Cyber Data Analysis (CS4035)
Lab Assignment 1

Harvey van Veltom
4350073

Viktor Wigmore
4279638

May 10, 2019

1 Visualisation

In order to visualise the data set two visualisations were made. The first visualisation is a scatter plot that can be seen in Figure 1. This scatter plot compares the amount of a transaction t with the average transaction amount minus t of the same card used. This scatter plot looks only at cards for which at least two transactions are in the data set. As a result of the 354 fraudulent transactions only 73 transactions are visualised in the scatter plot. While there is no clear separator of fraudulent and non-fraudulent transactions there is a clear cluster of fraudulent transactions. This cluster indicates that fraudulent transactions are performed on cards with an average transaction amount of below 2000 and that the fraudulent transactions themselves have an amount of below 2000.

The second visualisation can be seen in Figure 2. This is a heat map that compares the fraudulent transactions based on the type of credit card used and the type of currency used. This heat map shows three combinations of credit card type and currency code where a lot of fraudulent transactions happen. These are mccredit with AUD, mccredit with MXN and visa classic with MXN. Because only certain combinations are used with the fraud cases, a machine learning algorithm should be able to use such patterns in order to estimate the probability of a transaction being fraudulent.

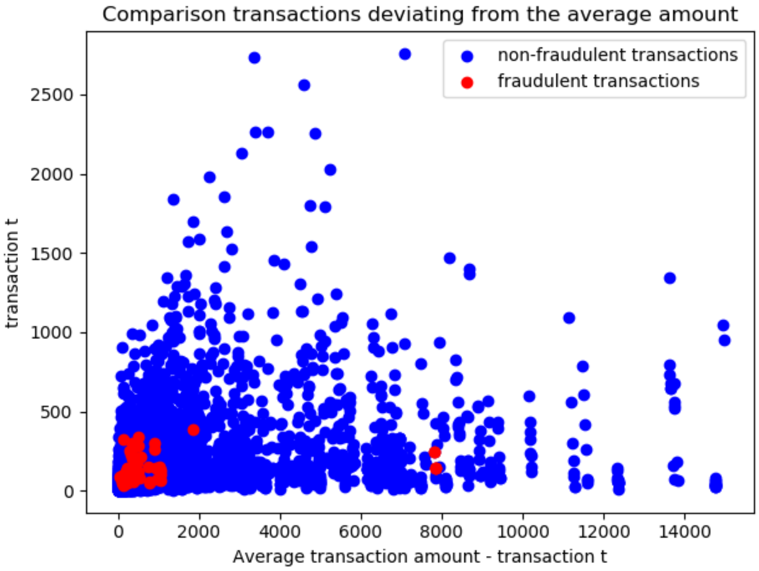


Figure 1: Scatterplot of transaction amounts compared to the average transaction amounts of the same card

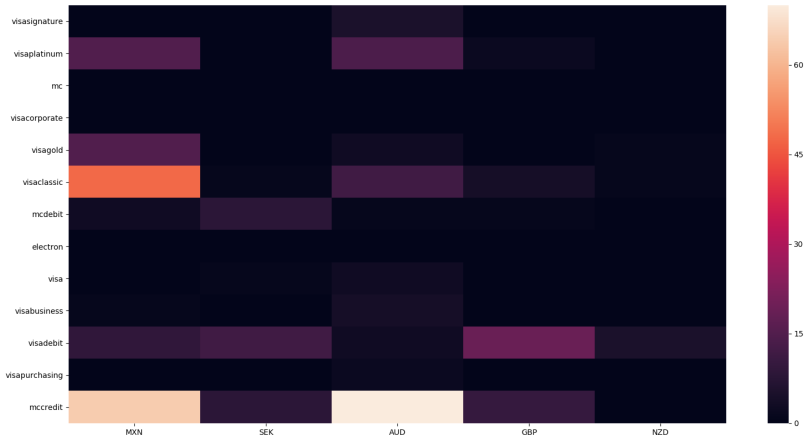
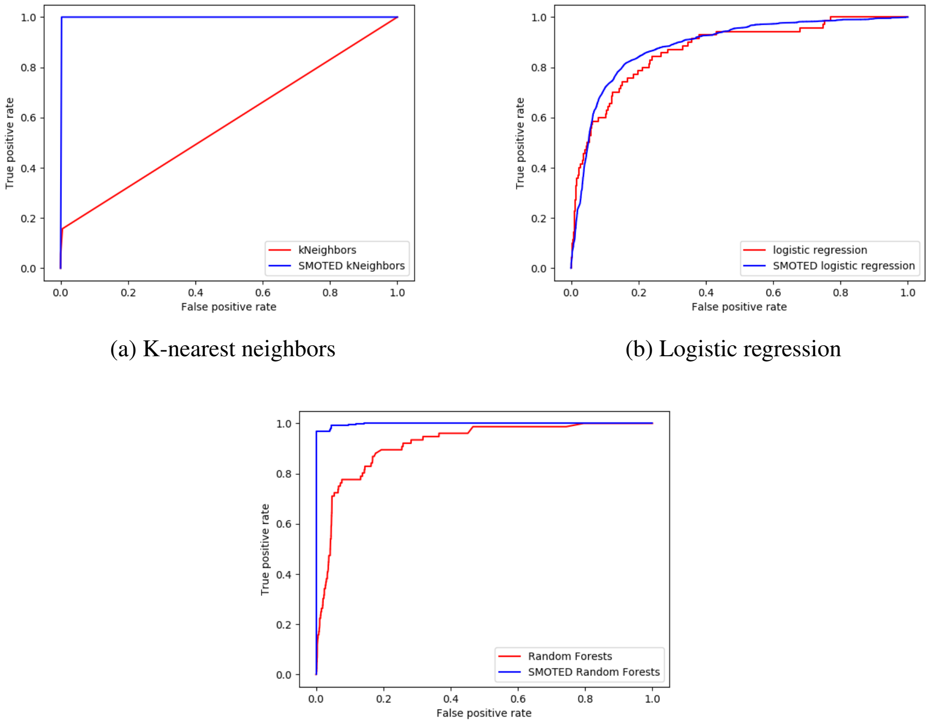


