Machine Learning Applications to Combat Credit Card Fraud: An Analysis by Regularly Scheduled Programming

Will Greenwood   
University of Tennessee, Knoxville  
wgreenwo@vols.utk.edu  
  
Maddie Gross  
University of Tennessee, Knoxville  
mgross10@vols.utk.edu  
  
Cinzia Pacione  
University of Tennessee, Knoxville  
cpacione@vols.utk.edu  
  
Ethan Weathers  
University of Tennessee, Knoxville  
dweathe5@vols.utk.edu

# Abstract

# *Credit card fraud is an increasing threat to financial security, requiring robust methods to detect and prevent fraudulent transactions. In this study, we analyze a comprehensive dataset containing various transaction attributes, including time, location, category, and monetary value, to identify patterns associated with fraudulent behavior. We perform extensive data cleaning and exploratory data analysis (EDA) to visualize trends in fraudulent transactions across different features. Our analysis highlights key fraud indicators, such as transaction time, merchant category, and age demographics of fraud victims.*

# Introduction

One of the most important parts about having financial peace of mind is to make sure your finances are safe. Unfortunately, credit card fraud is a major threat to that financial security, and is happening more and more frequently by the day. Our goal was to look into cases of credit card fraud, using predictable trends and collections of fraudulent data to try and come up with solutions to try and reduce the amount of potential credit card fraud cases.

# Dataset

The dataset we used for this project has a plethora of different categories for us to attempt to pull data from. For every transaction, it allows us to track the following for every purchase recorded in the dataset:

* Transaction date and time
* Merchant name
* Category of merchant
* Transaction amount
* Transaction number
* City of the credit card holder
* Population of credit card holder’s city
* State of the credit card holder
* Latitude/Longitude of the credit card holder
* Occupation of credit card holder
* Credit card holder date of birth
* Was the transaction fraud? (Y/N)

Data Cleaning

Using this extensive dataset, we looked at what the best way to clean up the data, and tailor it to our needs. We started by combing through it and seeing if there are any null data values in the dataset, which, to our pleasant surprise, there ended up being none. After that, we wanted to make 1 change to the categorization of the data set, and we combined the date and time of a transaction into a single column, We felt that this would be a more efficient way to look at the potential trends of fraudulent data going forward.

Exploratory Data Analysis

Once we felt that we had tuned and cleaned our dataset as needed, we opted to make different kinds of charts, graphs, and other visual aids to help us easily digest some of the highlights of the dataset. Below are some of the data analysis graphics we created:

Location Data

A close-up of a graph

AI-generated content may be incorrect.

A map of the united states

AI-generated content may be incorrect.

The folium library is first installed and imported, its plugins. The average longitude and latitude are calculated across all transactions to determine where to center the map, and Folium initializes a map object *m* centered at this location. To plot the transaction markers, the code iterates through each row of the datagrame and assigned red to fraudulent transactions and blue to non-fraudulent ones. An interactive popup window then displays the transaction category and amount when a marker is clicked. Then the marker creation and customization defines certain features like the color, opacity, radius, etc. The map is then exported to an html file that can be opened in any browser.

Time Data

A green and red graph

AI-generated content may be incorrect.

*[4] GeeksforGeeks was used for assistance in creating this histogram.*

A graph of a number of blue and red bars

AI-generated content may be incorrect.

[3] ChatGPT: to make the two fraud categories stack on each other

These 2 figures simply demonstrate how many fraud cases occur with our time-based data from our samples in the dataset. It shows that during the evening, coming off the heels of a weekend, and during the first quarter of the year, the chance of fraudulent data might be higher.

Category Data

A graph of sales and purchase

AI-generated content may be incorrect.

A pie chart of a number of items

AI-generated content may be incorrect.

These 2 figures simply demonstrate what kind of purchase might likely be a fraudulent purchase. It seems that grocery stores, and online shopping have the highest likelihood of being flagged as fraud.

Monetary Data

A graph with numbers and a line

AI-generated content may be incorrect.

A diagram of a graph

AI-generated content may be incorrect.

A graph of blue and red dots

AI-generated content may be incorrect.

These binary graphs display how much the different recorded transactions cost. With fraudulent ones being on the low end of around <$1500.

Age Breakdown

A pie chart with numbers and a number of people

AI-generated content may be incorrect.

One of the other things we wanted to see with our dataset was the age demographic breakdown of potential card holders who were victims of fraud. With the most vulnerable population being 31-35 year olds as well as 51-55 year olds.

Another odd thing we noticed was that, with this particular dataset, less than 1% of all transactions are considered fraudulent. This is a very interesting thing to note, and will change the way we measure the success of our model moving forward.

