Credit Card Fraud Anomaly Detection with Unsupervised Learning Techniques

Project 3: Milestone 3

Executive Summary

Credit card companies suffer significant losses from credit card fraud every year. Credit card companies must endeavor to identify fraudulent credit card activity. The goal of this project was to use the R programming language to analyze an actual credit card transaction dataset by using unsupervised learning methods to classify the credit card transactions as either fraudulent or non-fraudulent.

The credit cards transaction dataset included 31 features and was highly imbalanced, with only .17% of the transactions labeled as fraudulent. We under sampled the dataset for feature selection and identified significant features for the fraud classification. We then used the original unbalanced dataset in one class support vector machine and k means clustering unsupervised learning models.

The conclusions from this project are that the unsupervised One Class SVM model is a viable model for predicting credit card frauds. The clustering methods will take more work to become viable prediction methods. Further research should be performed to implement a specific cost matrix for evaluation,.

Introduction/Background

The Federal Trade Commission reported over $260 million of credit and debit card fraud losses in 2020 (Consumer Sentinel Network Data Book 2020, 2021). Credit card companies must endeavor to identify fraudulent transactions to prevent future fraudulent transactions. Fraud detection has always been challenging because the target class is very imbalanced (i.e. the incidences of fraud is far less than the total number of transactions) (Comparative Analysis of Machine Learning Techniques for Detecting Insurance Claim Fraud, n.d.). Also, there is a dearth of publicly available real financial transaction data for data scientists to analyze due to privacy concerns related to financial institutions. (Phua, 2010, Lopez-Rojas, 2016, Vadoodparast, 2015).

Credit card issuers have used traditional rule-based system to detect credit card fraud, but the trend is shifting to machine learning. Supervised learning is often used for credit card detection, but unsupervised learning is also used for anomaly detection and thus fraud detection (Bajaj, 2020).

The dataset contains transactions made by credit cards in September 2013 by European cardholders. The dimension of the dataset is 31 features and 284,807 records. For privacy purposes, the dataset contains only numerical input variables that are the result of a PCA transformation. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. The feature 'Class' is the target variable (1 in case of fraud and 0 for non-fraud). The dataset is highly imbalanced with respect to the percent of records with fraud labels (0.17%) versus records with non-fraud labels (99.83%). This imbalance in the target variable was addressed in feature selection but maintained for the unsupervised learning models.

The goal of this project was to use the R programming language to analyze an actual credit card transaction dataset by using unsupervised learning methods to classify the credit card transactions as either fraudulent or non-fraudulent.

Methods

The analysis is following the CRISP-DM stages for data science projects including Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. The

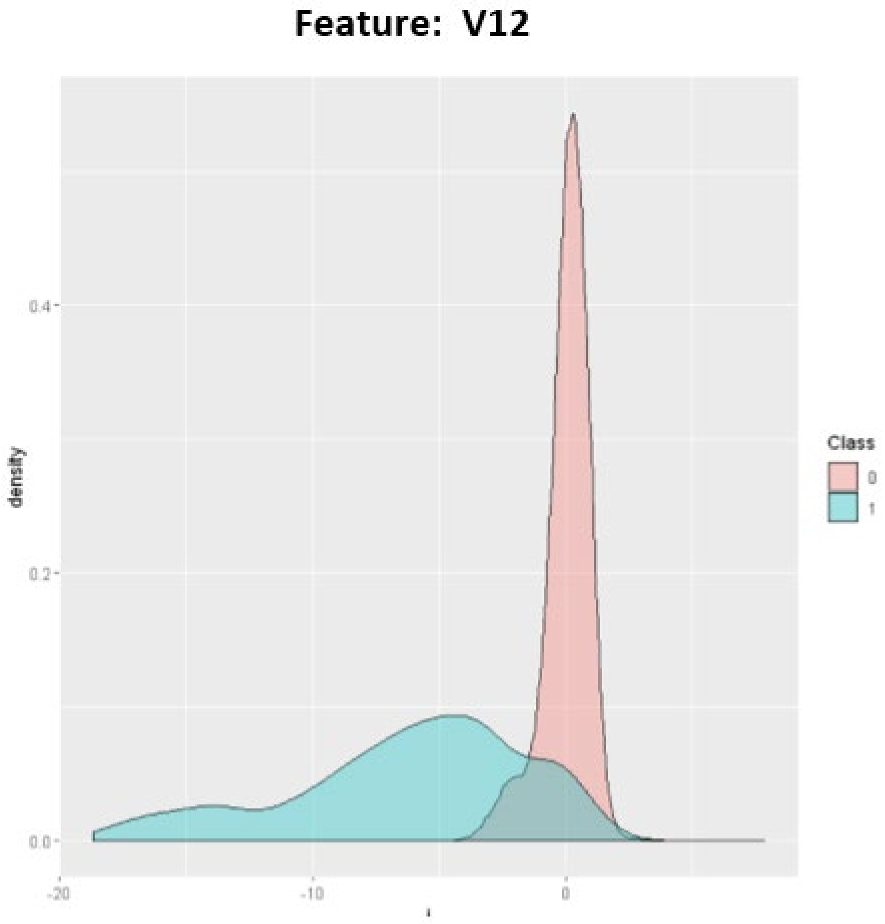
Business Understanding analysis is summarized in the Introduction / Background section above and deployment is beyond the scope of this project. The Data Understanding, Data Preparation, and Modeling, stages are discussed in this Methods section. Evaluation is subsequently discussed below.

