Predictive Models on Telecom data

LF\_Kua

11/26/2023

# Objective

Using logistic and tree-based models to predict the likelihood of customers canceling their service with a telecommunication company

In this experiment, the Random forest model demonstrated superior performance, achieving the highest roc\_auc, with an accuracy of 80%, specificity of 90% and sensitivity of 57%, outperforming all the models

# Loading worksheet  
telecom\_df <- readRDS("C:/Users/user/Downloads/telecom\_df.rds")  
  
# Have a look at the structure of this telecom dataset  
str(telecom\_df)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 975 obs. of 9 variables:  
## $ canceled\_service : Factor w/ 2 levels "yes","no": 1 1 2 1 2 2 1 2 1 2 ...  
## $ cellular\_service : Factor w/ 2 levels "multiple\_lines",..: 2 2 2 1 1 2 1 1 2 1 ...  
## $ avg\_data\_gb : num 7.78 9.04 10.32 5.08 8.05 ...  
## $ avg\_call\_mins : num 497 336 262 250 328 326 525 312 417 340 ...  
## $ avg\_intl\_mins : num 127 88 55 107 122 114 97 147 96 136 ...  
## $ internet\_service : Factor w/ 2 levels "fiber\_optic",..: 1 1 1 2 2 1 1 1 2 1 ...  
## $ contract : Factor w/ 3 levels "month\_to\_month",..: 1 1 2 2 3 1 1 2 1 1 ...  
## $ months\_with\_company: num 7 10 50 53 50 25 19 50 8 61 ...  
## $ monthly\_charges : num 76.5 94.9 103 60 75.2 ...

# Preparations

## Loading R packages

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'purrr' was built under R version 4.1.3

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## Warning: package 'lubridate' was built under R version 4.1.3

library(parsnip)  
library(rsample)  
library(recipes)  
library(tune)  
library(workflows)

## Warning: package 'workflows' was built under R version 4.1.3

library(yardstick)

## Warning: package 'yardstick' was built under R version 4.1.3

# Data resampling to create train/test dataset

# Include strata argument to ensure similar outcome distribution  
set.seed(100)  
telecom\_split <- telecom\_df %>% initial\_split(prop = 0.75, strata = canceled\_service)  
telecom\_training <- training(telecom\_split)  
telecom\_test <- testing(telecom\_split)  
  
# View the telecom\_split  
telecom\_split

## <Training/Testing/Total>  
## <731/244/975>

# EDA

# Check for missing values  
colSums(is.na(telecom\_df))

## canceled\_service cellular\_service avg\_data\_gb avg\_call\_mins   
## 0 0 0 0   
## avg\_intl\_mins internet\_service contract months\_with\_company   
## 0 0 0 0   
## monthly\_charges   
## 0

# Check numeric variables for multicollinearity  
telecom\_training %>% select\_if(is.numeric) %>% cor()

## avg\_data\_gb avg\_call\_mins avg\_intl\_mins months\_with\_company  
## avg\_data\_gb 1.0000000 0.15535994 0.12806660 0.4281637  
## avg\_call\_mins 0.1553599 1.00000000 0.04243445 0.0161803  
## avg\_intl\_mins 0.1280666 0.04243445 1.00000000 0.2236808  
## months\_with\_company 0.4281637 0.01618030 0.22368084 1.0000000  
## monthly\_charges 0.9577667 0.16077613 0.13118271 0.4551050  
## monthly\_charges  
## avg\_data\_gb 0.9577667  
## avg\_call\_mins 0.1607761  
## avg\_intl\_mins 0.1311827  
## months\_with\_company 0.4551050  
## monthly\_charges 1.0000000

# Logistic regression and fitting model

# Create model specification  
lr\_model <- logistic\_reg() %>%  
 set\_engine("glm") %>%  
 set\_mode("classification")  
# fitting model  
lr\_fit <- lr\_model %>%   
 fit(canceled\_service ~ ., data = telecom\_training)  
  
# use augment instead to yield both predicted class and probability values  
lr\_aug <- augment(lr\_fit, telecom\_test)  
colnames(lr\_aug)

## [1] ".pred\_class" ".pred\_yes" ".pred\_no"   
## [4] "canceled\_service" "cellular\_service" "avg\_data\_gb"   
## [7] "avg\_call\_mins" "avg\_intl\_mins" "internet\_service"   
## [10] "contract" "months\_with\_company" "monthly\_charges"

# customize own metric set function  
tel\_metric <- metric\_set(accuracy, roc\_auc)  
  
lr\_no\_tune <- tel\_metric(lr\_aug, truth = canceled\_service, estimate = .pred\_class, .pred\_yes)  
lr\_no\_tune

## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.783  
## 2 roc\_auc binary 0.842

# Decision tree and fitting model

dt\_model <- decision\_tree() %>%  
 set\_engine("rpart") %>%  
 set\_mode("classification")  
# fitting model  
dt\_fit <- dt\_model %>%   
 fit(canceled\_service ~ ., data = telecom\_training)  
# One drawback of using a tree model is no longer have model coefficients to help interpret the model  
  
# use augment to yield both predicted class and probability values  
dt\_aug <- augment(dt\_fit, telecom\_test)  
colnames(dt\_aug)

## [1] ".pred\_class" ".pred\_yes" ".pred\_no"   
## [4] "canceled\_service" "cellular\_service" "avg\_data\_gb"   
## [7] "avg\_call\_mins" "avg\_intl\_mins" "internet\_service"   
## [10] "contract" "months\_with\_company" "monthly\_charges"

dt\_no\_tune <- tel\_metric(dt\_aug, truth = canceled\_service, estimate = .pred\_class, .pred\_yes)  
dt\_no\_tune

## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.795  
## 2 roc\_auc binary 0.790

# Compare lr and dt models

bind\_cols(lr\_no\_tune, dt\_no\_tune) %>%  
 select(1,3,6) %>%  
 rename(metric = .metric...1, lr\_no\_tune = .estimate...3, dt\_no\_tune = .estimate...6)

## New names:

## # A tibble: 2 x 3  
## metric lr\_no\_tune dt\_no\_tune  
## <chr> <dbl> <dbl>  
## 1 accuracy 0.783 0.795  
## 2 roc\_auc 0.842 0.790

*Logistic regression model perform better than decision tree model, with a slightly higher roc\_auc*

# Feature Engineering

# Define column roles and determine variable data type  
telecom\_recipe <- recipe(canceled\_service ~ ., data = telecom\_training) %>%  
 # and Data preprocessing steps  
 step\_corr(all\_numeric\_predictors(), threshold = 0.8) %>%  
 step\_normalize(all\_numeric\_predictors()) %>%  
 step\_dummy(all\_nominal\_predictors(), -all\_outcomes())  
  
# note that the roc\_auc = 0.674 when log transformed the numeric predictors  
# prep to train the recipe   
# bake to apply recipe to the same training data  
recipe\_object <- telecom\_recipe %>% prep()  
  
baked <- bake(recipe\_object, new\_data = NULL)  
str(baked)

## tibble [731 x 9] (S3: tbl\_df/tbl/data.frame)  
## $ avg\_data\_gb : num [1:731] 1.084 -0.105 0.55 0.602 0.896 ...  
## $ avg\_call\_mins : num [1:731] -1.135 -0.271 -0.297 -0.481 -0.114 ...  
## $ avg\_intl\_mins : num [1:731] -1.741 0.458 0.195 1.278 0.917 ...  
## $ months\_with\_company : num [1:731] 0.641 0.641 -0.359 0.641 1.081 ...  
## $ canceled\_service : Factor w/ 2 levels "yes","no": 2 2 2 2 2 2 2 2 2 2 ...  
## $ cellular\_service\_single\_line: num [1:731] 1 0 1 0 0 1 1 0 0 0 ...  
## $ internet\_service\_digital : num [1:731] 0 1 0 0 0 1 0 1 1 1 ...  
## $ contract\_one\_year : num [1:731] 1 0 0 1 0 0 0 0 0 1 ...  
## $ contract\_two\_year : num [1:731] 0 1 0 0 0 0 0 1 0 0 ...

# to explore the effects of each steps in the recipe after prepared  
tidy(recipe\_object, number = 1)

## # A tibble: 1 x 2  
## terms id   
## <chr> <chr>   
## 1 monthly\_charges corr\_h6Pon

tidy(recipe\_object, number = 3)

## # A tibble: 4 x 3  
## terms columns id   
## <chr> <chr> <chr>   
## 1 cellular\_service single\_line dummy\_EKyjl  
## 2 internet\_service digital dummy\_EKyjl  
## 3 contract one\_year dummy\_EKyjl  
## 4 contract two\_year dummy\_EKyjl

*monthly\_charges has been removed and the nominal variables have been dummy encoded*

# Create workflow object for lr\_model

### Tree-based models do not need much data preprocessing

# apply feature engineering to logistic regression model  
telecom\_lr\_wkfl <- workflow() %>%  
 add\_model(lr\_model) %>%  
 add\_recipe(telecom\_recipe)  
# view the workflow 1  
telecom\_lr\_wkfl

## == Workflow ====================================================================  
## Preprocessor: Recipe  
## Model: logistic\_reg()  
##   
## -- Preprocessor ----------------------------------------------------------------  
## 3 Recipe Steps  
##   
## \* step\_corr()  
## \* step\_normalize()  
## \* step\_dummy()  
##   
## -- Model -----------------------------------------------------------------------  
## Logistic Regression Model Specification (classification)  
##   
## Computational engine: glm

## Train, fit, predict and assess the lr workflow

# Train, fit and predict workflow 1  
telecom\_lr\_wkfl\_fit <- telecom\_lr\_wkfl %>%  
 last\_fit(split = telecom\_split)  
# view the performance metrics on the test set  
lr\_wkfl\_fit\_metrics <- telecom\_lr\_wkfl\_fit %>% collect\_metrics()  
lr\_wkfl\_fit\_metrics

## # A tibble: 2 x 4  
## .metric .estimator .estimate .config   
## <chr> <chr> <dbl> <chr>   
## 1 accuracy binary 0.775 Preprocessor1\_Model1  
## 2 roc\_auc binary 0.847 Preprocessor1\_Model1

# Compare 3 models

bind\_cols(lr\_no\_tune, dt\_no\_tune, lr\_wkfl\_fit\_metrics) %>%  
 select(1,3,6,9) %>%  
 rename(metric = .metric...1, lr\_no\_tune = .estimate...3, dt\_no\_tune = .estimate...6, lr\_preprocess = .estimate...9)

## New names:

## # A tibble: 2 x 4  
## metric lr\_no\_tune dt\_no\_tune lr\_preprocess  
## <chr> <dbl> <dbl> <dbl>  
## 1 accuracy 0.783 0.795 0.775  
## 2 roc\_auc 0.842 0.790 0.847

*The feature-engineered logistic regression model shows a minor rise in roc\_auc, but the decision tree model exhibits the highest accuracy*

# Bagged ensemble and fitting model

# Create model specification  
library(baguette)

## Warning: package 'baguette' was built under R version 4.1.3

bag\_model <- bag\_tree() %>%  
 set\_mode("classification") %>%  
 set\_engine("rpart", times = 20)  
# fitting model  
bag\_fit <- bag\_model %>%  
 fit(canceled\_service ~ ., data = telecom\_training)  
# view the model object for the feature importance derived from bag model  
bag\_fit

## parsnip model object  
##   
## Bagged CART (classification with 20 members)  
##   
## Variable importance scores include:  
##   
## # A tibble: 8 x 4  
## term value std.error used  
## <chr> <dbl> <dbl> <int>  
## 1 avg\_data\_gb 116. 2.27 20  
## 2 avg\_call\_mins 106. 2.60 20  
## 3 months\_with\_company 104. 3.42 20  
## 4 monthly\_charges 103. 2.88 20  
## 5 avg\_intl\_mins 75.7 2.49 20  
## 6 contract 39.0 3.14 20  
## 7 internet\_service 22.5 1.63 20  
## 8 cellular\_service 16.0 1.26 20

*Avg\_data\_gb seems to be the most relevant predictor for customer canceling the service*

## Assessing performance metrics for bag\_model

bag\_aug <- augment(bag\_fit, telecom\_test)  
colnames(bag\_aug)

## [1] ".pred\_class" ".pred\_yes" ".pred\_no"   
## [4] "canceled\_service" "cellular\_service" "avg\_data\_gb"   
## [7] "avg\_call\_mins" "avg\_intl\_mins" "internet\_service"   
## [10] "contract" "months\_with\_company" "monthly\_charges"

bag\_no\_tune <- tel\_metric(bag\_aug, truth = canceled\_service, estimate = .pred\_class, .pred\_yes)  
bag\_no\_tune

## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.787  
## 2 roc\_auc binary 0.830

# Compare 4 models

bind\_cols(lr\_no\_tune, dt\_no\_tune, lr\_wkfl\_fit\_metrics, bag\_no\_tune) %>%  
 select(1,3,6,9,13) %>%  
 rename(metric = .metric...1, lr\_no\_tune = .estimate...3, dt\_no\_tune = .estimate...6, lr\_preprocess = .estimate...9, bag\_no\_tune = .estimate...13)

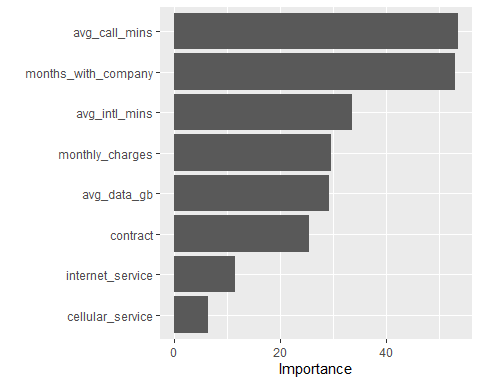
## New names:

## # A tibble: 2 x 5  
## metric lr\_no\_tune dt\_no\_tune lr\_preprocess bag\_no\_tune  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 accuracy 0.783 0.795 0.775 0.787  
## 2 roc\_auc 0.842 0.790 0.847 0.830

*Similarly, the feature-engineered logistic regression model shows the highest in roc\_auc, while the decision tree model exhibits the highest accuracy of 80%*

# Random Forest and fitting model

# Create model specification  
# rand\_forest() from parsnip package  
rf\_model <- rand\_forest() %>%  
 set\_mode("classification") %>%  
 set\_engine("ranger", importance = "impurity")  
# fitting model  
rf\_fit <- rf\_model %>%  
 fit(canceled\_service ~ ., data = telecom\_training)  
# view the model object for the feature importance derived from rf model  
vip::vip(rf\_fit)



*In contrast with the bag\_model, avg\_call\_mins seems to be the most relevant predictor for customer canceling the service*

## Assessing performance metrics for rf\_model

rf\_aug <- augment(rf\_fit, telecom\_test)  
colnames(rf\_aug)

## [1] ".pred\_class" ".pred\_yes" ".pred\_no"   
## [4] "canceled\_service" "cellular\_service" "avg\_data\_gb"   
## [7] "avg\_call\_mins" "avg\_intl\_mins" "internet\_service"   
## [10] "contract" "months\_with\_company" "monthly\_charges"

rf\_no\_tune <- tel\_metric(rf\_aug, truth = canceled\_service, estimate = .pred\_class, .pred\_yes)  
rf\_no\_tune

## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.795  
## 2 roc\_auc binary 0.850

# Compare 5 models

bind\_cols(lr\_no\_tune, dt\_no\_tune, lr\_wkfl\_fit\_metrics, bag\_no\_tune, rf\_no\_tune) %>%  
 select(1,3,6,9,13,16) %>%  
 rename(metric = .metric...1, lr\_no\_tune = .estimate...3, dt\_no\_tune = .estimate...6, lr\_preprocess = .estimate...9, bag\_no\_tune = .estimate...13, rf\_no\_tune = .estimate...16)

## New names:

## # A tibble: 2 x 6  
## metric lr\_no\_tune dt\_no\_tune lr\_preprocess bag\_no\_tune rf\_no\_tune  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 accuracy 0.783 0.795 0.775 0.787 0.795  
## 2 roc\_auc 0.842 0.790 0.847 0.830 0.850

*The random forest model outperform all the models with the highest roc\_auc of 0.848*

# Boosted Tree and fitting model

# Create model specification  
# rand\_forest() from parsnip package  
boost\_model <- boost\_tree() %>%  
 set\_mode("classification") %>%  
 set\_engine("xgboost")  
# fitting model  
boost\_fit <- boost\_model %>%  
 fit(canceled\_service ~ ., data = telecom\_training)

## Assessing performance metrics for boost\_model

boost\_aug <- augment(boost\_fit, telecom\_test)  
colnames(boost\_aug)

## [1] ".pred\_class" ".pred\_yes" ".pred\_no"   
## [4] "canceled\_service" "cellular\_service" "avg\_data\_gb"   
## [7] "avg\_call\_mins" "avg\_intl\_mins" "internet\_service"   
## [10] "contract" "months\_with\_company" "monthly\_charges"

boost\_no\_tune <- tel\_metric(boost\_aug, truth = canceled\_service, estimate = .pred\_class, .pred\_yes)  
boost\_no\_tune

## # A tibble: 2 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.783  
## 2 roc\_auc binary 0.833

# Compare 6 models

bind\_cols(lr\_no\_tune, dt\_no\_tune, lr\_wkfl\_fit\_metrics, bag\_no\_tune, rf\_no\_tune, boost\_no\_tune) %>%  
 select(1,3,6,9,13,16,19) %>%  
 rename(metric = .metric...1, lr\_no\_tune = .estimate...3, dt\_no\_tune = .estimate...6, lr\_preprocess = .estimate...9, bag\_no\_tune = .estimate...13, rf\_no\_tune = .estimate...16, boost\_no\_tune = .estimate...19) %>%  
 print(width = Inf)

## New names:

## # A tibble: 2 x 7  
## metric lr\_no\_tune dt\_no\_tune lr\_preprocess bag\_no\_tune rf\_no\_tune  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 accuracy 0.783 0.795 0.775 0.787 0.795  
## 2 roc\_auc 0.842 0.790 0.847 0.830 0.850  
## boost\_no\_tune  
## <dbl>  
## 1 0.783  
## 2 0.833

*The random forest model still outperform all the models with the highest roc\_auc of 0.850*

## Next we assess our selected model for its sensitivity and specificity metrics

tel\_metric\_1 <- metric\_set(roc\_auc, accuracy, sens, spec)  
  
rf\_no\_tune1 <- tel\_metric\_1(rf\_aug, truth = canceled\_service, estimate = .pred\_class, .pred\_yes)  
rf\_no\_tune1

## # A tibble: 4 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 accuracy binary 0.795  
## 2 sens binary 0.598  
## 3 spec binary 0.895  
## 4 roc\_auc binary 0.850

## Confusion Matrix of prediction using random forest

conf\_mat(rf\_aug, truth = canceled\_service, estimate = .pred\_class)

## Truth  
## Prediction yes no  
## yes 49 17  
## no 33 145

*Out of the 244 observations from the Testing data, the Random Forest model correctly classified 49 customers who canceled the service, while 33 customers who canceled the service were incorrectly classified as no canceled*

*In conclusion, the Random forest model demonstrated superior performance, achieving the highest roc\_auc, with an accuracy of 80%, specificity of 90% and sensitivity of 57%, outperforming all the models*