Bank Account Fraud

CSCI.420 - Principles of Data Mining - Group Project Report

Aidan Mellin, Gunnar Bachmann, TJ Praschunus, Aleksei Bingham

Project Design / Outline

Dataset:

- Bank Account Fraud Dataset Suite (NeurIPS 2022)
    - https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022?select=Base.csv

Notes

- The data in the data set deals specifically with real-world bank account opening fraud detection, but it is all anonymized.
    - There is a paper associated with this data set detailing its uses in ML accessible at: https://arxiv.org/abs/2211.13358

Information we're looking for

- Vulnerable/Susceptible demographics for victims of fraud
- Correlation between income and fraud
- Success rate of different fraud tactics

Phrased as questions

- As age increases, the likelihood of falling victim to fraud increases.
- As income increases, the likelihood of falling victim to fraud for a number of reasons (including but not limited to the chance of being approved for an account) increases.
- Not sure if the types of fraud (email, phone, etc) are actually visible.
- Employment status will impact likelihood of falling victim to fraud or the FPR (False Positive Rate) of denying bank accounts opening.

Preprocessing / EDA

Data Cleaning and Manipulation

The dataset was overall mostly clean. Some slight modifications were necessary with the non-trivial attributes of the data set. Some steps taken to clean the data include:

- Changing all negative values to zero
    - session_lengths_in_minutes had several entries with a negative value, all being -1. This was likely a placeholder because the data wasn't tracked.
    - device_distinct_emails_8w had a single negative value (-1). This was most likely a placeholder for no entry.

```
all_data.loc[all_data[Att.session_length_in_minutes] < 0, Att.session_length_in_minutes] = 0
all_data.loc[all_data[Att.device_distinct_emails_8w] < 0, Att.device_distinct_emails_8w] = 0
```

Image detailing how negative values were handled

Side note of Data Cleanings

- payment_type had anonymized values falling into a pattern AA, AB, AC, AD, or AE for the sake of protecting private data. While these values aren't able to really be extrapolated to new data, the values can still be used to show that there are trends with fraud based on specific payment type, though the specifics may not be known.
- Intended_balcon_amount had primarily negative numbers (which doesn't seem to make sense), so the column while not removed was essentially ignored when considering our data set using iloc.

Missing Data

While performing data cleaning, few missing values were encountered. These missing values were replaced by using the mean of the attribute.

- previous_address_months_count
    - One missing value
- current_address_months_count
    - One missing value
    - One zero value
- bank_months_count
    - One missing value

```
prev_address_avg = all_data[Att.prev_address_months_count].mean()
current_address_avg = all_data[Att.current_address_months_count].mean()
bank_avg = all_data[Att.bank_months_count].mean()

all_data.loc[all_data[Att.prev_address_months_count] == -1, Att.prev_address_months_count] = prev_address_avg
all_data.loc[all_data[Att.current_address_months_count] == -1, Att.current_address_months_count] = current_address_avg
all_data.loc[all_data[Att.bank_months_count] == -1, Att.bank_months_count] = bank_avg
```

