Detecting and Predicting Credit Card Fraud

Arizona State University

Data 490

Tristan McCoy

Table of Contents

Executive Summary

This study aims to investigate the occurrence of credit card fraud and identify the features of the dataset that serve as the best predictors of fraudulent transactions. The analysis involved employing a correlational analysis and coefficient & feature analyses of various models. These models include a logistic regression, a stochastic gradient decent classifier, a random forest classifier, and a histogram-based gradient boosting classifier. By leveraging these techniques, this study provides valuable insights into credit card fraud.

The dataset used in this study is a simulated dataset that is representative of real-world data. It contains over 1.8 million observations and has 23 features. Sensitive features like the cardholder's address will be removed before evaluation and extra target features will be engineered to gain further insights on the matter of credit fraud. The issue of data imbalance will be a main concern for this study as the count of legitimate credit transactions far outnumbers fraudulent credit transactions. Financial institutions and credit card companies can utilize these findings to develop more effective fraud prevention measures and improve the security of their systems assuming that this simulated dataset accurately reflects real-world transactional data within an acceptable degree of error.

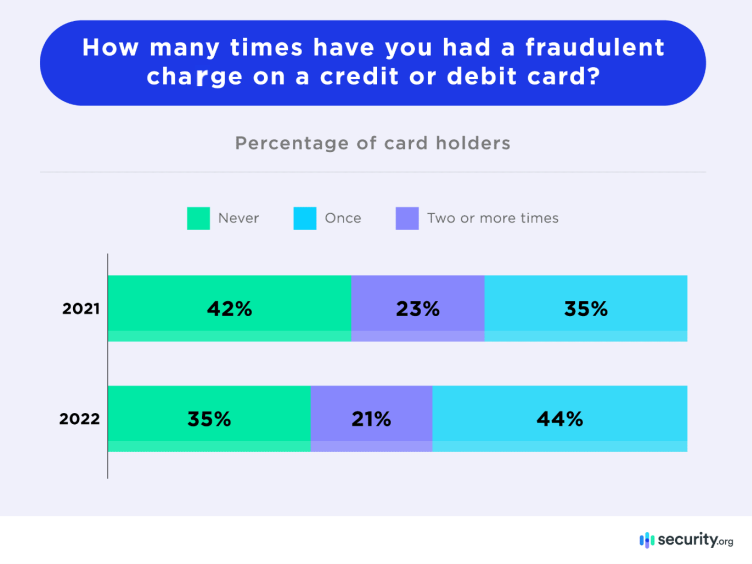
Based on the findings, the amount feature (the value of a credit card transaction in USD) demonstrates the strongest predictive power in determining credit fraud. Furthermore, the study explored the timeline of when fraud is most likely to occur. While the evidence is weak, it suggests a potential association between late night/early morning transactions and credit card fraud. The identification of the amount feature as the strongest predictor and the indication of potential time patterns contribute to the enhancement of fraud detection strategies.

Project Plan

Background:

Credit card fraud refers to the unauthorized or fraudulent use of someone else's credit card information or account for financial gain. It involves various illegal activities such as stealing credit card details, creating counterfeit cards, or making unauthorized transactions. Fraudsters can obtain credit card information through methods like phishing, skimming, hacking, or data breaches. Credit card fraud poses significant challenges for financial organizations and comprises billions of dollars wroth of fraudulent transactions. Consequently, these organizations are at risk of financial loss, repetitional damage, increased operational costs, and more.

The Federal Reserve Bank of San Francisco reported in 2019 that 51% of all transactions were made using credit or debit and this ratio is expected to have risen after the COVID-19 pandemic. With an increase in credit card transactions and the improvement of financial technology, it is essential for banks and credit card companies to develop models to predict potential fraudulent behavior. Data science and the use of machine learning can assist financial entities to develop methods of predicting fraudulent credit card behavior. In a nutshell, the main



goal of this project is to address the issue of class imbalance and to construct effective supervised machine learning models.

Research Questions:

Q1: When are fraudulent credit card transactions more likely to occur?

Credit card fraud accounts for billions of dollars in losses for card lenders. It is not practical to predict all fraud as the cost to identify every fraudulent transaction usually outweighs the financial losses. That being said, it is still necessary to know when fraud usually happens. Knowing this can help predict losses which is essential for budgeting purposes, risk management, stakeholder communication, and performance evaluations for card lenders. The aim of this question is determine when fraud occurs most.

Q2: What classifiers are more likely to predict fraudulent credit transactions?

The ability to predict fraud is an essential tool for credit card lenders. This allows these organizations to protect their customers and shields themselves from financial losses caused by fraud. An organization's reputation with their customers and regulatory bodies is on the line if fraud is not mitigated properly. By knowing which classifiers/indicators best predict fraudulent transactions, financial organizations can take proactive measure to prevent fraud before it happens. This question aims to identify the variables that correlate highly with fraud in order to understand how fraudsters operate.

Hypotheses:

H1: Months with major holidays have more instances of credit card fraud.

The total dollar amount of fraudulent transactions may also be higher during these times. The prediction may be impacted by the unbalanced nature of fraudulent transactions.

H2: One or two features strongly correlate with fraudulent transactions.

I suspect the distance between the merchant & cardholder's address and the category of the merchant will yield strong correlation with fraud. There will probably be a few more features that also predict well but with weaker correlation.

Data:

I have chosen the Credit Card Transactions Fraud Detection Dataset and can be found on Kaggle. This dataset contains 1.8 million observations over a timespan of about 2 years where each instance corresponds to a credit transaction labeled as fraudulent or legitimate (the target variable). It is a simulated data set since real-world credit card transaction data is hard to obtain due to privacy concerns. It does, however, contain 23 labeled features that have similar features to real-world credit card data. Some of the indicators include:
Transaction date & time, transaction category (ie. fuel, grocery, entertainment, etc.), transaction amount ($), card holder's latitude & longitude, birth date of card holder, card holder's address & zip code, merchant's latitude & longitude, etc.

A few of the indicators will be used for feature engineering (eg. Latitude and longitude of cardholder and merchant).

This dataset is unbalanced, it has an uneven distribution of legitimate transactions. The majority class is comprised of legitimate credit card transactions and fraudulent transactions are the minority class.

Methodology:

For Q1, the interest is to predict fraudulent credit card transactions will happen the most. To address this question, the date & time and the transaction amount feature will be useful in order to construct an analysis. Aside from monitoring the count of fraudulent transactions over time, an analysis of the total dollar amount of fraud transactions over time can also yield additional insights. A regression model could be constructed to capture these trends over time.

Q2 aims to determine the best predictors for fraudulent credit card transactions. To address this, I will use a majority of the features as predictors for the models after performing an analysis to assess the usefulness of each feature. EDA, feature analysis, and selection will be a part of this process before constructing any models.

The models most likely to be used will be supervised ML models like naive Bayes, random forest classifier, a deep neural net, etc. An ensemble approach to compare the different models is necessary.

Resampling is a crucial step in answering research question 2. Resampling handles imbalanced data and is performed before creating testing and training sets. Without resampling, the minority class may be mistaken for noise in the data and not be classified accurately. Under

sampling and over sampling are common resampling techniques. For under sampling, observations are removed from the majority class stochastically until the target class is balanced. Oversampling involves artificially increasing the number of observations in the minority class until the target class is balanced. There are advantages and disadvantages to both techniques so I will probably compare both techniques on the data or use a hybrid approach. I am also looking for other resampling techniques to use that other studies have not tried.

Computational Methods and Outputs:

For Q1, data visualization techniques can be used to see the monthly trends in fraud occurrence against the total money involved in those occurrences. The regression model would be evaluated using AUC/ROC.

For Q2, accuracy, recall, F-scores, and precision also will be used for evaluating the various models. AUC/ROC will most likely be unused due to the use of resampling; most studies do not use these two metrics and one explicitly advised against using them.

Outputs for Q1 and Q2 would be a prediction of the dollar amount of fraudulent credit card transactions for a period of time and the top features that best predict a fraudulent credit card transaction, respectively.

Another important output for both research questions is identifying a resampling method that optimizes a model for processing unbalanced data.

Output summaries:

For Q1, a time series plot of the dollar amount of each instance of fraud including a plot of every instance of a fraudulent transaction. A calculated table with count and the total dollar amount of fraudulent transactions for a given time period. A visualization of the regression model can be added as well.

For Q2, a heat map representing the correlational analysis of predictors and a confusion matrix for the model's performance. A confusion matrix would be created for each resampling method, as well, to summarize each method's performance.

Literature Review

Credit card transactions have become increasingly more popular with the development of financial and business technology like tap-to-pay and contactless payment methods. The Federal Reserve Bank of San Francisco reported in 2019 that 51% of all transactions were made using credit or debit and this ratio is expected to have risen after the COVID-19 pandemic. An increase in credit card transactions gives fraudsters more opportunity to steal personal credit card data. This gives banks and credit lenders an incentive to construct predictive models for fraudulent transactions.