Data Understanding

The dataset contains transactions made by credit cards in September 2013 by European cardholders. Appendix A includes full details on the dataset. For privacy purposes, 28 of the 31 features are the result of a PCA transformation. The three non-PCA features are 'Time', 'Amount', and the target feature ‘Class' (1 in case of fraud and 0 for non-fraud). Time' contains the seconds elapsed between each transaction and the first transaction in the dataset and 'Amount' is the transaction amount. As a result of the PCA transformation of 28 features, the actual nature of these features cannot be understood, but these features are independent of each other due to PCA transformation (Bajaj, 2020).

Each record in the dataset also includes a label of either fraud or non-fraud. The Data Understanding analysis utilized this target feature designation to examine differences in features when the target feature was designated as fraud vs non-fraud. We looked at differences in the data distributions and statistics for a subset of the fraud records versus a subset of the non-fraud records.

Point-biserial correlation is used for correlation between the binary target variable and the continuous predictive variables. None of the features have strong point-biserial correlations with the fraud target feature. The correlation statistics among the non-target features showed that the features in the non-fraud subset had very low correlation amongst each other; however, the fraud subset show high correlation between the following features: V3, V9, V10, V12, V16, V17, and V18. The distributions and density plots showed distinctions between the fraud records and the non-fraud records for following features: Amount, Time, V3, V4, V9, V10, V12, V14, V16, V17, AND V18. For example, the below overlapping density plot shows that for the V12 feature shows that the distribution for the non-fraud cases range between -5 and +5 and are mostly centered around point 0. In contrast, the distribution for the fraud cases range from about -18 through +5 and are not as centered around any one single number.



Finally, as previously mentioned, the dataset is imbalanced with respect to the fraud target variable with only 0.17% of the records labeled as fraud. This imbalance in the target variable was addressed in feature selection but maintained for the unsupervised learning models.

Data Preparation

A goal of feature selection is to find a reduced subset of the input features in which maximum redundant and irrelevant information is eliminated (Alam, 2020). The number of highly correlated features were already addressed with the PCA transformation of 28 of the 30 input features.

Feature selection also includes determining which predictive features will be used as inputs in the model. Identifying the key features in a dataset that results in accurate predictions is a goal of the predictive model. As discussed above, the dataset is extremely imbalanced. In order to have feature selection better identify the significant features in the fraudulent cases, we under sampled the data for feature selection but then used the original proportion of fraud vs non-fraud cases for the unsupervised models (Gao, Khoshgoftaar, & Napolitano).

In the modeling stage, one version of the models was tested with key features identified by the following tests/models: recursive feature elimination, learning vector quantization, Boruta, and LASSO regression. The results from these tests consistently identified the following features as significant: V4, V10, V12, V14 and V17. These five features were also identified in exploratory data analysis discussed above and we tested the unsupervised learning models with these five features.

Note, we also tried R's near zero variance test that removes features that have near zero variance but none of the features were identified as having too little variance.