Image detailing how missing data was handled

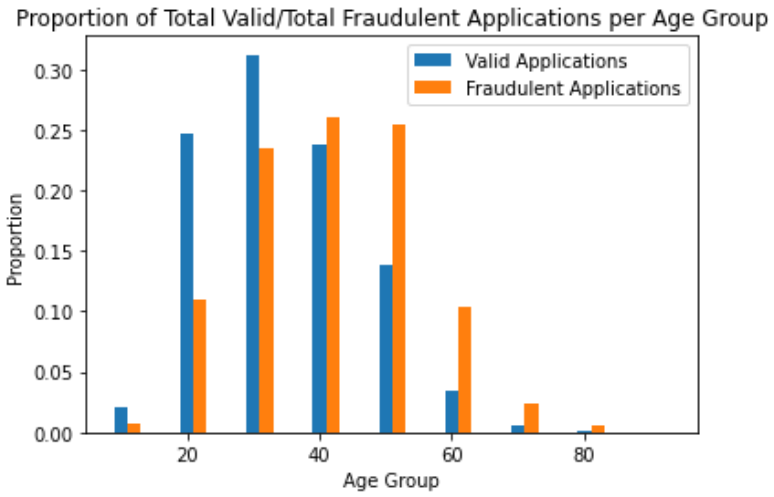
Outlier Searching

The first method for searching for an outlier was surprisingly obvious. We wanted to determine if age groups had any outliers (or specific trends) for falling victim to scams. With this in mind, we first looked for outliers in the data set corresponding to Age groups and proportion of valid and fraudulent applications submitted.

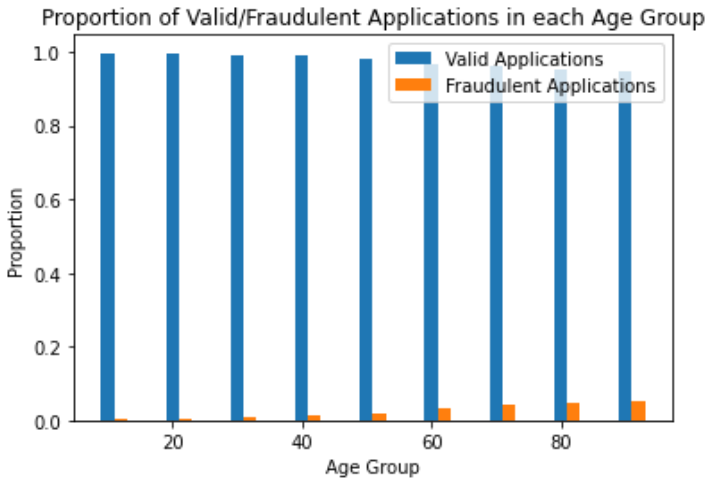
EDA Related to Hypothesis

1. Hypothesis: As age increases, the likelihood of falling victim to fraud increases.
   Description: Proportion of Total Valid/Total Fraud Applications per Age Group
   - ~75% of fraudulent applications were in their 30's - 50's
   - From the second graph we see that risk of fraud increases with age.
   - A comparatively much higher proportion of applicants in their 90's are victims of fraud than applicants in their late teens or 20's.

   Results:
   - Based on our findings, we are accepting our initial hypothesis that as age increases, the likelihood of falling victim to fraud increases.
   - As you can see in the graphs below, the proportion of fraudulent applications steadily increases with age. *Graph 2* gives a good visual representation of the steady increase in fraudulent applications based on age group.



Graph 1



Graph 2

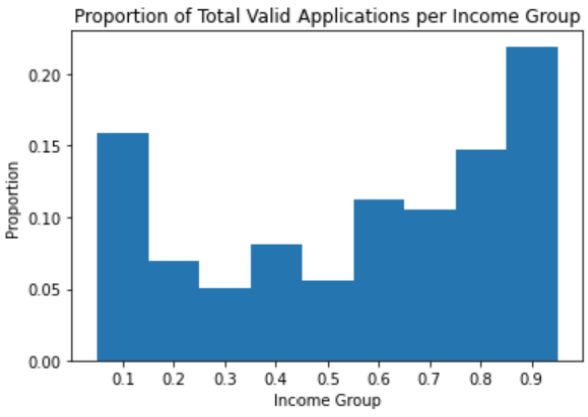
2. <u>Hypothesis:</u> As income increases, the likelihood of falling victim to fraud for a number of reasons (including but not limited to the chance of being approved for an account) increases.

