Fraud Detection Analysis in Customer Transactions through Clustering

Anim Ohene
College of Science and Technology
Temple University

Henry Nguyen
College of Science and Technology
Temple University

Introduction

Businesses and banking institutions have serious concerns about identifying and stopping fraud in consumer interactions. Fraudulent operations can cause large financial losses and harm to a company's reputation, including identity theft, account control, and unlawful transactions. The intricacy and sophistication of deceptive strategies keep changing as technology develops and digital transactions proliferate.

This study aims to solve the difficulties in detecting fraud in consumer transactions and analyze more efficient strategies for successfully preventing fraud. Financial institutions may limit financial losses, safeguard their brand, and improve customer contentment by creating effective fraud detection procedures. Effective fraud detection also helps keep financial systems secure and reliable, which is advantageous to both consumers and institutions.

The issue is how to quickly and efficiently detect fraudulent activity in customer transactions. Phishing schemes and pretexting are two common strategies used by fraudsters to fool people into giving personal information or committing fraudulent acts without realizing it. These techniques bypass security measures by preying on people's weaknesses and manipulating their minds to conduct fraud discretely. To detect sophisticated fraud schemes, traditional fraud detection approaches generally rely on rule-based systems (flag suspicious activities if the amount exceeds the limit) and manual review procedures, which may not be reliable, efficient, or flexible enough.While supervised machine-learning algorithms may be efficient in detecting fraudulent transactions, a more nuanced approach such as unsupervised learning, may prove more effective in detecting fraud. The work was divided evenly between the students. Anim Ohene was responsible for dataset selection, half of the feature engineering, half of the data preprocessing, optimizing the models through the elbow method, building half of the models, and interpreting the results. Henry Nguyen was responsible for data exploration, half of the feature engineering, building half of the models, half of the data preprocessing, and the evaluation of the results.

Approach and Methods

For the fraud analysis task presented, the proposed approach utilized unsupervised learning. While the alternative route-–supervised machine learning is useful in the task of predicting fraudulent transactions, it isn't tailored to uncovering hidden structures that may be present in fraudulent transactions–the objective of this task. To expand, through unsupervised learning approaches, the aim was to uncover key relationships between fraudulent and non-fraudulent transactions, and particularly find evidence of ranges of susceptibility to fraud through clustering. For the clustering task, the presented work consisted of a structured pipeline to properly engineer the data, to be fit for clustering. This machine learning pipeline consisted of, selecting a dataset, data exploration, feature engineering, data pre-processing, optimizing clustering parameters, partitioning data with contrasting clustering algorithms, and lastly evaluating the clustering performance. The first requirement–selecting a dataset–holds the greatest importance. The outcome of any data analysis task weighs heavily on the quality and integrity of the data. By selecting a dataset representative of fraudulent transactions, any insights gained from the fraud analysis may be credible. The next step is data exploration. The idea here is to understand the features and the target variable in the fraudulent dataset. In this issue, key questions may be, what are the

features associated with the fraud dataset, what are their distributions, are there categorical variables, and are there any dependencies between them? By exploring the fraud dataset, allows one to gain domain knowledge on the subject of fraud, and provides a great advantage in what decisions to make further into the pipeline and how to interpret the data.

The next two tasks would be to process the data and engineer the features. These two processes work hand in hand. In the preprocessing stage, it is important to deal with any potential missing values in the fraud dataset appropriately, convert any categorical features into numerical values by way of an appropriate encoding scheme, and normalize/standardize the numerical features. Another important approach in the preprocessing step is the check for any class imbalances. In the Fraud task, the dataset is highly likely to be imbalanced as there are more cases of non-fraudulent transactions than fraudulent transactions that occur. In tackling this approach performing oversampling is required, as the lack of fraudulent samples may be disadvantageous in the clustering task. In addition, depending on the size of the dataset, it may be necessary to apply a sampling criterion to downscale it while maintaining class distribution. Now, in the feature engineering stage, Mutual Information Gain Score–a statistical–method– may be employed to discern the features with the greatest dependencies on the target variable–the object of study– fraudulent or non-fraudulent transactions. This method is useful in selecting appropriate features, by removing the noisy features, and not having a strong association in revealing fraud or non-fraud transactions.

