Week8 Credit Card Fraud Detection Dataset

Kohei Nishitani

2024-03-04

1. Discuss the business problem/goal

In the realm of financial services, identifying fraudulent transactions is a key element in protecting businesses. However, if fraud detection systems wrongly stop trustworthy transactions, this can lead to a poor customer experience, which may result in customer churn in the future. Hence, having an accurate fraud detection model is crucial to sustain and protect both the business and customer experience

2. Identify where the dataset was retrieved from

This original dataset is retrieved from some credit card company's data and stored at Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud).

> The dataset contains transactions made by credit cards in September 2013 by European cardholders.

3. Identify the code that imported and saved your dataset in R

```
# set working directory
setwd("~/Workspace/McDaniel-Repository/515/week8")
data <- read.csv("creditcard.csv")
```

4. Describe your dataset

This dataset has 284807 credit card transactions with 31 variables. Due to confidentiality, most variables are already preprocessed using PCA, except for 'Time' and 'Amount'. Here's a sample of the dataset(pick up couple preprocessed columns as sample):

```
##   Time         V1          V2         V28 Amount Class
## 1    0 -1.3598071 -0.07278117 -0.02105305 149.62     0
## 2    0  1.1918571  0.26615071  0.01472417   2.69     0
## 3    1 -1.3583541 -1.34016307 -0.05975184 378.66     0
## 4    1 -0.9662717 -0.18522601  0.06145763 123.50     0
## 5    2 -1.1582331  0.87773675  0.21515315  69.99     0
```

and these are basic dataset summaries

```
##       Time           V1                 V2                V28          
##  Min.   :0.0   Min.   :-1.3598   Min.   :-1.34016   Min.   :-0.05975  
##  1st Qu.:0.0   1st Qu.:-1.3584   1st Qu.:-0.18523   1st Qu.:-0.02105  
##  Median :1.0   Median :-1.1582   Median :-0.07278   Median : 0.01472  
##  Mean   :0.8   Mean   :-0.7302   Mean   :-0.09086   Mean   : 0.04211  
##  3rd Qu.:1.0   3rd Qu.:-0.9663   3rd Qu.: 0.26615   3rd Qu.: 0.06146  
##  Max.   :2.0   Max.   : 1.1919   Max.   : 0.87774   Max.   : 0.21515  
##      Amount           Class        
##  Min.   :  2.69   Min.   :0        
##  1st Qu.: 69.99   1st Qu.:0        
##  Median :123.50   Median :0        
##  Mean   :144.89   Mean   :0        
##  3rd Qu.:149.62   3rd Qu.:0        
##  Max.   :378.66   Max.   :0
```

```
## 'data.frame':    5 obs. of  6 variables:
##  $ Time  : num  0 0 1 1 2
##  $ V1    : num  -1.36 1.192 -1.358 -0.966 -1.158
##  $ V2    : num  -0.0728 0.2662 -1.3402 -0.1852 0.8777
##  $ V28   : num  -0.0211 0.0147 -0.0598 0.0615 0.2152
##  $ Amount: num  149.62 2.69 378.66 123.5 69.99
##  $ Class : int  0 0 0 0 0
```

5. Discuss any data preparation

This dataset has NA value, it indicates this dataset is already preprocessed well. However, Amount and other columns has different unit, so we should use scale() to standardize.

```
# standardized Ammount column
data$Amount=scale(data$Amount)

# remove Time col since it's not major feature for this modeling
newdata=data[,-c(1)]
```

6. Modeling

To detect fraudulent transaction, using Logistic regression as a primary model. Firstly use caTools (https://cran.r-project.org/web/packages/caTools/index.html) and split dataset into train and test set by 80% vs 20%.

```
# set seed for randomisation for dataset split
set.seed(123)

# assign split label
data_sample = sample.split(newdata$Class,SplitRatio=0.80)
# create new variables based on split label
train_data = subset(newdata,data_sample==TRUE)
test_data = subset(newdata,data_sample==FALSE)
```

train dataset has 227846 rows for training and 56961 for training and those input data is standardized dataset. Because the target variable is binary(fraud or not), employ logistic regression model. Logistic regression uses logistic function to transform the output of the linear function into a probability value(fraud or not).

```
# Since we want to predict Class(0 or 1), put Class on the left hand side, and
# use train_data for fitting(changed from linked website, doesn't make sense using te
st set for training.)
# Also we set family as binominal() to tell model target is binary.
Logistic_Model <- glm(Class ~ ., data = train_data, family = binomial())
```

7. Result and Performance

Produce and discuss the results of the modeling. Was the modeling accurate? How do you know how well the modeling worked?