   <u>Description:</u> Proportion of Total Valid Applications per Income Group

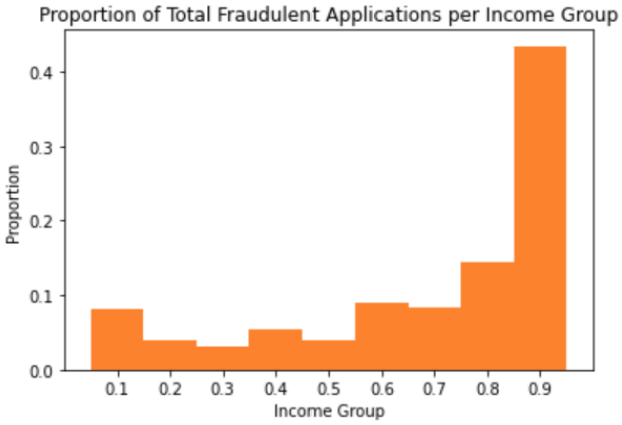
   - Supports our hypothesis that as income increases, the chance of fraud increases.
   - Fraud levels at their highest in group 0.9, but still relevant in groups 0.1, 0.6, 0.7, and 0.8.

   <u>Results:</u>

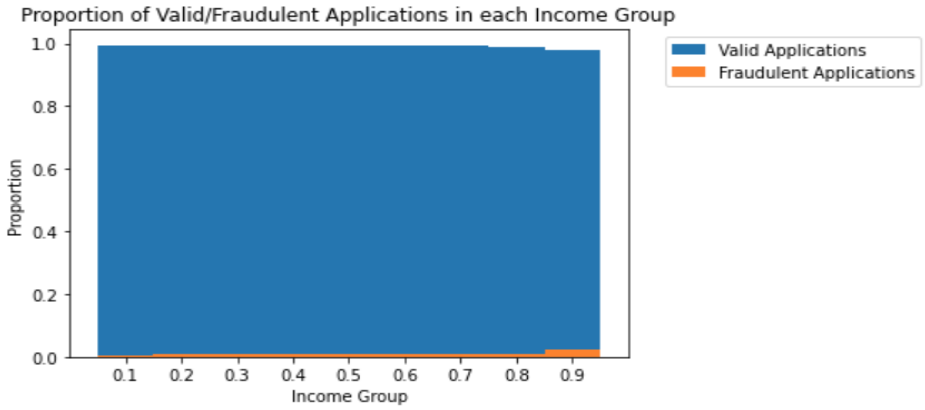
   - Based on our findings, we are accepting our initial hypothesis that as income increases, the likelihood of falling victim to fraud increases.
   - As we can see in our graphs, fraud levels are definitely at their highest in group 0.9 (the highest income group).
   - Relating back to our initial question of finding a correlation between income and fraud, it is important to note that fraud levels are still relevant in groups 0.1, 0.6, 0.7, and 0.8.