Modeling

Unsupervised learning models used for anomaly detection include One Class Support Vector Machine (SVM), k means clustering, and DBSCAN clustering (Bansal, 2019), (Garbade, 2020). We analyzed the One Class SVM and k means clustering models for this project. We also tried the DBSCAN clustering model.

For each of the One Class SVM and k means clustering models, we tested both the selected subset of 5 features identified above as well as a subset of all the features less three features that were not normally distributed. The three features that were not normally distributed were removed before modeling because each feature should be normally distributed in order to apply the unsupervised anomaly detection algorithm (Bajaj, 2020), (Berhane, n.d.).

Regarding the imbalanced dataset, one of the most important assumptions for an unsupervised anomaly detection algorithm is that the dataset used for the learning purpose is assumed to have an overwhelming percentage of non-anomalous records because the algorithm learns the normal pattern and then flags anomalous activity (Bajaj, 2020). See also (Gao, Khoshgoftaar, & Napolitano) (Garbade, 2020).

Accordingly, the datasets we used for the unsupervised models included the original proportion of fraud vs non-fraud cases. Because the size of the original dataset was too large for our computing capabilities, we used a smaller pro rata sample of the original dataset.

One Class SVM

One Class SVM models are unsupervised learning models because they only train on the positive records in the dataset. Accordingly, the dataset was subset to include only the fraud records. The data was split into training and test data so that we would have unseen data for testing the model. We used cross-validation for training the One Class SVM model and made predictions from the trained model on the training data. Following modeling, we tested the model by making predictions with the test data, which consisted of both positive and negative cases.

In order to analyze how well the One Class SVM model may have performed, we also tested a supervised SVM model on the data. In contrast to the unsupervised models, we needed to address the imbalance in the dataset for the supervised model. The training data was under sampled for the comparison supervised model.

K Means Clustering

We used the Elbow Method to determine the optimal number of clusters for the k means models. The Elbow Method determines the optimal number of clusters by calculating the percentage of variance explained by the clusters via the within-clusters sum of squares. The select features subset showed two clusters as the optimal number and the full normal dataset showed 10 clusters as the optimal number of clusters. We then ran a k means model for both the select subset of features and the full normal dataset.

We analyzed the k means clustering models in three ways: (1) visualization of the clusters, (2) using the clusters from a two-cluster model as the classification labels in a supervised model, and (3) adding the clusters to the dataset as a feature and training a supervised model with the clusters as a feature.

1. Visualization of Clusters

We visualized the results of the k means models by (1) plotting the clusters by classical discriminant coordinates, (2) plotting the clusters against two principal components, and (3) plotting scatterplots of the clusters for each pair of features.

2. Clusters as Classification Labels in Supervised Model

The k means model for the select features subset used only two clusters. As there were only two clusters, we decided to try these clusters as the label classification features in a supervised classification model. For comparison, we also reduced the number of clusters to two for the full normal dataset and tried these clusters as the label classification features in a supervised learning model.

3. Clusters as Input Features in Supervised Model

Including the k means clusters as classification labels in the datasets was mostly an experiment. The standard practice for using the k means clusters in the dataset is to use them as predictive features (Brendel, 2020). We added the optimal number of clusters to both datasets as predictive features and trained a supervised learning model with them.