The last step in the feature engineering stage will be to reduce the dimensionality of the dataset and capture more variance in the selected features. Upon completion of the data-driven engineering processes following task in the pipeline would be optimizing clustering parameters, partitioning data with contrasting clustering algorithms, evaluating the clustering performance, and interpreting the results. In the optimization task, a heuristic graphical approach known as the elbow method will be used to find the optimal number of clusters to be used in clustering the fraud dataset. After finding the optimal number of clusters, the next phase is to partition the dataset with contrasting clustering algorithms. K-means is a clustering algorithm that randomly initializes cluster centroids, classifies samples as members as part of its cluster based on their distance to its centroid in relation to other cluster centroids, relocates the cluster centroid to its center of mass, and repeats the process until the centroid center of mass remains the same. Spectral clustering is a graph-based approach where each data point is a node and an edge is a connectivity between the points. In spectral clustering, the goal is to maximize the similarity between the points in the clusters. Lastly, Agglomerative clustering is a form of hierarchical clustering that initializes each sample as a cluster, merges the nearest two clusters into a new cluster, and repeats until one cluster remains. Next, the clustering algorithms will be evaluated, based on the integrity of the clusters. One particular metric to be used is the silhouette score, which measures the association of a sample to its cluster as opposed to its attraction to another cluster. That is, silhouette scores assign values that range from -1 to 1, where -1 may be interpreted as a possible misclassification of a sample to a cluster, and 1 as a sample being assigned to its correct sample. After performing evaluation metrics, a bar plot will be made to compare the different clustering algorithms used and their silhouette scores, in selecting the most appropriate clustering algorithm for the given task. Finally, after partitioning with the clustering algorithms and performing the evaluations, clusters will be analyzed to uncover key relationships between fraudulent and non-fraudulent transactions, and particularly find evidence of ranges of susceptibility to fraud.

Results

It's important to note that, privacy concerns associated with financial transactions disallow the publication of big datasets containing fraudulent transaction metadata. As such, synthetic datasets derived from transaction simulators that mirror real-world financial transactions are made. One such dataset called

“Financial Fraud Detection Dataset” was selected by the students., as shown in Table 1a. To quote, “Derived from a simulator named PaySim, which utilizes aggregated data from actual financial logs of a mobile money service in an African country, this dataset aims to fill the gap in publicly available financial datasets for fraud detection studies. ”

Table 1a: Synthetic Financial Dataset For Fraud Detection

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0.00	0.00	0	0
...	...	...	...	...	...	...	...	...	...	...	...
6362615	743	CASH_OUT	339682.13	C786484425	339682.13	0.00	C776919290	0.00	339682.13	1	0
6362616	743	TRANSFER	6311409.28	C1529008245	6311409.28	0.00	C1881841831	0.00	0.00	1	0
6362617	743	CASH_OUT	6311409.28	C1162922333	6311409.28	0.00	C1365125890	68488.84	6379898.11	1	0
6362618	743	TRANSFER	850002.52	C1685995037	850002.52	0.00	C2080388513	0.00	0.00	1	0
6362619	743	CASH_OUT	850002.52	C1280323807	850002.52	0.00	C873221189	6510099.11	7360101.63	1	0

Table 1 shows the synthetic Financial dataset for fraud detection with dimensions 6362620 × 11

In the data exploratory stage, as shown in Table 1a, the dataset contains 6362620 rows and 11 columns. The column's features are type, amount, nameOrig, oldbalanceOrg newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud. Table 1a shows 3 categorical features: type, nameOrig, nameDest, and the rest are numerical features. The type feature represents the type of transaction that took place, the amount feature is the transaction amount in the local currency, nameOrig is the customer who initiated the transaction, and nameDest is the recipient of the transaction. The oldbalanceOrg feature is the initial balance before the transaction, and the newbalanceOrig is the balance after the transaction. The oldbalanceDest feature is the recipient's balance before the transaction, and newbalanceDest is the recipient's balance after the transaction. Now, isFraud is the target variable represented by binary values 0 and 1, for whether the transaction is fraudulent or not fraudulent. Lastly, the isFlaggedFraud feature, to quote, “Flags large-scale, unauthorized transfers between accounts, with any single transaction exceeding 200,000 being considered illegal.” To summarize the data exploratory phase, the correlations between the numerical features are taken, as well as the plot of the “type” feature. Remark, see the attached code file for distribution plots of the remaining features. As shown by Table 1b there are few correlations between each numerical feature. Foreseeably, there is a very high positive correlation between the OldbalanceOrg feature and the newbalanceOrig features. In addition, there is a fairly positive correlation between the transaction amount and the newBalanceDest feature. Figure 1b shows a bar plot of the categories within the “type” feature. 2237500 were CASH_OUT, 2151495 were PAYMENT, 1399284 were CASH_IN, 532909 were TRANSFER, and 41432 were DEBIT.

