MATH/STAT 4450/8456 Machine Learning Competition #3

Predicting the fraud in self-checkout stores

The final contest for this course is to predict the fraud in self-checkout stores. In self-checkout stores or smart stores (like Amazon Go), customers can enjoy their shopping experience without long lines at the checkout counter. They can even use their own smartphone to scan the products. The scenario for our competition is a grocery store that allows customers to scan their products using a hand-held mobile scanner while shopping. However, some customers do not scan all of the items in their cart intentionally or inadvertently. Our goal is to detect those fraudulent purchases.

Kaggle website

The data can be downloaded from the kaggle site and you will have to participate the competition through the link https://www.kaggle.com/t/fb34f0415a3244b08dbfaf3ddd4eefd2 as it is not open to public.

A few things:

- The maximum daily submission number is 15. So you will need to wait until the next UTC day after submitting 15 results.
- There is a public leaderboard (11% of the test set) and a private leaderboard (89% of the test set). Your final grade will be based on the private leaderboard score.
- Please use your real name as the "team" name when making submissions, as I need to know who you are to get your grade. You will work individually so you are the only member of your team.

Description of variables

- id: row ID. Has no meaning.
- credit: Customer's credit level. Either low or high.
- duration: The time in seconds between the first and last scans.
- total: The total amount of all scanned and not cancelled products.
- scans: The number of all scanned and not cancelled products.
- voidedScans: The number of voided scans.
- attemptsWoScan: The number of attempts to activate the scanner without scanning any products.
- modifiedQuantities: The number of modified quantities for the scanned products.
- fraud: Whether the purchase is fraudulent or not. 1: fraud. 0: not fraud.

Data

Here is a quick look at the data.

```
train = read.csv("data/train.csv")
str(train)
```

```
## 'data.frame':    16630 obs. of  9 variables:
##  $ id                : int  1000001 1000002 1000003 1000004 1000005 1000006 1000007 1000008 1000009 1
##  $ credit            : Factor w/ 2 levels "High","Low": 1 2 2 1 1 2 2 1 1 2 ...
##  $ duration          : int  891 352 712 1681 373 268 231 623 788 1580 ...
```

```
##  $ total             : num  18.1 10.6 84.2 24.9 10.4 ...
##  $ scans             : int  9 5 5 11 6 5 24 16 14 4 ...
##  $ voidedScans       : int  11 0 9 4 8 2 2 8 2 10 ...
##  $ attemptsWoScan    : int  10 9 6 6 1 10 1 6 5 8 ...
##  $ modifiedQuantities: int  0 0 1 4 0 0 4 4 1 1 ...
##  $ fraud             : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
table(train$fraud)
```

```
##
##     0     1
## 14280  2350
```

The number of non-fraud cases is about 6 times the number of fraud cases. So this is an unbalanced classification problem. Even if you predict all cases by 0, you will obtain around 85% correct rows. Therefore we will use the weighted accuracy to evaluate your result. You may consider techniques like upsampling, subsampling, class weights, regularization, etc., to deal with the unbalanced problem.

```
test = read.csv("data/test.csv")
str(test)
```

```
## 'data.frame':    149675 obs. of  8 variables:
##  $ id                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ credit            : Factor w/ 2 levels "High","Low": 2 1 1 2 1 2 1 1 1 1 ...
##  $ duration          : int  770 1545 725 870 125 71 1355 866 335 1397 ...
##  $ total             : num  11.1 22.8 41.1 32.5 25.5 ...
##  $ scans             : int  26 10 27 6 24 1 7 4 5 25 ...
##  $ voidedScans       : int  11 0 10 3 5 1 2 10 0 7 ...
##  $ attemptsWoScan    : int  5 8 2 1 6 4 0 9 9 9 ...
##  $ modifiedQuantities: int  2 4 4 5 2 4 4 1 5 4 ...
```

The test set is about 9 times the size of the training set. About 1/9 of the test set is used in the public leaderboard.

Format of submission

Your submission file should be in the csv format with two columns: `id` and `fraud`. Example of the submission:

```
id,fraud
1,0
2,0
3,1
...
149675,0
```

Evaluation metrics

Instead of using the accuracy for evaluation, we use weighted accuracy given by the formula below.

$$\text{Weighted accuracy} = \sum_{i=1}^{149675} w_i I(y_i = \hat{y}_i)$$

where $w_i = 1/w$ if the ith observation belongs to the non-fraud class, and $w_i = 2/w$ if the ith observation belongs to the fraud class. We have $\sum_i w_i = 1$.

Task

1. Create the most accurate classifier that you can for the data, as measured by the test data.
2. Write a 10-15 page slides summarizing your approach to
   (a) formulating the model (design) matrix,
   (b) building the classifier,
   (c) results from all attempts,
   (d) your findings from the data.

NOTE: This is an individual competition same as the final exam. You may not share your code to anyone except the instructor. If your code or slides are duplicated with another student's, both of you will fail in this competition.

Deadlines

- May 4 (11:59 pm): Final prediction submission.
- May 5 (4 pm): Presentations. Only the top 10 participants will present the result. Each presentation should be around 5 minutes.
- May 5 (11:59 pm): Slides and code submission.

Grading

- Total points: 50
  - Accuracy of classifier: 20
    - Score = 20 * (leaderboard score)
  - Progress made from multiple submissions: 10
    - Number of good submissions (decreasing error rate): 6
    - Amount of decreasing of the error rate: 4
  - Presentation: 15
    - Model matrix: 5
    - Model selection and assessment: 5
    - Results and findings: 5
  - Met the deadlines: 5