Figure 2: Heatmap showing cases of fraud with respect to currency used and creditcard used.

2 Imbalance

Before using SMOTE on the data set, the data has to be prepossessed, this is done using KNIME. In KNIME first all entries for which the label is Refused are removed from the data set as we only use guaranteed fraud and non-fraud cases. Then, we use one hot encoding on the columns which contains categorical data for improving the effectiveness of SMOTE. After this is done, we remove the categorical data columns from the data set. Now the data set only contains dates, ids and the one hot encoded data. No other columns are removed as we want the same amount of information when comparing the original and SMOTEd data sets.

In order to see the influence of oversampling using SMOTE on the data set, we used three different algorithms for classification. These algorithms are K-nearest neighbors, logistic regression and random forests. The results of these algorithms on the original data set and the SMOTEd data set can be seen in Figure 3. As can be seen, using SMOTE leads to a slightly better true positive rate in the case of logistic regression and random forests. However, for K-nearest neighbors the true positive rate significantly increases for the same false positive rate. When not oversampling the KNN algorithm performs worse than the others, however when using SMOTE the performance is actually better. It can be concluded that oversampling the data set using SMOTE is a good idea as the performance is increased in every case tested.



(a) K-nearest neighbors

(b) Logistic regression

(c) Random Forests

Figure 3: ROC curves for different algorithms on the original and the oversampled data set using SMOTE.

3 Classification

For our classification algorithms used, the data needs to be further preprocessed. For 10-fold cross validation the data set needs to be split into parts of equal size which all contain the same amount of fraud and non-fraud cases. After this, we obtain 10 training sets with a corresponding validation set. By separating the validation sets from the training sets before applying SMOTE, we guarantee that the learning algorithm has no information about the entries in the validation set. Columns which contain a date or an identifier are also removed from the data set. SMOTE can now be applied to each data set. On these data sets we run both our black-box and white-box algorithm.

Black-box

During the visualisation of the effect of oversampling, the K-nearest neighbors algorithm showed very good results. Because of these results we decided to use this classifier. For training purposes we did not use the whole training data set as it contains around 500.000 entries which is too large. From both the fraud and non-fraud cases 100.000 entries were used. After the algorithm finds a model using the training data, it predicts the probabilities of an entry being fraud for the transactions in the validation set by searching for k=5 nearest neighbors. This is done for every fold and the probabilities are stored. From the ROC curve a threshold of 0.8 was determined above which a transaction is labelled as fraud. Mapping all probabilities to their new label and comparing them with their real label gives the final result.

TP :	212
FP :	236
FN :	133
TN :	236455

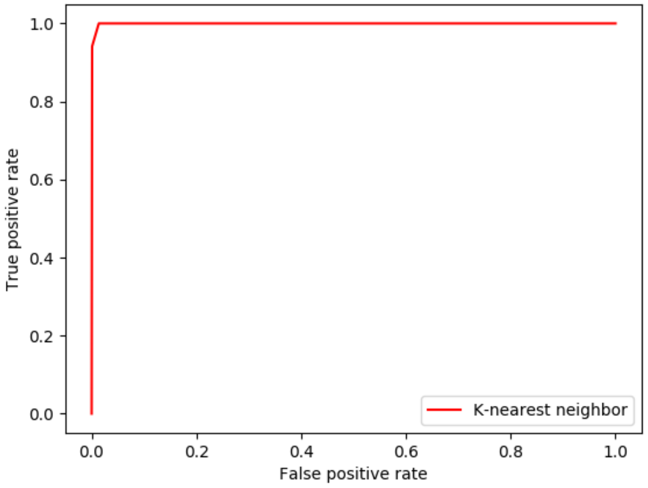


Figure 4: ROC curve for the black-box algorithm.

