Credit Fraud Detection Analysis

Arjun Sekhar

2023-12-26

Introduction

Data Preparation

As preparation we establish some of the packages that will come into use in this analysis. This is outlined in the below chunk of R.

```
library(dplyr)
library(tidyverse)
library(pander)
library(corrplot)
library(caret)
library(PRROC)
library(glmnet)
```

The next step involves introducing the `ratings_for_upload.csv` data set using the `read.csv()` function. Our aim here is to analyse and understand the data, as well as the variables that are at our disposal. This will segway into the exploratory data analysis (EDA), allowing us to pursue a model building strategy.

```
# Set the working directory
setwd("/Users/arjunsekhar/OneDrive/Knowledge/Courses/Kaggle/credit-fraud-analysis")

# Input the data
credit <- read.csv('creditcard.csv', header = TRUE)
head(credit)
```

```
##   Time         V1          V2        V3         V4          V5          V6
## 1    0 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.46238778
## 2    0  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.08236081
## 3    1 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.80049938
## 4    1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.24720317
## 5    2 -1.1582331  0.87773675 1.5487178  0.4030339 -0.40719338  0.09592146
## 6    2 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.02972755
##            V7          V8         V9         V10        V11         V12
## 1  0.23959855  0.09869790  0.3637870  0.09079417 -0.5515995 -0.61780086
## 2 -0.07880298  0.08510165 -0.2554251 -0.16697441  1.6127267  1.06523531
## 3  0.79146096  0.24767579 -1.5146543  0.20764287  0.6245015  0.06608369
## 4  0.23760894  0.37743587 -1.3870241 -0.05495192 -0.2264873  0.17822823
## 5  0.59294075 -0.27053268  0.8177393  0.75307443 -0.8228429  0.53819555
```

```
## 6  0.47620095  0.26031433 -0.5686714 -0.37140720  1.3412620  0.35989384
##          V13        V14        V15        V16         V17         V18
## 1 -0.9913898 -0.3111694  1.4681770 -0.4704005  0.20797124  0.02579058
## 2  0.4890950 -0.1437723  0.6355581  0.4639170 -0.11480466 -0.18336127
## 3  0.7172927 -0.1659459  2.3458649 -2.8900832  1.10996938 -0.12135931
## 4  0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.68409279  1.96577500
## 5  1.3458516 -1.1196698  0.1751211 -0.4514492 -0.23703324 -0.03819479
## 6 -0.3580907 -0.1371337  0.5176168  0.4017259 -0.05813282  0.06865315
##           V19         V20          V21          V22         V23         V24
## 1  0.40399296  0.25141210 -0.018306778  0.277837576 -0.11047391  0.06692807
## 2 -0.14578304 -0.06908314 -0.225775248 -0.638671953  0.10128802 -0.33984648
## 3 -2.26185710  0.52497973  0.247998153  0.771679402  0.90941226 -0.68928096
## 4 -1.23262197 -0.20803778 -0.108300452  0.005273597 -0.19032052 -1.17557533
## 5  0.80348692  0.40854236 -0.009430697  0.798278495 -0.13745808  0.14126698
## 6 -0.03319379  0.08496767 -0.208253515 -0.559824796 -0.02639767 -0.37142658
##          V25        V26          V27         V28 Amount Class
## 1  0.1285394 -0.1891148  0.133558377 -0.02105305 149.62     0
## 2  0.1671704  0.1258945 -0.008983099  0.01472417   2.69     0
## 3 -0.3276418 -0.1390966 -0.055352794 -0.05975184 378.66     0
## 4  0.6473760 -0.2219288  0.062722849  0.06145763 123.50     0
## 5 -0.2060096  0.5022922  0.219422230  0.21515315  69.99     0
## 6 -0.2327938  0.1059148  0.253844225  0.08108026   3.67     0
```

From the data presented, a total of 284807 transactions of 31 variables are recorded, which is a representation of credit card transactions in September 2013 by European credit card holders. As mentioned in the context of this task, the data has been dealt with Principal Component Analysis (PCA) transformations extensively, this limits the extent of information available.

