Branch: master kaggle-competitions / santander-customer-satisfaction / Create new file Upload files Find file History

caiomsouza Add Kaggle Santander Competion Latest commit 721ca6c a minute ago

..		
dat	Add Kaggle Santander Competion	14 hours ago
outputs	Add Kaggle Santander Competion	14 hours ago
src	Add Kaggle Santander Competion	43 minutes ago
README.md	Add Kaggle Santander Competion	a minute ago

README.md

Kaggle Santander Customer Satisfaction Competition

This project was created by Caio Fernandes Moreno, student at the faculty of statistics at UCM - Complutense University of Madrid - Universidad Complutense de Madrid.
The main purpose of this academic project is to learn data mining techniques, statistics, machine learning, R, SAS, data preparation, etc.

Neuronal Networks and Genetic Algorithms / Redes Neuronales y Algoritmos Genéticos
Master degree in Data Mining and Business Intelligence

Kaggle Competition dates:
Started: 7:43 pm, Wednesday 2 March 2016 UTC
Ended: 11:59 pm, Monday 2 May 2016 UTC (61 total days)

Competition link:
https://www.kaggle.com/c/santander-customer-satisfaction

The problem

Which customers are happy customers?

From frontline support teams to C-suites, customer satisfaction is a key measure of success. Unhappy customers don't stick around. What's more, unhappy customers rarely voice their dissatisfaction before leaving.

Santander Bank is asking Kagglers to help them identify dissatisfied customers early in their relationship. Doing so would allow Santander to take proactive steps to improve a customer's happiness before it's too late.

In this competition, you'll work with hundreds of anonymized features to predict if a customer is satisfied or dissatisfied with their banking experience.

Evaluation and Submission File

Evaluation

Submissions are evaluated on area under the ROC curve between the predicted probability and the observed target.

Submission File

For each ID in the test set, you must predict a probability for the TARGET variable. The file should contain a header and have the following format:

```
ID,TARGET
2,0
5,0
6,0
etc.
```

Dataset

The dataset contains 371 variables (all continuous variables).

A continuous variable is a variable that has an infinite number of possible values. In other words, any value is possible for the variable. A continuous variable is the opposite of a discrete variable, which can only take on a certain number of values.Sep 11, 2013

Numbers of observations (Row number):

- Train: 76020 rows
- Test: 75818 rows

Number of 1s (train): 3008 (3.95%) (Imbalanced Dataset Problem)

Variables:

- 34 variables with one single value; (Action: Delete all of them)
- 100 variables with two unique values; (binary variables)
- 157 variables with values between 3 y 101 unique values; (categorical variables)
- 80 variables has more than 101 distinct values; (continuous variables)

Unbalanced Dataset Problem

A dataset is said to be unbalanced when the class of interest (minority class) is much rarer than normal behaviour (majority class). The cost of missing a minority class is typically much higher that missing a majority class. Most learning systems are not prepared to cope with unbalanced data and several techniques have been proposed to rebalance the classes. This package implements some of most well-known techniques and propose a racing algorithm [2] to select adaptively the most appropriate strategy for a given unbalanced task [2].

Some links about Unbalanced Dataset Problem:

- http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
- http://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf
- http://florianhartl.com/thoughts-on-machine-learning-dealing-with-skewed-classes.html
- https://cran.r-project.org/web/packages/unbalanced/unbalanced.pdf
- https://www3.nd.edu/~dial/publications/chawla2005data.pdf

ML Algorithms

Gradient Boosting Machine

A Gradient Boosting Machine (GBM) is an ensemble of tree models (either regression or classification). Both are forward-learning ensemble methods that obtain predictive results through gradually improved estimates. Boosting is a flexible nonlinear regression procedure that helps improve the accuracy of trees. By sequentially applying weak classification

algorithms to incrementally changing data, a series of decision trees are created that produce an ensemble of weak prediction models. [1]

GBM is the most accurate general purpose algorithm. It can be used for analysis on numerous types of models and will always present relatively accurate results. Additionally, Gradient Boosting Machines are extremely robust, meaning that the user does not have to impute values or scale data (they can disregard distribution). This makes GBM the go-to choice for many users, as little tweaking is required in order to get accurate results. [1]

