Applied Analytics Project - Week 4 Assignment
Make Data Model Ready

Brady Levenson
ADAN 8888 - Applied Analytics Project
Professor Nurtekin Savas
September 29, 2024

Revisitation of Problem Statement for Motivation to Perform Exploratory Data Analysis

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.

Introduction

Last week I performed meaningful exploratory data analysis techniques to better understand the financial anomaly dataset that is under consideration for this project. Through this process, I searched for correlations between potentially related variables, learned more about the various distributions and spreads of those variables, and performed a split of the dataset to create training, validation and test sets. The EDA processes that were implemented in this project last week presented a great launching point for me to further clean this data this week and create datasets that are utilizable tools for modeling purposes. As part of this week's progression in the end-to-end modeling project, I made some adjustments to previous weeks' code to make the dataset more easily usable for this week, created a Date variable that will be used to bucketize information by specific dates of the year, transformed transaction amounts into a log form making them more usable for modeling purposes, removed columns that introduced unnecessary complexity to the dataset, performed one-hot encoding to represent categorical variables in a binary fashion, and created visualizations that will assist with the conceptual reasoning as to why certain steps have been taken thus far in the project. All data transformations that were executed this week were performed on the train, validation, and test sets respectively in a manner that would not allow for bias or data leakage in a model training process. I took an additional step to mask results from validation and test set cleaning to eliminate the potential for personal bias in the experimentation process of this project as well. After performing all of the aforementioned steps, my train, validation, and test sets are prepared for the final feature engineering stage of the project and for the modeling stage thereafter.

Minor Adjustments to Previous Weeks' Code

- Datetime object formatting
- The past few weeks I had not been reading my data file agnostically from my specific github directory. After correcting this error and rerunning my previous weeks' code cells, I received an error in my datetime formatting. I adjusted the formatting declarations of this cell as necessary and the code is once again operating smoothly.

- Uninterpretable and excessively large outputs
- Some of the exploratory data analysis techniques that I implemented last week were primarily intended to be a proof of concept for identifying what next steps I should take in this project. Keeping this in mind, some of my correlation analyses combined to produce over 200 pages of numerical outputs that are very difficult to read and do not add much value to the project as a whole. I have commented out a few of these code blocks so that we can still reflect on this process that was conducted, and if necessary, refer back to the outputs from last week's submission, without hampering the readability of this week's assignment.

Bucketization by Date

Prior to this week I had created time variables for hour of the day, day of the week, week of the month, and month of the year, but did not have a dedicated variable for date of the year. I have now introduced this variable and am exploring its use extensively. I have performed bucketization of the variables AccountID, Merchant, and Location by the Date variable to identify trends in the data based on daily temporal patterns. As we get into the modeling stage of the project I may incorporate other time interval bucketizations to identify trends in transactions by hour of the day or day of the week, for example.

Aggregation of Data

Going hand in hand with the bucketization step mentioned above, I have aggregated some columns of the dataset to identify the number of instances of certain categories in financial transactions per specific time intervals as well as descriptive statistics of the transaction amount values those transactions exhibited. As my dataset is fairly limited in terms of its number of numerical columns, the counts of categories per time interval will be considered heavily when identifying the presence of anomaly transactions by temporal means.

AccountID with Aggregated and Bucketized Components

	Date	AccountID	total_transactions	total_amount	average_amount	max_transaction	min_transaction
0	2023-01-01	ACC1	45	465.643415	10.347631	11.455237	6.655865
1	2023-01-01	ACC10	39	401.756535	10.301450	11.486849	4.735672
2	2023-01-01	ACC11	45	466.881311	10.375140	11.449300	6.820377
3	2023-01-01	ACC12	48	494.428680	10.300598	11.504645	7.298147
4	2023-01-01	ACC13	51	537.094450	10.531264	11.502063	5.600198
...	...	...	...	...	...	...	...
2260	2023-12-05	ACC5	75	795.515336	10.606871	11.503390	5.160261
2261	2023-12-05	ACC6	53	564.398349	10.649025	11.505050	7.599867
2262	2023-12-05	ACC7	60	632.902432	10.548374	11.503703	6.454727
2263	2023-12-05	ACC8	70	737.592979	10.537043	11.490852	6.755257
2264	2023-12-05	ACC9	80	829.224181	10.365302	11.489467	6.778694

