Applied Analytics Project - Week 3 Assignment
Perform Exploratory Data Analysis

Brady Levenson
ADAN 8888 - Applied Analytics Project
Professor Nurtekin Savas
September 22, 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

Performing exploratory data analysis is a crucial step of the analytics modeling solution development process. Representing a foundational portion of the data preparation stage of our project, EDA is performed to make our dataset more easily interpretable for machine learning models so that those models can do a great job at producing accurate solutions to our problem statement. In this use case, EDA is used to transform a raw dataset downloaded from the internet into a robust tool that is getting closer to being fully prepared to be fed through Machine Learning Models, Exploratory data analysis creates a blueprint for upcoming actions to be taken to further extend the robust nature of our dataset.

Dataset Partition Strategy Explained

To prepare our financial anomaly dataset for further analysis, it is necessary to parse the dataset into three separate sections. Conducting this process allows for machine learning models to be trained on a large portion of the whole dataset to gain a picture of the important properties that the dataset holds, validated on a smaller portion of the dataset to verify that the model went through an effective training process, and tested on the remaining portion of the dataset to ensure that the model performs well on unseen data and further could represent a viable solution to the problem of financial fraud detection in the real world.

```
#create train set (70%) and temporary other set (30%)
train_df, temp_df = train_test_split(new_financial_df_encoded, test_size=0.30, random_state=1)

#split the leftover temp set into validation and test sets (50% of 30% each- 15% each)
validation_df, test_df = train_test_split(temp_df, test_size=0.50, random_state=42)

#verify shape of train, validation, and test DataFrames
print(f'Training set shape: {train_df.shape}')
print(f'Validation set shape: {validation_df.shape}')
print(f'Test set shape: {test_df.shape}')
```

```
Training set shape: (151872, 183)
Validation set shape: (32544, 183)
Test set shape: (32544, 183)
```

Train Set

The train set is a subset of the whole dataset that has undergone exploratory data analysis that is composed of 70% of the available data. I strategically chose to make the train set 70% of the data because the dataset being used to solve our problem is rather large, consisting of 216,960 unique entries, meaning that the train set consists of 151,872 of those entries. I found this size of train set to be a large portion of the whole dataset that has the potential to capture trends of which transactions may be valid and which may be fraudulent while leaving a large enough portion for validating and testing to ensure the validity of our models can be verified accurately. As fraudulent financial transactions are fairly rare in the real world, it is essential that all of train, validate, and test sets have enough data present to identify some of the aforementioned anomaly transactions.

Validation Set

The incorporation of a validation set in the modeling process guarantees that we will have an opportunity to view the potential performance of a machine learning model on test-esque data and use our conclusions to make adjustments that may improve model performance accordingly prior to launching a final model that is intended to be used in live financial transaction markets. I made the validation set compose 15% of the total available data, hence 32,544 unique entries. This size of validation set is large enough for us to identify methods that we could execute in an effort to improve model performance while leaving enough data for our models to be trained on and enough data for our final model to essentially run a real world performance simulation.

Test Set

The test set in this project will be used to verify that our model performs to a standard with which we will decide whether or not it is a trustworthy resource for predicting the likelihood of a certain financial transaction being fraudulent. This set is the same size as the validation set, hence 15% of the total dataset or 32,544 unique entries. This size of test set is large enough for a final product model to perform to its greatest capabilities on and still leaves a substantial amount of data for the original models to be trained and validated on.

In sum, I find the proportions of each of the respective train, validation ,and test sets to be large enough to fulfill the purpose they are set out to fulfill within this project without hampering the performance of our machine learning models within the scope of any of their counterpart datasets.

