CS4035 Cyber Data Analytics: Assignment 1
Fraud Detection
Group 9

Georgios Dimitropoulos: 4727657
Emmanouil Manousogiannis: 4727517

May 13, 2018

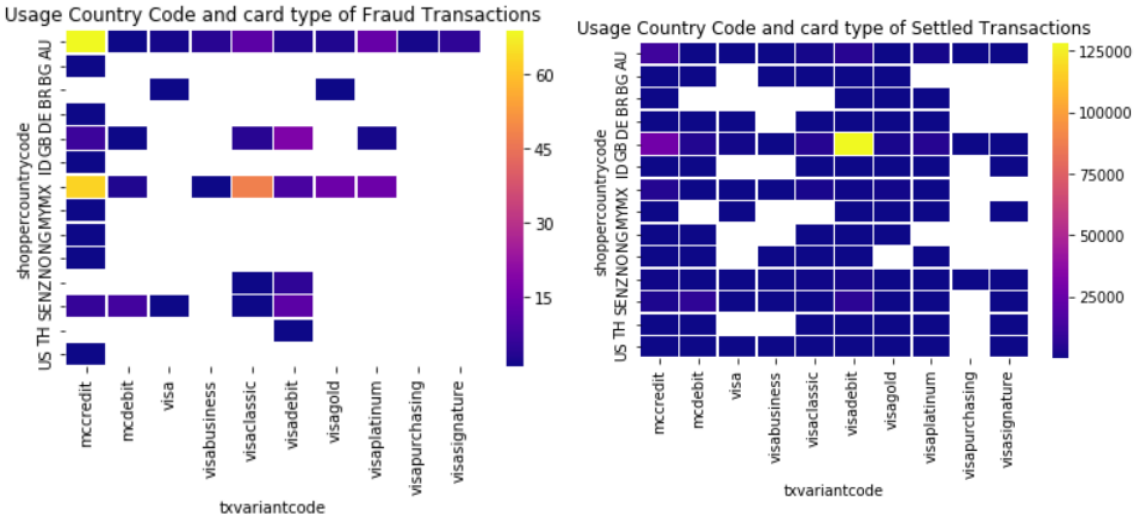
In this Assignment we will work on Fraud card transaction detection using machine learning techniques.Fraud card transaction detection is a typical case that we need to handle imbalanced data. This means that in our binary classification problem, the available samples of one class (fraud transactions) are a very small percentage of the total dataset, while the other class samples (normal transactions) are a majority.

1 Visualization task

In the first task of this assignment we tried to visualize a very useful relation between our data samples in terms of detecting fraud.Since the dataset is quite large in terms of features,as every transaction row in our dataframe has 17 different columns, this might be quite helpful.

Preprocessing:

Initially we performed some simple preprocessing steps in order to easier handle our data and make the classification procedure that will follow easier.Initially we **removed all transactions that were labeled as refused** as they were neither fraud or settled, and then we converted all features (columns) of each transaction to the **appropriate format(i.e. categorical, dateTime)** in order to be able to handle them. Finally we **converted all amounts to the same currency** (GBP) so that there are no inconsistencies. Below , we are presenting some interesting heatmaps and boxpots.



As can be seen in the figure above, there are too many fraud transactions taking place in Australia with the use of mccredit cards.Similarly in Mexico with the use of the same card type, as well as the use of visaclassic cards, the number of frauds is high.On the contrary, normal settled transactions have relatively low number of transactions in those countries with the use of those cards. This can probably be a good indicator of detecting a possible fraud. Apart from the heatmap, we also present an interesting comparison of the total amount of the fraud transactions against the normal ones.As can be easily noticed the amounts of fraud transactions are quite higher.

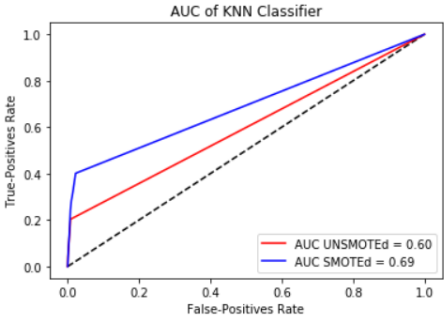
Figure 1: Boxplot of GBP spent on fraud and settled transactions

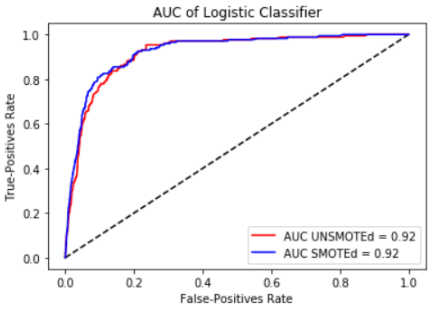
2 Imbalance task

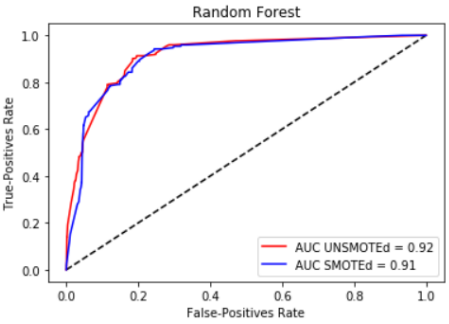
In this task we will try to handle the imbalanced data issue by performing SMOTE.Since we have very few samples of true positive (fraud) samples, we will try to use SMOTE to synthetically oversample this minority class.SMOTE takes each minority class sample and creates synthetic examples along the line segments joining the k minority class nearest neighbors.Depending on the amount of the oversampling needed, we can adjust the number of k neighbors considered.
In order to apply SMOTE we need to add further **pre-process** our data. Apart from the steps mentioned in section 1(removing unnecessary columns, converting to appropriate type etc) we need to implement the following steps.

