Credit Card Fraud Detection

Fernando José Velasco Borea

May 12th 2019

Contents

1 Introduction and Overview

An article conducted by Loss Prevention Magazine in 2018 showed that by 2020 the total monetary losses due to credit card fraud in the U.S. alone could excede the $10,000,000,000 mark (you can find the article here). With a constant growth on cardholders across the years, the concern about this type of fraud has also increased. On 2017 we saw an increment of 1.3 million credit card fraud victims, implying an increase of 8.4% when compared to the 2016 period (as reported by Javelin Strategy & Research). Taking this into account, I decided to conduct a supervised machine learning project with the goal of predicting potential fraudulent credit card transactions.

For this project, we will be using the data set provided by Machine Learning Group - ULB through Kaggle (you can find it through this link). The data set contains information about the time (relative to the frequency of the transactions when compared to the first one in the data set), amount, type of transaction (either fraudulent or non-fraudulent, represented by a 1 or a 0 respectively) and 28 numerical features resulting from a PCA Dimensionality Reduction to protect the users identity and sensitive information.

The project will be divided into 4 major sections, as follows:

1. Data Adquisition
2. Data Exploration and Wrangling
3. Modeling
4. Testing

Once we complete the sections mentioned above, we will create a *Conclusions* section with the insights we gathered throughout the project.

1.1 Side Notes

Although the data set used for this project is downloaded within the code, to improve the run time, it is recommended to clone the GitHub repository as it contains the `csv` file with the data set we used.

To enhance code readability when viewing the Rmd version of this report and/or when viewing the Credit Card Fraud Detection Script file to see only the coding part of the project, you can *fold* the all the sections from RStudio to then just *unfold* the section you are currently viewing, therefore, easing the interpretation of the code.

You can quickly do this from RStudio going to *Edit > Folding > Collapse All* or simply with the shortcut *ALT + O* on windows. If you want to exapnd all the sections again, you can use the shortcut *ALT + SHIFT + O* on windows or from *Edit > Folding > Expand All*.

The code contained in this report can be found on the Credit Card Fraud Detection Script file. It follows the same structure and order as the report, therefore, making it easier to reproduce the results while maintaining code readability.

To render the Rmd version of this report you will need to have a LaTeX installation. If you don't have it, you can find more details on how to install it here.

2 Data Adquisition

This section is going be mainly intended to download or read the data set (depending if you have the repository cloned into your local machine) that we will be using throughout the project.

First, we will start by loading the required libraries for our project, and then proceed to read our data either from our working directory if we cloned the repository, or from Git LFS if we have not. Note that because of formatting purposes, we will not show the output messages from the code below on the report.

Executing this code section might take some minutes depending on your internet connection.

```
if(!require(tidyverse)) install.packages("tidyverse",
                                         repos = "http://cran.us.r-project.org")

if(!require(RCurl)) install.packages("RCurl",
                                     repos = "http://cran.us.r-project.org")

if(!require(knitr)) install.packages("knitr",
                                     repos = "http://cran.us.r-project.org")

if(!require(caret)) install.packages("caret",
                                     repos = "http://cran.us.r-project.org")

if(!require(randomForest)) install.packages("randomForest",
                                     repos = "http://cran.us.r-project.org")

if(file.exists("creditcard.csv"))
{

  cc_dataset <- read_csv("creditcard.csv")

} else {

  URL_p1 <- "https://media.githubusercontent.com"
  URL_p2 <- "/media/FernandoBorea/Credit-Card-Fraud-Detection/master/creditcard.csv"
  datURL <- getURL(paste(URL_p1, URL_p2, sep = ""))

#We divided the entire URL in 2 string vectors and
#then used paste to maintain the report formatting

  cc_dataset <- read_csv(datURL)

}
```

2.1 Preliminary Data Exploration

Once the Data Adquisition process is finished, we will start performing some preliminary data exploration to make sure the data was downloaded and/or read correctly and to familiarize ourselves with the data set.

When calling the function `str()` to look for the data structure, it will result in quite a large and somewhat messy output within our report. We already know from the Kaggle Site where we got our data from, that we have several columns in our data set, therefore we are not going to include the output of the code below.

```
str(cc_dataset)
```

As we did not show the output of the code above, we will use another approach to still show some information about the data set within this section, more specifically, we will just check the amount of rows and columns as well as the class of each column:

```
data.frame(Columns = ncol(cc_dataset), Rows = nrow(cc_dataset)) %>%
  knitr::kable()
```

Columns	Rows
31	284807

```
col_classes <- data.frame(Column = colnames(cc_dataset)[1:16],
                  Class = unname(apply(cc_dataset, 2, class))[1:16],
                  Column = c(colnames(cc_dataset)[17:ncol(cc_dataset)],""),
                  Class = c(unname(apply(cc_dataset, 2, class))[17:ncol(cc_dataset)],""))

colnames(col_classes) <- c("Column", "Class", "Column","Class")

col_classes %>% knitr::kable()
```

Column	Class	Column	Class
Time	numeric	V16	numeric
V1	numeric	V17	numeric
V2	numeric	V18	numeric
V3	numeric	V19	numeric
V4	numeric	V20	numeric
V5	numeric	V21	numeric
V6	numeric	V22	numeric
V7	numeric	V23	numeric
V8	numeric	V24	numeric
V9	numeric	V25	numeric
V10	numeric	V26	numeric
V11	numeric	V27	numeric
V12	numeric	V28	numeric
V13	numeric	Amount	numeric
V14	numeric	Class	numeric
V15	numeric		

3 Data Exploration and Wrangling

4 Modeling

5 Testing

6 Conclusions