Fraudulent or not?

Sari Vesiluoma

12.11.2019

Introduction

The goal in this project is to learn how to predict a fraudulent financial transaction. The data used here is called Synthetic Financial Datasets for Fraud Detection generated by the PaySim mobile money simulator (https://www.kaggle.com/ntnu-testimon/paysim1).As described on the web page, the dataset is a synthetic one, generated using the simulator called PaySim. It uses aggregated data from a private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behaviour.

PaySim simulates mobile money transactions based on a sample of real transactions extracted from one month of financial logs from a mobile money service implemented in an African country. The synthetic dataset is scaled down 1/4 of the original dataset.

I have downloaded the dataset from the net (the link above) and I have unzipped it to the same folder where my R script and the rmd file are. Here, I am reading the data from my folder.

The dataset, here referred with a variable name fraud_or_not, has the following dimensions

```
## [1] 6362620      11
```

Next I will analyse the data and split it to training and test sets. I will use different machine learning algorithms to try to predict which transaction is fraudulent and which not. In this kind of a case the speciality is that the amount of fraudulent transactions is very minor compared to the amount of non-fraudulent transactions, as we will see.

Analysis

Understanding the data

Let's look the data first as is. As can be seen from the summary below, there are e.g. no NA values which would need to be cleaned. Zero values do exist, but those might be correct values needed, because the data seems to include different kind of transactios, where different features are relevant and the others might be zero as a value.

```r
summary(fraud_or_not)
```

```
##       step           type               amount            nameOrig
##  Min.   :  1.0   Length:6362620     Min.   :       0   Length:6362620
##  1st Qu.:156.0   Class :character   1st Qu.:   13390   Class :character
##  Median :239.0   Mode  :character   Median :   74872   Mode  :character
##  Mean   :243.4                      Mean   :  179862
##  3rd Qu.:335.0                      3rd Qu.:  208721
##  Max.   :743.0                      Max.   :92445517
##  oldbalanceOrg      newbalanceOrig       nameDest
##  Min.   :       0   Min.   :       0   Length:6362620
##  1st Qu.:       0   1st Qu.:       0   Class :character
##  Median :   14208   Median :       0   Mode  :character
##  Mean   :  833883   Mean   :  855114
##  3rd Qu.:  107315   3rd Qu.:  144258
##  Max.   :59585040   Max.   :49585040
##  oldbalanceDest      newbalanceDest          isFraud
##  Min.   :        0   Min.   :        0   Min.   :0.000000
```

```
##  1st Qu.:        0   1st Qu.:        0   1st Qu.:0.000000
##  Median :   132706   Median :   214661   Median :0.000000
##  Mean   :  1100702   Mean   :  1224996   Mean   :0.001291
##  3rd Qu.:   943037   3rd Qu.:  1111909   3rd Qu.:0.000000
##  Max.   :356015889   Max.   :356179279   Max.   :1.000000
##  isFlaggedFraud
##  Min.   :0.0e+00
##  1st Qu.:0.0e+00
##  Median :0.0e+00
##  Mean   :2.5e-06
##  3rd Qu.:0.0e+00
##  Max.   :1.0e+00
```

```r
str(fraud_or_not)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 6362620 obs. of  11 variables:
##  $ step          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ type          : chr  "PAYMENT" "PAYMENT" "TRANSFER" "CASH_OUT" ...
##  $ amount        : num  9840 1864 181 181 11668 ...
##  $ nameOrig      : chr  "C1231006815" "C1666544295" "C1305486145" "C840083671" ...
##  $ oldbalanceOrg : num  170136 21249 181 181 41554 ...
##  $ newbalanceOrig: num  160296 19385 0 0 29886 ...
##  $ nameDest      : chr  "M1979787155" "M2044282225" "C553264065" "C38997010" ...
##  $ oldbalanceDest: num  0 0 0 21182 0 ...
##  $ newbalanceDest: num  0 0 0 0 0 ...
##  $ isFraud       : num  0 0 1 1 0 0 0 0 0 0 ...
##  $ isFlaggedFraud: num  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   step = col_double(),
##   ..   type = col_character(),
##   ..   amount = col_double(),
##   ..   nameOrig = col_character(),
##   ..   oldbalanceOrg = col_double(),
##   ..   newbalanceOrig = col_double(),
##   ..   nameDest = col_character(),
##   ..   oldbalanceDest = col_double(),
##   ..   newbalanceDest = col_double(),
##   ..   isFraud = col_double(),
##   ..   isFlaggedFraud = col_double()
##   .. )
```

```r
fraud_or_not %>% head()
```

```
## # A tibble: 6 x 11
##    step type  amount nameOrig oldbalanceOrg newbalanceOrig nameDest
##   <dbl> <chr>  <dbl> <chr>            <dbl>          <dbl> <chr>
## 1     1 PAYM~  9840. C123100~        170136        160296. M197978~
## 2     1 PAYM~  1864. C166654~         21249         19385. M204428~
## 3     1 TRAN~   181  C130548~           181             0  C553264~
## 4     1 CASH~   181  C840083~           181             0  C389970~
## 5     1 PAYM~ 11668. C204853~         41554         29886. M123070~
## 6     1 PAYM~  7818. C900456~         53860         46042. M573487~
## # ... with 4 more variables: oldbalanceDest <dbl>, newbalanceDest <dbl>,
## #   isFraud <dbl>, isFlaggedFraud <dbl>
```