Exploratory Data Analysis (EDA)

A) Spread of values in 'Class' column

Firstly with the `Class` column, we can alter the summary by identifying the factors as `Non Fraudulent` and `Fraudulent` transactions. Upon doing so, this summary can be presented as a table using `pander`, which is a neater way of displaying the information in LaTeX format.

```
credit$Class <- as.factor(credit$Class)
levels(credit$Class) <- c('Non Fraudulent', 'Fraudulent')

credit_class <- credit %>%
  group_by(Class) %>%
  summarise(Total = n()) %>%
  mutate(Frequency = round(Total/sum(Total), 5)) %>%
  arrange(desc(Frequency))

pander(credit_class)
```

Class	Total	Frequency
Non Fraudulent	284315	0.9983
Fraudulent	492	0.00173

From the above we can see how the data provided is unbalanced. Despite the context provided acting as a foreshadow to this exploratory revelation, it can be foreshadowed from an everyday perspective since we would anticipate most transactions to be non fraudulent. Given the intention is to measure the accuracy, the Area Under the Curve (AUC) concept will be applied as the accuracy measure.

B) Missing Values

The next step is to analyse the proportion of missing values from the context of this data set.

```
# Quantity of Missing Values
credit %>%
  is.na() %>%
  sum()
```

```
## [1] 0
```

From the above we can see how there are no missing values. Although such a reading is rare, in this context it can be attributed to the data preparation when the data was provided.

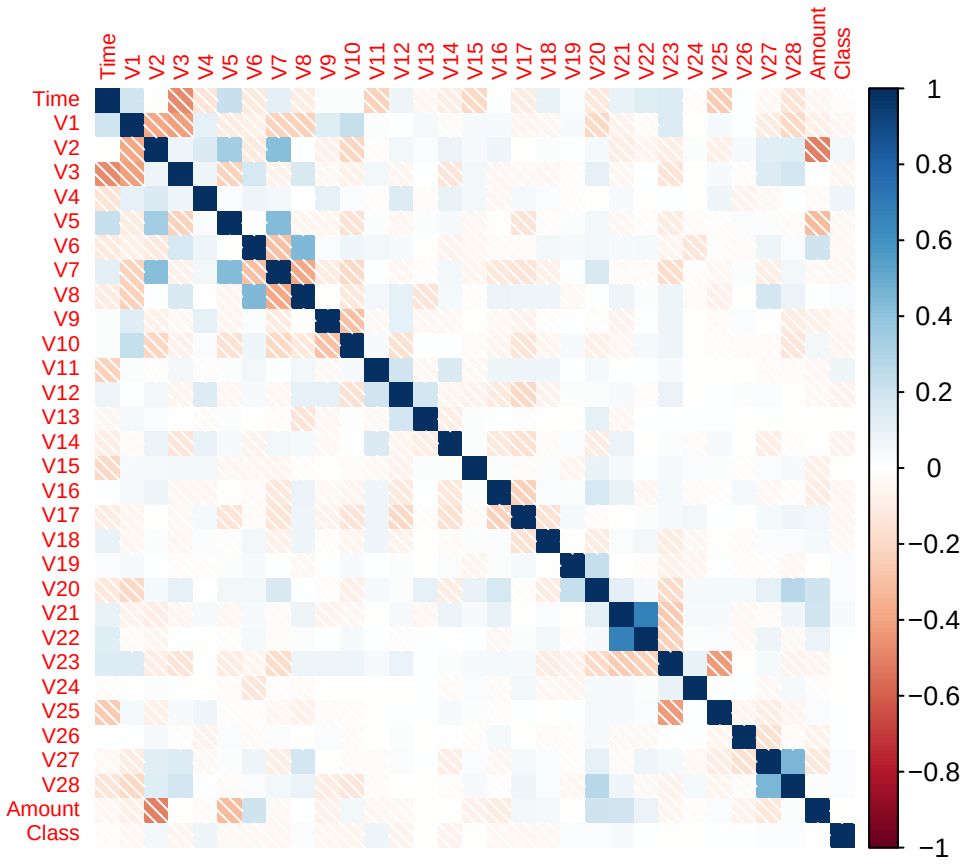
C) Correlation Matrix

```
credit %>% duplicated() %>% sum()
```

```
## [1] 1081
```

```
credit2 <- credit
credit2$Class <- as.numeric(credit2$Class)

credit_correlation <- cor(credit2[], method = 'spearman')
corrplot(credit_correlation, method = 'shade', tl.cex = 0.65)
```


Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
Class
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1