Ethically Responsible Fraud Detection

Mid-Semester Progress Report

Dylan Steimel

DSA 5900
Summer 2023
June 23, 2023
Faculty Supervisor: Dr. Trafalis

1. Introduction

As our society continues to embrace digital advancements, the risk of fraudulent activities also escalates. This has become a paramount concern for the banking industry, given the high stakes involved, as numerous individuals rely on these institutions to safeguard their life savings.

In addition to ensuring robust security measures, maintaining access to financial resources is equally crucial. If every transaction or credit application were indiscriminately flagged as fraudulent and subsequently blocked, it would undermine the trust and usability of banking services. Therefore, striking a balance that ensures both adequate security and accessibility is imperative.

Another facet that contributes to the complexity of this problem is the ethical dimension involved in model development. While it is possible to construct a model that maximizes AUROC (Area Under the Receiver Operating Characteristic curve), it may inadvertently lead to biased outcomes. For example, disproportionately denying individuals over the age of 50 access to their own funds. This discriminatory effect is inherently unjust and necessitates careful consideration before deploying any such model.

The primary objective of my project is to construct a model that comprehensively addresses these considerations. To achieve this, I will utilize the Bank Account Fraud Dataset Suite (NeurIDP 2022), which was developed by Feedzai, for the training and evaluation of the model based on the aforementioned criteria. This dataset has been specifically designed to emulate real-world credit applications and banking data. Notably, it encompasses additional attributes such as age, income, and employment status, which I will leverage to ensure fairness of the model.

2. Objectives

In addition, Feedzai offers baseline models along with their corresponding Receiver Operating Characteristic (ROC) curves. Among these models, the Logistic Regression model stands out as the most effective, achieving an AUROC of 0.877. Consequently, I will adopt this AUROC value as the benchmark to surpass while developing a robust classifier.

To mitigate the risk of discrimination within my model, I will observe the false positive rate (FPR) ratio between different groups. The false positive rate represents the rate at which legitimate transactions are falsely identified as fraudulent. To accept a model as fair I will test the FPR of sensitive groups for statistical parity.

In addition to the project's broader goals, I have several personal objectives that I look forward to accomplishing. Firstly, I aim to gain a deeper understanding of techniques specifically tailored for handling highly unbalanced datasets, considering that only approximately 1% of the data is labeled as fraud.

Moreover, I recognize the significance of ethical considerations and aspire for this project to enhance my awareness of such issues. By actively engaging with ethical questions throughout the project, I aim to develop a heightened sensitivity that will carry over into my future endeavors. This project serves as a valuable opportunity to ensure that ethical concerns remain at the forefront of my thinking, even in situations where they might not have been previously contemplated.

3. Data

The dataset utilized in this study, as documented by Feedzai (Jesus et al., 2022), comprises a comprehensive collection of 32 distinct features. These features encompass a range of data types, including categorical variables such as housing status, payment type, and transaction source, alongside numerical variables such as account velocity, session length, and credit risk score. The dataset encompasses a substantial sample size of 1,000,000 individual transactions, providing a rich and extensive foundation for analysis and model development.

3.1. Ingestion

The data ingestion process was straightforward for this project. As the purpose of the dataset provided by Feedzai was to promote accessibility to typically classified data, acquiring it involved a simple download and importation into my Jupyter notebook. However, it's worth noting that the dataset is quite large, and managing its memory became a challenge. To address this issue, I simply compressed the dataset into a zip file.

3.2. Exploration

The dataset utilized in this project comprises 30 features, in addition to the target column. These features can be categorized into different sets.

Firstly, there are six categorical features: "payment type," "employment status," "housing status," "application source," "month," and "device operating system." Figure 1 illustrates the distribution of these categories within the dataset.