Load data

../src/utils/load_datasets.R

```
rm(list = ls())
setwd("~/git/github.com/kaggle-competitions/santander-customer-satisfaction")
path <- setwd("~/git/github.com/kaggle-competitions/santander-customer-satisfaction")

train.dataset.path <- "~/git/github.com/kaggle-competitions/santander-customer-satisfaction/dat/train.csv"
test.dataset.path <- "~/git/github.com/kaggle-competitions/santander-customer-satisfaction/dat/test.csv"

train <- read.csv(train.dataset.path)
cat("Train dataset loaded")
cat("at ")
cat(train.dataset.path)
test  <- read.csv(test.dataset.path)
cat("Test dataset loaded")
cat("at ")
cat(test.dataset.path)

#head(train,5)
#head(test,5)
#nrow(train)
#nrow(test)
#olnames(train)
#olnames(test)
#summary(train)
#summary(test)
```

Code:
https://github.com/caiomsouza/kaggle-competitions/blob/master/santander-customer-satisfaction/src/utils/load_datasets.R

Data preparation

The script below will do data preparation for the Santander Customer Satisfaction dataset.

- Extracting TARGET
- 0 count per line
- Removing constant features
- Removing identical features

../src/utils/clean_datasets.R

```
##### Removing IDs
#train$ID <- NULL
#test.id <- test$ID
#test$ID <- NULL


##### Extracting TARGET
train.y <- train$TARGET
```

```
train$TARGET <- NULL

##### 0 count per line
count0 <- function(x) {
  return( sum(x == 0) )
}
train$n0 <- apply(train, 1, FUN=count0)
test$n0 <- apply(test, 1, FUN=count0)

##### Removing constant features
cat("\n## Removing constant features.\n")
for (f in names(train)) {
  if (length(unique(train[[f]])) == 1) {
    cat("-", f, "\n")
    train[[f]] <- NULL
    test[[f]] <- NULL
  }
}

##### Removing identical features
cat("\n## Removing identical features.\n")
features_pair <- combn(names(train), 2, simplify = F)
toRemove <- c()
for(pair in features_pair) {
  f1 <- pair[1]
  f2 <- pair[2]

  if (!(f1 %in% toRemove) & !(f2 %in% toRemove)) {
    if (all(train[[f1]] == train[[f2]])) {
      cat("-", f2, "\n")
      toRemove <- c(toRemove, f2)
    }
  }
}

feature.names <- setdiff(names(train), toRemove)

train <- train[, feature.names]
test <- test[, feature.names]

train$TARGET <- train.y
```

Code:
https://github.com/caiomsouza/kaggle-competitions/blob/master/santander-customer-satisfaction/src/utils/clean_datasets.R

Data preparation output:

```
## Removing constant features.
- ind_var2_0
- ind_var2
- ind_var27_0
- ind_var28_0
- ind_var28
- ind_var27
- ind_var41
- ind_var46_0
- ind_var46
- num_var27_0
- num_var28_0
- num_var28
- num_var27
- num_var41
- num_var46_0
- num_var46
```

```
- saldo_var28
- saldo_var27
- saldo_var41
- saldo_var46
- imp_amort_var18_hace3
- imp_amort_var34_hace3
- imp_reemb_var13_hace3
- imp_reemb_var33_hace3
- imp_trasp_var17_out_hace3
- imp_trasp_var33_out_hace3
- num_var2_0_ult1
- num_var2_ult1
- num_reemb_var13_hace3
- num_reemb_var33_hace3
- num_trasp_var17_out_hace3
- num_trasp_var33_out_hace3
- saldo_var2_ult1
- saldo_medio_var13_medio_hace3

## Removing identical features.
- ind_var29_0
- ind_var29
- ind_var13_medio
- ind_var18
- ind_var26
- ind_var25
- ind_var32
- ind_var34
- ind_var37
- ind_var39
- num_var29_0
- num_var29
- num_var13_medio
- num_var18
- num_var26
- num_var25
- num_var32
- num_var34
- num_var37
- num_var39
- saldo_var29
- saldo_medio_var13_medio_ult1
- delta_num_reemb_var13_1y3
- delta_num_reemb_var17_1y3
- delta_num_reemb_var33_1y3
- delta_num_trasp_var17_in_1y3
- delta_num_trasp_var17_out_1y3
- delta_num_trasp_var33_in_1y3
- delta_num_trasp_var33_out_1y3
```

References

1. H2o.ai - Gradient Boosting Machine. Avaiable at: http://www.h2o.ai/verticals/algos/gbm/
2. Dal Pozzolo, Andrea, et al. "Racing for unbalanced methods selection." Intelligent Data Engi- neering and Automated Learning - IDEAL 2013. Springer Berlin Heidelberg, 2013. 24-31.