Assignment 4

Jonah Muniz

Abstract

The objective of this research is to determine which unsupervised learning method detects fraud transactions by sales representatives the best. The unsupervised learning methods that will be compared are Local Outlier Factor score (lof) and Isolation Forest (iForest). Overall accuracy and F1 will be the metrics both algorithms will be evaluated on. The best performing algorithm will then be used to flag transactions reported by sales representatives as fraud.

Key Words: Anomaly detection, Fraud detection, Unsupervised Learning, Retail, Sales Representative Transactions

Introduction

Anomaly detection is an important capability for businesses across industries. For example, within banking being able to label each credit card transaction with an anomaly score is key to flagging potential fraud transactions. Within the retail industry being able to give each transaction by a sales representative an anomaly score is key to cutting down on fraud within a store by sales representatives. Reducing fraud transactions will save business money as well as build customer trust. The two main algorithms used in practice to assign anomaly scores to transactions are lof and iForest. Lof considers the distance of the neighbors around the observation to determine if its an outlier or not. Lof then assigns an anomaly score for each observation. IForest is a tree-based algorithm. Based on the average depth of each observation an anomaly score is assigned, the smaller the average depth the higher the anomaly score. Both algorithm's performance will be evaluated looking at accuracy and F1 score. The better performing algorithm would then be recommended to leadership to implement to flag fraud transactions.

Literature Review

The topic of anomaly detection is a well research subject. In the past anomaly detection algorithms look to model what "normal" occurrences look like and then they assign an anomaly score based on how much an observation resembled a normal observation. This approach to anomaly detection has some disadvantages. For instance, the algorithms are optimized to profile normal instances, not to detect anomalies, this leads to too many false alarms or too few anomalies being detected. Another disadvantage to the existing clustering-based methods is that they are limited to low dimensional data and small data sizes because of the high computational complexity which is not ideal when working with transactional data at scale. An alternative to the clustering-based anomaly detection is Isolation Forest (Iforest). This method builds an ensemble of iTrees for a given data set and then anomalies are those instances in which have a short average path length on the iTrees (Liu, Fei Tony 2016). As shown in the cited research. iForest outperformed common distance-based methods like Random Forest and Local Outlier Factor in AUC and execution time, especially with large datasets. A very similar approach will be taken in this research to discover if iForest performs better in regard to accuracy and F1 score compared to Local Outlier Factor when trained, test, and evaluated using the sales representative transaction dataset.

Another piece of research that was conducted that is related to the research being performed in the paper was done by Lin Xu, Yi-Ren Yeh, Yuh-Jye Lee and Jing Li (Xu, Lin, Yi-Ren Yeh, Yuh-Jye Lee, Jing Li 2016). In their research they looked to see if combining iForest and LOF would increase the overall performance of the anomaly detection algorithm when compared to both iForest and LOF alone. The application of their research was to detect anomalies in wireless sensor networks. The challenge for this type of application is that the data set is very large and complex leading to LOF alone not performing well. Anomalies within the wireless network are also considered local not global anomalies. This leads to iForest not performing well either. Lin and company were able to configure an ensemble anomaly

detection algorithm that first runs the data through an iForest algorithm, assigned anomaly scores to observations, then runs the high scoring observations through LOF. Lin and team were able to show that the ensemble algorithm was more efficient in computation and storage comparing to the individual methods. Although this research was applied to a different application then this research, understanding how the pros and cons of each algorithm can be negated by combining approaches is very applicable.

Methods

In order to evaluate which unsupervised learning method detects anomalies in sales representative transaction data the best the sales representative transaction data must be ingested in R. The sales representative transaction dataset consists of two different data sets, train and test. The train dataset consists of the sales representative id, product name, quantity sold, value and inspection value. All inspection values in the train data set are set to unknown. The test dataset contains the same five features, but the inspection value feature contains either ok or fraud. Now that both datasets have been ingested into R the necessary R libraries can be loaded.

Now that the datasets and R libraries have been loaded, data preprocessing can be performed prior to algorithm training. Summary statistics for each data set will be calculated to understand the min, max, mean and quartile values for each feature. If a feature contains any rows with NA this information can also be detected when calculating the summary statistics. If NA values exist within a feature, the rows should either be deleted, or the missing value should be imputed. The decision to either delete or impute the values for rows containing NA's depends on how many rows contain NA's and if removing the rows will skew the analysis. After all rows with NA are addressed each dataset should be analyzed to see if there are any duplicated rows. Each row will need to be unique to ensure proper anomaly detection. Now that each dataset no longer contains NAs or duplicated rows the product quantity and value feature can be normalized.

