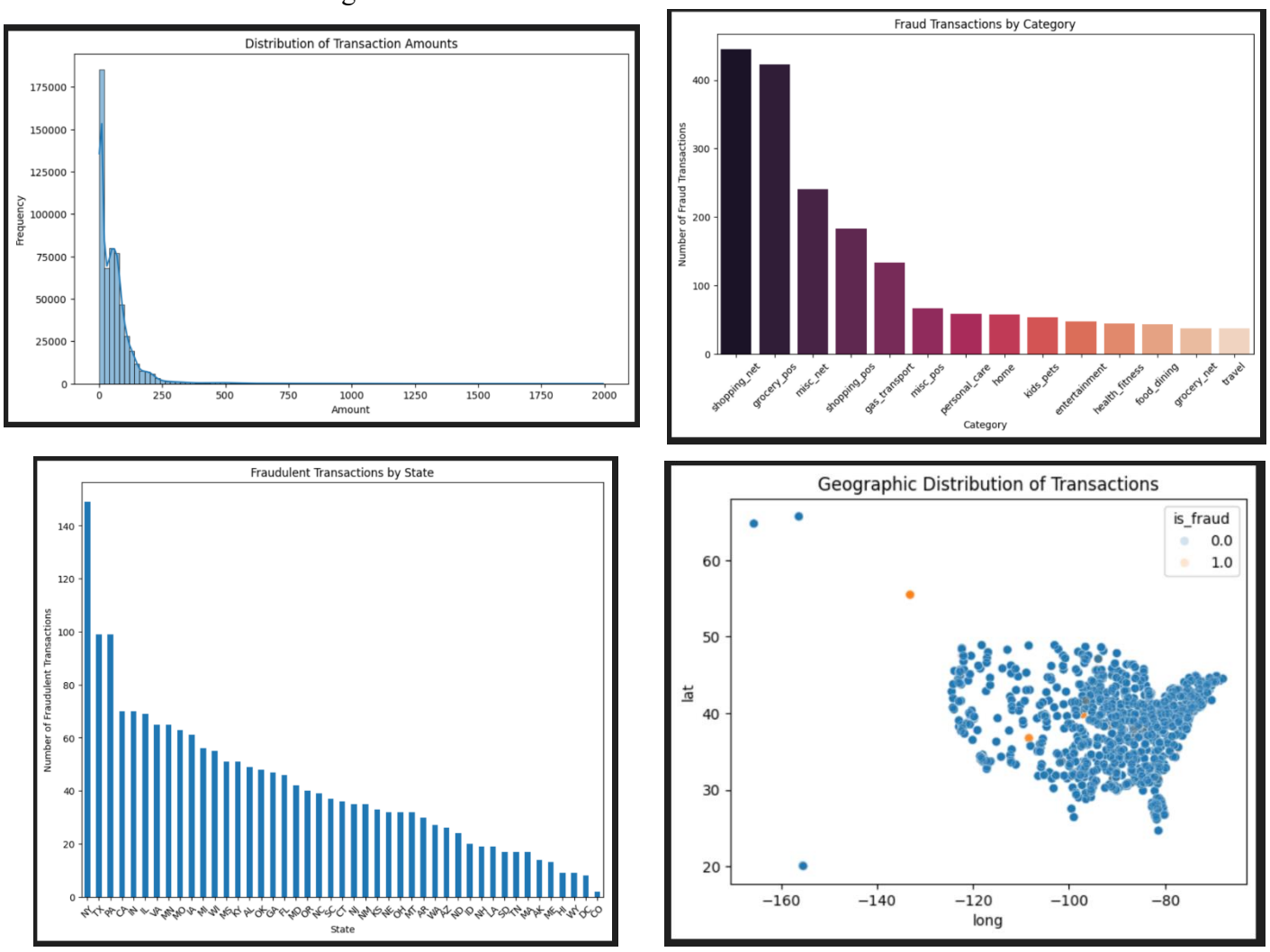
Midterm Report – Enhancing Fraud Detection in Credit Card Transactions

Kaggle Name: Longinus Her

Please refer to “starter_code.ipynb” for the code I used.

