Delft University of Technology

Cyber Data Analytics
CS4035

Lab 1

Authors:
Josefin Ulfenborg (4989988)
Wouter Zirkzee (4398858)

May 10, 2019

1 Visualization

After examining the provided data, most did not seem very interesting to use. Two stood out to us, the relationship between the how many e-mails are used for each card, as well as how many IP addresses exist per card used. The first visualization can be seen in figure 1. Likewise, in figure 2 is the code and scatter plot for the second visualization respectively. For both examples, *"data"* is *"data_for_student_case.csv"*.

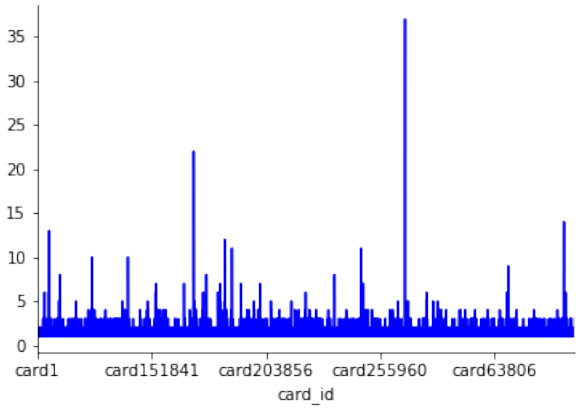

Figure 1: Scatter plot for number of e-mails used per card ID.

Most of the card IDs have been used by around three e-mail addresses, but one of them have been used by over 35, and another by almost 25. Most likely these are fraudulent cases as it does not seem reasonable a card should be used for this amount of e-mail addresses, and should give a cause for concern.

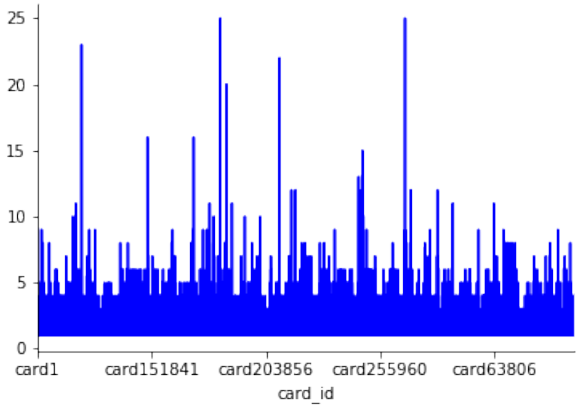

Figure 2: Scatter plot for number of IP addresses per card ID.

Similar to the chart above, most cards have been used at 5-10 different IP addresses, but some at 25 IP addresses.

These outliers could prove to be a good indication for fraud detection.

2 Imbalance Task

Prior to using SMOTE on the data, the number of legit cases was 174018 and fraudulent 211. After applying SMOTE, the number of legit cases and fraudulent were both 174018, showing a more balanced data set.

For ROC and PR analysis we used the following four classifiers: k-Nearest Neighbors (k=5), Decision Tree, Random Forest and AdaBoost. For all of them we used the *sklearn* library in Python. Figure 3 shows the ROC and PR curves before and after applying SMOTE, respectively, for KNN. Same structure applies to figure 4 for Decision Tree, figure 5 for Random Forest and finally figure 6 for AdaBoost. A Support Vector Machine has also been considered, but due to the runtime complexity this classifier also been dismissed.

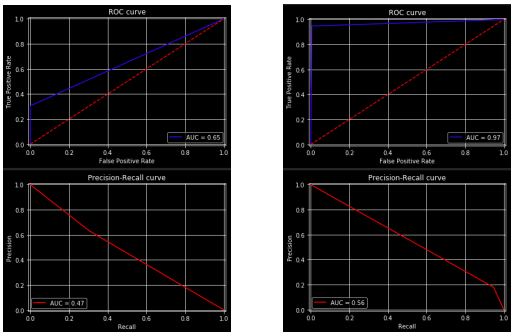

Figure 3: ROC and PR curve for KNN, before (left) and after (right) SMOTE.