The good news is that most cases of fraud can be easily identified by a customer or bank and have reliable labels (fraudulent vs legitimate). Because of this, Provost & Fawcett in *Data Science for Business* recommend using classic supervised data mining techniques to predict fraudulent cases — naive Bayes, logistic regressions, random forest, etc. Issues arise, however, when we consider the distribution of instances labeled fraudulent. Many times, the training sets are unbalanced as instances of legitimate transactions outnumber the fraudulent ones.

To address this imbalance, a study, *Unbalanced Credit Card Fraud Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques*, conducted by Gupta, et al. focuses on a few rebalancing techniques by way of resampling and feature selection before implementing various supervised machine learning algorithms. Over sampling was found to be the best approach to tackle the issue of imbalance. The F1-score was used to compare each model and a gradient-boosting classifier was found to be the most accurate model. They suggest from other studies that deep convolutional neural net yield high performance when using real time data.

Another study, *Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS)* by Ahmad, et al., compared random under sampling and clustering with the results indicating that the cluster approach is the most effective. They used fuzzy C-means clustering to resample the data based on selected similarities before implementing classic supervised ML algorithms and evaluating performance based on the usual metrics. Naive Bayes and kNN performed optimally and the artificial neural net outperformed all of the other models. This paper outlines a useful framework for modeling and processing unbalanced datasets, specifically with credit card fraud data.

As final note, other studies have discussed the economic impact of constructing models for fraud detection. Delamaire, et al. concludes in the study, *Credit card fraud and detection techniques: a review*, that it may be uneconomic for companies to invest too much into fraud detection systems as, in some cases, the cost of detecting fraud may outweigh the financial loss for certain cases of fraud. The security.org team published a report that states that 33% of fraudulent charges are less than $50. With this in mind, it may be beneficial to develop cost reductive methods to detect fraud and reevaluate how feature predictors are weighed within a model.

Exploratory Data Analysis

This artificial dataset was collected from kaggle.com and was simulated based on characteristics from real-world credit card transaction data. This is a fully structured, unbalanced dataset favoring legitimate credit card transactions with no missing values. The simulated time frame of this data spans from December of 2018 to December of 2020 and consists of multiple credit card transactions from different customers; some are fraudulent but most are legitimate (see Figure 1). Only 0.5% of credit card transactions are fraudulent. The data was already split between a test and a train set so I combined them into one CSV file to make EDA easier. I have left out a few variables such as the cardholder's name, address, and card number. The remaining are as follows:

Merchant: Name of the merchant

Category: Category of the merchant (eg. grocery, shopping, entertainment, etc.)

Amt: Transactional value in USD (not to be confused with a quantity or count)

Gender: Gender of the cardholder

City: City of the cardholder

State: State of the cardholder

Lat: Latitude of the cardholder's address

Long: Longitude of the cardholder's address

City Pop: Population of the cardholder's city

Job: Job title of the cardholder

Merch Lat: Latitude of the merchant's address

Merch Long: Longitude of the merchant's address

Is Fraud: Binary label indicating a legitimate or fraudulent transaction

Distance: Distance in miles between merchant's and cardholder's location

Month: The month the transaction occurred

Week: The week number the transaction occurred (1-52)

Hour: The hour the transaction occurred (0-24)

Birth Year: Birth year of the cardholder

The histogram of amounts (see Figure 2A) represents the quantity of total transactions (fraud and legit) within each amount bin. Each bin is $367 so from this figure we can see that the majority of transactions are under $367. Any transaction over $200 can be considered an outlier as 95% of transactions have an amount less than $200. Looking further into the distribution of transactions by the amount, I have determined that all fraudulent transactions involve a transaction amount of less than $1,400. Within this range using a bin size of 50, there appears to be a variable distribution of fraud cases with significantly more within the ranges of $0 - $50, $250 - $300, and $700 - $1,100 (see Figure 2B). 21% of fraud cases involved amounts less than $50, another 22% of cases for amounts between $250 and $350, and 40% between $700 and $1,100.

Next, I constructed a stacked bar graph to take a look at which category contains the most fraud (see Figure 3). While gas and transportation accounted for the most transactions out of any category, in-store grocery and online shopping (retail) were the two categories with the most fraudulent transactions. 46% of all fraud cases belong to these two categories.

Analyzing the date and time of transactions yielded comprehensive results. A display of fraud and legit transactions by week for the entire time series shows that legitimate transactions occur at a steady rate whereas fraudulent transactions occur at a more variable rate (see Figure 4A). Fraudulent transactions appear to ebb and flow every few weeks and gain more momentum during holiday seasons like Christmas time or halloween. The beginning of March seems to also be a time when fraud is more common possibly due to increased spending during the Super Bowl. An aggregate analysis by week shows that March, June/July, and December are more likely to have fraud.

An aggregate analysis by hour clearly shows that fraud cases dramatically increase from the hours of 9PM and 3AM (see Figure 4B). 9PM totaled about 100 cases of fraud while 10PM totaled about 2,400 cases. The time zone is not specified but I can still conclude that, because these are US-based transactions, late night transactions have higher rates of fraud.

A state-by-state analysis found that New York, Texas, and Pennsylvania are the top three contributors to fraud cases (see Figure 5). These three states account for nearly 20% of fraudulent transactions.

Taking a look at how fraud is distributed throughout different city populations sheds more light on what demographics to monitor. By using a bin size of 100, about 42% of all fraud involved card holders from towns and cities with a population of less than 1,000 people (see Figure 6). The average spending amount for these transactions are between $400 - $500, roughly. Cities with larger populations naturally have higher average amounts.

The last piece of analysis involves the distance between the merchant and the card holder’s home. I created a calculated field using the merchant and cardholder’s latitude and

longitude to determine the distance in miles with the help of the Python's Haversine package. Figure 7 displays the distribution of fraudulent transactions based on distance from a cardholder's home using 1 mile bin sizes. There appears to be a rough normal distribution with about 95% of the transactions occur between 10 and 80 miles from a cardholder's home. This fact corroborates the finding that rural populations are more likely to be targets of fraud. Moreover, the average transaction amount for each bin is uniform throughout ($500 - $600) except for a few transactions on the fringes which suggest high average amounts for fraud cases relatively close and far from the home.

Likely candidates for credit card fraud include late-night transactions, card holders who live in rural areas, and transactions made at grocery stores and online retail websites.

Methodology

Research Question 1: When are fraudulent credit card transactions more likely to occur?

The original dataset has date and time combined in one feature, therefore, further feature engineering is needed to create individual features for the month, week, hour, and possibly other time features. The EDA shows that the hour of the day is a significant predictor and so using a correlational analysis (Pearson's r) with a correlation matrix will be a effective way to corroborate this finding. Evaluating the performance of the ML models will further provide evidence of best predictors. See the list of models below as they will be used to address both research questions.

Research Question 2: What classifiers are more likely to predict credit card fraud?

From the initial EDA, I found three features that are likely to best predict fraud; city population, merchant category, and time (hour) of transaction. By performing a correlational analysis before training any models, similar to the methodology for the first research question, I will record which of the features have the greatest positive values and see if they match my predictions. It may also be useful to sequentially remove variables from the training set and retrain the models to see how this influences the metrics of each model. Some working models include:

- Logistic regression
  - Analyzing the standardized coefficients of the regression models will provide more information on which feature is the best predictor.

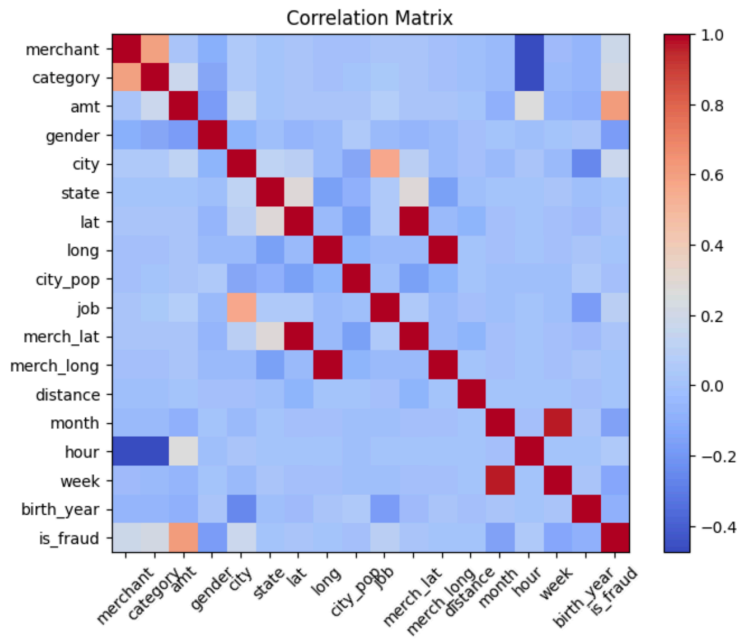
- Binary classifier: stochastic gradient decent classifier
  - Sklearn has a SGD classifier will be useful with a k-fold cross validation.
- Random Forest classifier
  - May perform optimally compared to SGD and will provide the benefit of adding weight to certain features.
- Histogram-based gradient boosting (HGB) classifier
  - HGB is optimized for large datasets and can reduce overfitting at the cost of precision but also provides early stopping capabilities.

