**Research Question:**

The research question I made for this capstone was, “Can I create an effective gradient-boosting model to detect fraud in e-commerce transactions?”. This research question is important in financial institutions and e-commerce companies, as gradient-boosting models have been used in real-world scenarios to detect fraud. Demonstrating these skills and answering this question will demonstrate my machine learning expertise, how I can assess and resolve a real-world business case, and help detect fraud for these companies (Kalia, 2024). My hypothesis is below:

**Null Hypothesis-** The gradient boosting model has a recall of less than 70%.

**Alternative Hypothesis-** The gradient boosting model has a recall greater than 70%.

I decided to use recall in my hypothesis. The reason is that in real-world scenarios, the importance of these models is in predicting as much fraud as possible. The recall metric will demonstrate how much fraud our model is finding. Another factor I must stay aware of is precision. While the recall is important, precision shows how well it properly detects fraud. If the model has low precision, we will get many false red flags in the system (Madashanian, 2024).

**Data Collection:**

I collected my dataset through Kaggle. The creator, Shriyash, created the synthetic dataset through Python’s Faker library. This dataset’s advantages are the size of the dataset being large, the data is based on actual transaction and fraud patterns and has multiple factors to use in the machine learning model. These advantages gave me plenty of data and allowed me to utilize random under-sampling for more accuracy. The disadvantages are that fraud, and legitimate transactions are not well balanced, and the computational cost may be high (Jagtap, 2024). I overcame these disadvantages with the large data size and utilizing random under-sampling to increase the metrics (Brownlee, 2021). The computational cost was not bad for me as my machine could handle these computational costs (Jagtap, 2024).

**Data Extraction and Preparation**

First, I utilized Pandas to extract the data from the CSV files as data frames and connected the two data frames. It was important to check duplicates and missing data, as these could lead to problems with the model’s accuracy and setup. I checked the amount of legitimate and fraudulent transactions to see if and by how much the data was imbalanced. I dropped all unnecessary columns like transaction ID and Customer ID, as these would not provide valuable information.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect.A white rectangular sign with black text

AI-generated content may be incorrect.

I continued this process by moving into EDA. This included univariate and bivariate analysis. First, I found that customer age had some unrealistic values that were also outliers. For the records of customers ages under 15, I dropped these rows. From there, I moved into deleting the IP addresses, as there weren’t many repeat IP addresses, so it wouldn’t provide much to the patterns. I also decided to pull out the zip codes for the shipping and billing addresses. It would allow me to maintain an important part of the data but allow the model to see more patterns via location without an overabundance of information.

A screenshot of a computer code

AI-generated content may be incorrect.A diagram of a box plot

AI-generated content may be incorrect.A white rectangular object with text

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.

After the univariate analysis and the changes, I made, I moved to bivariate analysis. In bivariate analysis, I compared the independent variables to our dependent variable. This further demonstrated the sparsity of fraudulent transactions in comparison to legitimate transactions. Here, I decided to pull the day of week, hour, and month out of the transaction date, as it may show more nuanced patterns.

A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

After EDA I moved into encoding and splitting my data. I encoded my ordinal values with label encoding, so they could be represented by numbers accordingly. I applied one hot encoding for my categorical data that had no order. Lastly, for encoding, I mapped my Boolean values to 0 and 1 to represent yes and no. Switching categorical variables to numerical is a vital part of machine learning, as it allows the ML model to understand the data in its own way. I continued by splitting my independent variables into my X values and my fraudulent dependent variable into my Y value for the model’s proper understanding of what variables are being used to predict the dependent variable. I finished by splitting the data in 80/20 for training/testing data, so the machine had a standard amount of data to train on and test with.

A screenshot of a computer code

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A white rectangular object with black text

AI-generated content may be incorrect.  
A white rectangular object with black text

AI-generated content may be incorrect.

One advantage of my data preparation was that I removed irrelevant data, and incorrect data and encoded all variables to be understood correctly. Another advantage of using these tools and techniques is that they were all industry-standard tools and techniques for extracting and preparing data for analysis. One disadvantage of my extraction and data preparation was that I did maintain outliers. I kept these outliers as they were all legitimate occurrences that could help my model, but they can also negatively impact the metrics I am using to model the success of the model.

**Analysis:**

My data analysis process was focused on creating a strong gradient-boosting model with 70% recall. At first, I tried using SMOTE to help balance the data more and help with metrics, but it did not provide good results. Realizing I was just creating more synthetic data from synthetic data, I decided to utilize random under-sampling as I had so many data points that the model could use (Brownlee, 2021). This method utilized a random 58,525 data points to create my gradient-boosting classifier model. While the recall of this first method was 100% for fraudulent charges, the rest of the metrics were terrible. I assumed the model was overfitting and needed to use some hyperparameter tuning. While this method had better metric results, it was still only grabbing a portion of the data. The disadvantage can be seen as not utilizing all the data provided and possibly missing out on valuable input for the model. I utilized RandomSearchCV for my hyperparameter tuning, as the computational cost was affordable, and it would help tune my model more accordingly (*Tuning XGBoost Hyperparameters With RandomizedSearchCV*, n.d.). The results were more realistic, with a recall of 72%, a precision of 26%, and an AUC of 80.51%. The advantage of using RandomSearchCV was the balance of increasing model performance while maintaining a reasonable computational cost. The disadvantage was that this model could benefit from a less random method that may increase its metrics.

**Baseline Model**

A screenshot of a computer screen

AI-generated content may be incorrect.

**Model with Hyperparameter Tuning**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Data Summary and Implications**

My data analysis implies that the model does pass the alternative hypothesis test with a recall of 72%. I have answered the business question, but there is room for improvement. There is room for growth in our weak precision and low F1 score. The precision of only 26% demonstrates that there will be a lot of false red flags. This model is limited in its accuracy, and I suggest further tuning of this model (Mardashanian, 2024). I could investigate the variables and see if some multicollinearity is affecting the results and perform some feature selection. Another method I can try is using more computationally heavy hyperparameter tuning for better metrics or even testing a neural network that may increase fraud detection. Either way, I suggest further improving fraud detection.

**A graph with numbers and a number of text

AI-generated content may be incorrect.A graph of a line graph

AI-generated content may be incorrect.**

References:

Brownlee, Jason. (2021, January 5). *Random Oversampling and Undersampling for Imbalanced Classification.* Machine Learning Mastery. <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>

Kalia, M. (2024, October 16). Why Gradient Boosting Machines (GBM) is Effective for Fraud Detection. <https://www.linkedin.com/pulse/why-gradient-boosting-machines-gbm-effective-fraud-detection-kalia-ybfqc/>

Madarshanian, Ramin. (2024, April 4). *Precision & Recall: When Conventional Fraud Metrics Fall Short.* Kount. <https://kount.com/blog/precision-recall-when-conventional-fraud-metrics-fall-short>

Scribbr. (n.d.) *Scribbr.* Scribbr. <https://www.scribbr.com>

*Tuning XGBoost Hyperparameters with RandomizedSearchCV.* (n.d.). Stack Overflow. <https://stackoverflow.com/questions/69786993/tuning-xgboost-hyperparameters-with-randomizedsearchcv>