**Introduction:**

Credit cards have become more common for making everyday purchases, but with these daily purchases there is an increased risk of your data being captured and used fraudulently. In fact, 60% of those that have a credit have experienced credit card fraud (Cruz, 2024). This number may increase depending on the current economic circumstances as the factors that attribute to fraud can ebb and flow with tougher economic circumstances (Examiners, 2024). That said, according to Examiners, fraud can come from anyone but is more likely to occur if someone is experiencing hardship. This means that fraudulent transactions may be more commonly done by people that are barely getting by in life and can be more commonplace if the economy is negatively impacted.

I am part of the population that has been affected by credit card fraud. In fact, just recently, all my commonly used bank accounts and credit cards had fraud transactions. I was left without a way to pay for food and gas for over a week and had to depend on backup credit cards to pay for these items. Because of this, I wanted to try and build a credit card fraud detection model to see how this works.

**Data Selection:**

The data set that was chosen for the project was the Credit Card Fraud Detection data set from Kaggle.com. This data set was chosen due to the substantial number of transactions that it contained. Also, this did have the data previously split into what

contained a fraud transaction and what was not a fraud transaction. This is important as we need this column to build the fraud detection model.

**Modeling & Methods:**

Before modeling it was important to review the data to ensure that there were no missing values, especially in the “Class” column. This column contained the instance of if a transaction was fraud. If a value were missing here it would have to be removed. Luckily, the data was clean and there were few steps needed in the data cleaning process. However, the fraudulent transactions only made up less than 1% of the data. This would be no good to build a model on, since the model may assume that all transactions are not fraudulent. To combat this, the data was split into two different data sets. One contained all the fraudulent transactions and the other all the non-fraud transactions. Once this was completed a sample was taken of the non-fraud transactions. I chose 492 transactions to match the number of fraud transactions. I then concatenated the data sets into a new data set. The “Class” statistics were then compared from a before and after of the subset of data was made to ensure there was no change in the data. After the data was cleaned, I moved to splitting the data into training and testing sets to prepare to build a model. While considering what works well to model data that has a categorical variable as the target, two models came to mind. The first model used was a logistic regression model and the second a random forest model. Both were completed using a standard scaler as the data seems to be normally distributed. The results were great, but I then attempted a best model algorithm with feature reduction for the random forest and weights for the logistic regression. This appeared to have improved the predictive power of the models.

**Results interpretation:**

While attempting to create a model without changes to the data, I was able to create two models that were reporting a 99% accuracy score. However, I thought this was slightly suspicious given that the fraudulent transactions only took up 1% of the data. I decided to test the fraud model on an unseen data set and only received 46% accuracy score. This is when I went back to the original data set and made a new data set with the sample of non-fraudulent transactions. With this my initial accuracy on the testing data was reduced to 96.95%, which is still excellent. On the unseen data the accuracy score was reported as 93.76%. The issue of overfitting has now been corrected.

The next step I made in choosing the best model was to use the best model method. However, this again led to overfitting and would take hours to compile with the more features that were added to it. I ended up having to remove the code entirely. I needed to choose between the linear regression model and the random forest model, so I decided to review their accuracy, precision, recall, f1- score and AUC.

Classification report for the logistic regression model.

The logistic regression classification report shows that our precision is at 100% for the fraud transactions and 94% for the non-fraud transactions. Recall has the reverse of these numbers and f-1 score for both are at 97%. As these values are above 0.9, they would be considered excellent.

A number of numbers on a white background

AI-generated content may be incorrect.

Classification report for the Random Forest Model.

For the random forest model, the results are like the logistic regression model as they are all above 0.9. However, they are only slightly worse off than the logistic regression model. The ROC was also reviewed. While the area under the curve for the random forest model was at 0.95, the area under the curve for the logistic regression model was 0.97. This means that the logistic regression model would have a stronger discriminatory power between what is fraud and what is not fraud.

A number of numbers on a white background

AI-generated content may be incorrect.

**Conclusion:**

Based on the accuracy scores, classifications report, and the ROC; we can move forward with the logistic regression model as the results were all slightly higher. This model would also be cheaper to maintain than the random forest model as a lot of time can be taken to update the model if the tree needs to make new nodes with newly added data. Future work on this model should be maintenance or adding new data to the model to keep its predictive power up to date. However, when doing this, we must ensure we are not overloading the model with non-fraud occurrences. Based on the results of the model that we have obtained I feel confident in moving forward to the deployment stage. Out of 95 fraud transactions in the testing, only 6 were not caught. This will aid in saving the company money and hardship to the customer due to the number of fraud cases that were caught.

References

Cruz, B. (2024). 52 million Americans experienced credit card fraud last year. Retrieved from https://www.security.org/digital-safety/credit-card-fraud-report/

Examiners, A. of C. F. (2024). Impact of Recession on Fraud. Retrieved from https://www.acfe.com/fraud-resources/impact-of-recession-on-fraud#:~:text=Intense%20financial%20pressures%20during%20the,Certified%20Fraud%20Examiners%20(ACFE).

ULB, M. L. G.-. (2018). Credit Card Fraud Detection. Retrieved from https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud