R.F. STAT 642 Kaggle Competition

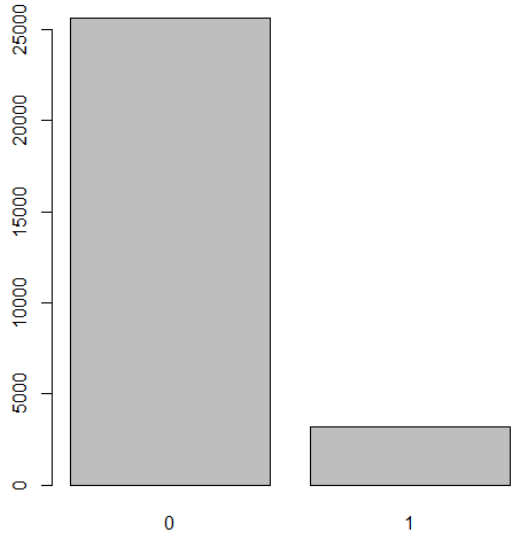
March 2023

Wang, Ruifeng
MS in Business Analytics
rw856@drexel.edu

The Dataset

- There are **28831** rows & **21** columns in training set, and **12357** rows & **20** columns in test set.
- The data in client information and target variable are imblanced.
- Unknown data can be recognized as NA, which decrease the predictive accuracy of algorithms.

```
table(x$ploan)
     no unknown     yes
  23795     663    4373

table(x$job)
      admin.      artisan entrepreneur    housemaid   management
        7255         6540         1010          759         2033

      retired selfemployed
        1218         1014

     services      student   technician   unemployed      unknown
        2779          597         4668          730          228

table(x$civil)
    divorced      married   single       unknown
        3198        17512     8065            56

table(x$edu)
          12K            4K            6K            9K
         6679          2982          1602          4250

apprenticeship     illiterate
         3656            15

    university        unknown
         8439          1208

table(x$credit)
     no unknown     yes
  22767    6061       3

table(x$hloan)
     no unknown     yes
  13088     663   15080
```



Data Preparation

- Dropping id column.
- Use kNN algorithm to impute NA(unknown).
- Using one hot encoding to convert categorical variables into binary vectors.
- Standarize Numeric variables from 0 to 1.
- Split training set into two parts to evaluate model.

kNN Imputation

```
# Missing Value imputation by kNN
## Replace unknown with NA
X$job[X$job == "unknown"] <- NA
X$civil[X$civil == "unknown"] <- NA
X$edu[X$edu == "unknown"] <- NA
X$credit[X$credit == "unknown"] <- NA
X$hloan[X$hloan == "unknown"] <- NA
X$ploan[X$ploan == "unknown"] <- NA

library(VIM)

df_imp <- kNN(X, variable = c("job", "civil", "edu",
                              "credit", "hloan", "ploan"), k = 10)
```

One Hot Encoding

```
## job column
X <- X %>%
  mutate(job_admin = ifelse(job == "admin.", 1, 0),
         job_artisan = ifelse(job == "artisan", 1, 0),
         job_entrepreneur = ifelse(job == "entrepreneur", 1, 0),
         job_housemaid = ifelse(job == "housemaid", 1, 0),
         job_management = ifelse(job == "management", 1, 0),
         job_retired = ifelse(job == "retired", 1, 0),
         job_selfemployed = ifelse(job == "selfemployed", 1, 0),
         job_services = ifelse(job == "services", 1, 0),
         job_student = ifelse(job == "student", 1, 0),
         job_technician = ifelse(job == "technician", 1, 0))

X = subset(X, select = -c(job))

## Civil column
X <- X %>%
  mutate(civil_divorced = ifelse(civil == "divorced", 1, 0),
         civil_married = ifelse(civil == "married", 1, 0))
```

Data Standardization

```
## cprice column
X <- X %>%
  mutate(scaled_cprice = (cprice - min(cprice)) /
           (max(cprice) - min(cprice)))

X = subset(X, select = -c(cprice))

## cconf column
X <- X %>%
  mutate(scaled_cconf = (cconf - min(cconf)) /
           (max(cconf) - min(cconf)))

X = subset(X, select = -c(cconf))
```

Modeling: XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly **efficient**, **flexible** and **portable**. It is a commonly used algorithms in machine learning, especially for classification.

This model can use parallel computing techniques to speed up the computational requirements. The more cores a computer have, the shorter time to require when run this model. In addition, XGBoost supports regularization in the form of both ridge and lasso, which can lead to further improvements.

Reference:
[1] XGBoost Documentation — xgboost 1.7.4 documentation
[2] Tutorial Week 10

Model Evaluation

- Using AUC to evaluate the model performance
- Tuning parameters to improve model performance.
- Retrain model by whole training set, output the result and submit.

```
xgb_params = list(
  objective = "binary:logistic",
  eta = 0.085,
  gamma = 0,
  max.depth = 2,
  min_child_weight = 1.9,
  eval_metric = "auc"
)
```

```
xgb <- xgboost(data = as.matrix(Xtrain[,1:46]),
               label = as.vector(Xtrain$outcome),
               params = xgb_params, nthread = 12,
               nrounds = 620, verbose = TRUE,
               early_stopping_rounds = 100,
               maximize = TRUE)

xgb.prob <- predict(xgb, as.matrix(Xtest[,1:46]),
                    type = "prob")
xgb.roc <- roc(Xtest$outcome, xgb.prob,
               plot = FALSE, quiet = TRUE)
auc(xgb.roc) # auc: 18:22 0.7968  18:23 0.7909  18
```

Managerial Implication

• Based on descriptive analysis and my observation, I found following insight:

1. Artisans are less likely to open an account, but retired people are more likely to open an account.
2. There are weak correlation between ages and target variable.
3. Change of employees (both ‘employee’ column and ‘employment’) will affect the consuming behaviors, which result in the decision on whether open an account.
4. As financial indicators, ‘cprice’, ‘cconf’ and ‘euri3’ reflect the economic condition, which also affect the number of potential customers.