Applied Analytics Project - Week 2 Assignment
Data Ingestion and Exploration

Brady Levenson
ADAN 8888 - Applied Analytics Project
Professor Nurtekin Savas
September 15, 2024

Revisitation of Problem Statement for Motivation to Analyze and Engineer Features

It is known that fraudulent financial transactions occur on a regular basis, but it is difficult to distinguish between them and legitimate financial transactions simply by viewing the transactions. The inability to correctly classify fraudulent financial transactions can promote uncontrolled and untraceable levels of financial losses for banks, investment firms, corporations, and numerous other entities. We are given a dataset of financial transactions and their unique properties. We will analyze this dataset using various machine learning techniques to determine effective measures to identify historical fraudulent transactions and prevent their occurrence in the future.

What is the Problem?

To efficiently unpack the premise of my problem statement, the problem is fraudulent financial transactions are oftentimes difficult to identify simply by viewing their surrounding variables and statistics with the naked eye, so it is pivotal that features are strategically engineered and considered in the scope of machine learning models to capture the key properties of the financial transactions in question and to make data-informed conclusions based off of those properties.

How is this the Right Dataset?

The Kaggle Financial Anomaly Dataset is a versatile basis for understanding the trends of diverse financial markets through an examination of distinct features and their associations with one another. This dataset contains a collection of financial transactions with their respective timestamps, transaction ID's, account ID's. amounts, merchants, transaction types, and locations, providing numerous unique opportunities to construct robust machine learning models and determine whether certain financial transactions are potentially fraudulent from multiple different perspectives.

How will it Help Solve the Problem?

It is not very possible for an individual to determine that a financial transaction is fraudulent by looking at its transaction amount alone. Keeping this in mind, the Kaggle Financial Anomaly Dataset will help solve the problem of efficiently identifying when a transaction should be flagged as potentially fraudulent by allowing for the creation of anomaly detection models that can capture the overarching trends of the dataset's inherent variables and manually-created categorical interaction variables and identifying instances of transactions that do not align with the expectations set forth by the constructed models. In order for the dataset to be incorporated in this project as effectively

as possible, it is essential that it is first cleaned, having any null values and irrelevant features handled. It is then equally essential that additional features are created to gain a comprehensive understanding of the trends of variables stemming from specific subcategories of the dataset. By conducting a dataset cleaning and feature engineering process in this manner, the dataset will be prepared for a machine learning model to interpret.

Analysis of the Dataset:

The dataset contains seven potential predictor variables by default, namely Timestamp, TransactionID, AccountID, Amount, Merchant, TransactionType, and Location. Each of these variables serves a purpose in capturing specific properties of the dataset:

Timestamp: This variable can assist with identifying temporal aspects of the dataset. This variable will be parsed into subcategories in an effort to classify our data points by specific time intervals they occur in.

TransactionID: This variable represents a categorization of specific transactions being made. It may be helpful to consider the presence of anomaly statistics within the scope of certain transaction ID's.

AccountID: Account ID, as it nominally appears, is the identification of the account that a certain transaction originated from. We can use this variable as a resource to analyze if suspicious transactions are occurring within specific accounts.

Amount: This is the raw dollar amount of each transaction. We will create subcategories for this variable to measure the distribution of dollar amounts of the transactions in our dataset.

Merchant: Merchant represents the specific one of ten merchants the transaction in our dataset involves. The Merchant variable can assist with the type of business the transaction occurred with respect to.

TransactionType: The transaction type variable takes on values of either Deposit, Withdrawal, or Transfer. It can be used in the process of identifying trends in specific types of transactions.

Location: The location variable will support the conduction of analyses into how financial market trends present themselves within the scope of specific geographies.

Stemming from the original predictor variables, we will create a series of categorical interaction variables that will allow for deeper analyses of what factors are most prevalent and deserve significant consideration when determining the presence of anomaly financial transactions within a large dataset. We will go into the specifics of the aforementioned categorical interaction variables shortly.