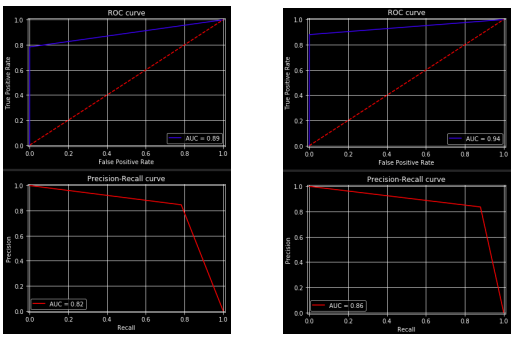

Figure 4: ROC and PR curve for Decision Tree, before (left) and after (right) SMOTE.

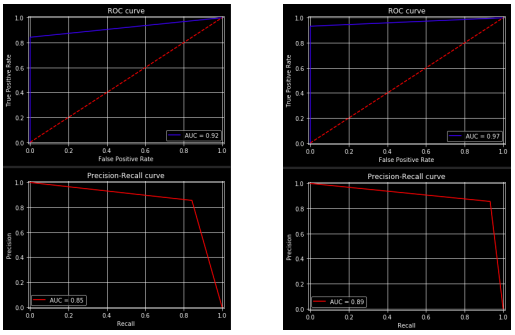

Figure 5: ROC and PR curve for Random Forest, before (left) and after (right) SMOTE.

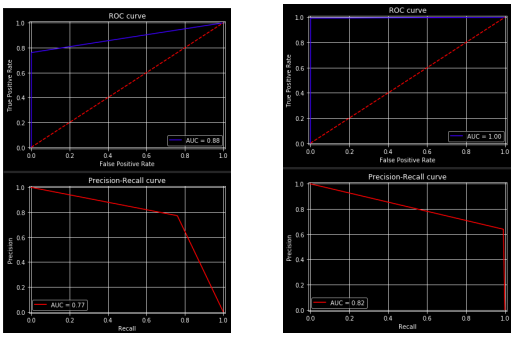

Figure 6: ROC and PR curve for AdaBoost, before (left) and after (right) SMOTE.

The ROC curve for KNN was the one most improved after SMOTE was applied to the data, but the PR curve still remains rather poor. Moreover, we would argue that the PR curve is of more interest than ROC when analyzing fraudulent card transactions, as we find *false positives* more important than *true negatives*. If a card transaction is falsely labeled as a fraud, then the card will be blocked for the actual card holder and they will not be able to use it. As the PR curve contains *precision*, which is the fraction of examples classified as positives that were actually positives, this will be more relevant for fraudulent card transactions. With this argument, the Random Forest Classifier performs best.

Using SMOTE is one way to deal with imbalanced data, and is specially good in case the performance of the algorithm is affected by imbalanced data. However, the way SMOTE is constructed may also be a problem, as the generated instances of the minority class may overlap too much with other classes.

3 Classification Tasks

3.1 Pre-Processing and features

For our implementation there is no pro-processing involved, besides factorizing the strings (i.e. country codes, mail ids, etc) as features need to be number based for the sci-kit implementation. We used the following fields as features:

- amount
- currencycode
- issuercountrycode
- shoppercountrycode
- shopperinteraction
- txvariantcode
- cardverificationcodesupplied
- cvcresponsecode
- accountcode
- mail_id
- ip_id
- card_id

3.2 Learning Algorithms

We used the previously listed features to train both a white-box algorithm and a black-box algorithm. For the white-box algorithm we have chosen a Decision Tree Classifier. As we observed during the imbalance task, the Random Forest Classifier seemed to perform best of the ones we evaluated.

3.3 Training and Evaluation

The are evaluated by using a stratified 10-fold. A stratified k-fold ensures that each fold has the same distribution of classes labeled as fraud and valid in the samples. The training data is then expand using *SMOTE* to balance the classes before training. Then the models are trained, all parameters are set to the default of sci-kit version 0.20.3, and used to predict the labels of the test dataset. In order to evaluate the predictions made by the classifiers a number of metrics are used: *accuracy*, *precision*, *recall* and the confusion matrix.