# Baseline Solution

Based off what we noticed with our initial views of patterns within our dataset, we looked around at others who has also dipped their toes in experimenting with this data set as well. We found one model titled “*Credit Card Fraud Analysis”* by Robin Singh Rawat[1], which at initial glance, seemed to fall in line with a lot of our goals and ambitions for this project. However, once we started playing around with it, we were pleasantly surprised. The performance of this model is extremely good, as evident from the results we gleaned from the SKLearn classification report produced by the model. We ended using the following features:

* Date and Time
* Card Holder Date of Birth
* Transaction Number
* Merchant Name
* City and State of Transaction

When identifying non-fraudulent transactions, the model has a recall of 1, and for fraudulent cases, a recall of 0.97. This means the model identifies non-fraudulent transactions with perfect accuracy, and only misses fraudulent transactions 3% of the time. Additionally, the highest accuracy the model reaches was .996, however, this is a misleading metric due to this dataset being highly skewed towards non-fraudulent transactions as previously discussed. The optimized maximum tree depth was 20, the minimum sample split was 10, and the minimum estimators was 200. Furthermore, we concluded through testing that the model performs best when using the full dataset as opposed to using randomly selected data points (indicated by the ‘bootstrap = false’ flag in the attached code)

# Proposed Extension

The pleasant surprise of the data set’s performance changed a lot of our game plan. We did still want to do a few other things to evaluate the model’s performance. We opted to create a confusion matrix, as well as a precision recall curve in order to reaffirm that the model was actually doing as good as we think it is. The confusion matrix was done just because we all thought it would help in visualizing the data without having to look at an extremely long list of output data, and the recall curve was done because we knew it would fit for an extremely skewed dataset like this one.

A green and red squares with numbers

AI-generated content may be incorrect.

*[2] Confusion Matrix for the implemented model. Assistance from W3Schools was used in creation*

The matrix shows the model’s high accuracy with the predicted data we wanted. We were all extremely shocked that there were 0 false positives, and the false negatives being less than 15 out of more than 14000 entries in the data set is incredibly good for the model we are using for this project. This will help us moving forward when we want to consider making further changes that we have a strong foundation to build off of.

A graph of a graph

AI-generated content may be incorrect.

*[5] Precision Recall Curve for the model. W3Schools was used for assistance in its creation.*

While it was meant to be supplementary with our confusion matrix, the graph still gives us a good sense that our data is being working well with the model and is giving us accurate predictions with the data that we are testing with it.

What we are thinking of doing next is implementing one-hot encoding or feature engineering. With One-Hot encoding, we understand that the random forest approach of the chosen model was really effective, but there could be a chance for a small performance improvement if applied to a category with a small number of possible values. This is extremely important as while the model is good, it was annoying for all of us when we wanted to test something with it and had to run the model because it took around 7-7.5 minutes to run to completion. However, we will soon likely be implementing some form of feature engineering. We saw a lot of other models that had some form of this, and they seemingly had a significantly lower runtime, and could not only improve runtime performance, it might be able to pick up on fraud more accurately if we were to mess around with a different dataset.

Additionally, we wanted to see how applications of clustering were appropriate, if at all to the model. We threw out the ideas of also adding a K-means or DBSCAN algorithm to further enhance the model, but that is something we are wanting to try for the final version if the project.

# Distribution of Work

Will:

* Data cleaning
  + Verify correct data types/handling missing values
  + \*Scaling and normalization
  + Parse dates
  + Lookup tools (labels dictionary)
* Bar chart
  + Fraud Index by State
* Pie charts
* Fraud by Age
  + Most Common Categories of Fraudulent Transactions

Maddie:

* Feature 1: Transaction Time
  + Formatted transaction time to exclude the date.
  + Grouped transactions by the hour they occurred.
  + Categorized transactions as Fraud or Not Fraud and counted occurrences.
  + Created a bar chart:
    - Left y-axis: Number of fraudulent and non-fraudulent transactions.
    - Right y-axis: Percentage of fraudulent transactions within each hour.
* Feature 2: Transaction Category
* Grouped transactions by category.
* Created Fraud and Not Fraud attributes.
* Used the same bar chart format as above to display the data.
* Feature 3: Transaction Amount
  + Explored different ways to best visualize the data.
  + Created multiple charts to compare effectiveness.
  + Ensured each transaction was represented while keeping granularity.
* Used different plot types:
  + Scatter plot
  + Violin plot
  + Strip plot
  + Distinguished fraudulent and non-fraudulent transactions in the plots.

Cinzia:

* Library & Setup:
  + - Installed and imported the Folium library and its plugins.
    - Used Pandas to clean transaction data.
* Map Initialization:
  + - Calculated the average latitude and longitude of all transactions to center the map.
    - Initialized a Folium map object at this location.
* Marker Creation:
  + - Iterated through each row of the DataFrame to plot transactions.
    - Assigned red for fraudulent transactions and blue for valid transactions.
    - Customized marker color, opacity, and radius to emphasize spatial clustering.
    - Interactivity:
    - Added a popup window for each marker displaying the transaction category and amount.
* Exporting the Map:
  + Saved the interactive map as an HTML file, viewable in any browser.

Ethan:

* Histograms:
  + Fraud Cases by Month
  + Fraud Cases by Week
* Wrote Project Report
* Confusion Matrix
* Precision Recall Curve

##### References

1. robinsinghrawat, “Credit\_Card\_fraud\_analysis,” *Kaggle.com*, Aug. 23, 2024. https://www.kaggle.com/code/robinsinghrawat/credit-card-fraud-analysis (accessed Mar. 30, 2025).
2. “Python Machine Learning - Confusion Matrix,” *www.w3schools.com*. https://www.w3schools.com/python/python\_ml\_confusion\_matrix.asp
3. OpenAI, "ChatGPT," OpenAI, Mar. 14, 2025. [Online]. Available: <https://openai.com/chatgpt>
4. “Matplotlib Histograms,” *www.w3schools.com*. https://www.w3schools.com/python/matplotlib\_histograms.asp
5. “Precision-Recall Curve | ML,” *GeeksforGeeks*, Jul. 19, 2019. https://www.geeksforgeeks.org/precision-recall-curve-ml/