Table 1b: Correlations of numerical features

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
step	1.000000	0.022373	-0.010058	-0.010299	0.027665	0.025888	0.031578	0.003277
amount	0.022373	1.000000	-0.002762	-0.007861	0.294137	0.459304	0.076688	0.012295
oldbalanceOrg	-0.010058	-0.002762	1.000000	0.998803	0.066243	0.042029	0.010154	0.003835
newbalanceOrig	-0.010299	-0.007861	0.998803	1.000000	0.067812	0.041837	-0.008148	0.003776
oldbalanceDest	0.027665	0.294137	0.066243	0.067812	1.000000	0.976569	-0.005885	-0.000513
newbalanceDest	0.025888	0.459304	0.042029	0.041837	0.976569	1.000000	0.000535	-0.000529
isFraud	0.031578	0.076688	0.010154	-0.008148	-0.005885	0.000535	1.000000	0.044109
isFlaggedFraud	0.003277	0.012295	0.003835	0.003776	-0.000513	-0.000529	0.044109	1.000000

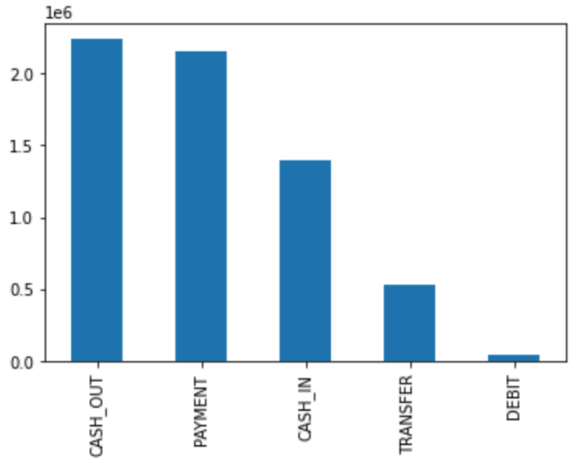

Figure 1b:
This shows a bar plot of the type of transactions presented in the dataset

Now, the results of the feature engineering stage and the data pre-processing overlapped in this study due to the nature of the data. After the data exploration, the first step was to check for any missing values. After the check, the dataset contained no missing values. Next, the categorical variables shown in Figure 1b were transformed into numerical values through label encoding. This was necessary for the clustering and for applying the Mutual information gain score. As stated in the approach, mutual information score is a method of finding the importance of each feature in relation to the target variable isFraud. Table 2a shows the Mutual information scores of each feature. Table 2a reveals the lack of relationship between most features and the target variable as the Mutual information scores are near 0. Table 2a reveals that the “step” and “type” features have the greatest mutual information score and thus the highest relation to the target variable “isFraud”. As such, only “step” and “type” were used in the study. After the feature selection, the step and type features were normalized using standardization, to preserve the variation of the trend shown in the “step feature”. After standardization, PCA was used to capture more variance. The PCA transformation may have been deemed unnecessary since only 2 features were selected, however, it captured more variance in the dataset than without utilizing PCA. Upon completion of the feature engineering stage, the pre-processing phase was resumed. Next, was to check for class imbalance within the target variable. Figure 2b shows the great discrepancy between the fraud and non-fraud classes. Specifically, there were 6354407 non-fraud cases and 8213 fraud cases. In this study, the imbalanced classes were addressed. This is because in the clustering task clusters may be dominated by the majority class, and fraudulent cases can not be properly studied. To resolve these potential problems, the minority class was oversampled to create a more balanced dataset and optimize the clustering task.

Remark, oversampling has some drawbacks as well due to the addition of synthetic data points. Figure 2c shows the result of the oversampling. Another notable obstacle was addressed in the pre-processing stage. With over 6 million samples, manipulation of the data, as well as visualizing it becomes a hard task. As such, stratified random sampling was performed. In this task, stratified random sampling was more appropriate than simple random sampling in order to preserve the class distribution. If left to a mere simple random sampling, there's no guarantee that the new sample dataset would contain the same distribution as the preceding dataset it drew samples from since samples are taken at random in simple random sampling. However in stratified random sampling, the data is divided into mutually exclusive subgroups–called strata– and samples are taken from each stratum. The result is a sample dataset more representative of the parent set. This concludes the pre-processing stage. The plot of the fully engineered, and preprocessed data, is shown in Figure 2d.

Table 2a: Mutual Information Score of Features

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFlaggedFraud
0	0.010813	0.170882	0.002459	7.006855e-07	0.002611	0.000642	0.000001	0.000177	0.000092	0.0

Table 2a shows the mutual information Scores of all features concerning the target variable. Higher scores represent higher dependency with the target variable and lower scores show less dependency(less relation).

