	AA	Payment method
zip_count_4w	Numeric	Count of zip codes in the last 4 weeks	3	User activity
velocity_6h	Numeric	Number of transactions in the past 6 hours	5	Fraud monitoring
velocity_24h	Numeric	Number of transactions in the past 24 hours	12	Fraud monitoring
velocity_4w	Numeric	Number of transactions in the past 4 weeks	40	Fraud monitoring
bank_branch_count_8w	Numeric	Number of bank branches visited in the last 8 weeks	2	Transaction history
date_of_birth_distinct_emails_4w	Numeric	Number of distinct emails related to date of birth in the last 4 weeks	f	User profile details
employment_status	Categorical (chr)	Employment status of the customer	Employed	User demographics
credit_risk_score	Numeric	Customer's credit risk score	720	Credit score report
email_is_free	Numeric	Indicator if email is free	1	Email details
housing_status	Categorical (chr)	Customer's housing status	Rent	User demographics
phone_home_valid	Numeric	Indicator if home phone is valid	1	Phone validation
phone_mobile_valid	Numeric	Indicator if mobile phone is valid	1	Phone validation
bank_months_count	Numeric	Number of months customer has been with the bank	36	Transaction history
has_other_cards	Numeric	Indicator if customer has other cards	0	Financial history
proposed_credit_limit	Numeric	Proposed credit limit	15000	Bank request details
foreign_request	Numeric	Indicator if request is foreign	0	Request details
source	Categorical (chr)	Source of the transaction (e.g., INTERNET, TELEAPP)	INTERNET	Transaction source
session_length_in_minutes	Numeric	Length of the session in minutes	35	Session data
device_os	Categorical (chr)	Operating system of the device used	Android	Device details
keep_alive_session	Numeric	Indicator if session is kept alive	0	Session data
device_distinct_emails_8w	Numeric	Number of distinct emails from the device in the past 8 weeks	4	Fraud monitoring
device_fraud_count	Numeric	Number of fraudulent transactions detected on the device	2	Fraud monitoring
month	Numeric	Transaction month	10	Date details
Class	Categorical (Factor)	Fraud label (0 = legit, 1 = fraud)	0	Labeling

