Data Analytics Project

Jai Chi Cham

2022-10-08

# discuss the business problem/goal

As a merchant/bank, we want to build a model to detech credit card fradulent activity. Credit card fraud is one forms of identify theft that made fraudulent payment by using a credit card in an unauthorized manner.

# identify where the dataset was retrieved from

The dataset contains transactions made by credit cards in September 2013 by European cardholders. <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download>

# identify the code that imported and saved your dataset in R

Downloaded Csv file from the Kaggle.com -> search for working directory folder of R Studio -> moved the file to the active working directory folder

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.7 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(dplyr)  
library(ggplot2)  
library(ranger)

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

getwd()

## [1] "C:/Users/jaich/OneDrive/Documents"

data <- read\_csv('creditcard.csv')

## Rows: 284807 Columns: 31

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (31): Time, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13, V14,...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# describe your data set

nrow(data)

## [1] 284807

ncol(data)

## [1] 31

names(data)

## [1] "Time" "V1" "V2" "V3" "V4" "V5" "V6" "V7"   
## [9] "V8" "V9" "V10" "V11" "V12" "V13" "V14" "V15"   
## [17] "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23"   
## [25] "V24" "V25" "V26" "V27" "V28" "Amount" "Class"

colnames(data)

## [1] "Time" "V1" "V2" "V3" "V4" "V5" "V6" "V7"   
## [9] "V8" "V9" "V10" "V11" "V12" "V13" "V14" "V15"   
## [17] "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23"   
## [25] "V24" "V25" "V26" "V27" "V28" "Amount" "Class"

summary(data$Amount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 5.60 22.00 88.35 77.17 25691.16

# discuss any data preparation, missing values and errors

Due to confidentiality issues, there is no original features nor more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are ‘Time’ and ‘Amount’. Feature ‘Time’ contains the seconds elapsed between each transaction and the first transaction in the dataset.

# find missing value using is.na function  
missing\_data <- c(sum(is.na(data$Time)), sum(is.na(data$V1)),sum(is.na(data$V2)),sum(is.na(data$V3)),sum(is.na(data$V4)),sum(is.na(data$V5)),sum(is.na(data$V6)),sum(is.na(data$V7)),sum(is.na(data$V8)),sum(is.na(data$V9)),sum(is.na(data$V10)),sum(is.na(data$V11)),sum(is.na(data$V12)),sum(is.na(data$V13)),sum(is.na(data$V14)),sum(is.na(data$V15)),sum(is.na(data$V16)),sum(is.na(data$V17)),sum(is.na(data$V18)),sum(is.na(data$V19)),sum(is.na(data$V20)),sum(is.na(data$V21)),sum(is.na(data$V22)),sum(is.na(data$V23)),sum(is.na(data$V24)),sum(is.na(data$V25)),sum(is.na(data$V26)),sum(is.na(data$V27)),sum(is.na(data$V28)),sum(is.na(data$Amount)),sum(is.na(data$Class)))  
table.df <- data.frame (missing\_data)  
print(table.df)

## missing\_data  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0  
## 7 0  
## 8 0  
## 9 0  
## 10 0  
## 11 0  
## 12 0  
## 13 0  
## 14 0  
## 15 0  
## 16 0  
## 17 0  
## 18 0  
## 19 0  
## 20 0  
## 21 0  
## 22 0  
## 23 0  
## 24 0  
## 25 0  
## 26 0  
## 27 0  
## 28 0  
## 29 0  
## 30 0  
## 31 0

There is no missing value in the data, most likely the data has been cleaned to remove the missing value.

# Data Modeling and Evaluation

We will first use caTools function to split the data into training and test sets. Then , we use set.seed function to make sure that we get the same result for randomization. Next, we will use sample.split and subset function to create a sample set. We set 80% of the data as training set and 20% as testing set. Finally, we use the dim function to check the dimensions of the sample dataset.

library(caTools)

set.seed(123)  
  
data\_sample = sample.split(data$Class,SplitRatio=0.80)  
trainset\_data <- subset(data,data\_sample==TRUE)  
testset\_data <- subset(data,data\_sample==FALSE)  
dim(trainset\_data)

## [1] 227846 31

dim(testset\_data)