Resampling will be done before training any of the models and will consist of a hybrid approach of oversampling and undersampling techniques so as to not cause significant information loss. The SMOTE method is ideal for oversampling and random undersampling should be sufficient for undersampling.

Data Visualizations

Correlational analysis:

The correlational analysis was performed before and after resampling. The coefficients remained proportional for both analyses. A generally weak correlation exists among most of the features but my focus is on ‘is fraud’ feature. The amount feature has the highest correlation with fraudulent transactions and has a 0.62 correlation coefficient after resampling. The top-5 most correlated features with fraud are amount, category, city, merchant, and job title with coefficients of 0.62, 0.22, 0.18, 0.18, and 0.11 respectively (see the table 1). Gender, month, week, and birth year were found to have a negative correlation with fraudulent transactions. The remaining features share very low correlation with fraudulent transactions.



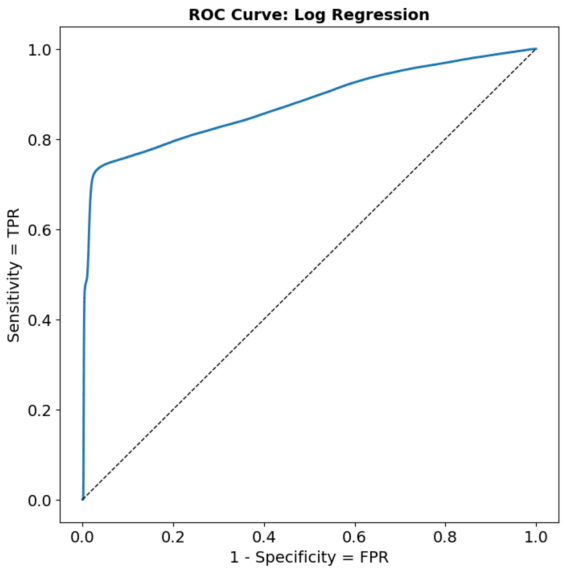
Logistic Regression:

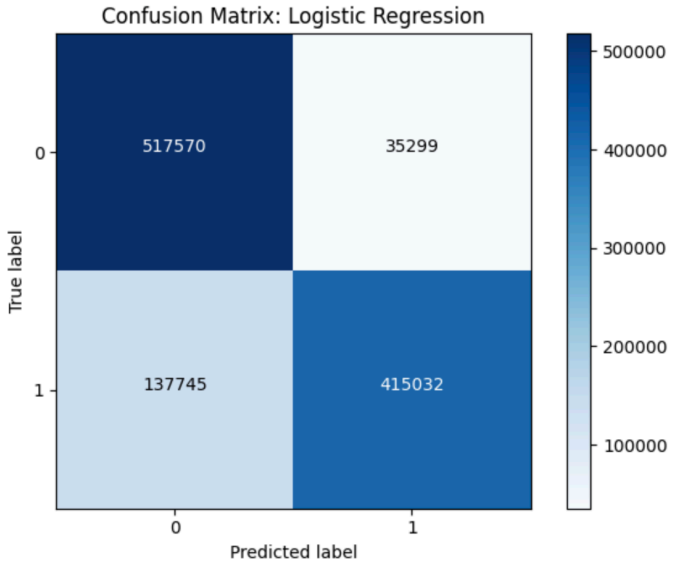
Using cross-validation with 5 folds yielded an optimal model with stronger regularization and satisfactory results. The accuracy, precision, recall, and f1 score are 84.4%, 92.2%, 75.1%, and 82.8% respectively (see the confusion matrix below). AUC for this model is 89.2% (see ROC curve below) and the r2 score is 37.4%. Table 3 contains the metrics for all of the models for comparison.

The 5 highest-weighted features in the model are amount, week, merchant, longitude, city with coefficients of 2.931, 1.413, 0.447, 0.421, and 0.288 respectively (see table 2 for the remaining coefficients). The remaining features have low magnitudes with most of them being negative.

Evaluating the model after removing the 'amount' feature significantly compromised the model to yield only a 60% accuracy score and a negative r2 score.

Partitioning the models by category yields similar results as the model with the 'amount' feature removed.



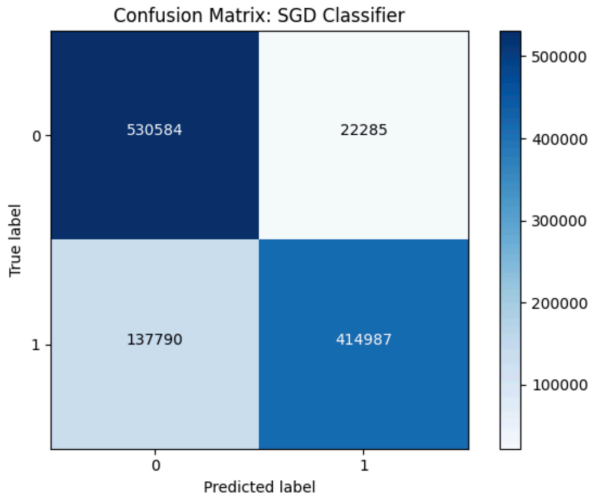


Stochastic Gradient Descent (SGD) Classifier:

The SGD model, again, performed satisfactorily after using a 5-fold cross-validation to select an 'l1' penalty and an alpha of 0.01. The accuracy, precision, recall, and f1 score are 85.5%, 94.9%, 75.1%, and 83.8% respectively (see the confusion matrix below). The top-3 features with the greatest weight in the model are amount, merchant and city at 2.52, 0.124, and 0.092 respectively. The remaining variables were either negative with low magnitude or left out of the model (see table 2 for the remaining values). State, latitude & longitude (for both merchant and cardholder), city population, job title, distance, hour, and week were not selected for this model.

The model was significantly compromised after removing the 'amount' feature and had similar metrics to the logistic model post-removal.

Partitioning the models by category yields similar results as the model with the 'amount' feature removed.

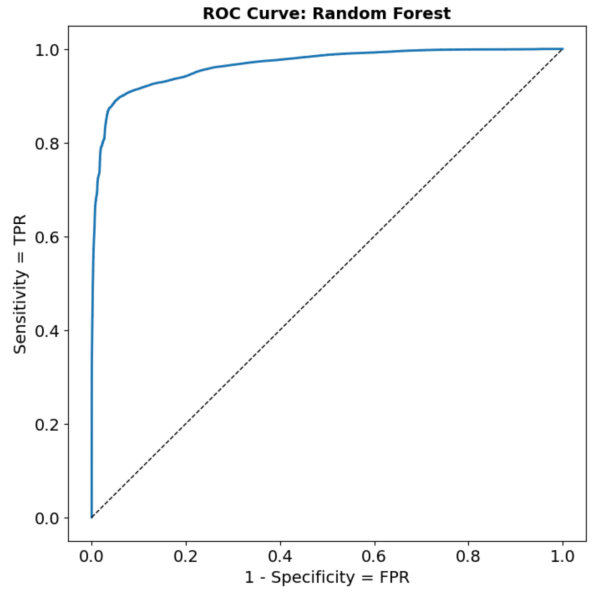


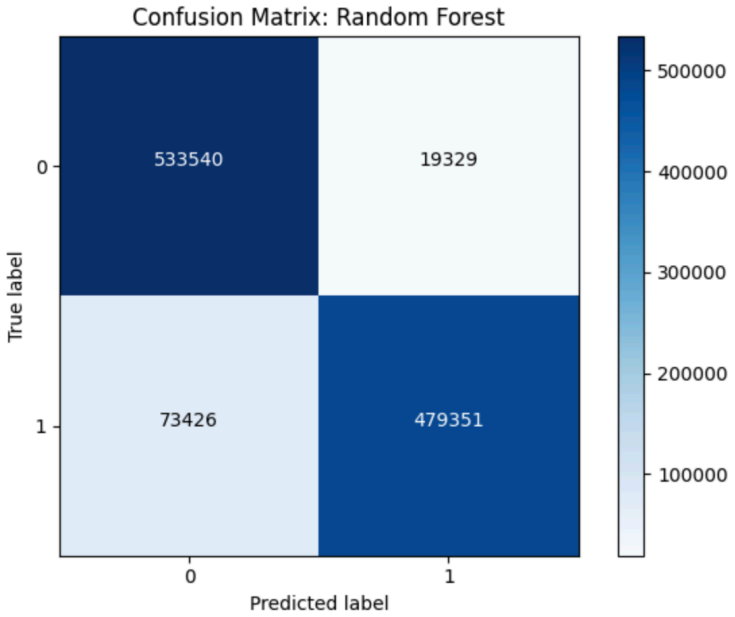
Random Forest Classifier:

The random forest model, using 200 estimator and max leaf nodes 16, yielded an accuracy, a precision, a recall, and an f1 score of 91.5%, 96.3%, 86.4%, and 91.1% respectively (see the confusion matrix below). The AUC is 96.6% (see ROC curve below) and the r2 score is 66.4%. The feature with the most importance is amount (68%) followed by hour (12.3%), category (9.2%), merchant (2.9%), and city (2.3%). The remaining features have very low importance and a few were 0. The random forest model was the most computationally expensive model.

Removing the 'amount' feature and reevaluating the model compromised the model to a 77% accuracy rate and 9% r2 score. The distributions of the feature importances remained proportional to those of the unaltered model. The top-three most important features for the altered model are hour (39.3%), category (26.8%), and city (7.4%).

Partitioning the models by category yields higher scores than the altered model and lower scores than the unaltered model.



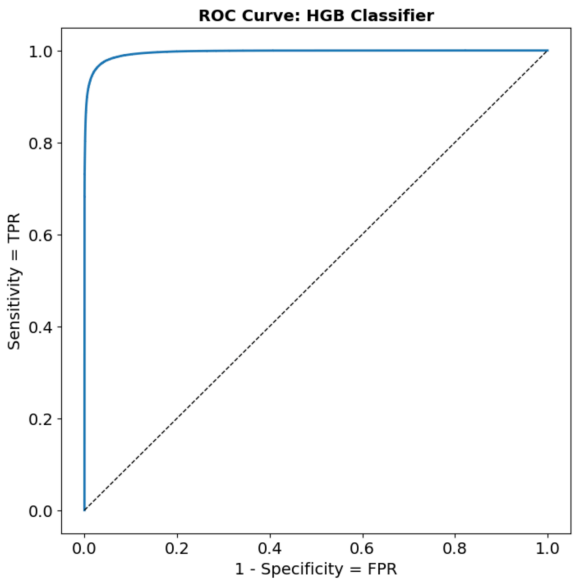


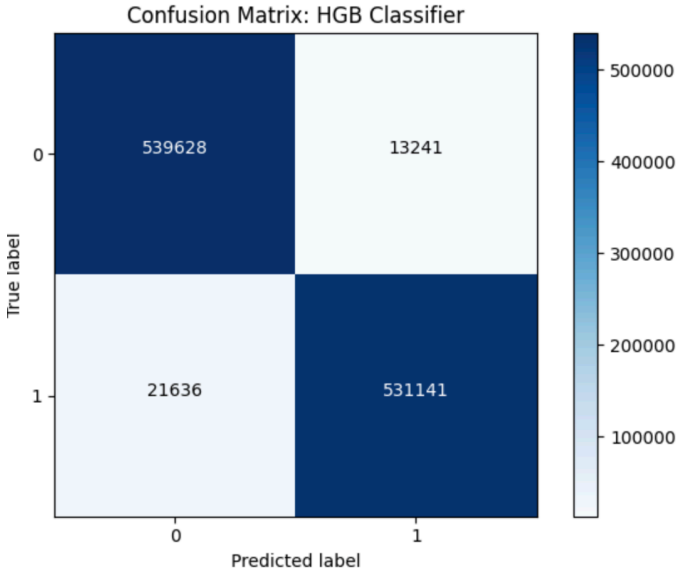
Histogram-based Gradient Boosting (HGB) Classifier:

The HGB model outperformed the previous models and yielded an accuracy, a precision, a recall, and an f1 score of 96.7%, 97.5%, 96%, and 96.7% respectively (see the confusion matrix below). It has a AUC of 99.6% (see ROC curve below) and an r2 score is 87.4%.

Evaluating the model after removing the 'amount' feature showed that it still performed satisfactorily with a compromised r2 score and an 89% accuracy rate.

Partitioning the models by category yields very similar results as the unaltered model.





Analysis

Model Performances

The HGB model outperformed the rest of the models and it strikes a balance between computational expense and accuracy — random forest was the most computationally expensive. It is an optimal model for this large dataset and other ad hoc analyses of large datasets for credit fraud. The other three models are potential candidates for discovering specific features that best predict fraud. Otherwise, the remaining models have low R2 scores and so may not be completely reliable models for classification. This is especially true for the logistic model and SGD model as they have low recall.

Research Question 1: When are fraudulent credit card transactions more likely to occur?

Hypothesis 1: Months with major holidays have more instances of credit card fraud.

The exploratory data analysis showed that later hours of the night and early morning had significantly more fraudulent credit card transactions than daylight hours. A correlational analysis yielded a relatively weak correlation between hour and fraud compared to other features. The random forest model being the second-best performing model selected hour as an important feature with a value of 0.1232. This finding suggests that the hour of the day is a moderate predictor but must rely on other features to correctly predict fraud.

The EDA did not provide strong evidence that the week feature is a strong predictor. The correlational analysis did not confirm a positive correlation to fraud but the logistic model yielded a coefficient of 1.413 for week. This suggests that in some cases the week of the year is a weak predictor for credit fraud or that a majority of the weeks of the year have steady rates of

credit fraud. Marginal increases in credit card fraud during certain holiday weekends or weekdays preceding/following holidays may explain this finding.

Ultimately, the hypothesis suggesting that certain months with major holidays would've higher instances of credit fraud was not supported by the findings of this study. Instead, the analysis revealed that late hours of the night and early morning exhibited a stronger correlation to credit card fraud compared to specific months of the year. This finding implies that fraudulent actors may exploit the relative quietness and reduced surveillance during these time periods. Additional attention should be given to these hours as they may warrant closer scrutiny and security measures.

Research Question 2: What classifiers are more likely to predict credit card fraud?

Hypothesis 2: One or two features strongly correlate with fraudulent transactions.

The initial correlational analysis shows that only a few features are correlated with fraudulent transactions. The amount feature is the highest and most significant predictor for fraud. Merchant, category, city, and job title are the following features with the next greatest correlations. Comparing these to the feature coefficients for the models slightly changes the results. Job title no longer holds significance in the models while the rest of the features hold their significance. I can conclude for this particular dataset that the amount feature contributes most to the overall performance of all the models given that it was the top features in all of the models. The merchant and city features also demonstrate significant contribution to the models' performances which implies that these features are moderate predictors of credit card fraud.

In some models, other features like hour and week also had notable impact to the models' performances. The logistic model and random forest model were able to suss out information on time predictors better than the other models.

Contrary to the hypothesis that the category of transaction and the distance between the cardholder and the merchant would be strong predictors of credit card fraud, the findings of this study indicted that the monetary value of the transaction emerged as the best predictor. Based on this finding, transactions under $50, between $250 and $350, and between $700 and $1,100 were associated with an increased likelihood of fraudulent activity. The lack of significant correlation between fraud and transaction category or the distance suggests that it is possible that fraudsters employ various techniques to obfuscate their activities, making it challenging to detect patterns based solely on transaction category or geographical distance. These findings highlight the importance of considering the transaction amount as a key feature in credit card fraud detection. Financial companies can leverage this insight to develop more accurate fraud detection models and enhance their monitoring systems by placing greater emphasis on transaction amounts within the aforementioned ranges.

Ethical Recommendations

Fraudulent credit card transactions create an ethical imperative for credit lenders to address not only on behalf of the borrowers but for themselves as well. First, credit card data, including other personal information, is very sensitive. Credit lenders would be held accountable if any of this data found its way into the hands of fraudsters and are expected to foot the bill. Therefore, fraud management and reporting is an essential business function of credit lenders. In the US, they conduct their reports and share them with the Consumer Financial Protection Bureau which supervises banks, lenders, and credit reporting agencies.

The ethical considerations of credit fraud work on multiple levels. Credit lenders want to minimize financial losses due to fraud and maintain good standing with regulatory agencies. Additionally, they need to uphold their reputation with borrowers. Even though credit fraud impacts the health of lenders, completely eliminating it is not entirely feasible. It is important to notice that some business objectives, although ideal, are not efficient or economic. Identifying cases of credit fraud becomes costly when scaled to every transaction and outweighs the money the could be saved if every case of fraud was detected.

The purpose of creating a model that identifies credit fraud is to identify major trends and indicators in fraudulent transactions. This suggests that there is an acceptable level of fraud for credit lenders to account for. At worst, this perpetuates the credit fraud industry and allows new methods of fraud to be constructed. Moreover, this also provides motivation for future research into efficient models that can cheaply and efficiently identify fraud at a large scale.

Challenges

Running the random forest model required more computational power than the rest of the models even while using the default parameters. This model took around 3-4 minutes to finish training. Performing cross-validation on this model costs significantly more computational power to a degree where cross-validation is not useful. Finding optimal parameters for this model poses a legitimate challenge because this model yields clearer results for feature importances compared to the rest of the models. The HGB model was also slow to train but did not take as long as random forest. The problem of computational expensiveness can be difficult to overcome for large datasets that are oversampled and trained on computers with lower processing power.

Another challenge involved a limited feature extraction with the HGB model. Extracting information about the features used to train a HGB model proved to be challenging. Since this model is the optimal model, a breakdown of feature importances would have provided more insight. Obtaining coefficients or feature importance measures from such models can be intricate since the focus is on creating an ensemble of decision trees rather than individual feature coefficients.

I was not able to determine a ROC curve for the SGD model due to the nature of the model. I suspect the score would follow the trend of the other metrics compared to the other models.

Recommendations

Future work for similar cases of credit card fraud could benefit from using neural networks. Decision trees, ensemble learning, and classifiers are optimal strategies for predicting fraud and for discovering important predictors of fraud. It would be worth comparing the results of different neural networks like recurrent neural nets and convolutional neural nets to the results of this study. Neural nets may be more effective in capturing the complex patterns of fraudulent transactions.

The dataset used in this study is restricted to credit card transactions within the US. Future work should include international transactions as well as domestic transactions. This dataset is also simulated and may not reflect real credit card transactional data in real time. Anonymized real-world data, albeit difficult to obtain, can be of great value to further research on credit card fraud. Proprietary studies done internally by banks and credit lenders is a safer approach considering the sensitive nature of ethical handling of customer data.

As models become more complex, it may also become more important to prioritize the explainability and interpretability of these models. Future research could aim to develop techniques that provide clear explanations for model decisions and predictions. This can enables stakeholders to understand the reasoning behind fraud detection outcomes and enhance trust in the system.

References

Ahmad, H., Kasasbeh, B., Aldabaybah, B., & Rawashdeh, E. (2023). Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS). *International journal of information technology : an official journal of Bharati Vidyapeeth's Institute of Computer Applications and Management*, *15*(1), 325–333. https://doi.org/10.1007/s41870-022-00987-w

Delamaire, L., Abdou, H., & Pointon, J. (2009). Credit card fraud and detection techniques: a review. *Banks and Bank systems*, *4*(2), 57-68.

Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*.

Gupta, Varshney, A., Khan, M. R., Ahmed, R., Shuaib, M., & Alam, S. (2023). Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques. *Procedia Computer Science*, *218*, 2575–2584. https://doi.org/10.1016/j.procs.2023.01.231

Kumar, R. & O'Brien, S. (June, 2019). 2019. Findings from the Diary of Consumer Payment Choice. *Federal Reserve Bank of San Francisco*. https://www.frbsf.org/cash/publications/fed-notes/2019/june/2019-findings-from-the-diary-of-consumer-payment-choice/?af=10545

Provost, F., & Fawcett, T. (2013, July 31). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*.

Security.org Team. (January 31, 2023). Credit Card Fraud Report. https://www.security.org/digital-safety/credit-card-fraud-report/

Appendix

Figure 1:

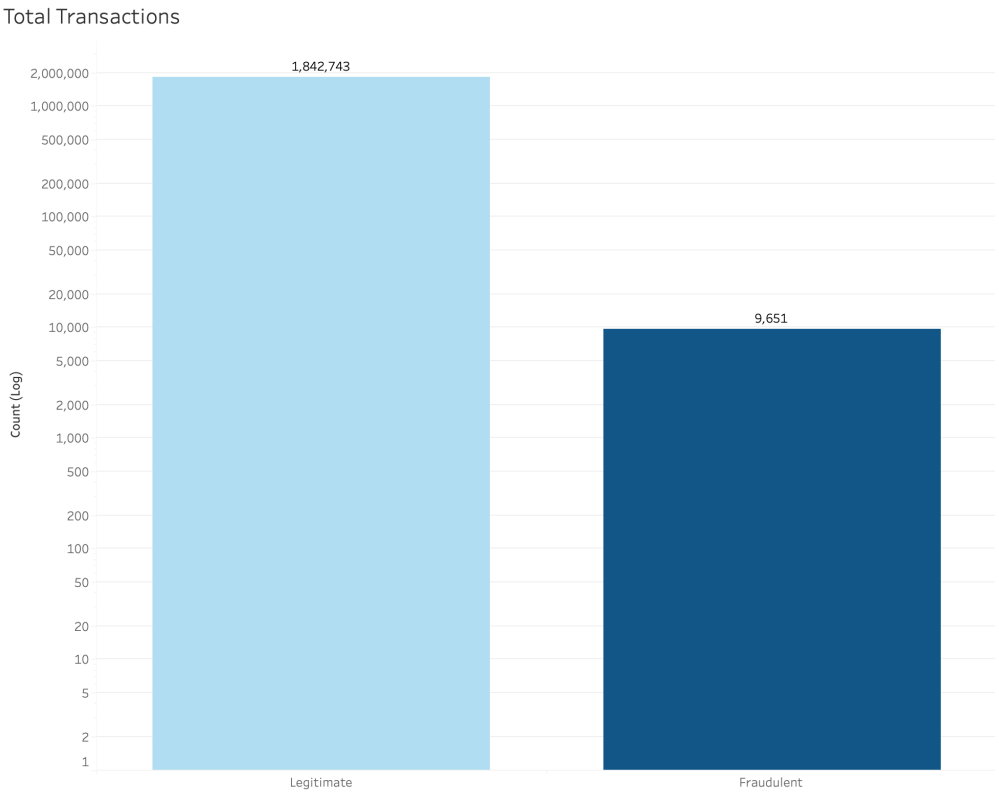


Figure 2A:

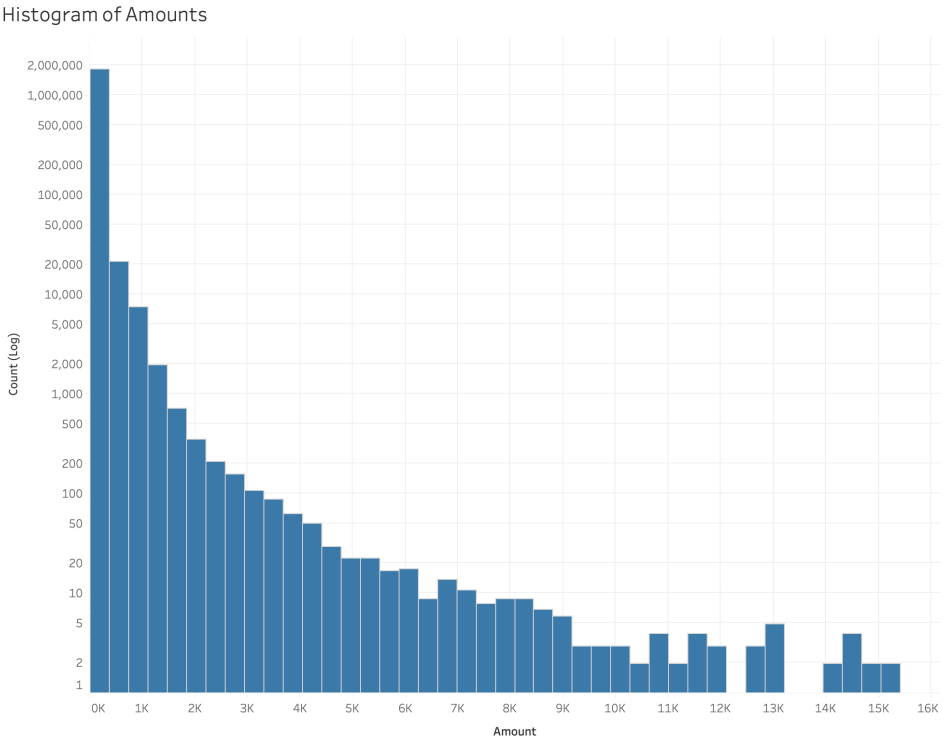


Figure 2B:

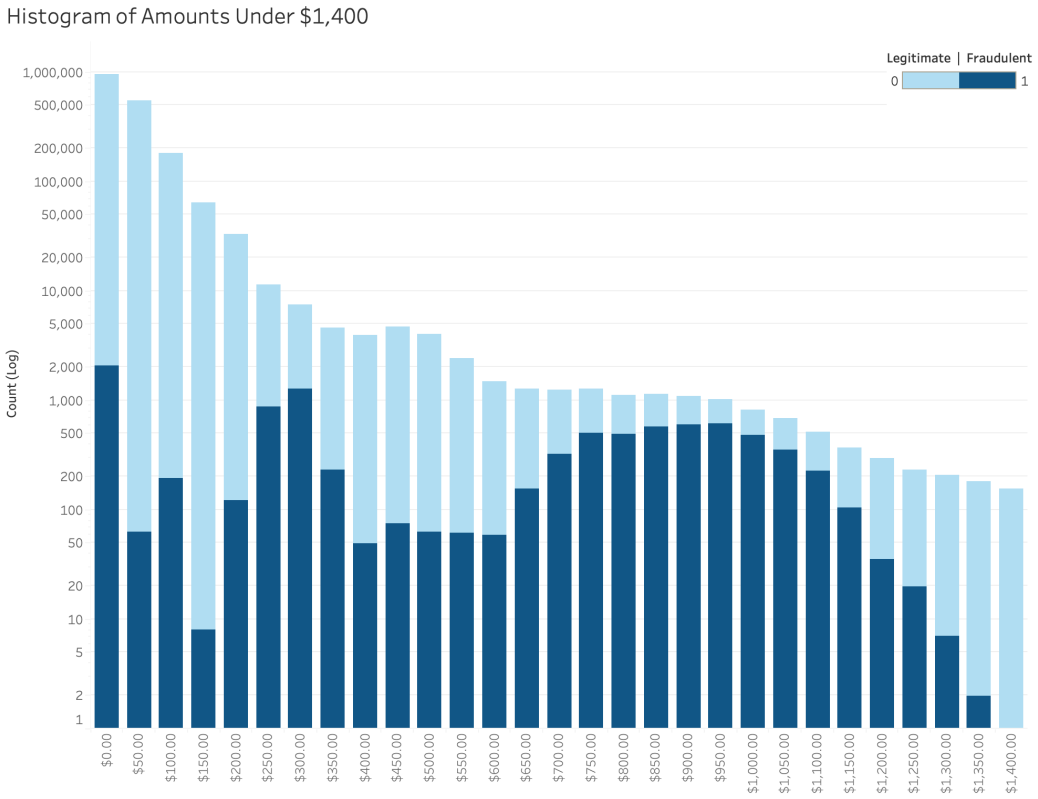


Figure 3:

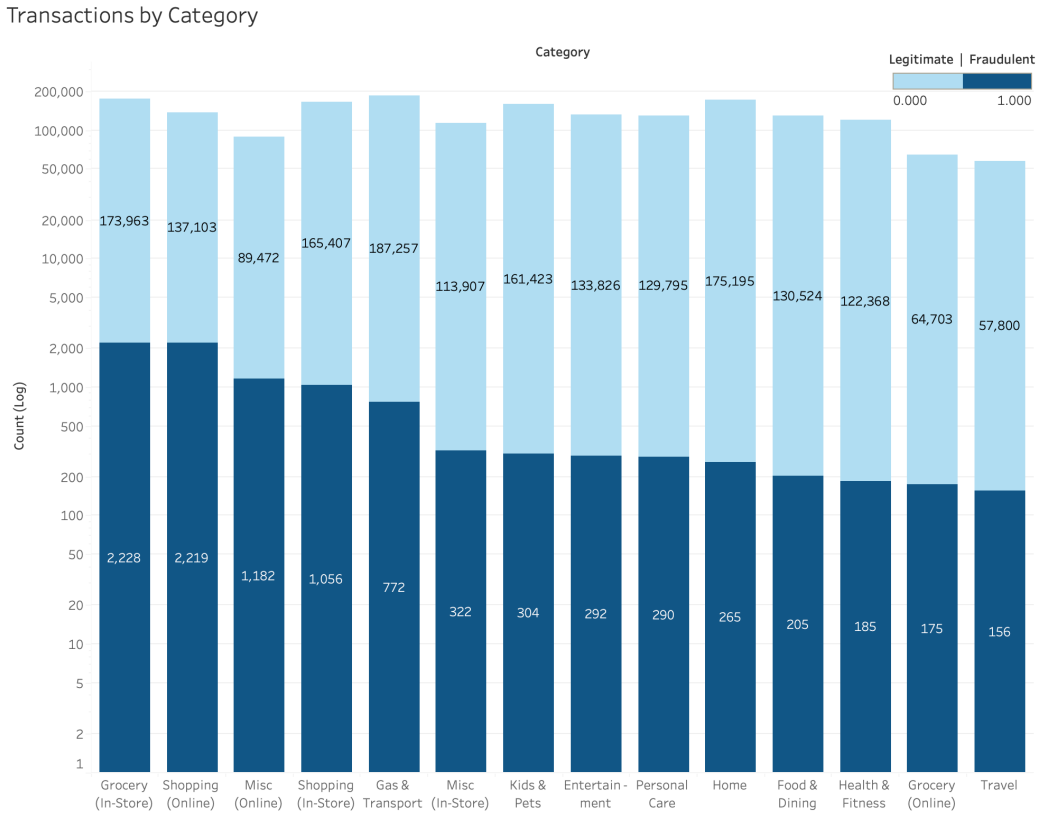


Figure 4A:

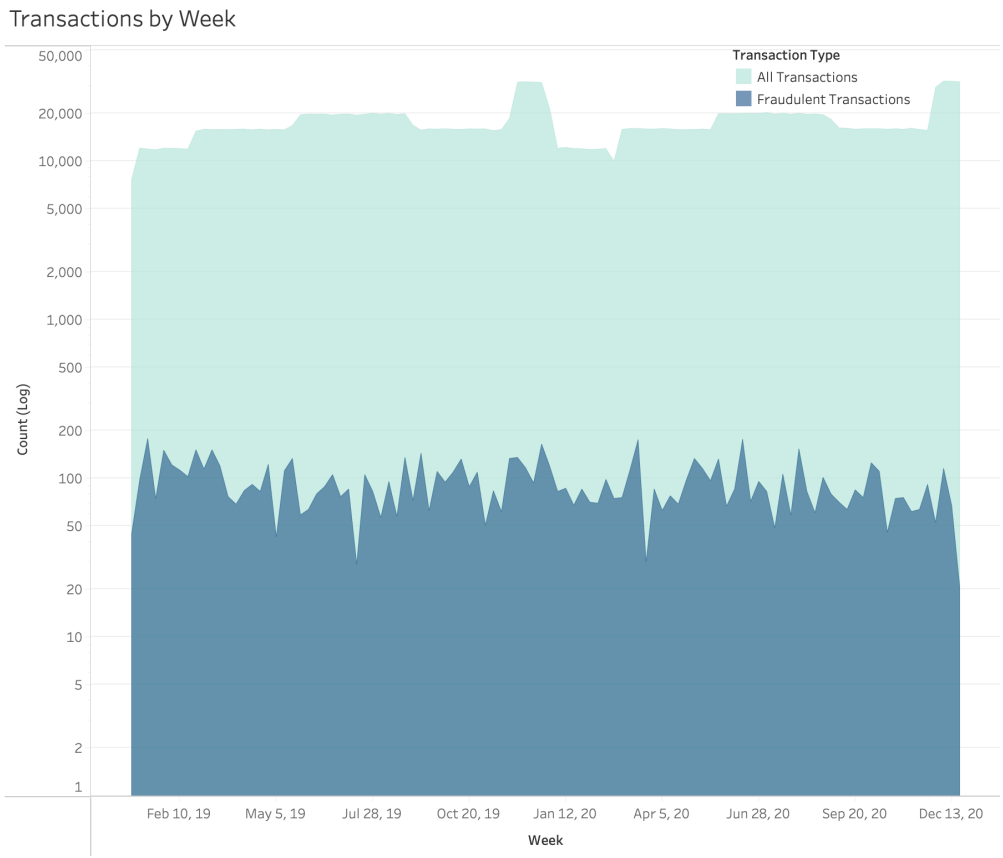


Figure 4B:

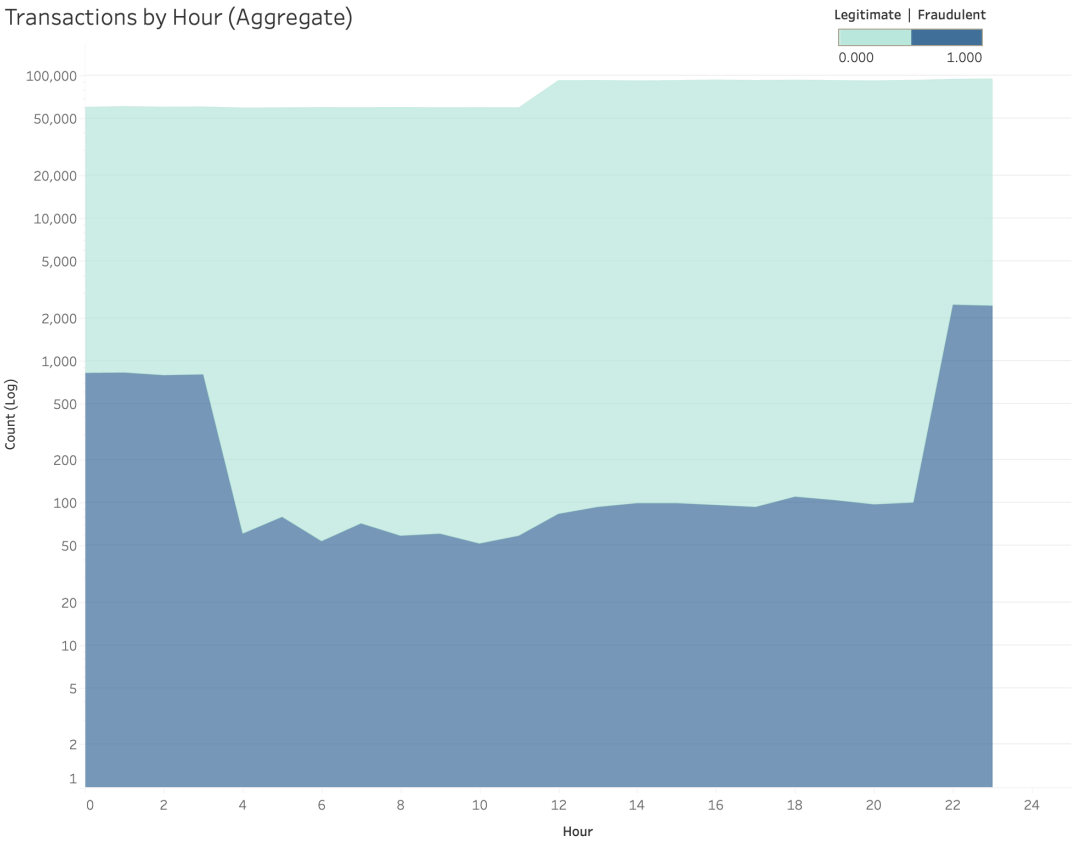


Figure 5:

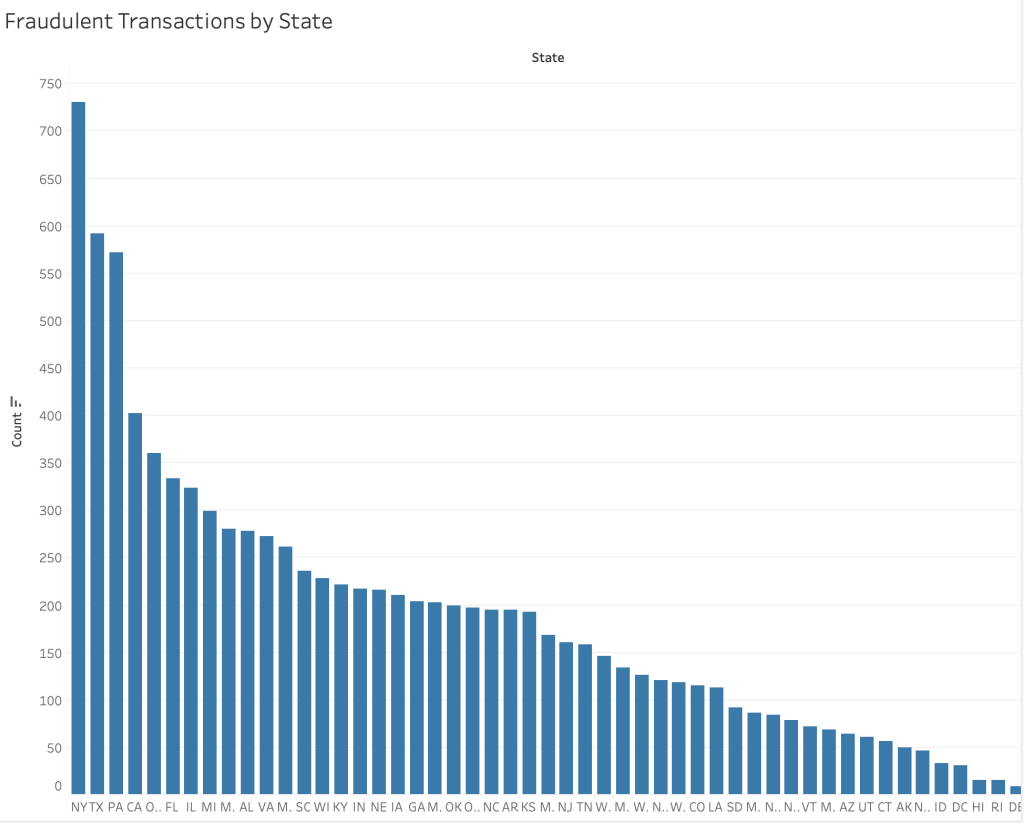


Figure 6:

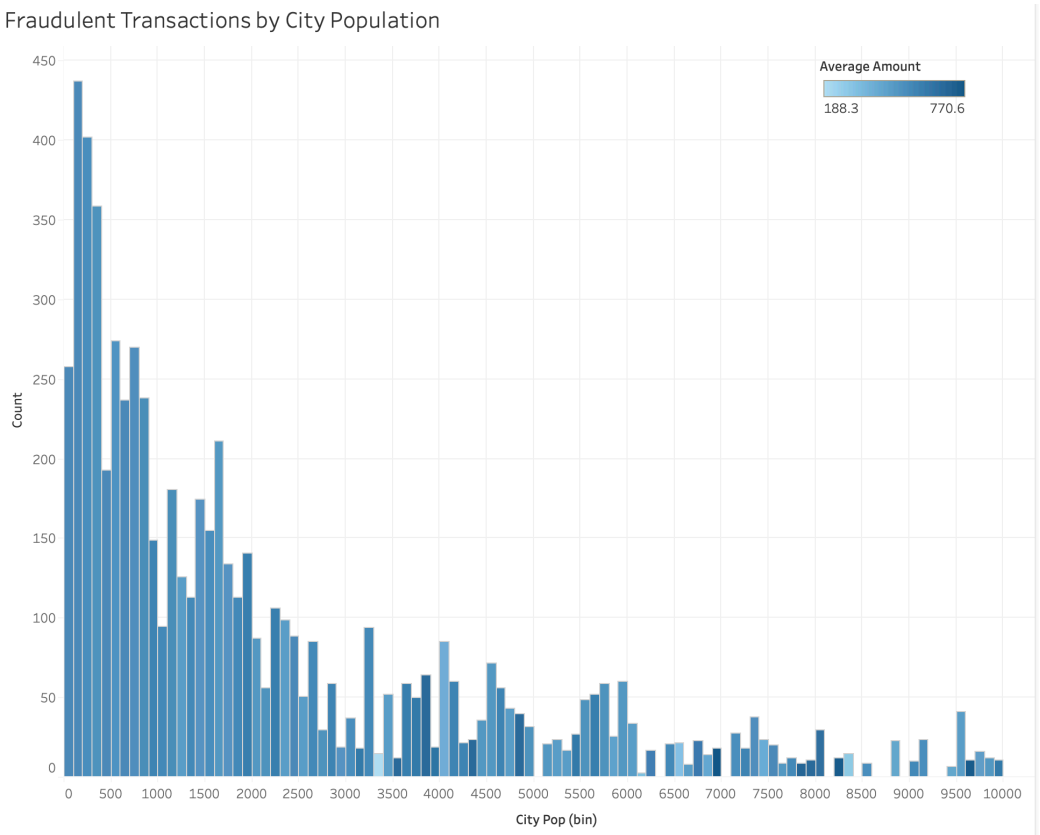


Figure 7:

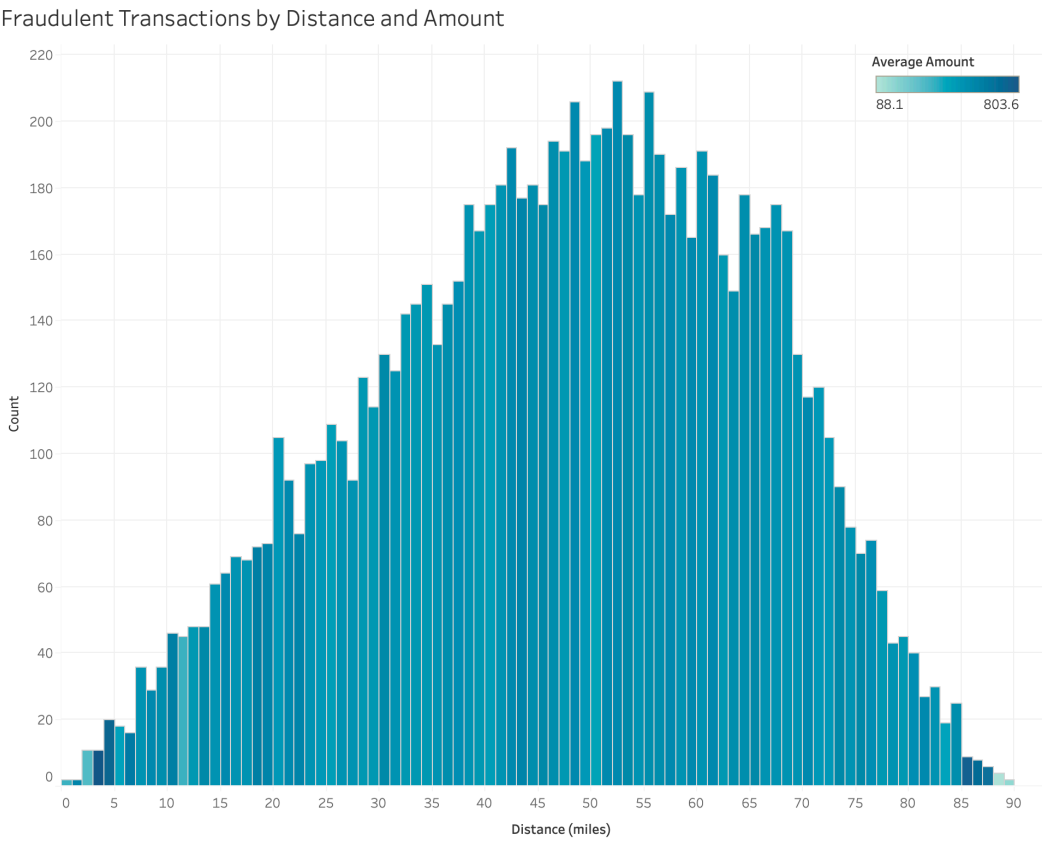


Table 1:

Feature	Is-fraud correlation coefficient	Is-fraud correlation coefficient (post-resampling)
Merchant	0.025	0.183
Category	0.033	0.219
Amount	0.209	0.619
Gender	-0.006	-0.177
City	0.028	0.176
State	0.002	0.008
Latitude	0.003	0.02
Longitude	0.001	0.012
City population	0.0003	0.002
Job title	0.015	0.105
Merchant latitude	0.003	0.02
Merchant longitude	0.001	0.012

Feature	Is-fraud correlation coefficient	Is-fraud correlation coefficient (post-resampling)
Distance	0.0004	0.005
Month	-0.016	-0.155
Hour	0.013	0.052
Week	-0.017	-0.128
Birth year	-0.011	-0.086

Table 2:

Feature	Logit Coefficients	SGD Coefficients	Random Forest Feature Importances
Merchant	0.447	0.116	0.0292
Category	-0.150	0.0	0.0924
Amount	2.931	2.508	0.6803
Gender	-0.143	-0.049	0.0154
City	0.288	0.093	0.0234
State	-0.031	0.0	0.0001
Latitude	0.087	0.0	0.0002
Longitude	0.421	0.0	0.0
City population	-0.005	0.0	0.0017
Job title	0.015	0.0	0.0018
Merchant latitude	-0.092	0.0	0.0
Merchant longitude	-0.419	0.0	0.0
Distance	0.013	0.0	0.0011
Month	-1.711	-0.096	0.0196
Hour	-0.026	0.0	0.1232
Week	1.413	0.0	0.0066
Birth year	-0.008	-0.012	0.0048

Table 3:

	Logit Regression	SGD Classifier	Random Forest	HGB Classifier
Accuracy	0.844	0.855	0.916	0.967
Precision	0.922	0.949	0.959	0.975
Recall	0.751	0.751	0.87	0.96
F1	0.828	0.838	0.912	0.967
AUC	0.892		0.964	0.996
R2	0.374	0.421	0.664	0.874

```
# import dependencies
import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

import sklearn
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
from sklearn.metrics import recall_score, f1_score, r2_score, roc_auc_score

import os

cwd = os.getcwd()
```

Feature Engineering & Cleaning

```
df = pd.read_csv(cwd + '/CreditFraud.csv')
```

```
from haversine import haversine

df['distance'] = 0

for i in range(len(df)):
    df['distance'][i] = haversine((df['lat'][i], df['long'][i]),
                                  (df['merch_lat'][i], df['merch_long'][i]),
                                  unit = 'mi')
```

```
from datetime import datetime

df['trans_date_trans_time'] = pd.to_datetime(df['trans_date_trans_time'])

df['month'] = df['trans_date_trans_time'].dt.month
df['hour'] = df['trans_date_trans_time'].dt.hour
df['week'] = df['trans_date_trans_time'].dt.isocalendar().week
df['week'] = df['week'].astype(int)

df['birth_year'] = df['dob'].str[:4]
df['birth_year'] = df['birth_year'].astype(int)

# Label encoding
df['category'] = pd.factorize(df['category'])[0]
df['gender'] = pd.factorize(df['gender'])[0]
df['city'] = pd.factorize(df['city'])[0]
df['state'] = pd.factorize(df['state'])[0]
df['job'] = pd.factorize(df['job'])[0]
df['merchant'] = pd.factorize(df['merchant'])[0]
```

```
df = df.drop(columns = ['cc_num', 'unix_time', 'first', 'last', 'street', 'dob',
                        'zip', 'trans_num', 'trans_date_trans_time'])
```

Resampling

SMOTE

```
: from imblearn.over_sampling import SMOTE

Xsamp = df.drop(columns='is_fraud')
ysamp = df['is_fraud']

smote = SMOTE()
X_resampled, y_resampled = smote.fit_resample(Xsamp, ysamp)

resampled_data = pd.concat([pd.DataFrame(X_resampled, columns=Xsamp.columns),
                            pd.Series(y_resampled, name='is_fraud')], axis=1)

resampled_data
```

Correlational Analysis

Before Resampling

```
corr_matrix = df.corr()

plt.figure(figsize=(8, 6))
plt.imshow(corr_matrix, cmap='coolwarm', interpolation='nearest')
plt.colorbar()
plt.xticks(np.arange(len(corr_matrix.columns)), corr_matrix.columns, rotation=45)
plt.yticks(np.arange(len(corr_matrix.columns)), corr_matrix.columns)
plt.title('Correlation Matrix')
plt.show()
```

```
for i in df.columns:
    correlation_coefficient = round(corr_matrix.loc['is_fraud', i], 5)

    print("is_fraud and " + i + f": {correlation_coefficient}")
```

After Resampling

```
corr_matrix_resamp = resampled_data.corr()

plt.figure(figsize=(8, 6))
plt.imshow(corr_matrix_resamp, cmap='coolwarm', interpolation='nearest')
plt.colorbar()
plt.xticks(np.arange(len(corr_matrix_resamp.columns)), corr_matrix_resamp.columns, rotation=45)
plt.yticks(np.arange(len(corr_matrix_resamp.columns)), corr_matrix_resamp.columns)
plt.title('Correlation Matrix')
plt.show()
```

```
for i in df.columns:
    correlation_coefficient = round(corr_matrix_resamp.loc['is_fraud', i], 3)

    print("is_fraud and " + i + f": {correlation_coefficient}")
```

Model Construction

```
# test train split
from sklearn.model_selection import train_test_split

X = resampled_data[resampled_data['category']==0]
X = X.drop(columns='is_fraud')
y = resampled_data[resampled_data['category']==0]['is_fraud']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

```
# rescaling
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
# helper function to grab the metrics for each model
def get_metrics(predictions):
    # tn,fp|fn,tp

    return(print(f"Confusion matrix:\n {confusion_matrix(y_test, predictions)}\n",
                 f"Accuracy: {round(accuracy_score(y_test, predictions), 3)}\n",
                 f"Precision: {round(precision_score(y_test, predictions), 3)}\n",
                 f"Recall: {round(recall_score(y_test, predictions), 3)}\n",
                 f"F1: {round(f1_score(y_test, predictions), 3)}\n",
                 f"R2: {round(r2_score(y_test, predictions), 3)}"))
```

Logistic Regression

```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression

param_grid = {'C': [0.1, 1, 10]}

grid_search = GridSearchCV(LogisticRegression(random_state=42), param_grid, cv=5)

grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_

print(best_params)
```

```
lr_model = LogisticRegression(C=0.1, random_state=42)

lr_model.fit(X_train, y_train)

lr_model.coef_
```

```
y_preds = lr_model.predict(X_test)

get_metrics(y_preds)
```

SGD Classifier

```
from sklearn.linear_model import SGDClassifier

param_grid = {'loss': ['hinge', 'modified_huber'], 'penalty': ['l1', 'l2'], 'alpha': [0.0001, 0.001, 0.01]}

grid_search = GridSearchCV(SGDClassifier(random_state=42), param_grid, cv=5)

grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_

print(best_params)
```

```
sgd_clf = SGDClassifier(penalty='l1', alpha=0.01, random_state=42)

sgd_clf.fit(X_train, y_train)

sgd_clf.coef_
```

```
y_preds_sgd = sgd_clf.predict(X_test)

get_metrics(y_preds_sgd)
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier

rnd_clf = RandomForestClassifier(n_estimators=200, max_leaf_nodes=16,
                                 n_jobs=-1, random_state=42)

rnd_clf.fit(X_train, y_train)
```

```
y_preds_rf = rnd_clf.predict(X_test)

y_prob = rnd_clf.predict_proba(X_test)

fpr, tpr, thresholds = metrics.roc_curve(y_test, y_prob[:,1], pos_label=1, drop_intermediate=False)

get_metrics(y_preds_rf)

print(f" AUC: {round(metrics.auc(fpr, tpr), 3)}")
```

```
for score, name in zip(rnd_clf.feature_importances_, X.columns):
    print(round(score, 4), name)
```

HGB Classifier

```
from sklearn.ensemble import HistGradientBoostingClassifier

hgb_clf = HistGradientBoostingClassifier(random_state=42)

hgb_clf.fit(X_train, y_train)
```

```
y_preds_hgb = hgb_clf.predict(X_test)

y_prob = hgb_clf.predict_proba(X_test)

fpr, tpr, thresholds = metrics.roc_curve(y_test, y_prob[:,1], pos_label=1, drop_intermediate=False)

get_metrics(y_preds_hgb)

print(f" AUC: {round(metrics.auc(fpr, tpr), 3)}")
```