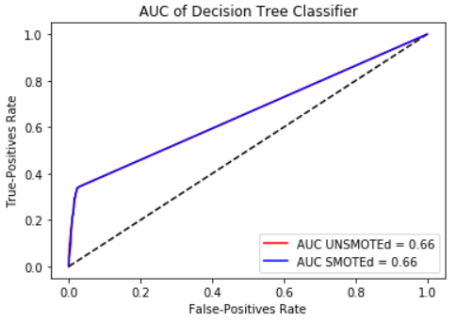
First, we need to take an informative subset of our dataframe,leaving out columns such as mail-id , ip-id and card-id. In addition, we set the $'simplejournal$ value to 1 if the transaction is fraud and 0 if it is settled.After that, we split the labels column (*simplejournal*) from the rest of the dataFrame, so that we can pass it to the classifiers, as a feature vector and a labels vector respectively.Now the next step is to take this vector and for each column create as many columns as the values in that column. For instance, regarding issuer country code, we will have one column for each country that issued a card.The values of that columns will be 1 if this characteristic is present in one row (transaction) and 0 otherwise.Finally, we will split this dataframe in two halves.The first half will be used as a training set and the rest as the test set for each one of our classifiers.

In total we tested 4 different classifiers and compared their performances. Those are kNN,logistic classifier,one decision tree and Random Fosrest classifier. We trained the classifiers on the initial training set, tested their performance on the test set and after that we performed SMOTE on the training set and tested again it again.
After various experimentations we concluded that a reasonable percentage of fraud data after SMOTE is a total of 20 % of the existing non-fraud data samples. The results of our experiments are displayed in the figures below.









As can be seen in the figures, logistic regression classifier and Random Forest perform much better than kNN and our decision tree classifier.However, despite the fact that their performance is really high SMOTE does not seem to help in further improvement. On the contrary, kNN, with k set to 7,which is a simpler model seems to improve its performance after SMOTE is applied to the training set.kNN is a relatively simple model and seems to improve a lot with SMOTE,while Random Forest and logistic are more complex and are more sensitive to over fitting,which can be possibly caused by SMOTE.

3 Classification task

3.1 Black-box algorithm

In this question we built one classifier which is used for fraud detection. The classifier which we employ in this case is a Random Forest classifier. Random forest model [1],[2] is an ensemble of classification decision trees. It is preferred in comparison to decision trees as, single tree models may be unstable and more sensitive to specific training data. Thus, Random forest as an ensemble method strives to address this problem through aggregating various models which it uses in its class labels predictions. Moreover, Random forests combine the concepts of bagging and random subspace method. In the first one, individual models in an ensemble are developed through sampling with replacement from the training data. Whereas, in the latter one each tree in an ensemble is built from a random subset of attributes. Based on this, Random forests are computationally efficient as each tree is built independently of the others and it is worth mentioning that providing a large number of trees in the ensemble are robust to overfitting and noise in the data.
Our main intuition behind the choice of the Random Forest classifier is its superior performance in similar works [3] (i.e. credit card fraud detection) in comparison to other classifiers. Furthermore, based on the difficulty in the explainability of the function modeled by this machine learning scheme and in combination with its superior performance, (as we are focusing explicitly in the performance in this question) we decided to employ the Random Forest classifier as a Black Box algorithm.
In order to be able to implement this classification algorithm, a Machine Learning pipeline was employed. Initially, the aforementioned pre-processing steps were used in order to be able to process our data in a meaningful way. After the successful completion of our processing steps, we ended up with the nine most discriminative features. This technique facilitates our effort in achieving better results in the evaluation metrics which we used.
In the sequel, we built the Random Forest classifier and we chose a selection in its parameters which it takes after experimentation and we identified the best combination of them. Finally, we performed 10-fold cross-validation in order to obtain performance criteria that are relevant in practice. Namely, we calculated the True Positives, True Negatives, False Positives, False Negatives and evaluation metrics such as Accuracy, Precision, Recall, F-measure and the AUC score. We performed this analysis for two cases (SMOTED and UNSMOTED), where we employed or not the SMOTE algorithm respectively.