Cleaning the Dataset:

The first step to cleaning our dataset was to create a pandas DataFrame for the data, which was previously being stored in a csv file. Then, we listed the core information of the dataset the better understand the properties of each of its inherent features, e.g. class, RangeIndex, columns, non-null count, data type, and memory usage information. From there, we identified the number of null occurrences within the dataset. In this case, each transaction in the dataset already had values for each of their predictor variables, but the DataFrame contained 481 empty rows. I decided it would be appropriate to remove these irrelevant rows, so I did so in the following step of the dataset cleaning process. Typically we would remove any irrelevant features at this point in the data cleaning process. I elected to hold onto all of the original features of the dataset for now considering that the dataset itself does not naturally contain an abundant amount of predictor variables. As we begin to undergo the modeling steps of this experimentation process, I will revisit the data cleaning step and ensure that only essential variables are being incorporated in our respective models.

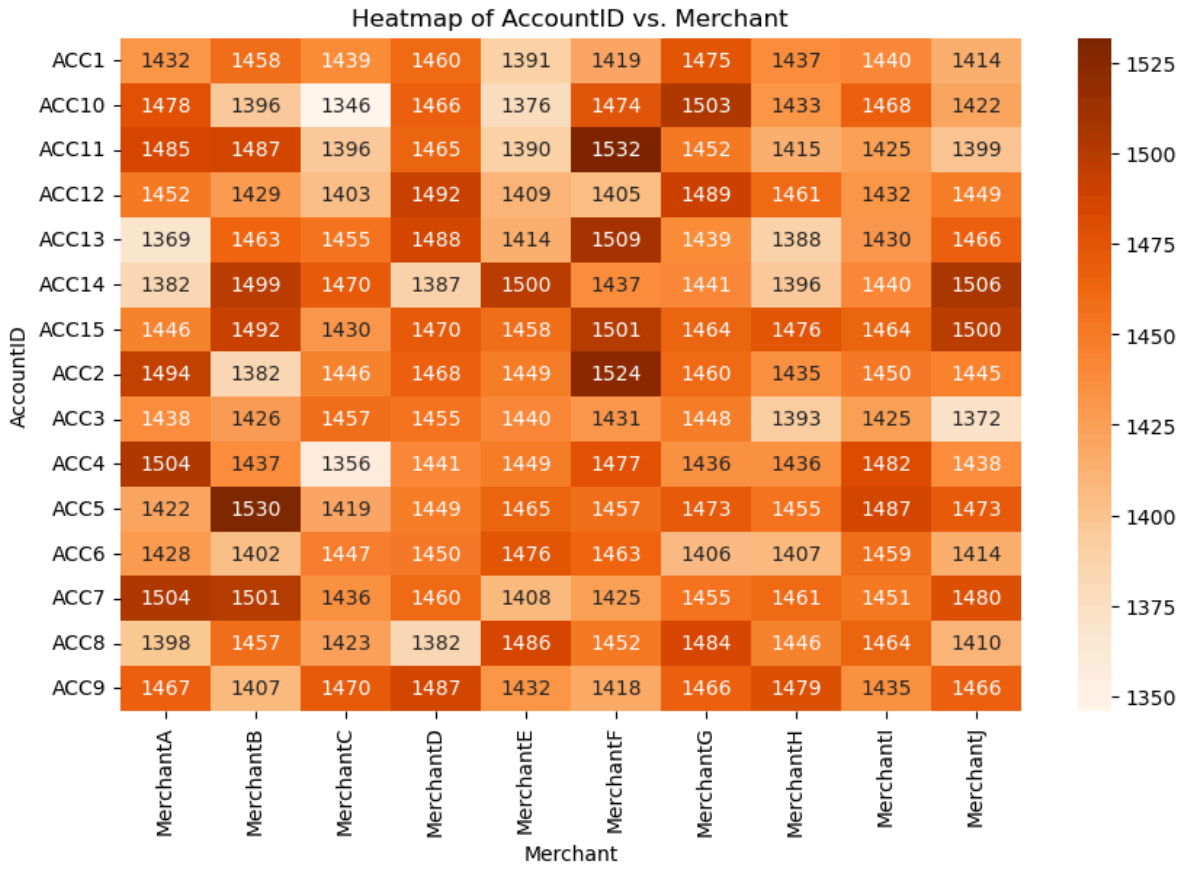
Why Should We Engineer New Features?