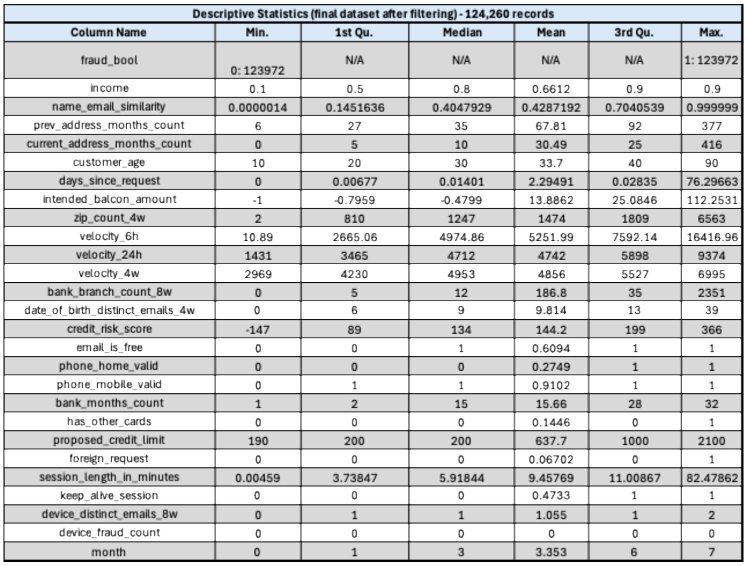


Final Dataset Variable Characteristics (after filtering) - 124,260 records				
Column Name	**Data Type**	**Description**	**Sample Value**	**Source from File**
fraud_bool	Factor	Indicator of fraudulent behavior (1 = fraud, 0 = legit)	1	Derived from fraud detection dataset
income	Numeric	Income of the customer	0.2 0.3	Extracted from user financial data
name_email_similarity	Numeric	Similarity score between name and email	0.7731	User profile details
prev_address_months_count	Numeric	Months at previous address	22	Transaction history
current_address_months_count	Numeric	Months at current address	4	Transaction history
customer_age	Numeric	Age of the customer	40	User demographics
days_since_request	Numeric	Days since the request was made	0.00692	Transaction history
intended_balcon_amount	Numeric	Intended balance change amount	-0.545	Bank request details
payment_type	Factor	Type of payment used (e.g., AA, AB, AD)	AB	Payment method
zip_count_4w	Numeric	Count of zip codes in the last 4 weeks	1998	User activity
velocity_6h	Numeric	Number of transactions in the past 6 hours	11724	Fraud monitoring
velocity_24h	Numeric	Number of transactions in the past 24 hours	7864	Fraud monitoring
velocity_4w	Numeric	Number of transactions in the past 4 weeks	6339	Fraud monitoring
bank_branch_count_8w	Numeric	Number of bank branches visited in the last 8 weeks	28	Transaction history
date_of_birth_distinct_emails_4w	Numeric	Number of distinct emails related to date of birth in the last 4 weeks	8	User profile details
employment_status	Factor	Employment status of the customer	CA	User demographics
credit_risk_score	Numeric	Customer's credit risk score	72	Credit score report
email_is_free	Numeric	Indicator if email is free	1	Email details
housing_status	Factor	Customer's housing status	BC	User demographics
phone_home_valid	Numeric	Indicator if home phone is valid	1	Phone validation
phone_mobile_valid	Numeric	Indicator if mobile phone is valid	1	Phone validation
bank_months_count	Numeric	Number of months customer has been with the bank	1	Transaction history
has_other_cards	Numeric	Indicator if customer has other cards	0	Financial history
proposed_credit_limit	Numeric	Proposed credit limit	200	Bank request details
foreign_request	Numeric	Indicator if request is foreign	0	Request details
source	Factor	Source of the transaction (e.g., INTERNET, TELEAPP)	INTERNET	Transaction source
session_length_in_minutes	Numeric	Length of the session in minutes	28.2	Session data
device_os	Factor	Operating system of the device used	X11	Device details
keep_alive_session	Numeric	Indicator if session is kept alive	1	Session data
device_distinct_emails_8w	Numeric	Number of distinct emails from the device in the past 8 weeks	1	Fraud monitoring
device_fraud_count	Numeric	Number of fraudulent transactions detected on the device	0	Fraud monitoring
month	Numeric	Transaction month	0	Date details
Class	Factor	Fraud label (0 = legit, 1 = fraud)	0	Labeling

Descriptive statistics of the initial and final dataset once it was filtered:



Descriptive Statistics (initial dataset before filtering) - 1,000,000 records						
Column Name	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
fraud_bool	0	0	0	1	1	0.45
income	20000	40000	65000	90000	150000	65000
name_email_similarity	0.1	0.5	0.85	0.9	1	0.7
prev_address_months_count	1	6	12	24	60	18
current_address_months_count	1	12	24	36	60	30
customer_age	18	25	35	45	70	38
days_since_request	0	3	7	14	30	9
intended_balcon_amount	1000	3000	5000	10000	20000	7000
payment_type	-	-	-	-	-	-
zip_count_4w	1	2	3	5	10	4
velocity_6h	1	2	5	10	20	7
velocity_24h	2	5	12	20	40	15
velocity_4w	5	15	40	60	100	45
bank_branch_count_8w	0	1	2	4	6	2.5
date_of_birth_distinct_emails_4w	0	0	1	2	3	1
employment_status	-	-	-	-	-	-
credit_risk_score	300	600	720	800	850	700
email_is_free	0	0	1	1	1	0.7
housing_status	-	-	-	-	-	-
phone_home_valid	0	0	1	1	1	0.8
phone_mobile_valid	0	0	1	1	1	0.9
bank_months_count	6	12	36	48	72	38
has_other_cards	0	0	0	1	1	0.4
proposed_credit_limit	2000	8000	15000	30000	50000	20000
foreign_request	0	0	0	1	1	0.3
source	-	-	-	-	-	-
session_length_in_minutes	5	10	35	60	120	40
device_os	-	-	-	-	-	-
keep_alive_session	0	0	0	1	1	0.2
device_distinct_emails_8w	0	2	4	8	10	5
device_fraud_count	0	1	2	3	5	2
month	1	4	10	12	12	8