Graph 1



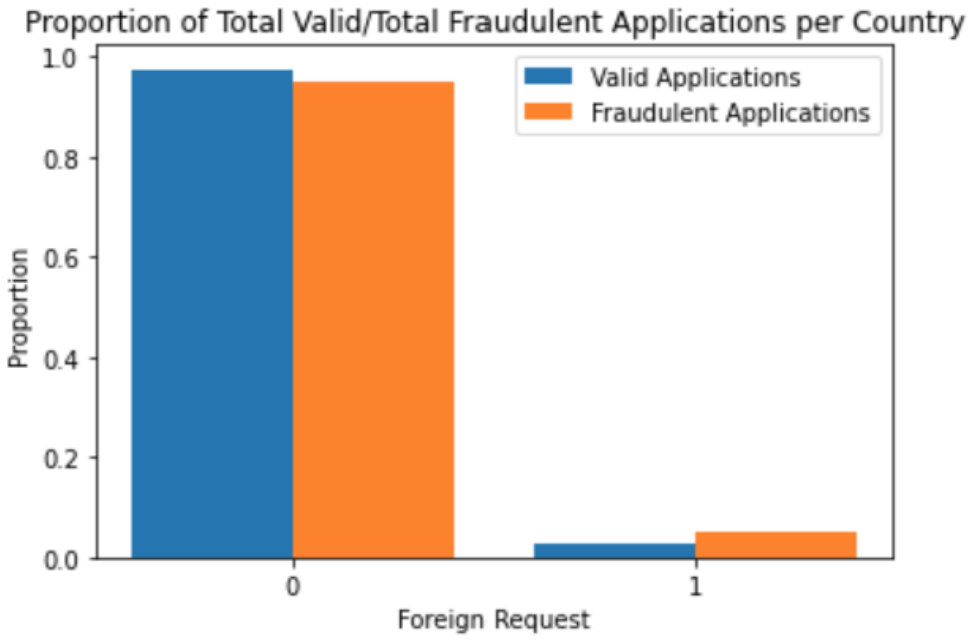
Graph 2



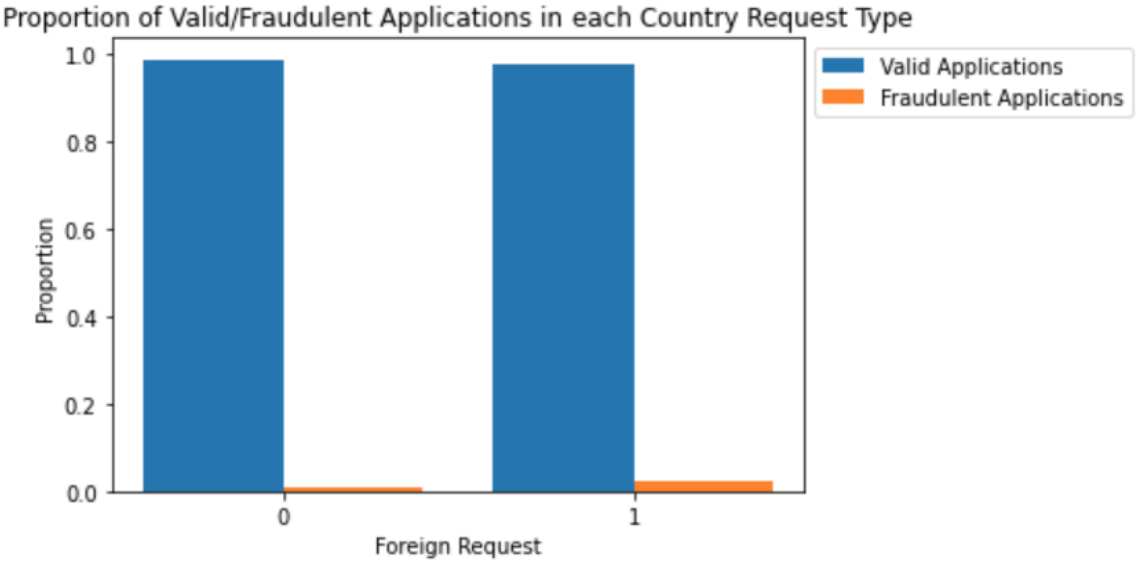
Graph 3

EDA Unrelated to Hypothesis

1. Proportion of Total Valid/Total Fraud Applications per Country of Request Origin Flag.
    - Most requests are from inside the country so that's where the most fraudulent requests originate from.
    - However, there is a higher chance of a request being fraudulent if it originates from a foreign country versus the same country.