The data has 11 columns which are:

Table 1: Explanations of the features

feature	expl
step	Maps a unit of time in the real world. 1 step is 1 hour of time. Total steps 744 (30 days simulation).
type	CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.
amount	Amount of the transaction in local currency.
nameOrig	Customer who started the transaction
oldbalanceOrg	Initial balance before the transaction
newbalanceOrig	New balance after the transaction
nameDest	Customer who is the recipient of the transaction
oldbalanceDest	Initial balance recipient before the transaction
newbalanceDest	New balance recipient after the transaction
isFraud	Transactions made by the fraudulent agents inside the simulation
isFlaggedFraud	An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

The feature isFraud refers to a fraudulent transaction. The feature isFlaggedFraud refers just to a transaction, which is a suspect for a fraud because it is an attempt to transfer more than 200 000. Normally the amount of fraudulent transactions is very small compared to the non-fraudulent transactions. That is the case here also and the prevalence of a fraud is close to one from thousand transactions.

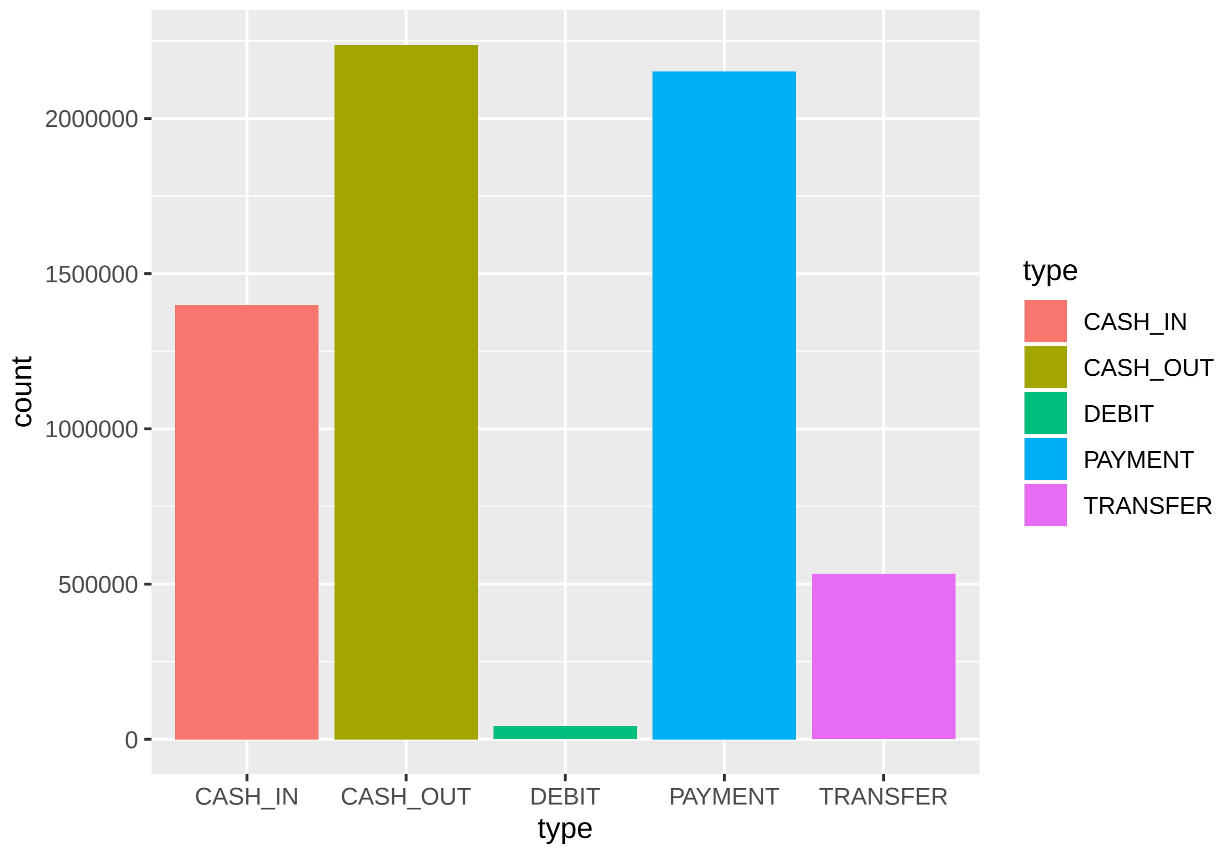
```
mean(fraud_or_not$isFraud)
```

```
## [1] 0.00129082
```