Secondly, there are six boolean features: "email is free," "valid home phone," "valid mobile phone," "has other cards," "foreign request," and "keep alive session." Figure 2 showcases the counts of "True" and "False" within each boolean category. [INSERT PIC HERE]

The dataset also includes nine features representing count data. These features include "months at previous address," "months at current address," "customer age," "number of

applications in zip code over 4 weeks," "bank branch count 8 weeks," "date of birth distinct emails," "credit risk score," "bank months count," and "device distinct emails." Additionally, there are nine continuous value features: "income," "name email similarity," "days since request," "intended balcony amount," "velocity 6 hours," "velocity 24 hours," "velocity 4 weeks," "proposed credit limit," and "session length in minutes." To compare the distributions of fraud versus non-fraud cases, kernel density plots were generated for these continuous value features. [INSERT IMAGE HERE]

3.3. Preparation

The initial step in data preparation involved addressing missing values. The "device fraud count" column, which was not included in the previous description, was entirely composed of 0 values and thus provided no valuable information. Consequently, this column was removed from the dataset.

Three columns, namely "current address month count," "session length in minutes," and "distinct device emails 8 weeks," contained placeholder values of -1 to represent missing data. To determine whether these missing values occurred randomly or non-randomly, a chi-squared contingency test was conducted for each column. The test compared the distribution of missing values between instances labeled as fraud and those labeled as legitimate. A significance level (alpha) of 0.01 was chosen for the test.

Results indicated that two columns, "session length in minutes" and "device distinct emails," exhibited missing values randomly and had relatively few instances with missing data compared to the overall dataset size. Consequently, instances with missing values in these columns were removed from further analysis.

However, the third column showed non-random missingness, with a p-value of 0.004. To address this, the missing values in this column were imputed using scikit-learn's IterativeImputer class, which leverages an iterative approach to estimate missing values based on the observed data.

Following the handling of missing values, the next step was to scale and encode the data. One-hot encoding was applied to the categorical features, while the numerical features underwent min-max scaling. Additionally, a log transform was performed on columns exhibiting significant skewness to mitigate the impact of outliers and improve model performance.

For the baseline model, no feature engineering was performed. However, during the testing of different fairness methods, some features were removed. Further details on these methods and their impact on feature selection will be discussed in the "Methodology" section of this paper.

4. Methodology

Due to the large size of the dataset, I was able to split it into training, validation, and test sets without concern about insufficient samples. I used a 64/16/20% split for the train/validation/test sets.

The artificial neural network (ANN) I developed as a binary classifier currently consists of a single hidden layer with a size of 10 and utilizes the ReLU activation function. However, these architecture choices are subject to change and will be fine-tuned along with other hyperparameters using the grid search technique.

Initially, my approach involved extensive feature engineering to enhance fairness within the model. I experimented with various methods, such as removing features associated with sensitive groups, performing dimension reduction on highly correlated features, and eliminating highly correlated features. Although these techniques did improve fairness to some extent, they did not reach the desired level and led to significant data loss, compromising the model's performance.

To overcome these challenges, I adopted adversarial learning, a method described in the literature (Zhang et al., 2018) for promoting fairness in machine learning models. Adversarial learning involves training two separate models: the deployment model and the adversarial model. The deployment model is trained using standard techniques for the desired task, such as employment prediction. The adversarial model is trained on the predictions made by the deployment model, focusing on the sensitive groups that should be treated fairly.

After the initial training of both models, I adjusted the loss function of the deployment model to incorporate the loss of the adversarial model. Specifically, I used binary cross-entropy (BCE) as the loss function for the deployment model and categorical cross-entropy (CCE) for the adversarial model. Therefore, the updated deployment loss function can be expressed as follows:

$$loss = BCE(deployment) - CCE(adversarial)$$

To train both models effectively, I set up a training loop with the following steps: 1. Train the adversarial model using the deployment model's predictions while keeping the deployment model's weights constant. 2. Train the deployment model with its new loss function, taking into account the adversarial model's predictions and keeping the adversarial model's weights constant.

By minimizing this loss function during optimization, the model is encouraged to increase the loss of the adversarial model, making it more challenging to identify which sensitive group the deployment model is making predictions for.