DBSCAN Clustering

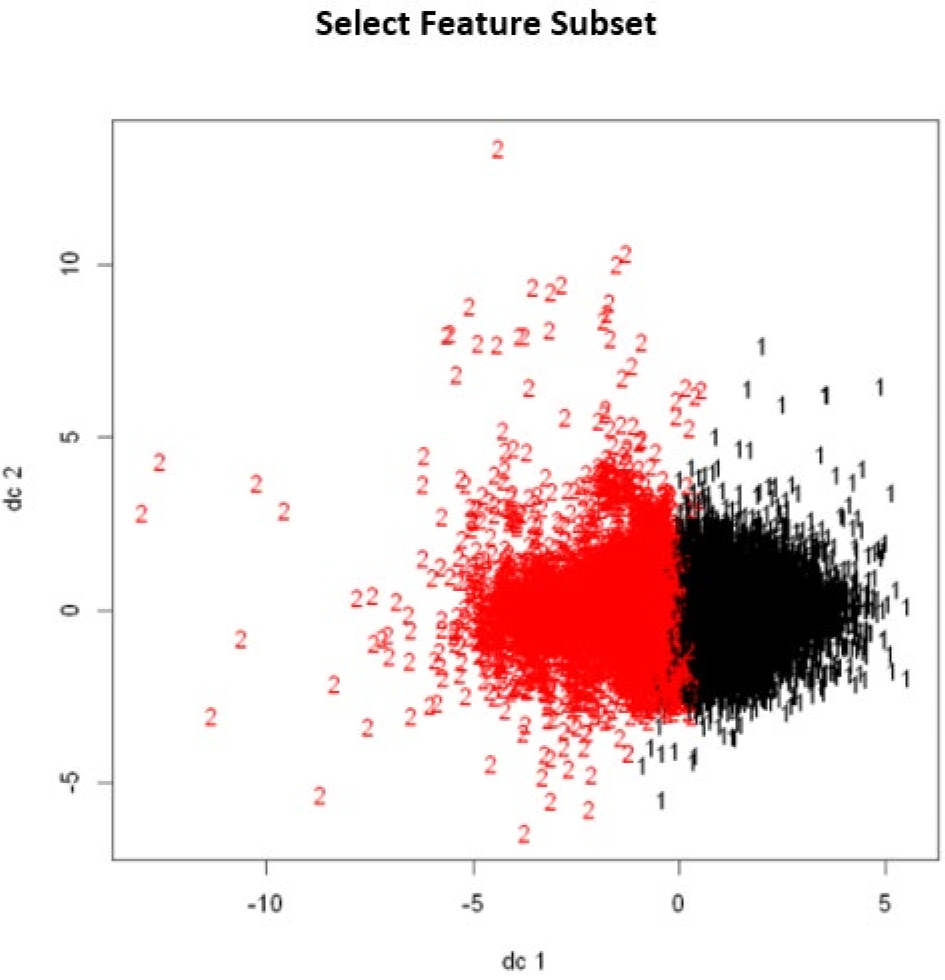
We also looked at DBSCAN clustering and visualized its clusters. The result was very similar to the k means clustering visualizations.

Results

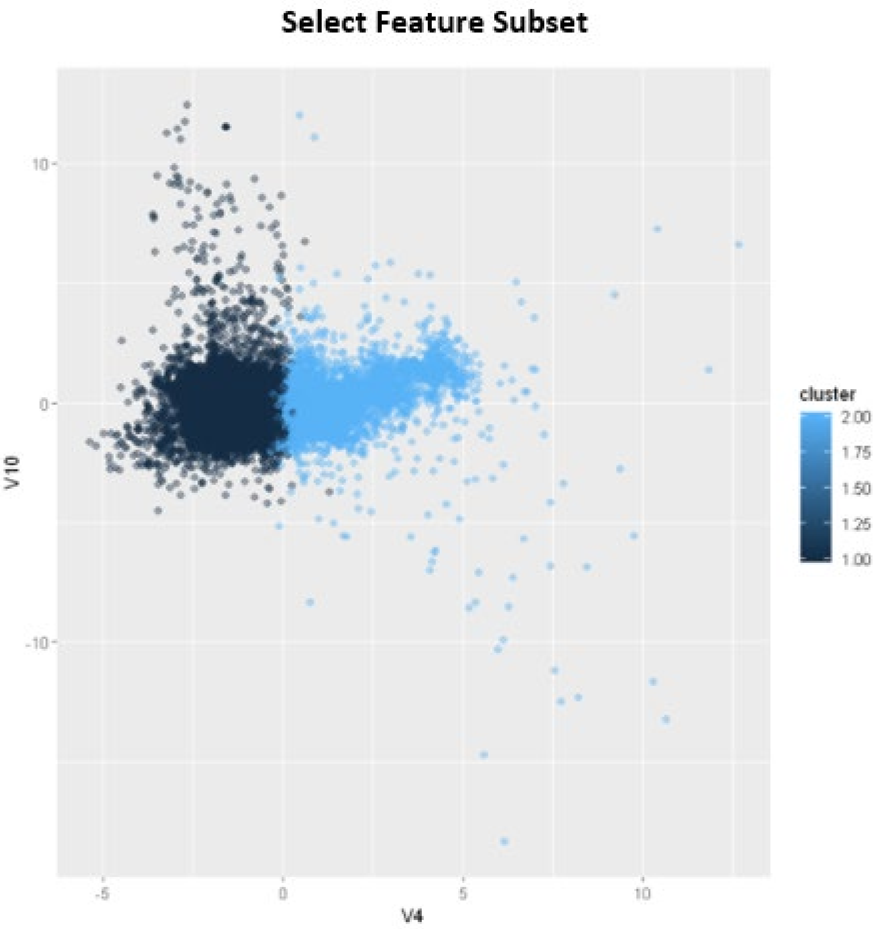
As discussed above, we analyzed the k means models with both visualizations and by inserting the clusters into the datasets and both training classification models on datasets with the clusters and without the clusters (for comparison purposes). A classification model is evaluated by Recall scores when, as in credit card fraud cases, there is a high cost associated with false negative predictions (Rosen, 2020). Recall scores calculate how many of the actual positive default cases are labeled as true positives by the model (Classification: Precision and Recall, n.d.) (Saxena, 2018). We also looked at the F1 score to see the combination of the Precision and Recall error metrics.