This means that 99.870918 % of the transactions are non-fraudulent. We would get a very high accuracy if predicting that a transaction is always non-fraudulent.

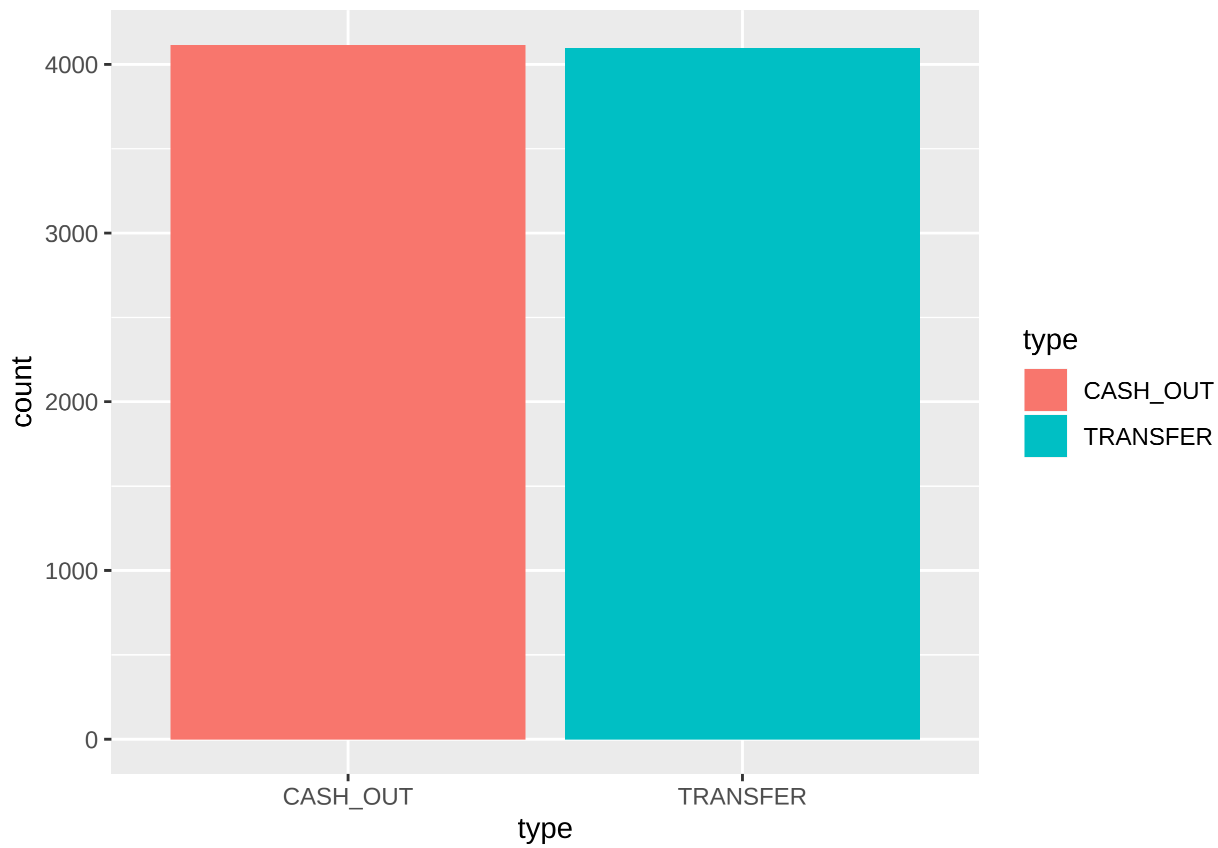
The amount of transactions per type is described at the next histogram. The biggest amount being CASH_OUT transactions and the smallest amount being DEBIT transactions.

```
fraud_or_not %>% ggplot(aes(type , fill = type)) + geom_bar()
```