## [1] 56961 31

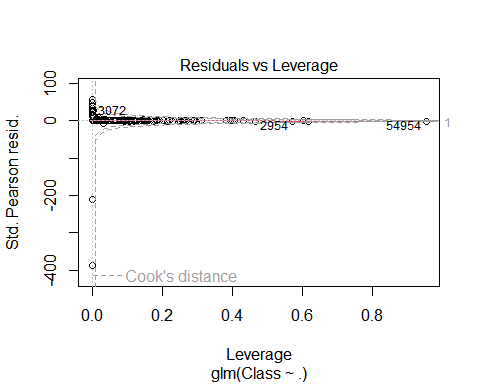
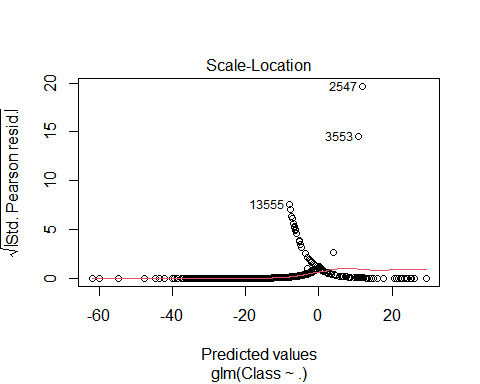
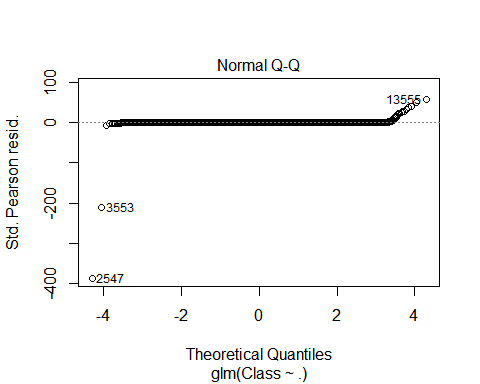
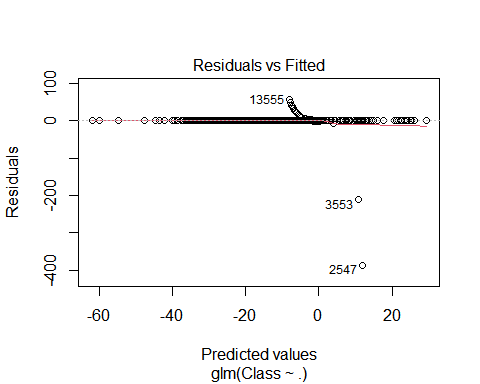
## Logistic Regression Model  
#we use class as the dependent variable, and all others as independent variable  
# we expect the result to be binomial response variable: fraud or not fraud  
Logistic\_Model=glm(Class~.,testset\_data,family=binomial())

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(Logistic\_Model)

##   
## Call:  
## glm(formula = Class ~ ., family = binomial(), data = testset\_data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.8815 -0.0246 -0.0142 -0.0069 4.0192   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.116e+01 1.013e+01 -1.101 0.2707   
## Time -1.173e-05 5.309e-06 -2.210 0.0271 \*  
## V1 -6.966e-02 1.254e+00 -0.056 0.9557   
## V2 1.204e+00 4.181e+00 0.288 0.7734   
## V3 3.305e-02 2.356e-01 0.140 0.8885   
## V4 2.822e+00 7.078e+00 0.399 0.6901   
## V5 1.387e+00 3.754e+00 0.369 0.7118   
## V6 -1.150e-01 2.006e-01 -0.573 0.5665   
## V7 1.280e+00 4.160e+00 0.308 0.7584   
## V8 -3.915e-01 1.699e-01 -2.305 0.0212 \*  
## V9 2.559e+00 8.556e+00 0.299 0.7649   
## V10 -2.639e+00 6.528e+00 -0.404 0.6860   
## V11 -2.458e-01 2.869e-01 -0.857 0.3915   
## V12 1.753e+00 6.478e+00 0.271 0.7866   
## V13 -7.236e-01 1.244e+00 -0.582 0.5607   
## V14 1.670e-02 3.245e+00 0.005 0.9959   
## V15 7.992e-01 2.850e+00 0.280 0.7791   
## V16 -2.666e+00 7.018e+00 -0.380 0.7040   
## V17 -1.663e+00 4.929e+00 -0.337 0.7358   
## V18 2.442e+00 8.015e+00 0.305 0.7606   
## V19 -1.358e+00 4.709e+00 -0.288 0.7730   
## V20 -6.967e-01 1.141e+00 -0.611 0.5415   
## V21 -2.854e-01 1.971e+00 -0.145 0.8849   
## V22 -1.002e+00 5.128e+00 -0.195 0.8451   
## V23 1.486e-01 5.984e-01 0.248 0.8039   
## V24 -1.001e-01 4.299e-01 -0.233 0.8159   
## V25 -6.329e-01 1.915e+00 -0.331 0.7410   
## V26 2.275e+00 9.222e+00 0.247 0.8051   
## V27 -5.138e-01 8.223e-01 -0.625 0.5321   
## V28 -7.352e-02 3.371e-01 -0.218 0.8274   
## Amount 7.281e-04 2.443e-03 0.298 0.7657   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1443.40 on 56960 degrees of freedom  
## Residual deviance: 373.61 on 56930 degrees of freedom  
## AIC: 435.61  
##   
## Number of Fisher Scoring iterations: 18

plot(Logistic\_Model)



# produce and discuss the output

we will use ROC curve to evaluate the performance of the model.

library(pROC)

## Type 'citation("pROC")' for a citation.

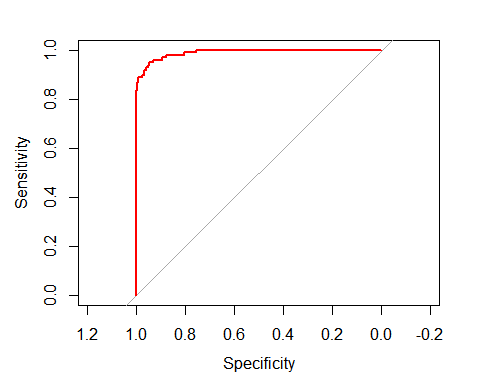
##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

data\_predict <- predict(Logistic\_Model,testset\_data, probability = TRUE)  
auc.gbm = roc(testset\_data$Class, data\_predict, plot = TRUE, col = "red")

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

 Per result,the curve is located at the top left, this mean that logistic model is a good predictive model to classify credit card transactions as fraudulent or not.