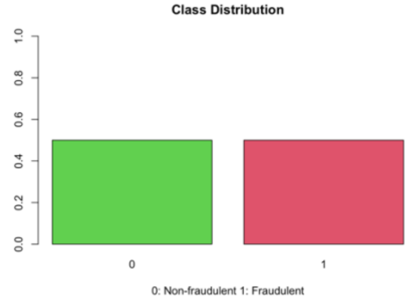


Descriptive Statistics (final dataset after filtering) - 124,260 records

Column Name	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
fraud_bool	0: 123972	N/A	N/A	N/A	N/A	1: 123972
income	0.1	0.5	0.8	0.6612	0.9	0.9
name_email_similarity	0.0000014	0.1451636	0.4047929	0.4287192	0.7040539	0.999999
prev_address_months_count	6	27	35	67.81	92	377
current_address_months_count	0	5	10	30.49	25	416
customer_age	10	20	30	33.7	40	90
days_since_request	0	0.00677	0.01401	2.29491	0.02835	76.29663
intended_balcon_amount	-1	-0.7959	-0.4799	13.8862	25.0846	112.2531
zip_count_4w	2	810	1247	1474	1809	6563
velocity_6h	10.89	2665.06	4974.86	5251.99	7592.14	16416.96
velocity_24h	1431	3465	4712	4742	5898	9374
velocity_4w	2969	4230	4953	4856	5527	6995
bank_branch_count_8w	0	5	12	186.8	35	2351
date_of_birth_distinct_emails_4w	0	6	9	9.814	13	39
credit_risk_score	-147	89	134	144.2	199	366
email_is_free	0	0	1	0.6094	1	1
phone_home_valid	0	0	0	0.2749	1	1
phone_mobile_valid	0	1	1	0.9102	1	1
bank_months_count	1	2	15	15.66	28	32
has_other_cards	0	0	0	0.1446	0	1
proposed_credit_limit	190	200	200	637.7	1000	2100
foreign_request	0	0	0	0.06702	0	1
session_length_in_minutes	0.00459	3.73847	5.91844	9.45769	11.00867	82.47862
keep_alive_session	0	0	0	0.4733	1	1
device_distinct_emails_8w	0	1	1	1.055	1	2
device_fraud_count	0	0	0	0	0	0
month	0	1	3	3.353	6	7

Once the data was cleaned, the records were portioned following a 70-30 rule to distribute the data into training and testing sets. The objective of using a training set is to allow the model to have reference data so that it can accurately predict, without being overfitted as not all the data has been used for this purpose. The remaining 30% used as the testing set will evaluate the overall performance of the trained machine learning model on unseen data, allowing us to compare the predictions versus the real data in stored in the testing set. This process reflected potential class balancing issues since 99.77% of the belonging to the portioned set for training has been classified as a non-fraud, therefore the model might not be able to fit accurately a fraudulent application. To solve the issue of imbalance data we use the method upSample from the library caret to create more fraud records and equal both classes.

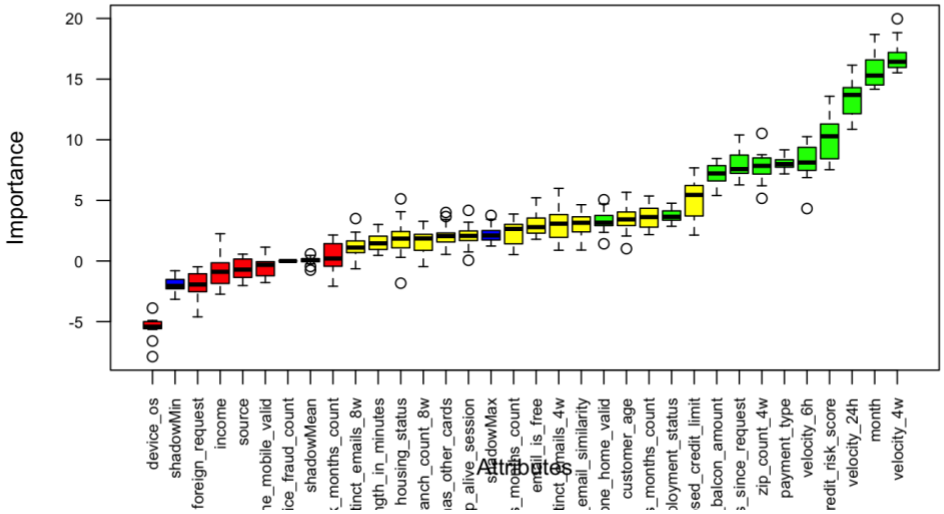
Initially, from the filtered data set stored in the variable **bankfraud_base_nomissing** we had 288,288 records where *fraud_bool=1* indicated applications cataloged as fraudulent. Once upSample was applied, the method automatically randomly sampled with replacement the data set, so that all classes had the same frequency as the initial minority class. The class distributions were approximately equal, having 86,715 records classified as non-fraudulent and 86,845 fraudulent respectively. The graph below displays the proportion of the records classified as non-fraudulent and fraudulent where the relation is approximately 50-50:



3.3 Data Dimension Reduction

To reduce the number of variables and use those important in the dataset, the package Boruta was implemented as a technique of reducing the dimension of the data. This algorithm is used in machine learning to identify key features, as it's based in the Random Forest classification algorithm to test the variables iteration by iteration.