The initial step in any data analysis project is to understand the data, which I approached through exploratory data analysis (EDA). This phase aimed to uncover the dataset's characteristics, identifying patterns and anomalies. Several observations stood out:

- A majority of transactions were under $250
- Fraudulent transactions predominantly occurred from longitude of -120 to -80 and latitude of 30 to 50
- Certain states exhibited higher frequencies of fraudulent activities.
- The volume of transactions and incidence of fraud varied significantly across different categories.



Figures regarding the observations

Informed by these insights, my objective was to craft new features that could provide deeper context and enhance the model's accuracy. I strived to encapsulate these insights within a "process" function, streamlining the progression to the modeling phase. When done properly, the model would only have to be created through numerical values. Beyond the initial findings, I also considered attributes that could enrich the analysis without leading the model to overfit or underfit the data.

I first created a model with Decision Tree Classifier, since it was given with the starting code. It only picked out the numerical variables and calculated the score. During the lecture I learned that the model itself does not have a significant impact on F1 score compared to adding features; however, exploring a range of models was instrumental in identifying the most fitting one for our dataset. I tried PCA, Random Forest, SVC, and Gradient Boosting as my other options. Among them, Random Forest consistently yielded superior results, solidifying its use throughout the competition. Despite periodic reassessments with other models, none surpassed the efficacy of Random Forest

The more I thought about it, the more I was convinced that there is a strong correlation among fraud and latitude and longitude. I implemented Haversine formula to calculate the distance between two points (two latitudes = 30, 50 and two longitudes = -120 ad -80). I decided the center point based on my empirical finding. I tried to set up a limitation that would block out transactions above $250. In addition, I separated out the states such NY, TX, and PA as they had highest transactions. I also focused on top 5 categories (shopping_net, grocery_pos, misc_net, shopping_pos, and gas_transport) that displayed highest fraudulent transactions. I created multiple features regarding the frequencies and deviations as well.

At this point, the F1 score was about 0.78. For the next step, I attempted model tuning and validation. I tried cross-validation through k-fold cross-validation. I also tried grid search and randomized search for hyperparameter optimization. However, I do not have any results regarding these because my laptop crashed while doing so. After running for a few hours, it just could not handle anymore. Large datasets could be a problem, so I made a few changes. I switched to randomized search from grid search, reduced hyperparameter space, limited the number of parameter combinations sampled, and set up the parameter to use all available CPU core for the search. Even after these changes, I was not able to get the desired results.

Since I cannot just buy a new computer, I thought about what I could do to achieve a high F1 score without exhausting my current laptop. I shifted focus towards optimizing feature selection. This process of iterative refinement and evaluation underscored the complexity of the dataset. Even for this step, I had an obstacle that I could not overcome until the end. There is “trans_date_trans_time” column, which is date and time of the transaction. Although separating it out in the format of “%d/%m/%Y %H:%M” was successful, I was not able to perform other exploratory analysis regarding this column outside the function. I kept on getting error saying that there is something wrong with the column. I was not able to identify what was wrong with the column until the end.

I believe if I was able to deal with this column without an error, I would have been able to come up with features that would capture better than the existing ones. In my effort to

cover up for not able to create features that did not deal with time but still do their job, I made numerous features. Ultimately, the "process" function evolved to include a broad range of features aimed at capturing a nuanced understanding of fraudulent transactions.

This methodical approach to feature development and model evaluation significantly improved the F1 score to 0.81506 on the public leaderboard. However, transitioning to the private leaderboard revealed a decrease to 0.73863, signaling potential overfitting. I did not think too much about model tuning after my CPU failed on me. Now that I am looking at the final result, I feel like I could have tried different methods to tune the model. There would have been multiple steps I could have taken in different directions. Although my score is not great, I have learned a few valuable lessons of what to do and how think when it comes to this kind of task.