K Means Visualizations

The visualization method of analysis was much more effective for the select features subset model because it only used two clusters and five input variables. The plots show marked distinction in the two clusters:



The scatterplots of the clusters for each set of pairs of variables show clear distinctions in the clusters when the V4 feature is one of the two plotted features; however, the scatterplots of feature pairs not involving the V4 feature show highly mixed clusters.



The 10-cluster model for the full normal dataset made visualizing the results more difficult. The plot of the clusters against two principal components does show some distinction in the clusters but overall, most of the points are non-distinctive. Many additional plots would be necessary to plot each pair of the 27 input features to identify cluster distinctions.

As mentioned above, the DBSCAN plots were similar to the k means plots.

K Means Clusters as Classification Labels

The following are the results of the classification model with the cluster results as the classification label and the classification model with the original classification label.

Dataset	Cluster as Label	F1 Score	Recall Score
Full Normal	Yes	.0917	.9587
Full Normal	No	.1112	.9041
Select Features	Yes	.0011	.0112
Select Features	No	.1665	.8400

K Means Clusters as an Input Feature in Classification Model

The following are the results of the classification model with the cluster results as an input predictive feature and the classification model without the cluster as a feature.

Dataset	Cluster as Feature	F1 Score	Recall Score
Full Normal	Yes	.6087	.5000
Full Normal	No	.1112	.9041
Select Features	Yes	.3810	.2667
Select Features	No	.1665	.8400

One Class Support Vector Machine

The following are the error metrics of the Once Class Support Vector Machine.

Dataset	SVM Type	F1 Scores	Recall Scores
Full Normal	One Class	.9119	.9072
Full Normal	Two Class	.0201	.9359
Select Features	One Class	.9239	.8763
Select Features	Two Class	.1710	.8500

Discussion/Conclusion

We can see from the visualizations of the k means cluster plots that used just five features as input features that there is a marked distinction in the two clusters. From the scatterplots for these select feature clusters, we see that the V4 feature appears to be the most distinctions. The visualizations for 28 features are not showing this level of distinction.

When the k means clusters were used as the classification labels in the supervised model, it performed just as well as the supervised model with the original Class labels.

When the k means clusters were used as an input feature in the supervised model, the Recall score was worse than the supervised model without the clusters as input features (.5000 vs .9041). The Precision of the model with clusters as input features was much better than the model without the cluster feature (.6087 vs .1112). However, the cluster model should not be used without improvements to the Recall because not enough fraud cases were identified correctly (Bajaj A. , 2019).

The One Class SVM models' Recall scores were very similar to the supervised (two class) SVM model. The Recall scores for all the SVM models ranged from .8500 to .9359. As such, the fraud and non-fraud actual positives were identified at good rates. Also, the One Class SVM's False Negative rates were only 9% and 12%. This is a favorable result for fraud prediction cases because false negative predictions are costly to the credit card company. The higher Precision score for the One Class SVM indicates that the model did not identify as many non-fraud cases as fraud cases (Bajaj A. , 2019).

