Applied Analytics Project - Week 5 Assignment

Engineer Features

Brady Levenson
ADAN 8888 - Applied Analytics Project
Professor Nurtekin Savas
October 6, 2024

Revisitation of Problem Statement for Motivation to 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.

Introduction

This stage of the end-to-end applied analytics modeling project was dedicated to engineering new features that have the capability of uncovering more or less hidden patterns within our dataset to later be used for modeling purposes. During this portion of the assignment, creating embeddings for categorical variables was also explored, and will be an ongoing point of consideration for the beginning portion of Week 6. This technique allows for a reduction in the dimensionality of our dataset while maintaining the integral trends and properties held by the dataset as a whole. During this portion of the project, many newly engineered features were derived from already existing features in the dataset and introduced as new columns, yielding our eventual models with wider varieties of perspectives on the story the data has to tell.

Why Should We Perform Feature Engineering?

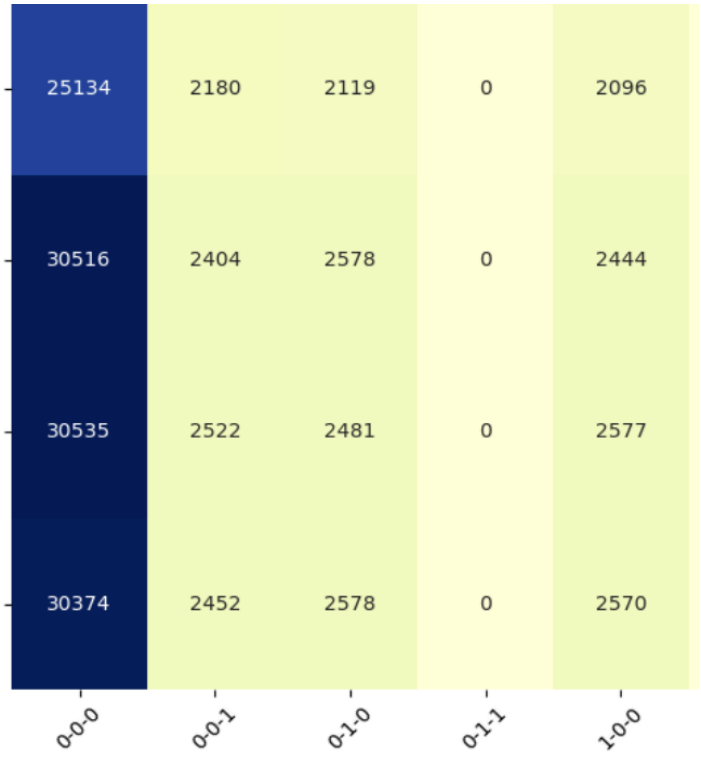
Our original financial anomaly dataset downloaded directly from Kaggle contained 7 total columns with 6 of those columns being categorical. Through transforming data types, one-hot encoding, and engineering new features, we have created vast datasets that have the ability to capture temporal patterns of data, categorize categorical variables in a binary fashion, and determine the relative position and importance of certain features and their respective transaction entries.

Features That We Have Engineered

Hour_Group: This feature allows for our models to take temporal patterns of financial transaction data into consideration without the risk of overfitting to too many unique time values. We have condensed the Hour feature that we had previously created, consisting of 24 unique values (one for each hour of the day) to a feature that has 4 unique values, namely buckets of hour ranges 12:00AM-5:59AM, 6:00AM-11:59AM, 12:00PM-5:59PM, and 6:00PM-11:59PM. By categorizing variables by time using the Hour_Group feature, we may be able to extract more robust results from our anomaly detection algorithms than what may have been possible by categorizing variables by each of the 24 hours of the day individually.

Hour_Group_AccountID: This feature creates an interaction between the aforementioned Hour_Group feature and the AccountID feature that we one-hot encoded in last week's assignment. By incorporating an interaction variable to connect Hour_Group and AccountID, we can gain a deeper understanding of the temporal patterns that each of financial accounts in our dataset exhibit, creating a greater likelihood of our models accurately identifying when a financial transaction that is outside of the relative " normal" transaction pattern of a certain account is attempted.

Frequencies of transactions of 3 accounts and hour groups from the dataset (the column of zeros is due to the one-hot encoding format. There is no account that has two values of 1 in its one-hot encoded representation, so this column prints zeros.



AccountID_mean (amount): This feature calculates the mean of financial transaction amounts for individual financial accounts. I am primarily utilizing this feature as an intermediary step towards calculating each individual transaction's deviation from its respective account's mean.

AccountID_std (amount): AccountID_std calculates the value of the standard deviation of financial transaction amounts per specific account. In the same light as AccountID_mean, AccountID_std is primarily being utilized to calculate the amount that a certain financial transaction deviates from its account's mean.