While it is possible to learn a lot about our given dataset through a thorough examination of the variables that are naturally present within it, our analysis potential is limited by an inability to isolate patterns inherent to specific combinations of features outside of scope of all of the datasets features being considered together. In order to conduct a deeper analysis on this dataset and recognize trends and patterns pertinent to underlying subcategories of the dataset, we will engineer new features that represent categorical variable combinations that have the greatest potential for identifying instances of anomaly financial transactions. Below is the set of newly constructed features for this project and their respective purposes in the modeling process. Note that due to the vast amount of combinations of individual feature variables that can possibly be created, this list is

not exhaustive of all possible combinations. If we continue with this project and determine that other feature combinations would provide a greater benefit than the ones listed below, we will revisit this section and construct more new features accordingly.

AccountID/ Merchant: A combination of Account ID and Merchant that will allow for an analysis of certain accounts having biases towards specific industries.

Heat Map of AccountID vs. Merchant:



AccountID/ TransactionID: A combination of Account ID and Transaction ID that will allow us to visualize certain accounts having biases towards making specific transactions.

AccountID/ Merchant/ TransactionID: A combination of all three of the aforementioned variables that will allow for a deeper dive into bias and anomaly detection.

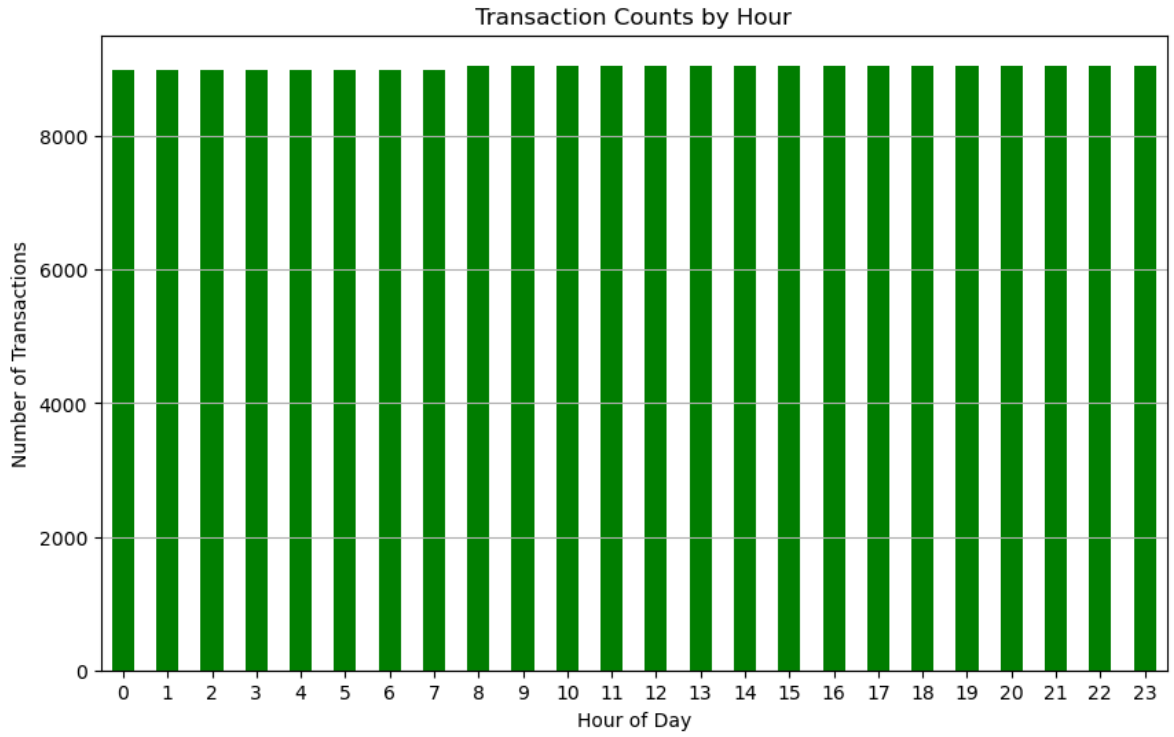
TransactionType/ Merchant: A combination of Transaction Type and Merchant to determine if there are specific types of transactions being made more or less often in specific industries.