More interesting is actually how the fraudulent transactions are positioned among the transactions. As can be seen from the next histogram, only the categories CASH_OUT and TRANSFER have fraudulent transactions.

```
fraud_or_not %>% filter(isFraud==1) %>%
  ggplot(aes(type , fill = type)) + geom_bar()
```



Training and test sets

For the prediction we need training and test sets. In this project those were created based on the fraud_or_not data the following way, checking that the dimensions of the resulting sets are correct and that the prevalence of fraudulent transactions is similar between the cases.

```
# Creating training and test sets
set.seed(1234, sample.kind = "Rounding")
```

```
## Warning in set.seed(1234, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
test_index <- createDataPartition(y = fraud_or_not$isFraud, times = 1,
                                  p = 0.2, list = FALSE)
train_set <- fraud_or_not[-test_index,]
test_set <- fraud_or_not[test_index,]
# Checking the dimensions and prevalence of fraud in these sets
dim(train_set)
```

```
## [1] 5090096      11
```

```
mean(train_set$isFraud == "1")
```

```
## [1] 0.001288581
```

```
dim(test_set)
```

```
## [1] 1272524      11
```

```r
mean(test_set$isFraud == "1")
```

```
## [1] 0.001299779
```

Machine learning

Now we are ready to start trials with machine learning algorithms. To start with, we should get a pretty accurate results if guessing that none of the transactions are fraudulent. Let's try.

```r
# Algorithm 1: quess that none of the transactions are fraud
# Define mu_hat as a predition of no fraudulent transactions
mu_hat <- replicate(length(test_set$isFraud), 0)
# Check the results with a confusionMatrix - ensure the same levels to be used
cm <- confusionMatrix(data = factor(mu_hat,
                                    levels=min(test_set$isFraud):max(test_set$isFraud)),
                      reference = as.factor(test_set$isFraud))
cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction       0       1
##          0 1270870    1654
##          1       0       0
##
##                Accuracy : 0.9987
##                  95% CI : (0.9986, 0.9988)
##     No Information Rate : 0.9987
##     P-Value [Acc > NIR] : 0.5065
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.9987
##          Neg Pred Value :    NaN
##              Prevalence : 0.9987
##          Detection Rate : 0.9987
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```r
# Store the results of this algorithm
algorithm_results <- data.frame(method="1: Assume none is fraud",
                                accuracy=cm$overall['Accuracy'],
                                specificity=sensitivity(factor(mu_hat), factor(test_set$isFraud), posit
```

As we can see from the confusion matrix, the accuracy is really high 0.9987002 but, the specificity is 0. In practice, this means that we have failed in predicting any of the frauds. It is important to notice, that in addition to following accuracy getting better, we are actually even more worried in having a very high specificity, meaning that we have correctly predicted the fraudulent actions being fraudulent.

Results

Conclusion