Deviation_From_Mean_Account: As alluded to with the previous two newly engineered features, this feature calculates the number of standard deviations that a certain financial transaction deviates from that account's mean transaction amount. This feature is notably versatile as it may

assist with identifying financial transactions that do not align with the trends and expectations of certain financial accounts, which may ultimately make us inclined to flag some of those transactions as being potentially fraudulent.

Train Set Info through this assignment.

```
<class 'pandas.core.frame.DataFrame'>
Index: 151872 entries, 9230 to 128037
Columns: 111 entries, Timestamp to AccountID
dtypes: category(1), datetime64[ns](1), float64(45), int32(49), object(15)
memory usage: 100.4+ MB
```

Sample head of dataset (not inclusive of all columns due to the present extensive nature of the one-hot encoded columns

	Timestamp	Amount	Minute	Hour	Day	Month	Date	AccountID_ACC1	AccountID_ACC10	AccountID_ACC11	...	Deviation_From_Mean_6	Deviation_From_
9230	2023-07-01 17:50:00	9.064231	50	17	5	7	2023-07-01	0	0	0	...	-1.462772	-1
41764	2023-01-30 08:04:00	10.757187	4	8	0	1	2023-01-30	0	0	0	...	0.242158	0
136513	2023-06-04 03:13:00	10.996651	13	3	6	6	2023-06-04	0	0	1	...	0.483316	0
158548	2023-04-21 10:28:00	11.204528	28	10	4	4	2023-04-21	0	0	0	...	0.692663	0
9929	2023-08-01 05:29:00	9.295688	29	5	1	8	2023-08-01	1	0	0	...	-1.229678	-1

5 rows × 111 columns

_8	Deviation_From_Mean_9	Deviation_From_Mean_10	Deviation_From_Mean_11	Deviation_From_Mean_12	Deviation_From_Mean_13	Deviation_From_Mean_14	AccountID
39	-1.441531	-1.461196	-1.451646	-1.457638	-1.475667	-1.429012	12
07	0.248660	0.242888	0.257266	0.242051	0.236121	0.250782	9
36	0.487733	0.483926	0.498987	0.482467	0.478249	0.488385	11
51	0.695270	0.693169	0.708823	0.691171	0.688438	0.694646	12
26	-1.210451	-1.228217	-1.218007	-1.225260	-1.241634	-1.199354	1

Consideration of Data Augmentation

Data augmentation is a useful tool for relatively small datasets that need to gather and generalize additional synthetic data beyond the data that is already present in the dataset in order to produce more robust and reliable results from modeling applications. In the case of my specific financial

anomaly dataset, we already have 216,960 unique financial transactions and numerous newly engineered features to analyze them with, Using data augmentation for this project would primarily increase the risk of overfitting, creating less efficient models, and consequently increasing training time and necessary computing power. Since we already have a sufficiently-sized dataset I elected to mitigate the aforementioned risks by not pursuing data augmentation in this project.

Embeddings Implementation

I decided to implement embeddings as my primary dimensionality reduction technique for the data preprocessing/ feature engineering portion of this project. My thought process for incorporating this method as opposed to more standard methods such as Principal Component Analysis was that this financial anomaly dataset contains notably uniform data, meaning that to a human viewer not using machine learning methodologies to parse through the dataset, the data looks relatively unsuspecting in that it is difficult to discern which financial transactions may have the highest likelihood of being fraudulent. With this significant factor in mind, I reasoned that a dimensionality reduction technique that primarily captures linear trends of data (e.g. PCA) may not be effective at reducing the dimensionality of the dataset while retaining the value held within it. Embeddings on the other hand have the ability to capture more nuanced patterns within the dataset that may not be evident from strictly considering linear patterns in the data. I found the embeddings approach to reducing the dimensionality of this dataset to be the most sensible option for the aforementioned reasons. By incorporating embeddings at this stage in the end-to-end modeling project, I can feel confident in my eventual models' capabilities to perform to their greatest potential on data that is most indicative of subtleties present in the data of large financial markets globally.

Conclusion

During this stage of the modeling project, we have engineered a plethora of new features that will assist us in identifying the presence of anomaly financial transactions from a crowd of thousands of these transactions that each look natively similar to one another. We have also created embedding representations of many features of the dataset to reduce dimensionality and aid in better understanding the true nature of the financial transaction dataset we have at our disposal.

Next Steps

We are quickly approaching the model development phase of the project. Prior to building my first model, I would like to refine and quality-check my embeddings structure and implementation to ensure that the dataset is fully prepared to be used as input for modeling purposes. Once I complete this my focus will be shifted towards producing models that have the ability to realize the key trends and indicators held within my dataset that I have been preparing for over the past month or so. I will be looking forward to sharing my final dimensionality cleaning updates early next week and producing my first model.