In conclusion, we have determined that the unsupervised One Class SVM model is a viable model for predicting credit card frauds. Utilizing the k means clusters as features or labels in supervised models are also viable methods. The clustering visualization method will take more work to become viable prediction methods. We have further identified follow up items to further analyze this project to include (1) tuning of hyperparameters for the models, (2) optimizing the classification thresholds, (3) making probabilistic predictions based on optimizing the classification threshold, (4) implementing a cost matrix for evaluation, (5) trying different resampling methods for feature selection and supervised models, (6) try different approaches to balancing the dataset, and (6) improving computing capacity to be able to employ the full dataset.

References

Alam, T. M. (2020). An Investigation of Credit Card Default Prediction in the Imbalanced Datasets. *IEEE Access*, Vol. 8 201173-201198.

Bajaj, A. (2019, December 25). *What does your classification metric tell about your data*. Retrieved from Towards Data Science: https://towardsdatascience.com/what-does-your-classification-metric-tell-about-your-data-4a8f35408a8b#:~:text=Now%2C%20a%20high%20F1%2Dscore,about%20performance%20at%20a%20threshold.

Bajaj, V. (2020, August 8). *Unsupervised Learning For Anomaly Detection*. Retrieved from Towards Data Science: https://towardsdatascience.com/unsupervised-learning-for-anomaly-detection-44c55a96b8c1

Bansal, H. (2019, October 16). *Credit Card Fraud Detection: Neural Network vs. Anomaly Detection Algorithms*. Retrieved from Analytics Vidhya: https://medium.com/analytics-vidhya/credit-card-fraud-detection-c66d1399c0b7

Berhane, F. (n.d.). *Anomaly Detection with R*. Retrieved from Data Science Enthusiast: https://datascience-enthusiast.com/R/anomaly_detection_R.html

Brendel, C. (2020, February 10). *Cluster-then-predict for classification tasks*. Retrieved from Towards Data Science: https://towardsdatascience.com/cluster-then-predict-for-classification-tasks-142fdfdc87d6

Classification: Precision and Recall. (n.d.). Retrieved from Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

Comparative Analysis of Machine Learning Techniques for Detecting Insurance Claim Fraud. (n.d.). Retrieved from wipro: https://www.wipro.com/en-US/analytics/comparative-analysis-of-machine-learning-techniques-for-detectin/

(2021). *Consumer Sentinel Network Data Book 2020.* Federal Trad Commission.

Gao, K., Khoshgoftaar, T., & Napolitano, A. (n.d.). Combining Feature Subset Selection and Data Sampling for Coping with Highly Imbalanced Software Data.

Garbade, M. (2020, December). *How to use Machine Learning for Anomaly Detection and Conditional Monitoring*. Retrieved from https://www.kdnuggets.com/2020/12/machine-learning-anomaly-detection-conditional-monitoring.html: https://www.kdnuggets.com/2020/12/machine-learning-anomaly-detection-conditional-monitoring.html

Rosen, D. B. (2020, August 1). *How To Deal With Imbalanced Classification, Without Re-balancing the Data*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/how-to-deal-with-imbalanced-classification-without-re-balancing-the-data-8a3c02353fe3

Saxena, S. (2018, May 11). *Precision vs Recall*. Retrieved from medium.com: https://medium.com/@shrutisaxena0617/precision-vs-recall-

386cf9f89488#:~:text=While%20precision%20refers%20to%20the,correctly%20classified%20by%20your%20algorithm.&text=For%20problems%20where%20both%20precision,maximizes%20this%20F%2D1%20score.

Appendix A

Here is the link to get to the dataset: https://www.kaggle.com/mlg-ulb/creditcardfraud

- The dataset contains transactions made by credit cards in September 2013 by European cardholders. For privacy purposes, it contains only numerical input variables that are the result of a PCA transformation.
- Features V1, V2, … V28 are the principal components obtained with PCA.
- The only features which have not been transformed with PCA are 'Time' and 'Amount'.
    - The feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset.
    - The feature 'Amount' is the transaction Amount.
- The feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.