This approach results in a model that provides fairer predictions across different classes without sacrificing significant amounts of data due to feature engineering.

To validate my findings, I will utilize the test set that was set aside before any modeling was performed. I will compare the performance and fairness of my model before and after the adversarial training was completed.

5. Process Validation

Having had two Zoom meetings with Dr. Trafalis, I received valuable guidance and feedback on my approach. Dr. Trafalis emphasized the effectiveness of artificial neural networks in generalizing complex tasks, and he was particularly intrigued by the concept of adversarial debiasing. After discussing the theory behind it and sharing my preliminary results, he found it to be a fascinating approach and encouraged me to continue using it as my primary method for creating a fair model. Additionally, Dr. Trafalis generously provided me with numerous resources on neural networks (Abadi et al., 2015), imbalanced training (He & Garcia, 2009), and fairness evaluations (Barocas et al., 2019), which I plan to leverage in my work.

Based on the resources provided, I have identified several techniques that I will implement. For example, I will set an initial output bias on the deployment model. This bias will help the model by precluding the need for it to spend the initial epochs learning that the data is imbalanced. This technique can potentially speed up the training process.

Furthermore, I have decided to modify my evaluation criteria for assessing fairness in the model. Initially, I had planned to rely on the metrics provided by Feedzai, which primarily focused on the maximum difference between the false positive rates (FPR) of sensitive groups. However, I now understand that a more widely accepted measure of fairness is statistical parity across groups. Therefore, I will incorporate statistical parity as my fairness evaluation metric, aiming to achieve equitable outcomes across different sensitive groups.

By incorporating these techniques and adjusting my fairness evaluation, I aim to further enhance the fairness and performance of my model, as well as align with industry standards for evaluating fairness.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Barocas, S., Hardt, M. & Narayanan, A. (2019). *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org. `http://www.fairmlbook.org`.

He, H. & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21(9)*, 1263–1284.

Jesus, S., Pombal, J., Alves, D., Cruz, A., Saleiro, P., Ribeiro, R. P., Gama, J. & Bizarro, P. (2022). Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation.

Zhang, B. H., Lemoine, B. & Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning.