[2265 rows x 7 columns]

Merchant with Aggregated and Bucketized Components

	Date	Merchant	total_transactions	total_amount	average_amount	max_transaction	min_transaction
0	2023-01-01	MerchantA	65	693.193900	10.664522	11.484058	7.952207
1	2023-01-01	MerchantB	71	738.537511	10.401937	11.503438	3.548180
2	2023-01-01	MerchantC	83	862.766604	10.394778	11.479025	5.491950
3	2023-01-01	MerchantD	77	807.919846	10.492466	11.488507	5.600198
4	2023-01-01	MerchantE	51	529.932174	10.390827	11.491695	6.264293
...	...	...	...	...	...	...	...
1505	2023-12-05	MerchantF	116	1209.703998	10.428483	11.506536	7.657349
1506	2023-12-05	MerchantG	94	966.141520	10.278101	11.507756	6.653495
1507	2023-12-05	MerchantH	103	1102.075913	10.699766	11.503703	7.587767
1508	2023-12-05	MerchantI	110	1165.206592	10.592787	11.511759	5.992239
1509	2023-12-05	MerchantJ	104	1094.717977	10.526134	11.512097	5.024472

[1510 rows x 7 columns]

Location with Aggregated and Bucketized Components

	Date	Location	total_transactions	total_amount	\
0	2023-01-01	London	146	1530.237727	
1	2023-01-01	Los Angeles	139	1454.136479	
2	2023-01-01	New York	135	1402.617642	
3	2023-01-01	San Francisco	138	1445.854505	
4	2023-01-01	Tokyo	133	1410.359817	
..	...	...	...	...	
750	2023-12-05	London	173	1808.436815	
751	2023-12-05	Los Angeles	193	2028.860958	
752	2023-12-05	New York	219	2297.486070	
753	2023-12-05	San Francisco	205	2163.578111	
754	2023-12-05	Tokyo	232	2440.682868	

	average_amount	max_transaction	min_transaction
0	10.481080	11.502063	5.491950
1	10.461414	11.504645	3.548180
2	10.389760	11.507789	5.600198
3	10.477207	11.504023	4.735672
4	10.604209	11.499541	7.197413
..	...	...	...
750	10.453392	11.506536	5.992239
751	10.512233	11.503571	5.160261
752	10.490804	11.512097	5.024472
753	10.554040	11.507744	6.454727
754	10.520185	11.507756	6.924711

[755 rows x 7 columns]

Logarithmic Representation of Amount Variable

Transforming my amount variable into its logarithmic representation will allow for my eventual models to be trained in a robust manner that reduces skewness and the impact of outliers on the datasets overall stature. This adjustment to the amount variable redistributes the weight of impact that any specific data point may have on the training process of a machine learning model, which creates a more optimal environment to identify properties of our dataset and apply those properties towards identifying potentially fraudulent financial transactions due to their “sticking out” nature amongst their neighboring data points.

Dropping Columns

Upon the conclusion of the exploratory data analysis stage of this project, I recognized that my dataset had become very large and potentially overly complex for the purpose of the models I would like to implement. Thus, I found it appropriate to remove some columns from my datasets that were creating this sense of false complexity. The columns I removed from each of my respective train, validation, and test sets are TransactionID, AccountID/TransactionID, AccountID/Merchant/TransactionID, AccountID/Merchant, Location/TransactionType, and Merchant/Location.

My reasoning for removing the individual column, TransactionID, was that it contains roughly 2,000 unique values that do not have any relative numerical relationship to one another, making it difficult if not impossible to bucketize or identify any real conclusions from.