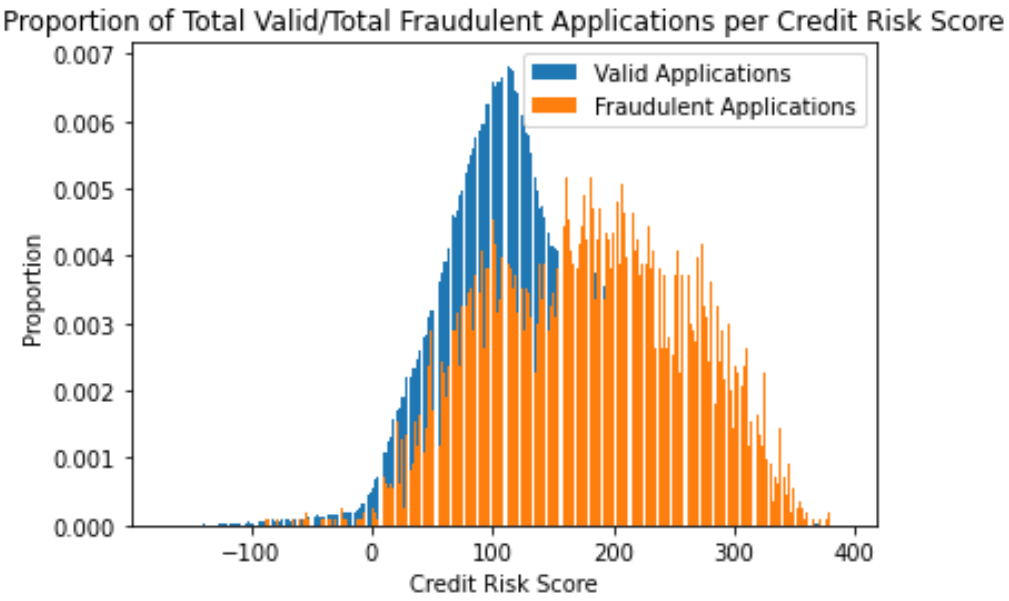


Graph 1



Graph 2

2. Proportion of Total Valid/Total Fraud Applications per Credit Risk Score
    - Average valid application seems to have a CRS of ~100 while the average fraudulent application is ~200.



Graph 1

EDA Findings Summary

- Accepted Hypotheses:

    - As income increases, the likelihood of falling victim to fraud for a number of reasons (including but not limited to the chance of being approved for an account) increases.
    - As age increases, the likelihood of falling victim to fraud increases.
- The likelihood of falling victim to fraud, based on income, is highest in the largest income group.
- ~75% of fraudulent applications were in their 30's - 50's (age).
- Proportion of fraudulent applications is much higher in the age range 60-90 than 10-30.
- Most requests are from inside the country so that's where the most fraudulent requests originate from (Country of Request Origin).
- Average valid application seems to have a Credit Risk Score of ~100 while the average fraudulent application is ~200.

Data Mining

Technique 1: Random Forest (Decision Tree)

The first technique used was Random Forest, which is a supervised learning algorithm. In the Random Forest algorithm, decision trees are built on different samples and take their majority vote for classification (and average in case of regression). In this case, the metric *fraud_bool* is used as our target variable. Random Forest is an extremely popular and commonly used technique for classification. We decided to use this technique, not only because of that, but also due to the fact that Random Forest uses a large number of models (decision trees) working as a committee (which will outperform any individual model).

When sampling, we needed to ensure that the training set consisted of ~30% fraudulent data so that the technique would not just guess "not fraud" (while still achieving high accuracy). This was our initial finding when working with Random Forest. The raw data set was given without resampling, and the technique would guess "Not Fraud" while still achieving ~98% accuracy. In this initial data set, *fraud_bool* = *true* only made up ~0.15% of the data set and adjustments needed to be made accordingly.

As mentioned, we needed to ensure that the training set consisted of ~30% fraudulent data so that the technique could accurately predict results. After making these adjustments, the Random Forest technique used is able to achieve ~98% accuracy. As you can see below (*Figure 1 - Results*), the statistics for the best model are

```
accuracy: 0.938 (±0.00072), avg score time: 0.145, for {'max_depth': 16, 'max_features': 1, 'n_estimators': 2}
```