3.4 Results

Both the models have shown to obtain very good scores as shown in table 1, 2, 3. These high results sparked our interest to see how much SMOTE actually contributed to this performance. The experiment was repeated without applying SMOTE to balance the test data to obtain the results in table 4, 5, 6. From this we can conclude that SMOTE does not provide a significant increase in this case. Figure 7 shows the decision tree created in order to make classifications. While quite large and hard to decipher, the most important combination seems to be the combination of merchant and the amount.

	Decision Tree	Random Forest
Accuracy	0.9996	0.9997
Precision	0.8757	0.8847
Recall	0.8377	0.8896

Table 1: Obtained accuracy, precision and recall by the Decision Tree and Random Forest using SMOTE

		Predicted	
		Valid	Fraud
Actual	Valid	289997	40
	Fraud	50	295

Table 2: Confusion matrix obtained using a stratified 10 fold prediction using a Decision Tree Classifier with SMOTE

		Predicted	
		Valid	Fraud
Actual	Valid	289997	40
	Fraud	38	307

Table 3: Confusion matrix obtained using a stratified 10 fold prediction using a Random Forest Classifier with SMOTE

	Decision Tree	Random Forest
Accuracy	0.9996	0.9997
Precision	0.8650	0.8798
Recall	0.8174	0.8492

Table 4: Obtained accuracy, precision and recall by the Decision Tree and Random Forest without SMOTE

		Predicted	
		Valid	Fraud
Actual	Valid	289993	44
	Fraud	63	282

Table 5: Confusion matrix obtained using a stratified 10 fold prediction using a Decision Tree Classifier without SMOTE

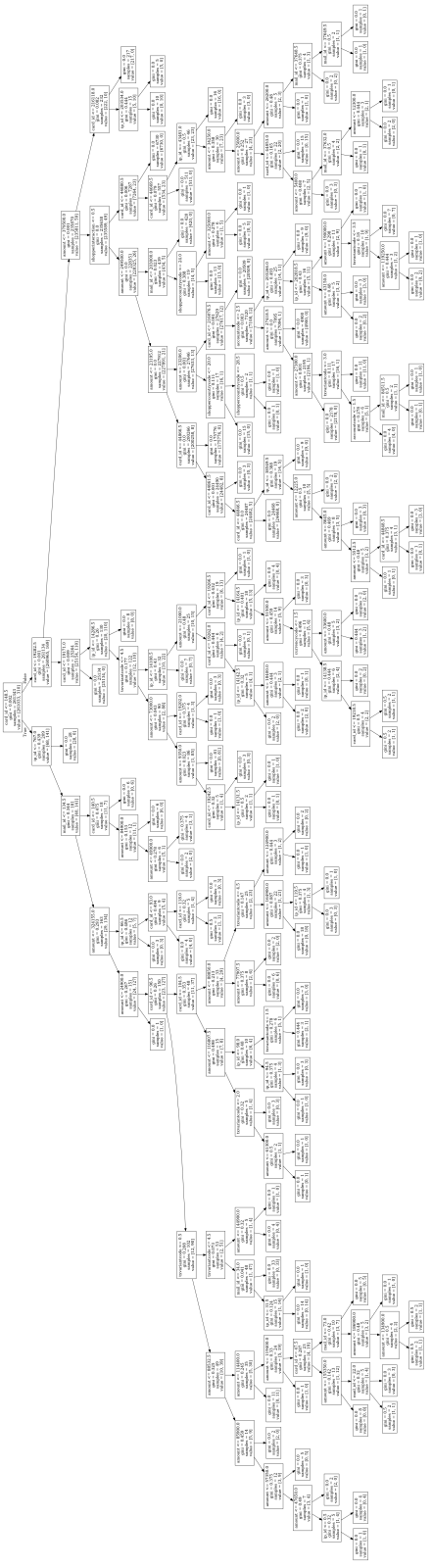

Figure 7: Visualization of the Decision Tree classifier

		Predicted	
		Valid	Fraud
Actual	Valid	289997	40
	Fraud	52	293

Table 6: Confusion matrix obtained using a stratified 10 fold prediction using a Random Forest Classifier without SMOTE