If the range of the product quantity and value listed is large, a min-max normalization can be used to normalize these features improving overall model performance. Product name can also be converted to a numeric value to allow product name to be incorporated into the models. Converting product name to a numeric value now allows both iForest algorithm and lof to leverage product name, quantity, and value to categorize sales as either ok or fraud.

Now that all data preprocessing of the train and test dataset are completed both algorithms can be trained and tested. First the iForest algorithm can be defined and trained. The iForest algorithm will consist of 500 trees, a variable that can be altered if needed. Once the iForest algorithm has been constructed the algorithm will be trained on product id, product quantity and product value. Once the iForest algorithm has been trained inspection values can be predicted for each observation. An anomaly score and average depth length will be predicted for each observation in the train dataset. A cutoff value for anomaly score will then be used to classify whether an observation is okay or should be labeled fraud. The overall accuracy of the iForest algorithm will be tested by predicting inspection values for the test dataset using the trained iForest algorithm and comparing the predicted inspection values to the actual values in a confusion matrix. Once this process has been completed the lof algorithm can be used to assign predicted inspection values.

The lof algorithm can now be used to identify values who have a large distance to their neighbors and assign an overall outlier value. The lof algorithm is run on the test set in order to compare predicted outlier values to actual. A cutoff value is selected for outlier value in order to classify test observations as either ok or fraud. A confusion matrix is then created to understand the overall accuracy of the lof function in detecting anomalies in sales representative sales. Now that both iForest and lof accuracies have been evaluated on the sales representative datasets the overall objective of understanding which algorithm performs the best can be concluded.

Results

Following the above methodology, the train and test datasets of the sales representative data was ingested into R and the necessary R libraries for the analysis were loaded into the R session. Once the datasets were ingested in R the summary statistics for both datasets were calculated. As can be seen in the appendix as Figure 1, both datasets contain rows with NAs present in product quantity and product value. The largest amount of NA values was present in the training dataset for product quantity with 4,479 rows containing NA. The overall train dataset contains 133,731 observations so that means ~3.35% of the values in the dataset contained NA values. Due to such a low percentage containing NA values it was decided that all rows containing NA values in either product quantity or value can be removed from the test and train datasets. This process removed 4,640 rows from the train dataset and removed 186 rows from the test dataset. Next both datasets were evaluated to see if any rows were duplicated. If a row was duplicated it was removed from the dataset. This further reduced the train dataset by 26,050 leaving a total of 103,041 observations in the training dataset. The test dataset was reduced by 2,468 leaving a total of 13,078 observations in the test dataset. Now that all data cleansing activities are complete the product names can be converted to numeric values to enable the incorporation of the product into both algorithms. The product quantity and product value features were also normalized using the min-max normalizer.

Now that there is a fully transformed and cleansed train and test dataset the iForest algorithm can be constructed. The iForest algorithm will be consist of 500 trees. The iForest algorithm once built was then trained on three features of the training set, product id, product quantity and product value. Once the iForest algorithm was trained anomaly score and average depth values were predicted for each observation. The lower the average depth of an observation in the tree algorithm the higher the anomaly score. The summary statistics of the average depth and anomaly scores for each value for the train dataset can be seen in the appendix as Figure 2. Based on the summary statistics, all observations

with an anomaly score greater then 0.5887 were assigned fraud while the rest were assigned as ok. A scatter plot of product number and anomaly score with the iForest prediction value as the coloring can be seen in the appendix as Figure 3. As can be seen in Figure 3, there are clear observations that have an abnormally high anomaly score for the product number which lead to them being labeled fraud. Now that we have trained the iForest algorithm, the same algorithm can be used to predict average depth and anomaly scores for the test dataset. The same cutoff value of 0.5887 was used to classify whether a test observation was fraud or ok. A confusion matrix was then constructed to compared how the predicted values of fraud and ok compared to the actual values. The confusion matrix can be seen in the appendix as Figure 4. As can be seen in Figure 4, the iForest algorithm was able to detect fraud and okay sales representative transactions with 78.08% accuracy. The fscore for the iForest algorithm was 0.8764. Now that the accuracy and fscore of the iForest has be calculated the accuracy and fscore of the lof algorithm can be calculated.