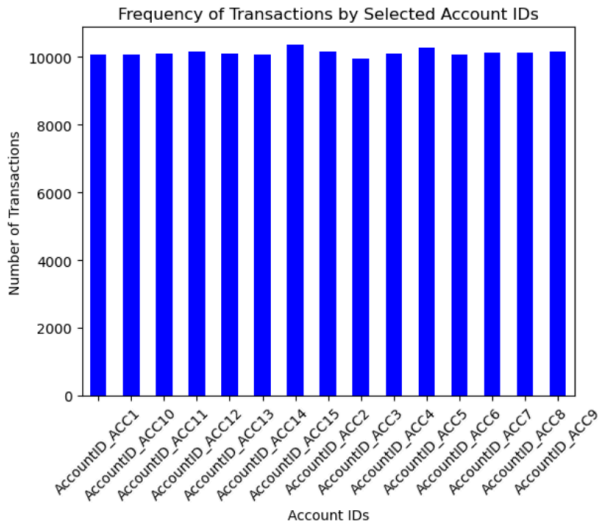
My initial thought process of creating categorical interaction variables was to identify connections between the respective columns of the dataset. The issue with this process is it leads to variables that can take on many unique values, which in a similar light to TransactionID, makes those variables difficult to derive interpretations from. With the feature engineering stage of the modeling project on the horizon, I would like to explore implementing categorical interaction variables or other diverse bucketizations that could set my eventual models up for producing strong conclusions.

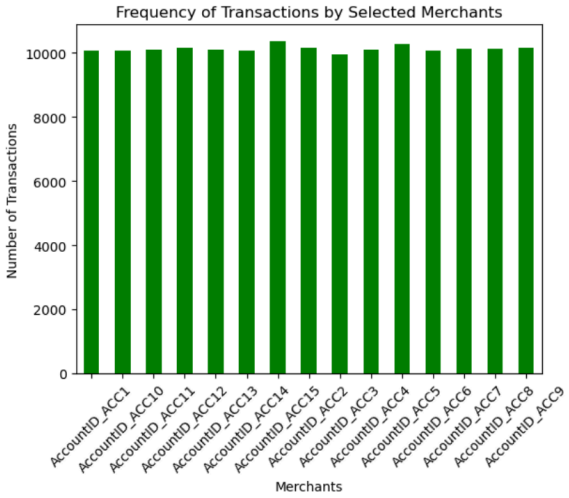
One-Hot Encoding Categorical Variables

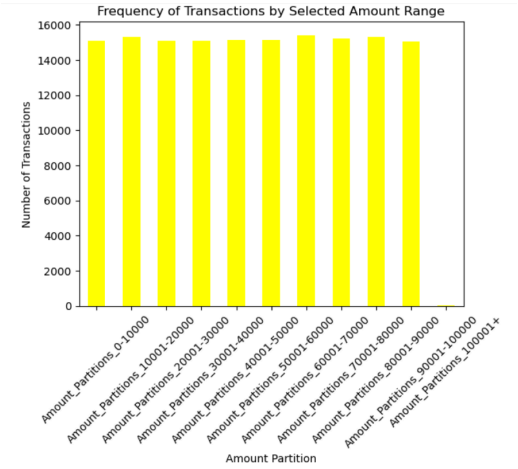
I have one-hot encoded five variables, namely AccountID, Merchant, TransactionType, Location, and Amount_Partitions. The purpose of executing this technique is to make these variables more easily inputtable for machine learning models while maintaining their respective meanings. I briefly explored what one-hot encoded variable representations would look like for this dataset last week and decided these variables made the most sense to one-hot encode as they do not have overly large dimensionalities as is. This factor makes them more flexible in terms of maneuvering the structure of a model around them and overall maintaining model versatility.

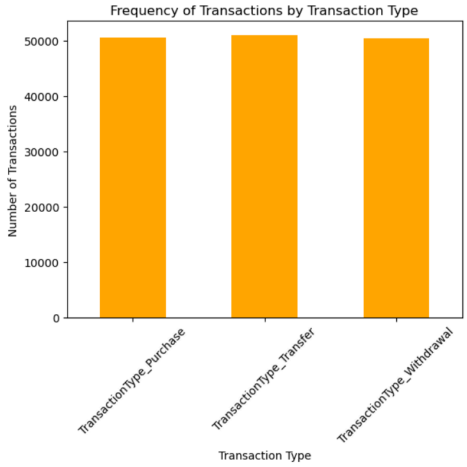
```
#One hot encode categorical variables
train_encoded_df = pd.get_dummies(train_df, columns=['AccountID', 'Merchant', 'TransactionType', 'Location', 'Amount_Partitions'])
```