EDA Analysis Discussion

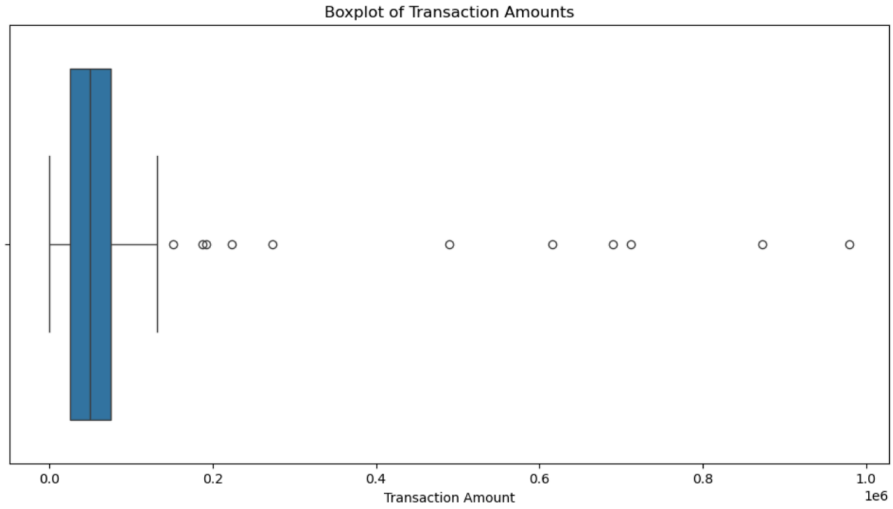
Each of the exploratory data analysis steps that I have taken this week have been executed with the intent of uncovering what data processing practices would be most recommended for the dataset that we presently have in order to make the data model ready. Keeping this in mind, I have identified areas where the data may perform very effectively in a modeling scenario and I have also uncovered some challenges with this dataset that I will need to overcome to increase the likelihood of achieving viable results from this experimentation process.

Types of EDA Conducted:

- Statistical analysis of numeric data
- Calculations of mean, quartiles, min, max, count and standard deviation.
- The statistical analyses conducted here yield a better understanding of the spread of transaction amount and time variables. From this portion of EDA, I learned further that this dataset is constructed very uniformly with respect to time, hence some time series analysis may prove to be a viable technique to uncover seasonal trends and patterns in the financial transaction data.

	Timestamp	Amount	Minute	Hour	Day	Month
count	216960	216960.000000	216960.000000	216960.000000	216960.000000	216960.000000
mean	2023-03-17 15:59:30	50090.025108	29.500000	11.517699	2.973451	3.017699
min	2023-01-01 08:00:00	10.510000	0.000000	0.000000	0.000000	1.000000
25%	2023-02-07 23:59:45	25061.242500	14.750000	6.000000	1.000000	2.000000
50%	2023-03-17 15:59:30	50183.980000	29.500000	12.000000	3.000000	3.000000
75%	2023-04-24 07:59:15	75080.460000	44.250000	18.000000	5.000000	4.000000
max	2023-05-31 23:59:00	978942.260000	59.000000	23.000000	6.000000	5.000000
std	NaN	29097.905016	17.318142	6.918770	2.008659	1.421907

Boxplot detailing the spread of financial transaction amounts in our dataset



- Counts of specific values within each column of DataFrame
- This factor yields a clearer picture of how common or uncommon instances of certain features are within the dataset. I learned that instances of some features are very common while others are not very common. In some categories there is little to no difference in the counts of specific values within a given column. In any case, all of this information will be taken into account when making our data model ready next week.

Sample of one column's (Amount_Partitions) counts:

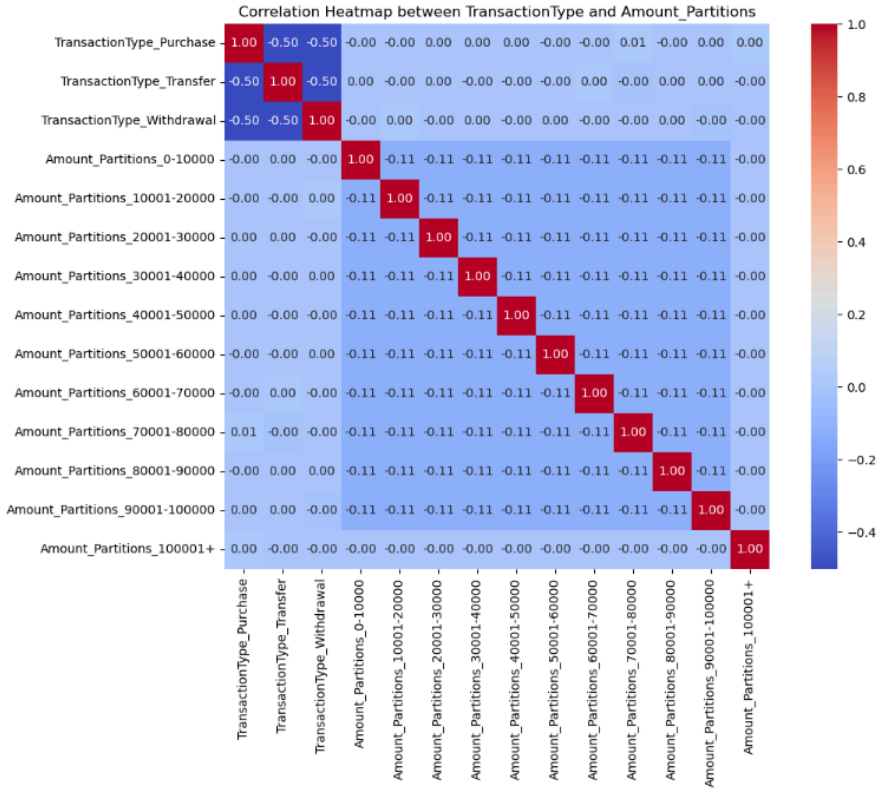
```
Amount_Partitions
60001-70000     22015
80001-90000     21938
10001-20000     21743
70001-80000     21736
50001-60000     21661
0-10000         21651
40001-50000     21605
20001-30000     21601
90001-100000    21530
30001-40000     21466
100001+            14
Name: count, dtype: int64
```

- One-hot encoding to measure correlation between certain categorical variables
- One-hot encoding allows us to view relationships between different categorical variables that may not be obvious by viewing them in a non-numeric sense. I was originally going to conduct this step for all of the categorical variables in my DataFrame, although it would have required over 31.2 GB of RAM to execute, which rendered my personal laptop incapable of performing this operation. Thus, I performed one-hot encoding on four of my categorical variables and measured the correlation between them.