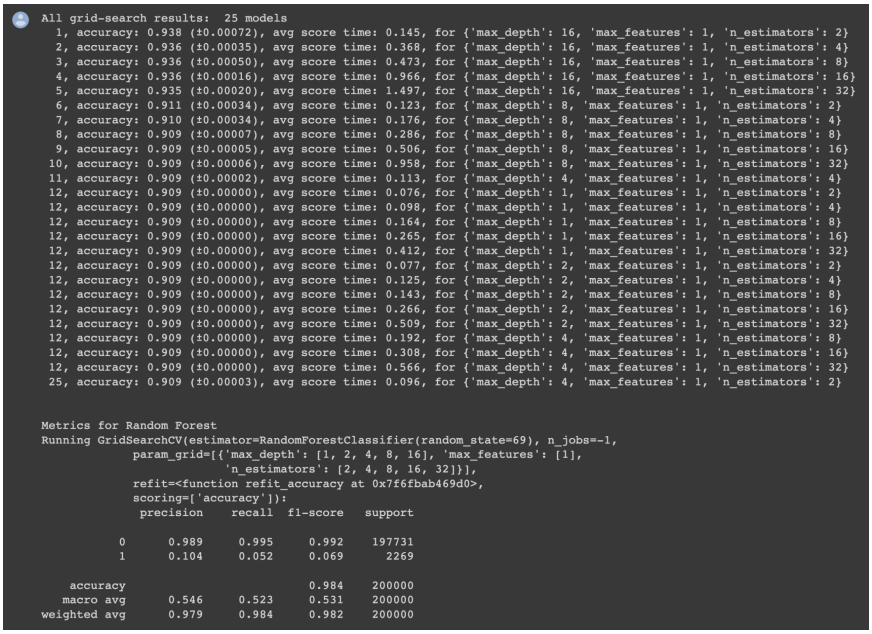


Figure 1 - Results

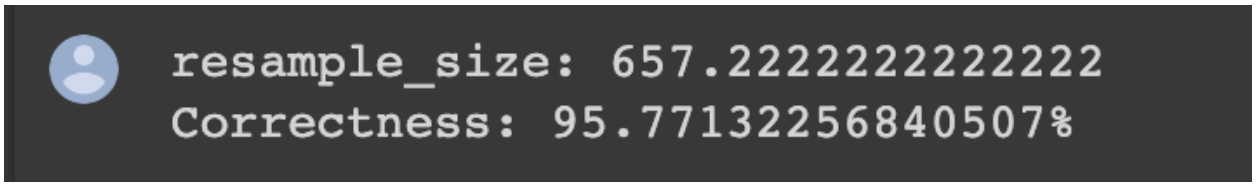
Technique 2: Neural Network to predict *fraud_bool*

The second data mining technique we used was a Neural Network, using Scikit Learn's MLPClassifier (multi-layer perceptron), to predict *fraud_bool* (our target variable). This technique is considered an unsupervised learning algorithm.

This particular technique was chosen because the functions provided with Scikit Learn makes the process a lot easier to implement, which means that we could focus on building out a more robust set of input nodes.

While using this technique, we learned that MLPClassifier actually has three possible "solvers". The first, Adam: can converge faster, but generalize poorly. The second, Lbfgs: Unsuitable for large data sets, given our limited memory. And lastly, Sgd (the solver used in our implementation). Even though Sgd takes longer, it allows the network to generalize faster.

In our process, we used Min-Max to normalize the data. When resampling, we ensured that fraud data made up 30% of our data set (as we discussed in the prior section on our Random Forest implementation). After using Scikit Learn's MLPClassifier and the sgd solver, we resulted in a correctness of ~95.77%.



Project Summary

Initial Proposal and what the expected outcome was

In our project, we were working with a Bank Account Fraud dataset. Our group was focused on looking for three key things:

1. Vulnerable/Susceptible demographics for victims of fraud.
2. Correlation between income and fraud.
3. Success rate of different fraud tactics.