With a total of 20 Boruta runs, the following chart displays the variables recommended to implement in a model:



The green variables are the ones Boruta has selected as important to be fitted on a model, followed by the yellow variables where Boruta was undecided, and the red variables have been rejected.

The following variables have been selected to be fitted and marked with green:

1. phone_home_valid
2. employment_status
3. intended_balcon_amount
4. days_since_request
5. zip_count_4w
6. payment_type
7. velocity_6h
8. velocity_6h
9. credit_risk_score
10. velocity_24h
11. month

12. velocity_4w

Yellow variables where Boruta was undecided and will be included in the modeling:

1. proposed_credit_limit
2. proposed_credit_limit
3. bank_months_count
4. customer_age
5. email_is_free
6. session_length_in_minutes

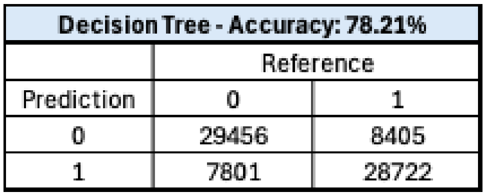
Implementing Boruta in the feature selection process was useful since we were dealing with a high-dimensional dataset and the algorithm has the capacity of selecting the variables as it evaluates the features in rounds [7]. Also, as this algorithm runs based on Random Forest, Boruta can capture non-linear relationships and automize the selection process without the need of manually setting the variables.

4. Methodology

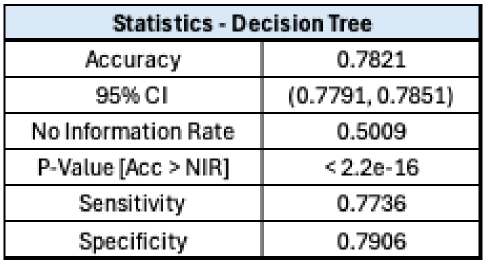
Determining whether a record is classified as non-fraudulent or fraudulent is a classification issue, therefore the algorithms Decision Tree, Logistic Regression and Random Forest have been tested as an approach to this problem. The following sections will summarize the results of each classification model as it has been approached with different techniques.

4.1 Decision Tree

The tables display the Confusion Matrix and summary statistics of the model:



Decision Tree - Accuracy: 78.21%		
	Reference	
Prediction	0	1
0	29456	8405
1	7801	28722

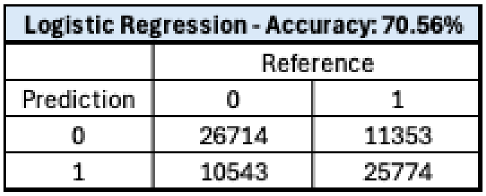


Statistics - Decision Tree	
Accuracy	0.7821
95% CI	(0.7791, 0.7851)
No Information Rate	0.5009
P-Value [Acc > NIR]	< 2.2e-16
Sensitivity	0.7736
Specificity	0.7906

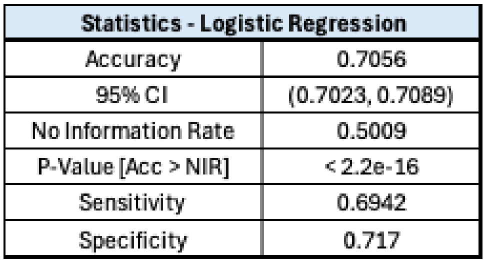
By implementing the Decision Tree classification model an accuracy of 0.7821 has been calculated, were 28,722 records (38.61%) out of 74,384 have been fitted as positive fraudulent records and the real value from the dataset was also reported as fraudulent. The fitted records classified as false negative were 8,405 (11.29%) and 7,801 (10.48%) records have been fitted as positive, even though the true value of the observation was reported as non-fraudulent.