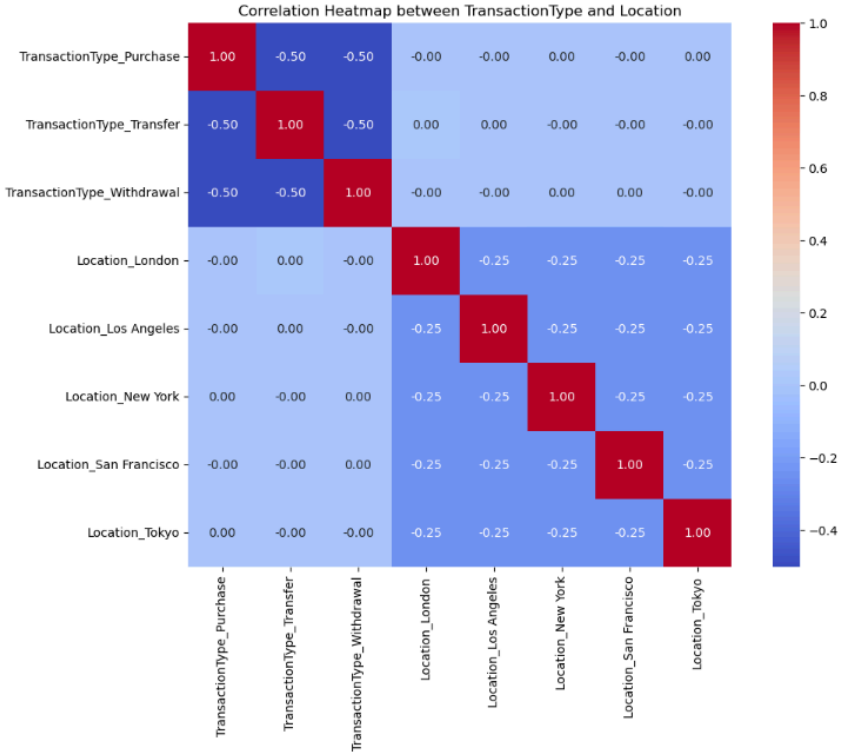
  Sample of correlation values from one-hot encoded variables

```
Correlation between AccountID/Merchant_ACC10_MerchantA and TransactionType_Purchase: -0.0032207883322759832
Correlation between AccountID/Merchant_ACC10_MerchantA and TransactionType_Transfer: 0.001793310647927313
Correlation between AccountID/Merchant_ACC10_MerchantA and TransactionType_Withdrawal: 0.0014255429485284879
Correlation between AccountID/Merchant_ACC10_MerchantB and TransactionType_Purchase: 0.002105593961643959
Correlation between AccountID/Merchant_ACC10_MerchantB and TransactionType_Transfer: 0.0022736608589373534
Correlation between AccountID/Merchant_ACC10_MerchantB and TransactionType_Withdrawal: -0.004388247006339169
Correlation between AccountID/Merchant_ACC10_MerchantC and TransactionType_Purchase: -0.0028821986432473723
Correlation between AccountID/Merchant_ACC10_MerchantC and TransactionType_Transfer: -0.00019905063275461838
Correlation between AccountID/Merchant_ACC10_MerchantC and TransactionType_Withdrawal: 0.003084886992434382
Correlation between AccountID/Merchant_ACC10_MerchantD and TransactionType_Purchase: -0.0007273312838229094
Correlation between AccountID/Merchant_ACC10_MerchantD and TransactionType_Transfer: 0.0009696077051246796
Correlation between AccountID/Merchant_ACC10_MerchantD and TransactionType_Withdrawal: -0.0002443939485877638
```

Example of Correlation Heatmap for TransactionType and Amount_Partitions



Example of Correlation Heatmap for TransactionType and Location



Insights Gained as a Result of EDA

This dataset is spread very uniformly. Each datapoint has its own minute that it occurs within, which continues consistently for a duration of about 5 months. Due to the fact that the dataset is spread evenly over time, it may make sense to batch certain time intervals in the next stage of our modeling process with the intent of capturing the trend of specific categorical or numeric features over time. Along with the uniform spread of time, many other categorical variables are spread fairly evenly across the dataset. Each unique Merchant, AccountID, Location, and Amount Partition basket instance, among others, have roughly the same amount of occurrences within them, making it even more important that trends of this dataset are captured from multiple different angles and perspectives. To touch on the one-hot encoding process we conducted this week, the categorical variables in this dataset seem to have some level of correlation between one another, even if that level is relatively small. Keeping this factor in mind, exploring further one-hot encoding applications as we continue to make this data model ready will be very useful in the context of preparing our data to produce meaningful outputs from which data-driven conclusions and solutions can be derived.

How Does Our EDA Relate to the Problem Statement?

The synopsis of our problem statement is that it is difficult to identify fraudulent financial transactions without exploring them through the lens of machine learning. The EDA processes that we have executed through this week have provided us with an understanding of exactly why we cannot pinpoint fraudulent financial transactions with a high degree of accuracy simply by looking at them. EDA has also provided us with a sequence of upcoming steps to take to further preprocess our data so that our machine learning models cna effectively perform the actions that we would like them to.

Opportunities to Solve Our Problem Statement Presented by EDA

The EDA we have conducted in this assignment has presented us with some unique opportunities. There are some combinations of categorical interaction variables present in our dataset that occur with a higher frequency than others. In a dataset that appears to be very uniform on the surface, recognizing some factors that could present themselves as varying from the majority trend of the dataset could be pivotal in flagging specific financial transactions as anomalies, and hence as potentially fraudulent.