3.2 White-box algorithm

In this question we built one white-box classifier used for fraud detection. The classifier which we employ in this case is a Logistic Regression classifier. In this work, our dependent variable 'fraud', is binary, and logistic regression is a widely used technique in similar problems like studying fraud [4]. Logistic regression algorithm uses a maximum-likelihood estimation in order to estimate its weights from the training data. Hence, it is a minimization algorithm which is used to optimize its weights for the training data.
Our main intuition behind this algorithm is that in this scheme, there is a very simple relationship between inputs and outputs. Thus, we are able to identify why a certain sample point of the data was classified as fraud or benign (i.e. it depends on a specific value of a term in the weight matrix) and hence we can explain why a transaction is labeled as being fraudulent.
Similarly to the black-box case, in order to be able to implement this classification algorithm, a Machine Learning pipeline was employed. We implemented our standard pre-processing steps in order to be able to process our data in a meaningful way.Same as before, we ended up with the nine most discriminative features .
In the sequel, we built the Logistic Regression classifier and chose its parameters after some experimentation, identifying the most suitable combination for them. After that, in order to be able to understand its inner workings, we computed the best value of the probability threshold which is used in order to assign the class label of each test sample .
After finding the optimal threshold, we obtained all the weights of each feature. The feature with the larger positive weight has the highest contribution in order to assign objects to the label"fraud". On the contrary, the feature with the larger negative coefficient has the highest contribution in the assignment of samples to "benign" class.Thus, this is the main reason that logistic regression is explainable as a white box algorithm.Below we plot the features in order to justify our choice.

It can be depicted from figure 2, that in case that the issuer country code is AR (i.e. Argentina), then the corresponding weight has a high value and there is a strong indication that a transaction with this feature may be a fraud one.

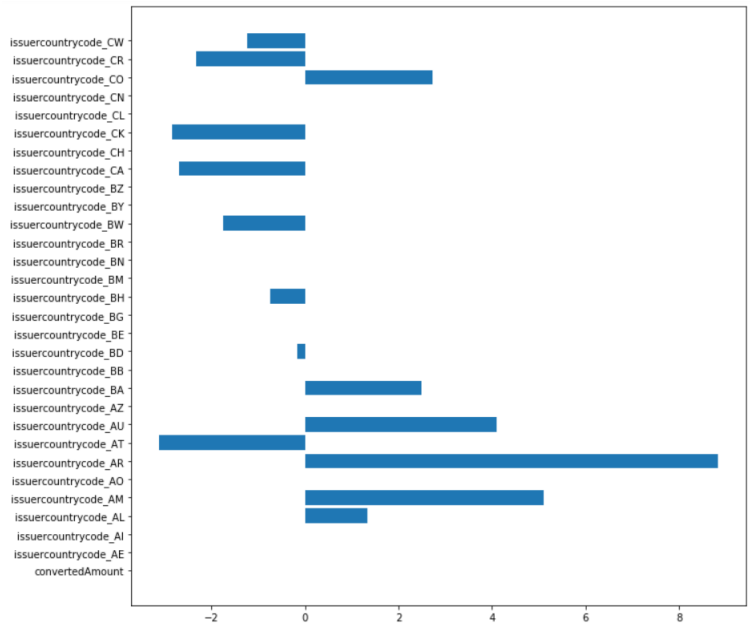


Figure 2: 2-Norm Difference over Number of Interactions with system, $\alpha = 0.9$

Finally, we performed again 10-fold cross-validation in order to obtain performance criteria that are relevant in practice. Namely, we calculated the True Positives, True Negatives, False Positives, False Negatives and evaluation metrics such as Accuracy, Precision, Recall, F-measure and the AUC score. We performed this analysis for the SMOTED case. A Comparison of the performance of the two algorithms (White-Black box) can be depicted in table 1.

Table 1: Comparison of the performance of White-Black box

Evaluation Metric	**White Box**	**Black Box**
Accuracy	95.34	96.85
Recall	2.07	2.56
Precision	66.92	34.27
F-Measure	40.19	33.58
AUC	92.34	92.37
TP	231	118
FP	10921	7219
TN	225770	229472
FN	114	227

As it can depicted from this table, the performance of the Black box algorithm in terms of accuracy and recall is better in comparison to the White box case. Nonetheless, in terms of Precision, F-Measure and the total number of TP (which are the most relative measures in the fraud detection case), the White box is the most dominant one. Finally, the UNSMOTED case was examined, where our algorithm had a worser performance in the evaluation metrics.

References

[1] L. Breiman. *Random forest, Machine Learning 45,2001, 5–32.*

[2] Siddhartha Bhattacharyya , Sanjeev Jha , Kurian Tharakunnel and J. Christopher Westland. *Data mining for credit card fraud:A comparative study. In Decision Support Systems. v.50 n.3, p.602-613, February, 2011*

[3] C. Whitrow, D.J. Hand, P. Juszczak, D. Westona and N.M. Adams. *Transaction aggregation as a strategy for credit card fraud detection. In Data Mining and Knowledge Discovery 18 (1) (2009) 30–55*

[4] Y. Jin, R.M. Rejesus, B.B. Little, Binary choice models for rare events data: a crop insurance fraud application, Applied Economics 37 (7) (2005) 841–848.