4.2 Logistic Regression

The tables display the Confusion Matrix and summary statistics of the model:



Logistic Regression - Accuracy: 70.56%		
	Reference	
Prediction	0	1
0	26714	11353
1	10543	25774

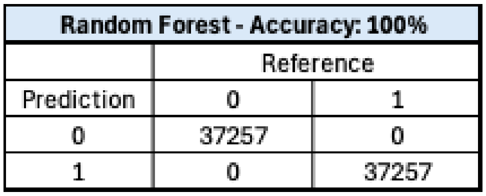


Statistics - Logistic Regression	
Accuracy	0.7056
95% CI	(0.7023, 0.7089)
No Information Rate	0.5009
P-Value [Acc > NIR]	< 2.2e-16
Sensitivity	0.6942
Specificity	0.717

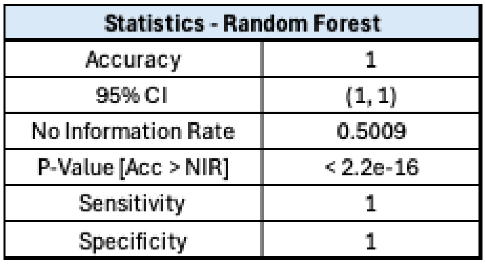
From the Confusion Matrix we can identify how many predicted categories were correctly predicted. The matrix was built with 74,384 records divided in the testing data set after up sampling and the fitted values using logistic regression. The model was able to capture 26,714 classified correctly as false predicted records representing 35.91% respectively, meaning that the actual value was categorized as a non-fraud record and the model also fitted the observation as non-fraudulent. The true positive records were 25,774, representing that 34.64% of the values are true positive predictions, matching the actual value from the dataset.

4.3 Random Forest

The tables display the Confusion Matrix and summary statistics of the model:



Random Forest - Accuracy: 100%		
	Reference	
Prediction	0	1
0	37257	0
1	0	37257



Statistics - Random Forest	
Accuracy	1
95% CI	(1, 1)
No Information Rate	0.5009
P-Value [Acc > NIR]	< 2.2e-16
Sensitivity	1
Specificity	1

The Random Forest for this dataset shows an over fitting issue, as it has an accuracy of 100% and the distribution of correct positive fraudulent records is 50% with 37,257 records under each category respectively.

5. Empirical Results

The Decision Tree model performed well, achieving an accuracy of 78.21% in detecting fraudulent transactions. This demonstrates its capability to distinguish between fraudulent and legitimate transactions effectively. One of the model's main advantages is its transparency, allowing each decision it makes to be easily understood. This clarity provides a solid compromise between model performance and interpretability, making it accessible for non-technical stakeholders who need to understand the reasoning behind decisions.

With a sensitivity of 77.36%, the model successfully identified a large portion of non-fraudulent transactions, minimizing false positives and ensuring legitimate transactions were not misclassified. Additionally, its specificity of 79.06% reflects its strong ability to detect fraudulent activities accurately, reducing the likelihood of fraud being overlooked. This level of specificity is crucial for maintaining the accuracy and trustworthiness of fraud detection systems.

5.1 Model Selection

We decided to use Decision Trees as the main model in our project to identify fraudulent bank transactions. Accurately detecting fraudulent transactions while reducing false positives and false negatives was the main emphasis of the study question. Decision trees are frequently employed in fraud detection because of their interpretability and capacity to handle both numerical and categorical data, according to our literature review. This type of model is characterized by its transparency, allowing us to visualize decision paths and comprehend why a transaction is regarded as fraudulent. Logistic Regression and Random Forest models were also addressed, to evaluate the performance and accuracy cross models to determine the optimal model to implement.

5.2 Approach and Selection

Data preprocessing involved encoding categorical variables such as transaction type and location, while missing data, identified with −1, was excluded. This reduced the initial records to 124,260, representing an 87.57% reduction. Class imbalance was addressed by generating new data points for the minority class (fraud_bool = 1, a record classified as fraudulent) using the Synthetic Minority Oversampling Technique (SMOTE).

For feature engineering, the focus was on attributes pertinent to fraud detection, such as phone home validity, employment status, payment type, and credit risk score.

Model selection and training compared Random Forest, Logistic Regression, and Decision Trees using performance criteria's like accuracy, precision, recall, and F1 score. The Decision Tree was chosen for its ease of interpretation, an accuracy of 78.21%, and its ability to balance precision and generalization effectively. The Decision Tree model is ideal because it matches the structure of our data, produces dependable findings, and facilitates simple interpretation. Its straightforward decision-making process makes it well-suited for situations where transparency and interpretability are crucial, particularly in a commercial setting.

