Applied Analytics Project - Week 6 Assignment
Develop First Modeling Approach

Brady Levenson
ADAN 8888 - Applied Analytics Project
Professor Nurtekin Savas
October 13, 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.

Dimensionality Reduction Update

At the conclusion of last week's assignment, I had yet to finish implementing my dimensionality reduction strategies. My datasets at the time contained about 112 unique columns, which would have made this dataset extremely difficult to use for modeling tasks without first addressing this issue. After successfully creating embedding representations of the features AccountID, Merchant, TransactionType, Location, and Hour_Group, consolidating my unique Deviation from Mean variables into one clean variable, and removing numerous columns that I either decided were noncontributory to resolving the problem statement of this assignment or that I considered in a more efficient manner in another portion of the datasets, I have reduced my datasets to 16 unique columns.

First Model Approach

My first model approach for this project is an unsupervised K-Means Clustering Model. There are a couple reasons why I decided this modeling approach may be appropriate for this task. Firstly, the structure of my chosen dataset indicates that this problem should be solved in an unsupervised manner. While the target variable of this project is the binary classification of yes, a financial transaction is an anomaly, or no, a financial transaction is not an anomaly, we do not have a clearly defined labeled training set that our models can learn from directly. Thus, we needed to exercise a more elaborate approach. The K-Means Clustering technique is great for anomaly detection tasks as it allows for comparisons to be made between financial transactions within the same dataset, specifically aiding in the determination of which groups of transactions reside farthest away from the rest. This can help us to identify patterns of data that could potentially represent fraudulent transactions, which can then be isolated for further evaluation.

Complexity of Modeling Approach

Following my projected upward progression of modeling approach complexity, this model is the simplest of the three I have planned, and I would say it has a low to moderate level of complexity. A portion of the complexity of this model comes from the feature engineering that I conducted last week, though the model standing alone is fairly straightforward. The inclusion of features such as embeddings, mean amount deviations, and day and time considerations add versatility to the model, as its structure groups together transactions that are inherently similar to one another based on their feature values. Adjusting hyperparameter values can also increase the base complexity of the model, as we can perform hyperparameter tuning to discover setups that work the best for the specific feature set at hand.

Hyperparameters Evaluated

k- number of clusters

- The number of clusters considered in the model can determine the relative complexity of the model e.g. too few clusters = too simple; too many clusters = too complex.

init- initialization method

- Different initialization methods can lead to different starting points for cluster centroids, potentially impacting model performance.

n_init- number of initializations

- Higher values can improve model stability by considering more different initial centroids.

max_iter- maximum iterations

- Determines the convergence rate of the algorithm- varying levels may result in increased model stability.

tol- Tolerance for Convergence

- Threshold for determining that centroids have stabilized- high value indicates less precise results; low value indicates more precise results.
-

Model Performance Metrics

Silhouette Score: Measures a transaction's similarity to its own cluster versus others. This metric can be leveraged indirectly to determine if certain points are poorly clustered and may be more likely to be anomalies. Lower scores indicate poorer clustering, which indicates greater anomaly potential.

Dunn Index- Ratio between minimum inter-cluster distance and maximum intra-cluster distance. This metric yields a comparison between how close neighboring clusters are to each other and how far apart transactions within the same cluster are from each other. Low values signify poor clustering, which could indicate the presence of anomaly transactions in the data.

Inertia- Sum of squared distances between transactions and their nearest centroid. Greater inertia values may indicate a farther distance between a transaction and its nearest local center, which provides us with justification to say those points have high potential of being anomalies and should be considered further.

Metric Calculations Results Table

	Variation 1 Silhouette	Variation 1 Dunn Index	Variation 1 Inertia
Training	0.1814	0.0007	0.9682
Validation	0.2434	0.0023	0.972

	Variation 2 Silhouette	Variation 2 Dunn Index	Variation 2 Inertia
Training	0.242	0.0174	0.9461
Validation	0.2438	0.019	0.9745

	Variation 3 Silhouette	Variation 3 Dunn Index	Variation 3 Inertia
Training	0.4815	0.0176	0.459
Validation	0.4629	0.0177	0.4932

Model Performance in Different Variations

Variation 1: k=5, init=random, n_init=1, max_iter=300, tol=0.0001

Variation 2: k=5, init=k-means++, n_init=1, max_iter=100, tol=0.01

Variation 3: k=10, init=k-means++, n_init=1, max_iter=200, tol=0.001

Considering Factors for Hyperparameter Tuning

Due to the complexity and size of my dataset, I was originally waiting for multiple hours for my models to compile without signs of improvement. Thus I took more of a minimalist approach to

hyperparameter tuning here, focusing on specific differences in hyperparameters rather than trying many combinations of them as I had originally intended.

For the purpose of our modeling task, we will consider models that identify poorer clustering as stronger models, for we can identify anomaly data points more easily in this case. Hence, models with low Silhouette Score, low Dunn Index, and high Inertia should be given priority consideration.

Train Set Performance Metric Rankings (best to worst):
Silhouette Score: Variation 1, Variation 2, Variation 3
Dunn Index: Variation 1, Variation 2, Variation 3
Inertia: Variation 1, Variation 2, Variation 3

Validation Dataset Performance Rankings:
Silhouette Score: Variation 1, Variation 2, Variation 3
Dunn Index: Variation 1, Variation 3, Variation 2
Inertia: Variation 2, Variation 1, Variation 3

Week 6 Winning Model
When considering all variations across both training and validation data, I select Variation 1 as my winning model of the week. In 5 out of 6 performance categories, this model placed first amongst the three variations with a narrow second place finish in validation set Inertia. Based on the statistics calculated in this portion of the project, I would expect Variation 1's hyperparameter combination to perform the best on an unseen test set. An important consideration of these models is they were each trained on exactly one centroid initialization rather than variable amounts of centroid initializations, which may result in more variation in model results. I would have liked to include a wider variety of more stable results, though my embeddings-filled dataset proved to be too complex to accomplish this today.

Next Steps
As we progress through the next couple weeks of model development, I would also like to continue optimizing the model I have created for this week's assignment, as I see a lot of potential with it. This may include reducing the size of my training set as I notice validation set code compiles much faster due to its relatively smaller size. I would also like to explore more hyperparameter

combinations that may yield further insights into what factors provide the most guidance towards identifying financial anomalies amongst the transactions available to us in this dataset. Simultaneously, I will be developing my next modeling approach, which will be a tier up in complexity, which is why it is vital that my datasets are truly up to standard in their size and complexity level so that elaborate models can be processed on them in a timely manner.