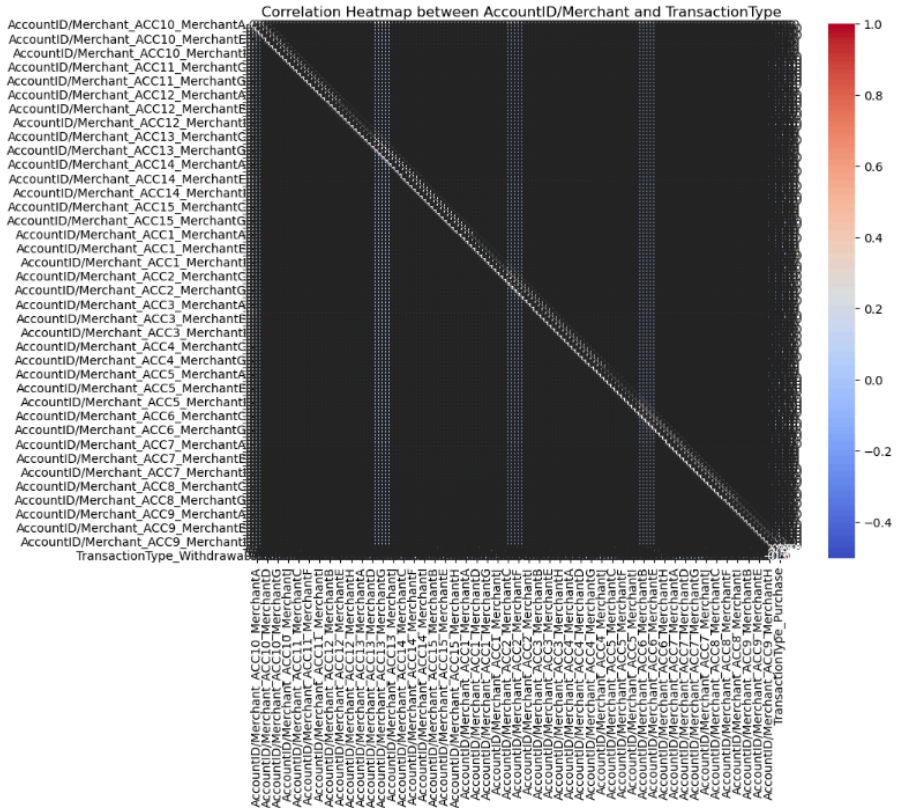
The uniform time spread of the data presents an opportunity to examine patterns exhibited within the dataset over constant intervals of time. This factor could ultimately help us identify if suspicious transactions are being made in consistent or inconsistent time intervals.

Challenges of Solving Our Problem Statement Presented by EDA

The uniform nature of many of the variables present in this dataset render some of these variables ineffective at identifying instances of data that appear to be abnormal in nature. This was evident in some of my correlation analyses from this week that visualized some variables having essentially no correlation with some others.

Some of the features of this dataset have many unique values, making it difficult to analyze each of them individually, especially when interacting with other features that also can take on many unique values. This factor will make the process of identifying correlating variables more challenging, although it also presents an opportunity to conduct even further preprocessing to make the data more model ready than it may have been otherwise.

Example of features that are too thinly spread to easily measure correlation between



Data Problems

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

Data Preprocessing Recommendations

I have come up with three data preprocessing recommendations based on the exploratory data analysis conducted for this assignment:

1. Create time interval batches to analyze trends of specific feature values over consistent intervals of time.
2. Remove any features that do not indicate abnormalities within the dataset, hence making the dataset relevant to anomaly detection and reducing the computing power necessary for a machine learning model to be trained, validated, and deployed.
3. Conduct further analyses to discover other variables that are correlated to one another beyond the ones that we considered this week.

Closing Remarks

Through executing the exploratory data analysis process, we have uncovered a significant amount of information about our dataset. We now are aware of the next steps that should be taken to further prepare our dataset to be model ready and have considered sensible approaches to this problem. We identified explicit ways that our dataset can be used to solve our problem statement, and we will continue to check this quality through each step of our model development process. The exploratory data analysis conducted at this stage of our project reassured us that there are practical means available to solve our problem statement, and helped us to map out the next steps that should be taken to get closer to accurately identifying fraudulent financial transactions in the real world.