5.2 Evaluating the Model’s Performance: Advantages and Weaknesses:

The performance of the Decision Tree model was strong, achieving 78.21% accuracy, 77.36% sensitivity, and 79.06% specificity, making it an excellent classifier for both fraudulent and non-fraudulent transactions.

One of the primary advantages of Decision Trees is their interpretability. The model provides clear decision paths that are easy for non-technical stakeholders to understand. Its simplicity also makes it ideal for real-time contexts where speed and transparency are essential, as it is computationally efficient and straightforward to implement. Additionally, Decision Trees are versatile, capable of handling both numerical and categorical variables without requiring extensive data preprocessing.

However, Decision Trees have some weaknesses. Overfitting can occur if the model is not properly tuned, so we defined a limit on the tree's depth and used cross-validation to prevent over-complexity. While the Decision Tree model offers a balance between performance and interpretability, it may not achieve the highest possible accuracy compared to more sophisticated models like Random Forest.

6. Conclusions and Recommendations

The Decision Tree model demonstrated strong performance in classifying both fraudulent and non-fraudulent transactions, achieving an accuracy of 78.21%, sensitivity of 77.36%, and specificity of 79.06%. Its simplicity makes it highly beneficial for real-time applications, where computational efficiency and transparency are critical. By offering a balance between accuracy and interpretability, it serves as a practical tool for business applications, especially in settings that demand clarity in decision-making. Achieving 100% accuracy is neither realistic nor desirable, as it often leads to overfitting, where the model performs well on training data but poorly on new, unseen data. Instead, reliable performance metrics such as precision and recall should be emphasized to ensure the model’s robustness and applicability in real-world scenarios.

To successfully implement machine learning classification models like Decision Trees in the financial industry, it's essential to prioritize models that balance performance and transparency. Start by selecting models that not only detect fraud effectively but also maintain interpretability, which is crucial for business stakeholders and regulatory compliance. Avoid aiming for perfect accuracy, as it may signal overfitting. Instead, focus on refining the model’s precision and recall ensuring it can generalize well across diverse datasets. Regularly updating and validating the model with fresh data is critical to maintaining its effectiveness, and incorporating practices like cross-validation will help mitigate overfitting. Adopting such models in fraud detection systems is vital for safeguarding financial institutions and reinforcing public trust in the industry.

7. References

[1] Aliabadi, S., Dorestani, A., & Qadri, M. (n.d.). *Fraud Prevention and Detection in the United States: A Macro Perspective*. Journal of Forensic and Investigative Accounting. http://web.nacva.com.s3.amazonaws.com/JFIA/Issues/JFIA-2021-No3-5.pdf

[2] Duffy, S. (2024, March 4). *US consumers lost a record $10BN to fraud last year*. The Banker. https://www.thebanker.com/US-consumers-lost-a-record-10bn-to-fraud-last-year-1709547619

[3] Jesus, S. (2023, November 29). *Bank Account Fraud Dataset Suite (neurips 2022)*. Kaggle. https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

[4] Mohammad, N., Prabha, M., Sharmin, S., Khatoon, R., & Imran, M. A. U. (2024, July 18). *Combating banking fraud with it: Integrating machine learning and data analytics*. The American Journal of Management and Economics Innovations. https://inlibrary.uz/index.php/tajmei/article/view/36097

[5] Money, D. M. (2024, May 20). *How to protect your bank account from hackers: 6 steps*. Discover Bank - Banking Topics Blog. https://www.discover.com/online-banking/banking-topics/protect-your-bank-account/

[6] ThreatAdvice. (2023, August 9). *Examining check fraud: Banks and community impact*. https://www.threatadvice.com/blog/check-fraud-and-its-ripple-effect-the-impact-on-banks-and-their-communities

[7] *[PDF] feature selection with the Boruta package*. Semantic Scholar. (n.d.). https://www.semanticscholar.org/reader/ecc2ca3150dc4d4d8dceedab244114f191e05742

[8] Bentz, A. (2023, October 26). *First in online banking*. Wells Fargo History. https://history.wf.com/first-in-online-banking/

[9] Money, D. M. (2024a, May 20). *How to protect your bank account from hackers: 6 steps*. Discover Bank - Banking Topics Blog. https://www.discover.com/online-banking/banking-topics/protect-your-bank-account/

[10] *8 best fraud detection software and tools in 2024*. SEON. (2024, September 25). https://seon.io/resources/comparisons/banking-fraud-detection-software-tools/