Location/ TransactionType: A combination of Location and Transaction Type that will help to determine if specific types of transactions are being made more or less often across different locations

Merchant/ Location: A combination of Merchant and Location that will help to determine if transactions are being made commonly or uncommonly in various combinations of business industries and locations.

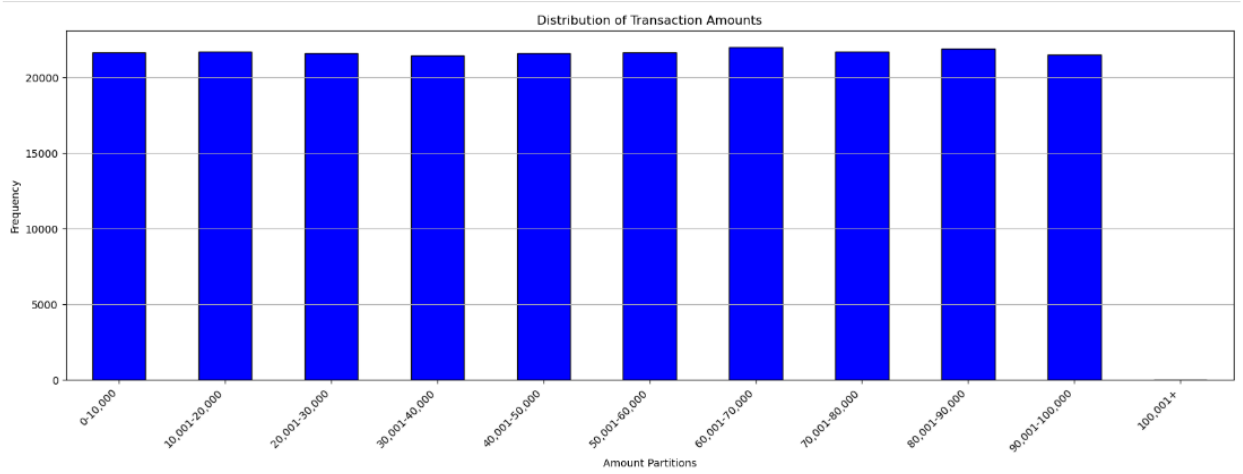
Time-Based Features - Minute, Hour, Day, Month: Each of these time-based features will serve to categorize financial transactions by unique times of the day, days they occurred on, or months they occurred in. These new variables may assist with identifying temporal trends within the financial transaction dataset.

Bar Plot for Transaction Counts by Hour:



Amount Partitions: This feature created $10,000 baskets to categorize the transaction amounts more broadly. We will attempt to determine if certain transaction amounts are more likely to contain anomaly data.

Bar Plot of Number of Transactions per Amount Partition (Note that the 100,001 basket contains only 14 transactions, which makes it difficult to see in this visual):



Sample of DataFrame with New Features:

Timestamp	TransactionID	AccountID	Amount	Merchant	TransactionType	Location	AccountID/Merchant	AccountID/TransactionID	AccountID/Merchant/TransactionID
2023-01-01 08:00:00	TXN1127	ACC4	95071.92	MerchantH	Purchase	Tokyo	ACC4_MerchantH	ACC4_TXN1127	ACC4_MerchantH_TXN1127
2023-01-01 08:01:00	TXN1639	ACC10	15607.89	MerchantH	Purchase	London	ACC10_MerchantH	ACC10_TXN1639	ACC10_MerchantH_TXN1639
2023-01-01 08:02:00	TXN872	ACC8	65092.34	MerchantE	Withdrawal	London	ACC8_MerchantE	ACC8_TXN872	ACC8_MerchantE_TXN872
2023-01-01 08:03:00	TXN1438	ACC6	87.87	MerchantE	Purchase	London	ACC6_MerchantE	ACC6_TXN1438	ACC6_MerchantE_TXN1438
2023-01-01 08:04:00	TXN1338	ACC6	716.56	MerchantI	Purchase	Los Angeles	ACC6_MerchantI	ACC6_TXN1338	ACC6_MerchantI_TXN1338
2023-01-01 08:05:00	TXN1083	ACC15	13957.99	MerchantC	Transfer	London	ACC15_MerchantC	ACC15_TXN1083	ACC15_MerchantC_TXN1083
2023-01-01 08:06:00	TXN832	ACC9	4654.58	MerchantC	Transfer	Tokyo	ACC9_MerchantC	ACC9_TXN832	ACC9_MerchantC_TXN832
2023-01-01 08:07:00	TXN841	ACC7	1336.36	MerchantI	Withdrawal	San Francisco	ACC7_MerchantI	ACC7_TXN841	ACC7_MerchantI_TXN841
2023-01-01 08:08:00	TXN777	ACC10	9776.23	MerchantD	Transfer	London	ACC10_MerchantD	ACC10_TXN777	ACC10_MerchantD_TXN777
2023-01-01 08:09:00	TXN1479	ACC12	49522.74	MerchantC	Withdrawal	New York	ACC12_MerchantC	ACC12_TXN1479	ACC12_MerchantC_TXN1479

TransactionType/Merchant	Location/TransactionType	Merchant/Location	Minute	Hour	Month	Amount_Partitions	Day
Purchase_MerchantH	Tokyo_Purchase	MerchantH_Tokyo	0	8	1	90,001-100,000	6
Purchase_MerchantH	London_Purchase	MerchantH_London	1	8	1	10,001-20,000	6
Withdrawal_MerchantE	London_Withdrawal	MerchantE_London	2	8	1	60,001-70,000	6
Purchase_MerchantE	London_Purchase	MerchantE_London	3	8	1	0-10,000	6
Purchase_MerchantI	Los Angeles_Purchase	MerchantI_Los Angeles	4	8	1	0-10,000	6
Transfer_MerchantC	London_Transfer	MerchantC_London	5	8	1	10,001-20,000	6
Transfer_MerchantC	Tokyo_Transfer	MerchantC_Tokyo	6	8	1	0-10,000	6
Withdrawal_MerchantI	San Francisco_Withdrawal	MerchantI_San Francisco	7	8	1	0-10,000	6
Transfer_MerchantD	London_Transfer	MerchantD_London	8	8	1	0-10,000	6
Withdrawal_MerchantC	New York_Withdrawal	MerchantC_New York	9	8	1	40,001-50,000	6

Each of these new features have been engineered with the intent of aiding us in our problem solving process. I expect that at least some of these features will provide us with trends that will lead us to forming accurate conclusions and solutions to our problem, but I also am practical in realizing that this is not a guarantee. The limitations present in these new categorizations are primarily that they are not inclusive of all of the possible combinations of variables in this dataset, so it is possible that we are missing key interaction variables that would most effectively solve this problem. As a backup plan, as mentioned previously, this section will be revisited and we will swiftly conduct further feature engineering to manipulate this dataset in a manner that is most advantageous to us.

What is the Target Variable and Why?

The target variable for this project is whether or not a financial transaction should be flagged as potentially fraudulent in real-time. More specifically, this is modeled through a binary classification of either:

Yes, this transaction is suspected of being fraudulent

Or

No, this transaction is not suspected of being fraudulent.