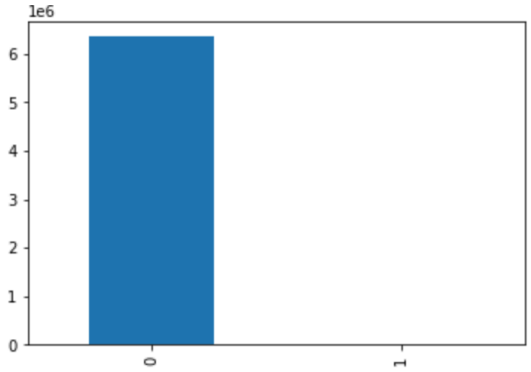

Figure 2b:
This figure shows the class imbalance of fraudulent and fraudulent classes. 0 represents a non-fraudulent case and 1 represents a fraudulent case

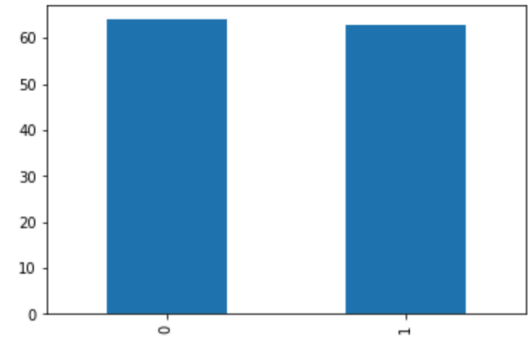

Figure 2c
This figure shows the result of oversampling the minority class–fraud cases. 0 represents a non-fraudulent case and 1 represents a fraudulent case

Figure 2d
The plot shows the result of the fully engineered, and preprocessed data. PCA1 is the preprocessed and transformed step feature, and PCA2 is the pre-processed and transformed type feature.

After Selecting the dataset, data exploratory/summary statistics, data preprocessing, and feature engineering, the next step in the pipeline was to perform the elbow method. The elbow method is a technique used to identify the optimal number of clusters to use. This is done by first obtaining the Within Cluster Sum of Squares(WCSS). WCSS value is the sum of the variance of each cluster and its data point. As the number of clusters increases WCSS decrease. This is because eventually if each data point becomes its own cluster the distance between that data point and itself is 0. Similarly, as the number of clusters decreases WCSS increases. The question becomes how can one obtain an optimal number of clusters such that clusters are not underfitted and no information is learned, but also that clusters aren't overfitted, such that the data has been partitioned into too many sub-categories. By visualizing the variation between clusters and data points as a function of the number of clusters, one can spot the point at which WCSS stops dropping rapidly–represented as an “elbow” in the graph and obtain the optimal amount of clusters. According to Figure 3, the students found that the optimal number of clusters was 4.

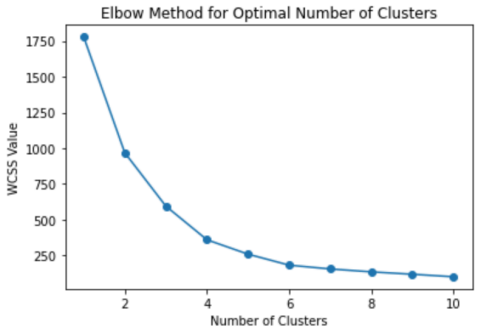

Figure 3

Figure 3 shows the elbow method. The X-axis is the number of clusters while the Y-axis represents the WCSS value for the corresponding amount of clusters.

After, obtaining the optimal clusters, the students performed the k-means clustering algorithm. In k-means, cluster centroids are randomly initialized among the data points. Data points are assigned to the initialized centroid based on their distance from a given centroid to other initialized centroids. Afterward, the centroids are relocated to their center of mass. This process of reassigning data points to their closest centroid and relocating each cluster to its center of mass is repeated until the center of mass of each centroid does not change signaling the final cluster assignment. Figure 4 shows the graph of using means to partition the data. Next was to perform Spectral Clustering on the dataset. Unlike the k-means algorithm that uses centroids, spectral clustering is a graph-based approach where each data point is a node and an edge is a connectivity between the points. In spectral clustering, the goal is to maximize the similarity between the points in the clusters. The steps are to first, construct the similarity matrix, compute the Laplacian matrix, conduct eigendecomposition to get the first C eigenvectors–U, then apply k-means to U. Figure 5 shows the result of the spectral clustering. The final clustering algorithm the students used was Aggolrmerative clustering. Agglomerative clustering is a form of hierarchical clustering that initializes each sample as a cluster, merges the nearest two clusters into a new cluster, and repeats until one cluster remains. The clustering results can be seen in Figure 6.