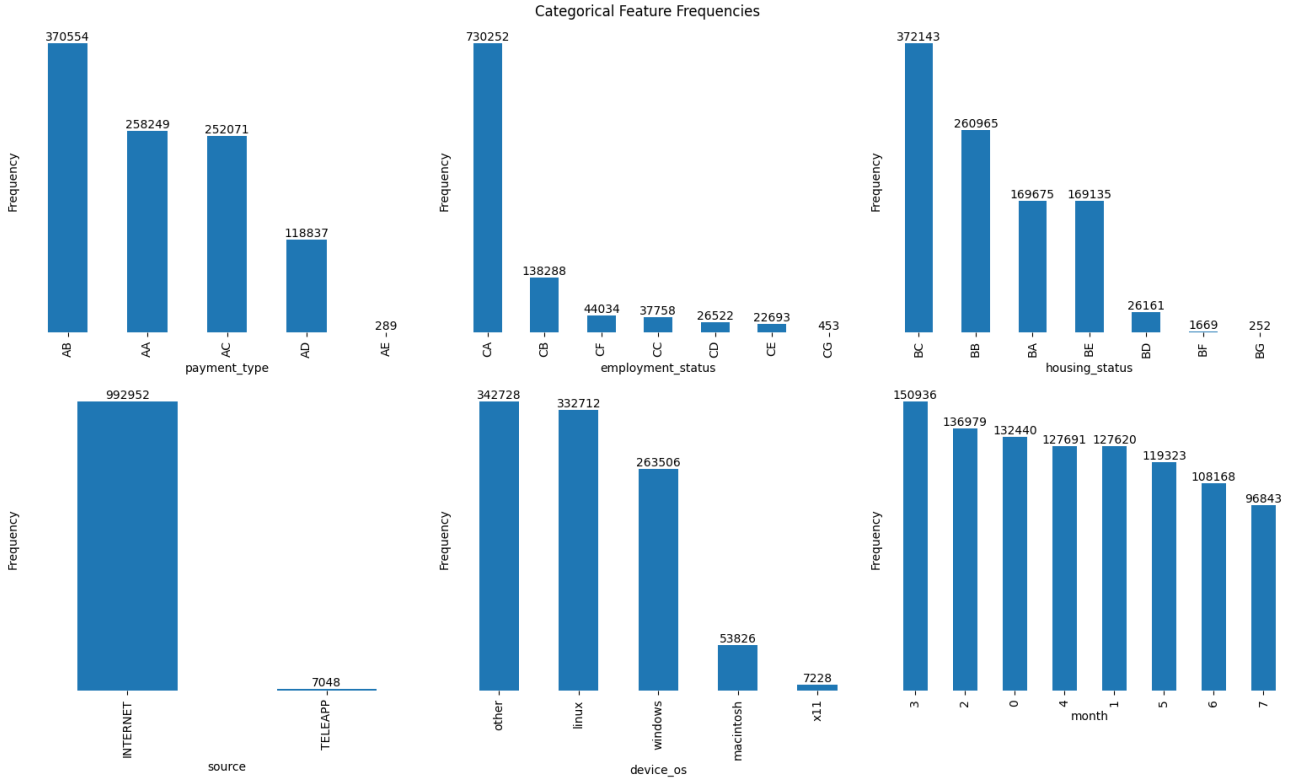

Figure 1. Categorical Feature Frequencies

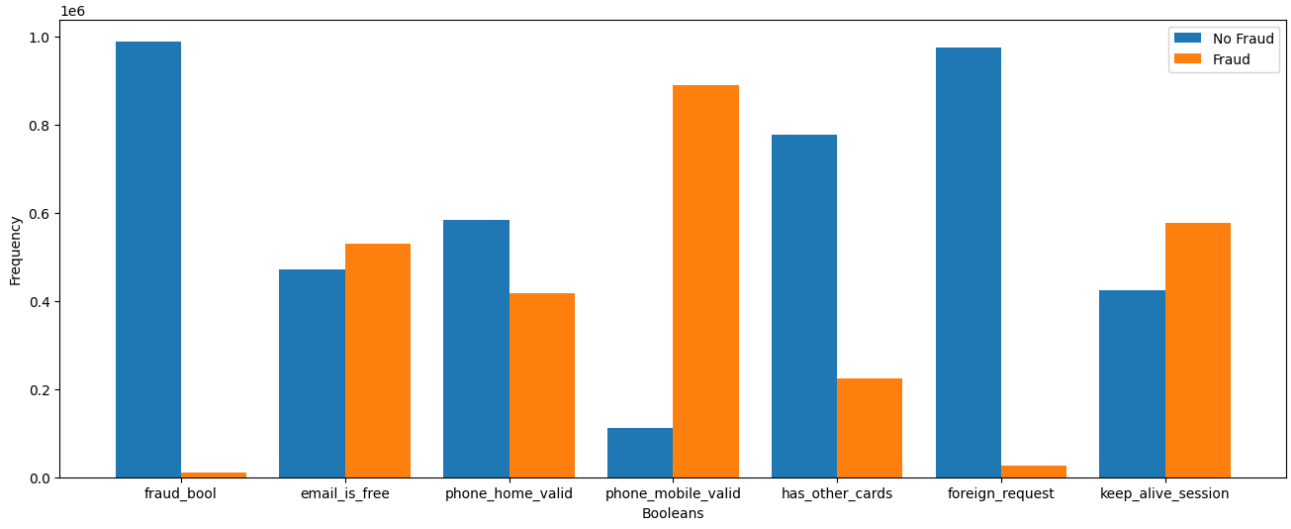