The black-box algorithm correctly classifies 212 fraud cases and wrongly classifies 236 non-fraud cases. In Figure 4 the ROC curve for the black-box algorithm used is shown.

White-box

To create a white box algorithm from which the decision making can be visualised and understood a random forest is used as a classifier. The random forest uses 20 decision trees each with a max depth of 5 to estimate the transactions. These parameters where chosen so that decision making process of all the trees can still be understood. While the classifier would be more accurate if more trees and a higher depth were used as a trade off it would mean that the decision making process of all the trees would grow so large that it would become to difficult to reasonably understand and be able to explain the decision process of the random forest for that reason these parameters were given. The random forest is trained using the SMOTEd data sets. Because these sets are too big to read in all at once 50000 entries are used creating an equal distribution of fraudulent and non fraudulent cases. for each of the folds the probabilities of it being fraud are stored. And using the ROC curve a decision threshold of 0.7 was determined to be used to determine whether a transaction was labelled as fraud or not. Mapping all predicted labels to there real label counter part gives the final result.

Decision visualisation

There were two ways to visualise the decision making process implemented the simplified and complex version. In the simplified version for each tree that labelled the transaction as fraudulent the decision path through all the nodes of the tree is visualised. From this decision path all the features used to determine the transaction to be fraudulent can be seen. An example of a decision path of a tree is given below. In the complex version a binary representation of each tree in the random forest is printed out as well as two decision path visualisations for each tree. One visualisation is the same as in the simplified version the other uses the columns of the transaction instead of the features and the decision thresholds of all nodes(In the simplified version only the decision thresholds relating to bin, amount and cvcresponse code are used because all other features are one hot encoded categorical features.)

```
decision path of the transaction in tree 19 of the random forest
node [0] of the tree made the decision because
```

```
cvcresponsecode of the transaction was 0.0 which is less than
the threshold of 0.9995706677436829
node [1] of the tree made the decision because amount of the transaction
was 84915.0 which is more than the threshold of 13450.240234375
node [13] of the tree made the decision because transaction
did not match the condition of SEK_currencycode
node [14] of the tree made the decision because transaction
did match the condition of MX_issuercountrycode
node [18] of the tree made the decision because transaction
did not match the condition of visaclassic_txvariantcode
```

TP :	231
FP :	396
FN :	114
TN :	236295

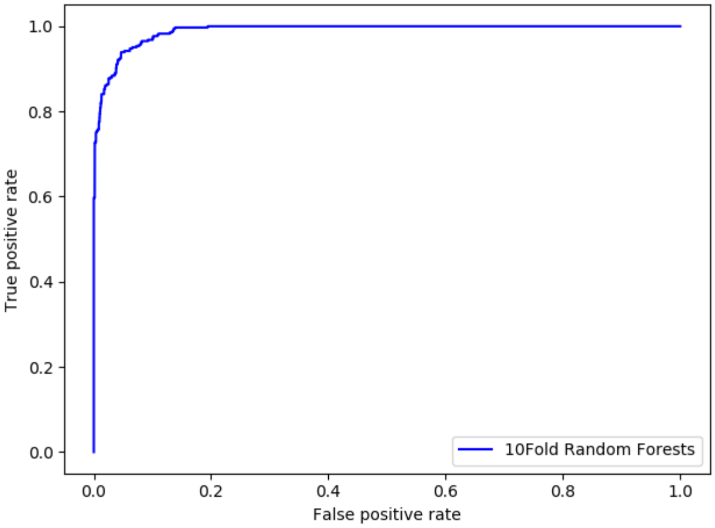


Figure 5: ROC curve for the white-box algorithm.

The white-box algorithm correctly classifies 231 fraud cases and wrongly classifies 396 non-fraud cases. In Figure 5 the ROC curve for the white-box algorithm used is shown. The performance of the white-box algorithm could be improved, but as a design choice to keep it understandable this is not possible.

Comparison

As the white-box algorithm cannot be too complicated, too many trees or too many nodes will result in an unclear decision path, the cost of being able to explain a certain outcome results in a performance loss. When comparing the white-box algorithm with the black-box algorithm this trade-off becomes visible. Both algorithms find approximately the same number of true fraud cases but the white-box algorithm wrongly classifies double the amount of transactions the black-box algorithm classifies wrongly. Therefore it is the choice of the developer on what is more important, being able to understand or having a higher performance.