The primary objective of conducting this in-depth machine learning analysis is to create versatile algorithms that support the detection of anomalies in financial transaction data. The algorithms created in this project through a rigorous training, validating, and testing process are intended for utilization in real-time financial transaction anomaly detection to prevent the processing of fraudulent financial transactions and ultimately lead to the saving/ making of significant sums of money for real financial institutions. As calculated in last week's report, this modeling project has a roughly estimated annual total potential economic value of $6 billion, so financial transaction anomaly detection is a target variable that has the potential to reap dividends globally to both businesses facilitating financial transactions and each of their respective clients.

What are the Predictors and Why?

We have discussed seven predictors present inherently within the given dataset (Timestamp, TransactionID, AccountID, Amount, Merchant, TransactionType, and Location) as well as eleven new features that we have created (AccountID/ Merchant, AccountID/ TransactionID, AccountID/ Merchant/ TransactionID, TransactionType/ Merchant, Location/ TransactionType, Merchant/ Location, Amount Partitions, Minute, Hour, Day, and Month), for a total of eighteen predictor features [at this point in time]. These predictors were determined and selected following an estimated analysis of which features/ combinations of features/ time intervals would provide the greatest utility towards predicting whether a financial transaction should be suspected of being fraudulent. Through the consideration of many diverse and unique combinations of features, we can mitigate the presence of biases in financial transaction anomaly detection and create models that can accurately flag financial transactions as potentially being fraudulent.

Exploration of the Dataset:

Following some preliminary data analysis in Python, I was able to determine some key statistical properties of our dataset, which we can refer back to at any time to ensure that we are achieving sensible results through our modeling process.

Breakdown of important properties of our DataFrame:

```
<class 'pandas.core.frame.DataFrame'>
Index: 216960 entries, 0 to 216959
Data columns (total 18 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Timestamp                         216960 non-null  datetime64[ns]
 1   TransactionID                     216960 non-null  object
 2   AccountID                         216960 non-null  object
 3   Amount                            216960 non-null  float64
 4   Merchant                          216960 non-null  object
 5   TransactionType                   216960 non-null  object
 6   Location                          216960 non-null  object
 7   AccountID/Merchant                216960 non-null  object
 8   AccountID/TransactionID           216960 non-null  object
 9   AccountID/Merchant/TransactionID  216960 non-null  object
 10  TransactionType/Merchant          216960 non-null  object
 11  Location/TransactionType          216960 non-null  object
 12  Merchant/Location                 216960 non-null  object
 13  Minute                            216960 non-null  int32
 14  Hour                              216960 non-null  int32
 15  Month                             216960 non-null  int32
 16  Amount_Partitions                 216960 non-null  category
 17  Day                               216960 non-null  int32
dtypes: category(1), datetime64[ns](1), float64(1), int32(4), object(11)
memory usage: 34.8+ MB
```

Number of Occurrences of Each Predictor Variable:

```
Number of unique Timestamp: 216960
Number of unique TransactionID: 1999
Number of unique AccountID: 15
Number of unique Amount: 214687
Number of unique Merchant: 10
Number of unique TransactionType: 3
Number of unique Location: 5

 Number of unique AccountID/Merchant: 150
 Number of unique AccountID/TransactionID: 29967
 Number of unique AccountID/Merchant/TransactionID: 154226
 Number of unique TransactionType/Merchant: 30
 Number of unique Location/TransactionType: 15
 Number of unique Merchant/Location: 50
 Number of unique Minute: 60
 Number of unique Hour: 24
 Number of unique Day: 7
 Number of unique Month: 5
 Number of unique Amount_Partitions: 11
```

Altogether, our DataFrame now contains 216,960 rows and 18 columns. I will attempt to use the whole dataset in some capacity in this modeling project, though if the training phase ends up taking an excessively long time due to the vast dataset, I will consider randomly sampling from the current dataset to obtain a smaller, unbiased dataset that could produce near-equivalent if not exactly equivalent results to those of the full dataset.

We are now prepared to conduct further exploratory data analysis as we begin to approach the modeling phase of the project.