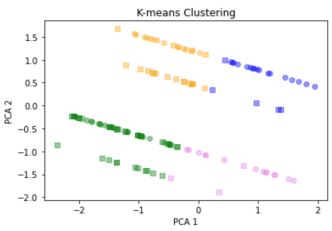

Figure 4

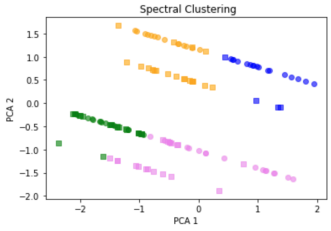

Figure 5

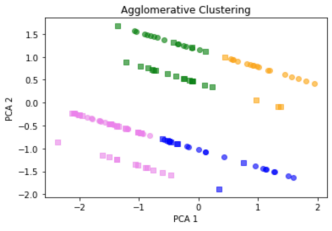

Figure 6

Figures 4, 5, and 6 show the use of k-means, Spectral and Agglomerative Clustering with 4 clusters, represented by pink, blue, orange, and green. Data points that are boxes represent non-fraud cases and circles represent fraud cases.

The last task performed was the evaluation of the clusters. The students used silhouette scores to measure the integrity of the clusters. Silhouette score measures how closely related the data points are to their assigned cluster, and how dissimilar they are to other clusters. The range of silhouette score is between -1 and 1, with a value closer to 1 meaning better clustering results. According to Figure 7, the k-means algorithm had the best partitioning, followed by agglomerative clustering, and lastly spectral clustering.

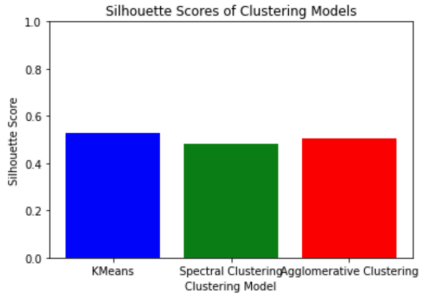

Figure 7

Figure 7 shows a bar plot of the silhouette scores for each of the clustering models. Blue is the score for k-means, green for spectral clustering, and red for agglomerative clustering.

Conclusion

In conclusion, the study showed how Fraud may be analyzed through unsupervised machine learning. In this study, to restate the goal, through unsupervised learning approaches, the aim was to uncover key relationships between fraudulent and non-fraudulent transactions, and particularly find evidence of ranges of susceptibility to fraud through clustering. This machine learning pipeline consisted of, selecting a dataset, data exploration, feature engineering, data pre-processing, optimizing clustering parameters, partitioning data with contrasting clustering algorithms, and lastly evaluating the clustering performance. To interpret the clusters, each slope represents a specific transaction, except for the debit transaction which was excluded in the sample plot shown. Cash-out and payment are the two slopes at the highest point in the graph, while transfer and cash-in are the two lowest slopes in the graph. There is a clear and "permanent" division of the clusters at a particular point in the x-axis which represents the PCA transformed step variable. The step variable represents a particular time in the simulation, ranging from 0-742 hours(0-30 days)-- the length of the simulation. This indicates that at a specific point in time in the simulation, the clusters change. This relation lays the groundwork for future studies of the precise time of day at which the fraud occurs. In the objective of finding evidence of ranges of susceptibility to fraud, different transaction types are more associated with fraud, such as cash out and payment transactions, and the most important finding was the shift in the clusters at a particular instance of time in the simulation. For example, in the k-means clustering, the orange and green clusters shift into blue and pink clusters about halfway into the simulation, so approximately day 15. These sudden change in clusters show less overlap between fraud and non fraud cases. Similar shifts are seen in the spectral and aggolemerative clusters. The questions to raise from this study are, what events or particular days invoked a sudden change in the clusters with more homogenous fraud and non fraud classes? Once again, this study showed a benchmark case of how fraudulent transactions may be studied through unsupervised machine learning, and out of the three clustering algorithms used, the k-means clustering algorithm was the best-unsupervised classification algorithm for measuring clusters' quality with a silhouette score of around 0.5. This suggests that the clusters are well separated and the data within each cluster are relatively similar to each other.

Acknowledgments

Abdallah, Aisha, et al. “Fraud Detection System: A Survey.” Journal of Network and Computer Applications, vol. 90–113, 1 June 2016, https://doi.org/10.1016/j.jnca.2016.04.007.

Vasan, K. Keerthi, and B. Surendiran. “Dimensionality Reduction Using Principal Component Analysis for Network Intrusion Detection.” Perspectives in Science, vol. 510–512, 1 Sept. 2016, https://doi.org/10.1016/j.pisc.2016.05.010.