The lof algorithm is an unsupervised learning method that assigns an anomaly score based on an observations distance to its neighbors. The higher the anomaly score the higher the probability the observation is an outlier. The lof algorithm was run on the train dataset on the same three features as was run for the iForest algorithm. An observation was considered fraud if it had an anomaly score higher then 2. A scatter plot can be seen in the appendix as Figure 5 showing the relationship between product number, anomaly score colored by the predict class. As can be seen in Figure 5, the scatter plot is slightly skewed due to one observation having a very high lof score compared to the others. The test dataset was then run through the lof algorithm and the same cutoff value of 2 was used to classify fraud observations. A confusion matrix comparing prediction to actual can be seen in the appendix as Figure 6. As can be seen in Figure 6, the lof algorithm was able to detect fraud and okay sales representative transactions with 83.71% accuracy. The fscore of the lof algorithm was 0.9057.

Conclusions

The overall objective of the research was to determine if the iForest algorithm or the lof algorithm performed better at detecting fraud transactions by sales representatives. As can be seen in the results section above, when comparing the two algorithms performance the lof algorithm outperformed the iForest algorithm in both accuracy and F1 score. One reason for why the lof algorithm outperformed the iForest algorithm is due to the local aspect of the lof algorithm. The score of abnormality only takes into consideration the scores of the nearest neighbors. This would enable quantity and value scores for the same products to be compared to one another and quantity and values scored for other products are not considered. Based on this results and how the two algorithms differ in how they evaluate anomaly observations I would recommend that lof be used to classify whether a sales representative transaction is ok or is fraud based on the advantages of comparing transactions to local transactions.

Appendix

```
> summary(dataset_train)
      ID                Prod               Quant              Val               Insp          
 Length:133731      Length:133731      Min.   :     100   Min.   :   1005   Length:133731     
 Class :character   Class :character   1st Qu.:     109   1st Qu.:   1420   Class :character  
 Mode  :character   Mode  :character   Median :     185   Median :   2885   Mode  :character  
                                       Mean   :    6644   Mean   :  12571                     
                                       3rd Qu.:     887   3rd Qu.:   8730                     
                                       Max.   :11011711   Max.   :3913920                     
                                       NA's   :4479       NA's   :516                         
> summary(dataset_test)
      ID                Prod               Quant               Val               Insp          
 Length:15732       Length:15732       Min.   :      100   Min.   :   1005   Length:15732      
 Class :character   Class :character   1st Qu.:      110   1st Qu.:   1260   Class :character  
 Mode  :character   Mode  :character   Median :      446   Median :  12595   Mode  :character  
                                       Mean   :   108132   Mean   :  63337                     
                                       3rd Qu.:     6010   3rd Qu.:  55908                     
                                       Max.   :473883883   Max.   :4642955                     
                                       NA's   :140         NA's   :49
```

Figure 1: *Summary statistics of both the Train and Test dataset*

```
> summary(scores_train)
       id         average_depth   anomaly_score   
 Min.   :     1   Min.   :2.358   Min.   :0.5820  
 1st Qu.: 25761   1st Qu.:7.832   1st Qu.:0.5827  
 Median : 51521   Median :7.956   Median :0.5837  
 Mean   : 51521   Mean   :7.760   Mean   :0.5920  
 3rd Qu.: 77281   3rd Qu.:7.982   3rd Qu.:0.5887  
 Max.   :103041   Max.   :8.000   Max.   :0.8525
```

Figure 2: *Summary statistics of the training set average depth and anomaly scores for the iForest algorithm*