Figure 2. Boolean Feature Frequencies

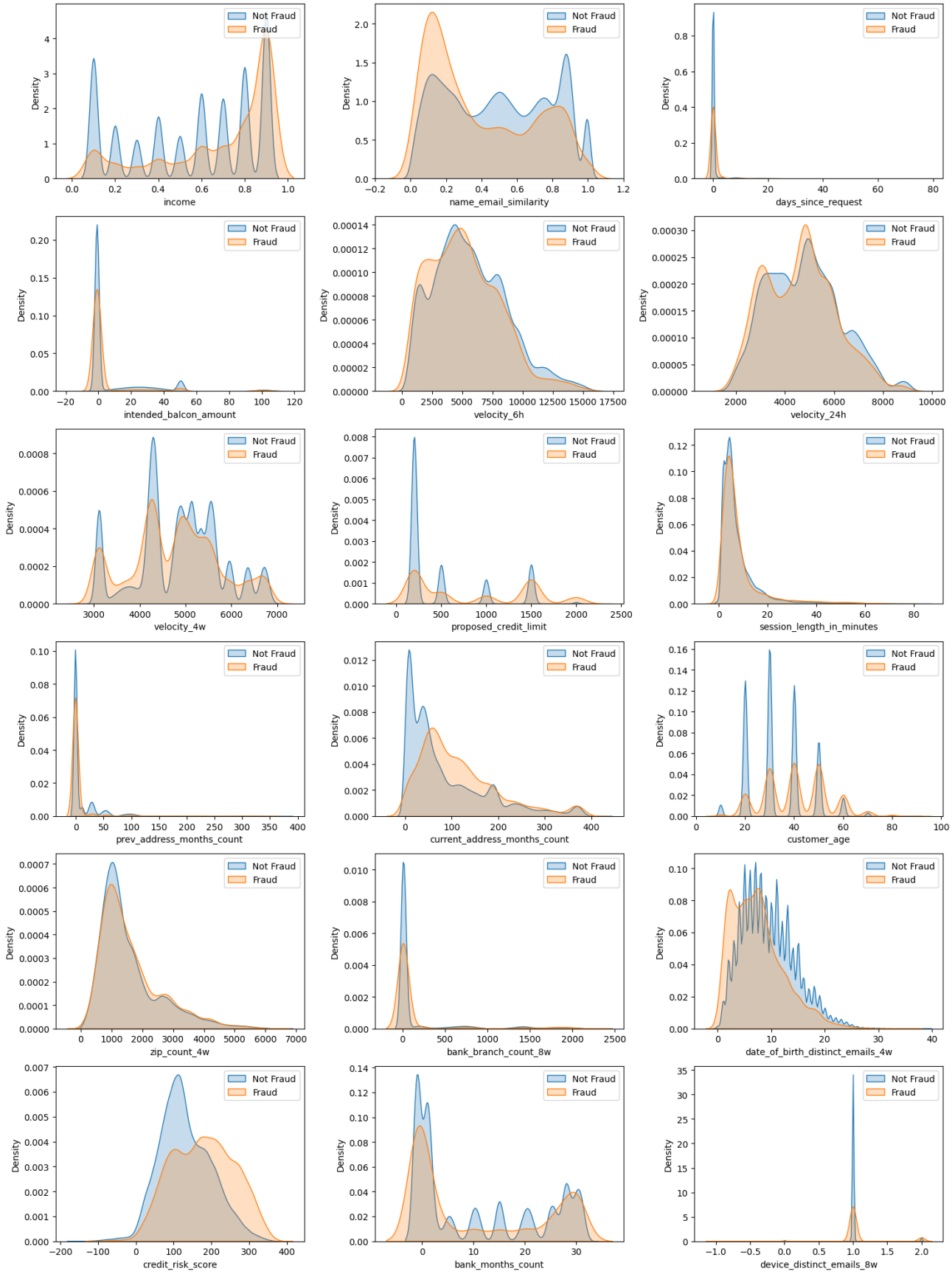

Figure 3. Numerical Feature Distributions