After some review, our group arrived at three main hypotheses:

1. As age increases, the likelihood of falling victim to fraud increases.
2. As income increases, the likelihood of falling victim to fraud for a number of reasons (including, but not limited to, the chance of being approved for an account) increases.
3. Employment status will impact likelihood of falling victim to fraud or the FPR (False Positive Rate) of denying bank accounts opening.

Observed Outcome

(TL;DR)

- Risk of fraud increases with age.
- Fraud levels are highest in the highest income group, but still relevant in the 0.1, 0.6, 0.7, 0.8 income groups.
- Age, income, and employment status are key markers in determining fraud likelihood.

(In-depth)

When we performed Exploratory Data Analysis (EDA) on two of the three mentioned hypotheses, we were able to accept them both. The hypotheses we decided to perform EDA on were the first and second in the list mentioned in the prior section. As far as our hypothesis relating to age goes, you can see in the second graph that risk of fraud increases with age. Our findings also showed that there is a comparatively much higher proportion of applicants older in age (specifically the 90's), in comparison to victims in their teens and twenties. Our second hypothesis was supported, through analysis, that as income increases the chance of fraud increases. We did notice, however, that although fraud levels are highest in the highest income group, fraud levels are still very relevant in the 0.1, 0.6, 0.7, 0.8 income groups.

The Random Forest allowed us to prove that there were predictable outcomes based on various variables; those falling prey to fraud did not do so randomly. Given the high accuracy in the predictions, one can conclude that there are specific categories that determine if someone will be targeted/susceptible to fraud, namely age, income, and employment status (as determined from previous phases). The Neural Network falls under the same realm as random forest, in that we decided to build out our models on predictions as opposed to classification. We knew that there were key markers in determining fraud likelihood, and wanted a generalizable model that could predict whether or not a given individual is at risk.

Possible Reasons for differences and/or details on the results

1. Data Imbalance: One of the main challenges we faced was the imbalance in the dataset. The number of fraudulent cases was significantly lower compared to non-fraudulent ones, which could have affected the model's ability to accurately predict fraud cases. We

had to deliberately resample the data to ensure enough fraudulent cases were included in the training set, which may have affected the generalizability of our models. Future research could explore more advanced techniques for handling imbalanced data, such as oversampling the minority class or using a combination of over and under-sampling techniques.

2. Anonymized Features: Some features, such as payment_type, were anonymized to protect private data. While we could still observe trends related to these features, it was difficult to make specific conclusions about their impact on fraud likelihood. This lack of detailed information may have limited our ability to fully explain the reasons behind some of our findings.
3. Feature Selection: Although our models achieved high accuracy, we may have missed some important features that could better explain the relationships between demographics and fraud likelihood. It is also possible that some features in the dataset were highly correlated, which could have affected the performance of our models.

Issues Faced During the Project & Project Process

One of the biggest issues we ran into was around data sampling sizes. The actual number of people who fall victim to fraud is low, at least compared to the number of people who didn't in this data set. When building out our models, we had to deliberately remove the entries with a positive fraud outcome, minimize the non-fraud data set, and then shuffle in fraud markers such that there was enough data to train the models on. This was a bit of a kink as it took a bit to remember how to do it, and it removed an aspect of generalizability for our project that we were hoping for. Other issues encountered include the bane of most data scientists' existence: null values. There were a number of null values in areas where there arguably should not have been, so figuring out data imputation, generalizing various columns, or simply removing the data when it was either not contributing or contributing negatively.

Earlier phases went swimmingly. Most data analysis went off without a hitch, and it was more a question of collating the data into a readable format. We learned a lot about specific analyses that would work best for our data set (which is a lot of numbers without any flexibility in how the data can be interpreted–the results were that it either was or was not fraud). Sci-kit was a godsend for the model-building process, and even though a lot of documentation reading was necessary it is something that can be applied out in the field, making it an easy decision to use the library. We also had to learn a lot about f1 convergence for our models, as we ran into some issues regarding categorized variables that impacted the model heavily, but those were rather easy fixes.