Credit Card Data Set

Shashi Roshan (NetID : sroshan2)

Abstract

Statistical learning models were applied to credit card fraud detection data in order to predict possible credit card frauds. A variety of learning techniques were explored and validated, and ultimately the predictive power of the models appears to be good.

Introduction

We use credit card fraud detection data to identify fraudulent transactions. It is important for credit card companies to correctly identify frauds in credit card transactions to help their customers not pay for them, and also to minimize the losses incurred by the credit card companies. We use data to build a statistical model to detect frauds in credit card transactions. The performance of these statistical models will be evaluated on unseen data, to determine if they are useful.

Methods

Data

The data has been collected by Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB during September 2013, and the transactions were made by European credit card holders. The dataset has transactions which span over a period of two days. The features/columns in the dataset are masked, due to confidentiality. PCA has been used to achieve this masking. All the feature columns in the data set are the Principal components retrieved on the original data set. The feature Amount is the transaction amount of the credit card transaction. Feature Class is the response variable, and it denotes if the transaction was fraudulent or genuine. The data is hugely imbalance, with majority of the transactions having the label of genuine transactions. We are using a sub sample of the original data set for this analysis. The purpose of the analysis is to detect fraud transactions.

```r
# load packages
library("tidyverse")
library("caret")
library("gbm")
library("ROSE")
library('kableExtra')

# set seed
set.seed(4619)

# read in data
cc = read_csv(file = "https://fall-2019.stat432.org/analyses/data/cc-sub.csv")

# randomly split data
```

```
cc$Class = factor(cc$Class, levels = c('fraud', 'genuine'))
trn_idx = sample(nrow(cc), size = 0.5 * nrow(cc))
cc_trn = cc[trn_idx, ]
cc_tst = cc[-trn_idx, ]
```

Modelling

In order to identify fraudulent vs genuine transactions, four different statistical models were used : - Gradient boosting model was trained on all the available predictors. Class imbalance was not handled in this case. - Gradient boosting model was trained on all the available predictors. To handle class imbalance, ROSE method was used. - Logistic regression model was trained using all available predictor variables. To handle class imbalance, ROSE method was used. - Decision tree model was trained using all available predictors. The choice of the complexity parameter was chosen using cross validation. To handle class imbalance, ROSE method was used.

Evaluation

To evaluate the performance of different models, we split the data into train and test set. On the training set, we did cross validation technique to find the optimal set of parameters. Sensitivity metric was used to select the best model while performing hyper parameter tuning. Once all the model's performance was recorded using cross-validation results, we picked the best model. Then, the best model was tested on the test data set.

```
set.seed(42)
gbm_model_without_sampling = train(
  Class ~ . - Time,
  data = cc_trn,
  method = "gbm",
  metric = "Sens",
  trControl = trainControl(
    method = "cv",
    number = 5,
    classProbs = TRUE,
    summaryFunction = twoClassSummary
  ),
  verbose = FALSE
)

set.seed(42)
gbm_model = train(
  Class ~ . - Time,
  data = cc_trn,
  method = "gbm",
  metric = "Sens",
  trControl = trainControl(
    method = "cv",
    number = 5,
    classProbs = TRUE,
    summaryFunction = twoClassSummary,
    sampling = 'rose'
  ),
  verbose = FALSE
```

```r
)

set.seed(42)
rpart_model = train(
  Class ~ . - Time,
  data = cc_trn,
  method = "rpart",
  metric = "Sens",
  trControl = trainControl(
    method = "cv",
    number = 5,
    classProbs = TRUE,
    summaryFunction = twoClassSummary,
    sampling = 'rose'
  )
)

set.seed(42)
glm_model = train(
  Class ~ . - Time,
  data = cc_trn,
  method = "glm",
  metric = "Sens",
  trControl = trainControl(
    method = "cv",
    number = 5,
    classProbs = TRUE,
    summaryFunction = twoClassSummary,
    sampling = 'rose'
  ),
  family = 'binomial'
)
```

Results

```r
model_names = c(
  'Decision Tree Model',
  'Gradient Boosting Model (using ROSE)',
  'Logistic Regression Model',
  'Gradient Boosting Model'
)

model_results = c(
  max(rpart_model$results$Sens),
  max(gbm_model$results$Sens),
  max(glm_model$results$Sens),
  max(gbm_model_without_sampling$results$Sens)
)

model_description = c(
  'cp = 0.05',
  'n.trees = 50, interaction.depth = 1, shrinkage = 0.1',
  '',
  'n.trees = 50, interaction.depth = 3, shrinkage = 0.1'
)

df = data.frame(model_names, model_results, model_description)
df$model_results = round(df$model_results, 2)

tibble(
  'Model' = df$model_names,
  'Sensitivity (using Cross-validation)' = round(df$model_results, 2),
  'Model Description' = df$model_description
) %>%
  kable(digits = 2) %>%
  kable_styling("striped", full_width = FALSE)
```

Model	Sensitivity (using Cross-validation)	Model Description
Decision Tree Model	0.88	cp = 0.05
Gradient Boosting Model (using ROSE)	0.82	n.trees = 50, interaction.depth = 1, shrin
Logistic Regression Model	0.18	
Gradient Boosting Model	0.56	n.trees = 50, interaction.depth = 3, shrin

```
p1 = predict(rpart_model, cc_tst, type = 'prob')[, 'fraud']
p1 = factor(ifelse(p1 >= 0.5, 'fraud', 'genuine'),
            levels = c('fraud', 'genuine'))
cf = confusionMatrix(data = p1,
                     reference = cc_tst$Class,
                     positive = 'fraud')
```

Discussion

We used credit card frauds data to detect fraudulent transactions using statistical model. Decision tree model seems to be giving the best performance.

Sensitivity : 87.04 %

We found that the predictive performance of the best model is quite satisfactory, and it predicts fraudulent cases with very good sensitivity.

We should note that the data which we have used is a subset from the larger population, and future work can include creation of statistical models on the original data set. There is a possibility that the model's performance will improve if it is trained on the entire data set. Future work can also include validating the models on other performance metric like AUC, Precision, Recall, etc. Also, different sampling methods can be tried (we used ROSE method).