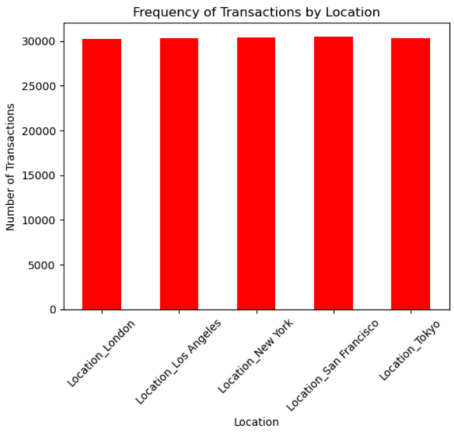
Aggregated Counts of One-Hot Encoded Variables











State of the Data after One-Hot Encoding

	Timestamp	Amount	Minute	Hour	Day	Month	Date	AccountID_ACC1	AccountID_ACC10	AccountID_ACC11	...	Amount_Partitions_10001-20000	Amount_Partit
9230	2023-07-01 17:50:00	9.064231	50	17	5	7	2023-07-01	False	False	False	...	False	
41764	2023-01-30 08:04:00	10.757187	4	8	0	1	2023-01-30	False	False	False	...	False	
136513	2023-06-04 03:13:00	10.996651	13	3	6	6	2023-06-04	False	False	True	...	False	
158548	2023-04-21 10:28:00	11.204528	28	10	4	4	2023-04-21	False	False	False	...	False	
9929	2023-08-01 05:29:00	9.295688	29	5	1	8	2023-08-01	True	False	False	...	True	

5 rows × 51 columns

As evidenced by the image, the dataset now contains 53 columns, with most taking on binary values due to their one-hot encoded representations.

Common Preprocessing Steps That Were Not Necessary For This Dataset

- Missing Value Treatment
- The financial anomaly dataset is a complete dataset that does not contain random missing values. If it had contained missing values, we would have decided on an appropriate metric to fill those missing values, either by using a kind of placeholder or removing those rows of the dataset entirely.

- Outlier Treatment
- Given the nature of this project attempting to identify anomaly transactions, it is vitally important not to throw away presumably outlier transactions due to their potential to fall under the anomaly classification. We applied a log transformation on the Amount variable to reduce the impact of outliers, though I did not find it appropriate to delete them from this dataset.

- Duplicate Row Removal
- Each of the 216,960 transactions contained in the original dataset are their own respective unique transactions, so there are no exact duplicates present to remove. I pondered the idea of removing very similar data points, though I reasoned that similar data points can indicate trends in the dataset that could ultimately help us to solve the objectives of this project, so I did not manipulate these rows.
-

Validation and Test Set Implementations

Each of the steps that I have taken in this stage of this modeling project have been performed exactly as they were on the train set on the respective validation and test sets as well. In careful consideration of biases and data leakage that can occur in large-scale modeling projects such as this one, I have independently performed each of those steps on the aforementioned datasets. Additionally, I intentionally did not produce any outputs from my validation and test set cleaning and preprocessing so that this experimentation can be conducted in a blind fashion without me having the potential of understanding traits surrounding the validation or test sets that could influence my decisions in the feature engineering and model construction processes.

Revisitation of Problems Identified Last Week

"Our dataset started out relatively large, containing over 200,000 unique entries. Across the past few weeks, we have introduced numerous new features and numerical representations of variables, further expanding and complicating our dataset. A very large dataset may prove to take an exorbitant amount of computing resources that we do not have available to us to fully and effectively process.

Aside from the sheer size of the dataset, our EDA has introduced us to limitations present in the data. As we progress through the next couple weeks, it will be essential that any variables we have identified do not correlate with the objective of our problem statement are removed from the dataset so that we have a platform to build machine learning models that are very pertinent to identifying anomalies in financial transaction data."

In the process of making our data model ready, we have addressed and resolved the above two issues. These issues were primarily resolved by reducing the raw size of our data files and eliminating the presence of complicated variables that are difficult or impossible to interpret or make meaningful connections between. Next week, we will continue to consider the relative correlations between variables in our datasets as we are engineering new features and preparing to feed them through a series of machine learning models.

Closing Remarks and Next Steps

Through this point in the end-to-end modeling project, our datasets are, for the majority, ready for the modeling stage. We have cleaned the data, reduced its complexity and dimensionality, and made it more easily usable for input to prospective machine learning models. Our next order of business will be to engineer new features that support our objective of identifying which financial transactions from our large dataset stand out as anomalies and could reasonably represent financial fraud.