Let’s check each feature’s coefficients.. we could find couple features are statistically significant.

```
# get details of model
summary(Logistic_Model)
```

```
##
## Call:
## glm(formula = Class ~ ., family = binomial(), data = train_data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.651305   0.160212 -53.999  < 2e-16 ***
## V1           0.072540   0.044144   1.643 0.100332
## V2           0.014818   0.059777   0.248 0.804220
## V3           0.026109   0.049776   0.525 0.599906
## V4           0.681286   0.078071   8.726  < 2e-16 ***
## V5           0.087938   0.071553   1.229 0.219079
## V6          -0.148083   0.085192  -1.738 0.082170 .
## V7          -0.117344   0.068940  -1.702 0.088731 .
## V8          -0.146045   0.035667  -4.095 4.23e-05 ***
## V9          -0.339828   0.117595  -2.890 0.003855 **
## V10         -0.785462   0.098486  -7.975 1.52e-15 ***
## V11          0.001492   0.085147   0.018 0.986018
## V12          0.087106   0.094869   0.918 0.358532
## V13         -0.343792   0.092381  -3.721 0.000198 ***
## V14         -0.526828   0.067084  -7.853 4.05e-15 ***
## V15         -0.095471   0.094037  -1.015 0.309991
## V16         -0.130225   0.138629  -0.939 0.347537
## V17          0.032463   0.074471   0.436 0.662900
## V18         -0.100964   0.140985  -0.716 0.473909
## V19          0.083711   0.105134   0.796 0.425897
## V20         -0.463946   0.081871  -5.667 1.46e-08 ***
## V21          0.381206   0.065880   5.786 7.19e-09 ***
## V22          0.610874   0.142086   4.299 1.71e-05 ***
## V23         -0.071406   0.058799  -1.214 0.224589
## V24          0.255791   0.170568   1.500 0.133706
## V25         -0.073955   0.142634  -0.519 0.604109
## V26          0.120841   0.202553   0.597 0.550783
## V27         -0.852018   0.118391  -7.197 6.17e-13 ***
## V28         -0.323854   0.090075  -3.595 0.000324 ***
## Amount       0.292477   0.092075   3.177 0.001491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5799.1  on 227845  degrees of freedom
## Residual deviance: 1790.9  on 227816  degrees of freedom
## AIC: 1850.9
##
## Number of Fisher Scoring iterations: 12
```

Set 0.5 as threshold and check the confusion matrix for model performance.

```
# use trained model for test dataset
predictions <- predict(Logistic_Model, newdata = test_data, type = "response")
# set a threshold to determine binary result
predicted_class <- ifelse(predictions > 0.5, 1, 0)

# generate confusionMatrix
confusionMatrix(data = as.factor(predicted_class), reference = as.factor(test_data$Cl
ass))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 56856    41
##          1     7    57
##
##                Accuracy : 0.9992
##                  95% CI : (0.9989, 0.9994)
##     No Information Rate : 0.9983
##     P-Value [Acc > NIR] : 1.581e-08
##
##                   Kappa : 0.7033
##
##  Mcnemar's Test P-Value : 1.906e-06
##
##             Sensitivity : 0.9999
##             Specificity : 0.5816
##          Pos Pred Value : 0.9993
##          Neg Pred Value : 0.8906
##              Prevalence : 0.9983
##          Detection Rate : 0.9982
##    Detection Prevalence : 0.9989
##       Balanced Accuracy : 0.7908
##
##        'Positive' Class : 0
##
```

Given above result, this model demonstrates high accuracy (99.92%) with exceptional Recall(Sensitivity) (99.99%), indicating it's very effective at identifying legitimate transactions.

8. Visualization

A ROC plot is powerful visualization for evaluating model performance, since it visually represents the trade-off between the true positive rate and the false positive rate (1-specificity) across various thresholds.

```
# get predicted result
lr.predict <- predict(Logistic_Model, newdata = test_data, type = "response")

# use test data label and predicted label for ROC plot
auc.gbm = roc(test_data$Class, lr.predict, plot = TRUE, col = "blue")
```

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

```
## Setting direction: controls < cases
```