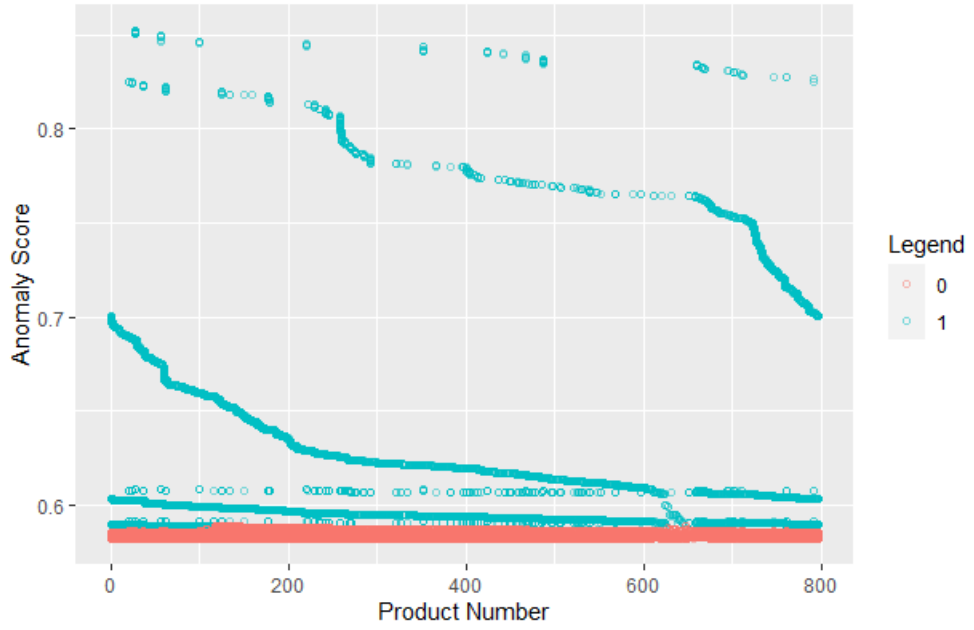

Figure 3: *Scatter plot showing the product number vs anomaly score with the prediction value as the coloring, 1 = fraud*

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10166  1153
         1  1714    45

               Accuracy : 0.7808
                 95% CI : (0.7736, 0.7878)
    No Information Rate : 0.9084
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0882

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.85572
            Specificity : 0.03756
         Pos Pred Value : 0.89814
         Neg Pred Value : 0.02558
             Prevalence : 0.90840
         Detection Rate : 0.77734
   Detection Prevalence : 0.86550
      Balanced Accuracy : 0.44664

       'Positive' Class : 0
```

Figure 4: *Confusion matrix of the iForest Algorithm and the Test dataset*

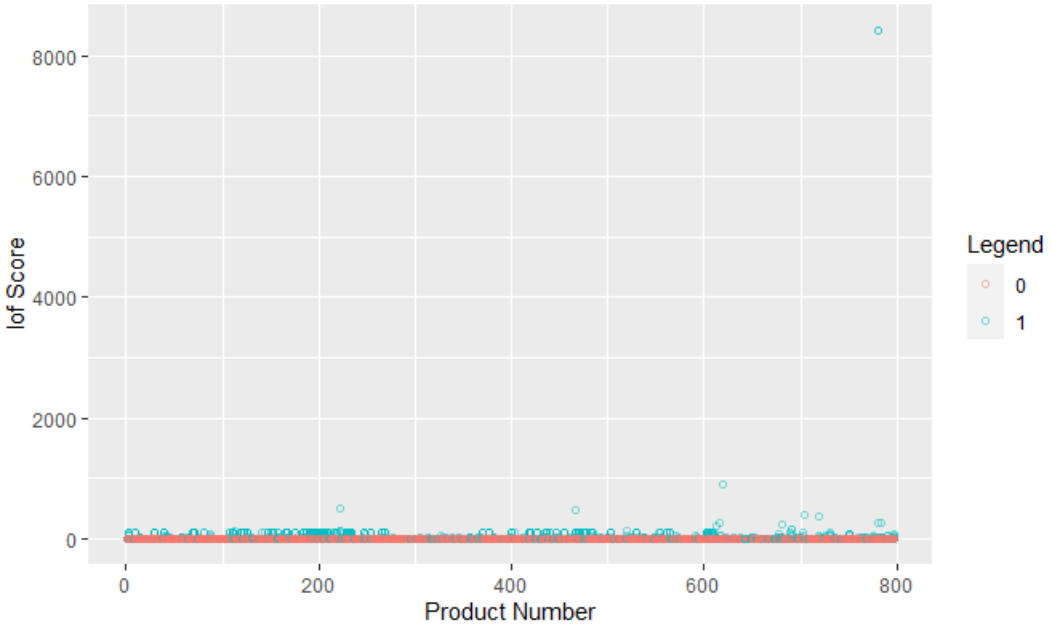

Figure 5: *Scatter plot showing the product number vs lof score with the prediction value as the coloring, 1 = fraud*

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 10229   479
         1  1651   719

               Accuracy : 0.8371
                 95% CI : (0.8307, 0.8434)
    No Information Rate : 0.9084
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3203

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8610
            Specificity : 0.6002
         Pos Pred Value : 0.9553
         Neg Pred Value : 0.3034
             Prevalence : 0.9084
         Detection Rate : 0.7822
   Detection Prevalence : 0.8188
      Balanced Accuracy : 0.7306

       'Positive' Class : 0
```

Figure 6: *Confusion matrix of the lof Algorithm and the Test dataset*

References

1. Tony Liu, Fei. “Isolation Forest - NJU.EDU.CN.” Accessed November 22, 2021. https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf.
2. Xu, Lin, Yi-Ren Yeh, Yuh-Jye Lee, and Jing Li. “A Hierarchical Framework Using Approximated Local Outlier Factor for Efficient Anomaly Detection.” Procedia Computer Science. Elsevier, June 24, 2013. https://www.sciencedirect.com/science/article/pii/S187705091300776X.