Titanic Survival Statistics

Thamma Tharasombat

Open library

```
library(titanic)
```

```
## Warning: package 'titanic' was built under R version 4.2.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.1     v forcats 0.5.2
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Drop NA

```
titanic_train <- na.omit(titanic_train)
```

Split Data

```r
set.seed(42)

n <- nrow(titanic_train)
id <- sample(1:n, size= n*0.7)
train_data <- titanic_train[id, ]
test_data <- titanic_train[-id, ]
```

Create model

```r
model <- glm(Survived ~ Pclass + Age + Sex + SibSp, data = train_data, family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Age + Sex + SibSp, family = "binomial",
##     data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9097  -0.6223  -0.3378   0.6033   2.4133
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.21585    0.69789   8.907  < 2e-16 ***
## Pclass      -1.49165    0.17867  -8.349  < 2e-16 ***
## Age         -0.05004    0.01031  -4.851 1.23e-06 ***
## Sexmale     -2.82454    0.27045 -10.444  < 2e-16 ***
## SibSp       -0.40560    0.15504  -2.616  0.00889 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 673.56  on 498  degrees of freedom
## Residual deviance: 424.77  on 494  degrees of freedom
## AIC: 434.77
##
## Number of Fisher Scoring iterations: 5
```

Train Data

```r
train_prob_su <- predict(model,type = "response")
train_data$pred_su <- ifelse(train_prob_su >= 0.5,1,0)

##Confuionmetric
conM <- table(train_data$pred_su, train_data$Survived,
              dnn = c("Predicted","Actuall"))
```

```r
##Model evaluation
train_a <- (conM[1,1] + conM[2,2])/ sum(conM)
train_p <- conM[2,2] / (conM[2,2] + conM[2,1])
train_r <- conM[2,2]/(conM[1,2]+conM[2,2])

cat("Acculacy:",train_a,"\nPrecision:",train_p,"\nRecall",train_r)
```

```
## Acculacy: 0.8136273
## Precision: 0.7853403
## Recall 0.7425743
```

```r
##Train F1
Train_f1 <- 2*((train_p*train_r)/(train_p+train_r))

cat("\nTrain F1 :", Train_f1)
```

```
##
## Train F1 : 0.7633588
```

Test Data

```r
test_prob_su <- predict(model, newdata = test_data, type = "response")
test_data$pred_su <- ifelse(test_prob_su >= 0.5,1,0)

##Confuionmetric
conM1 <- table(test_data$pred_su, test_data$Survived,
               dnn = c("Predicted","Actuall"))

##Model evaluation
test_a <- (conM1[1,1] + conM1[2,2])/ sum(conM1)
test_p <- conM1[2,2] / (conM1[2,2] + conM1[2,1])
test_r <- conM1[2,2]/(conM1[1,2]+conM1[2,2])

cat("Acculacy:",test_a,"\nPrecision:",test_p,"\nRecall",test_r)
```

```
## Acculacy: 0.8
## Precision: 0.7848101
## Recall 0.7045455
```

```r
#Test F1
Test_f1 <- 2*((test_p*test_r)/(test_p+test_r))

cat("\nTest F1 :", Test_f1)
```

```
##
## Test F1 : 0.742515
```

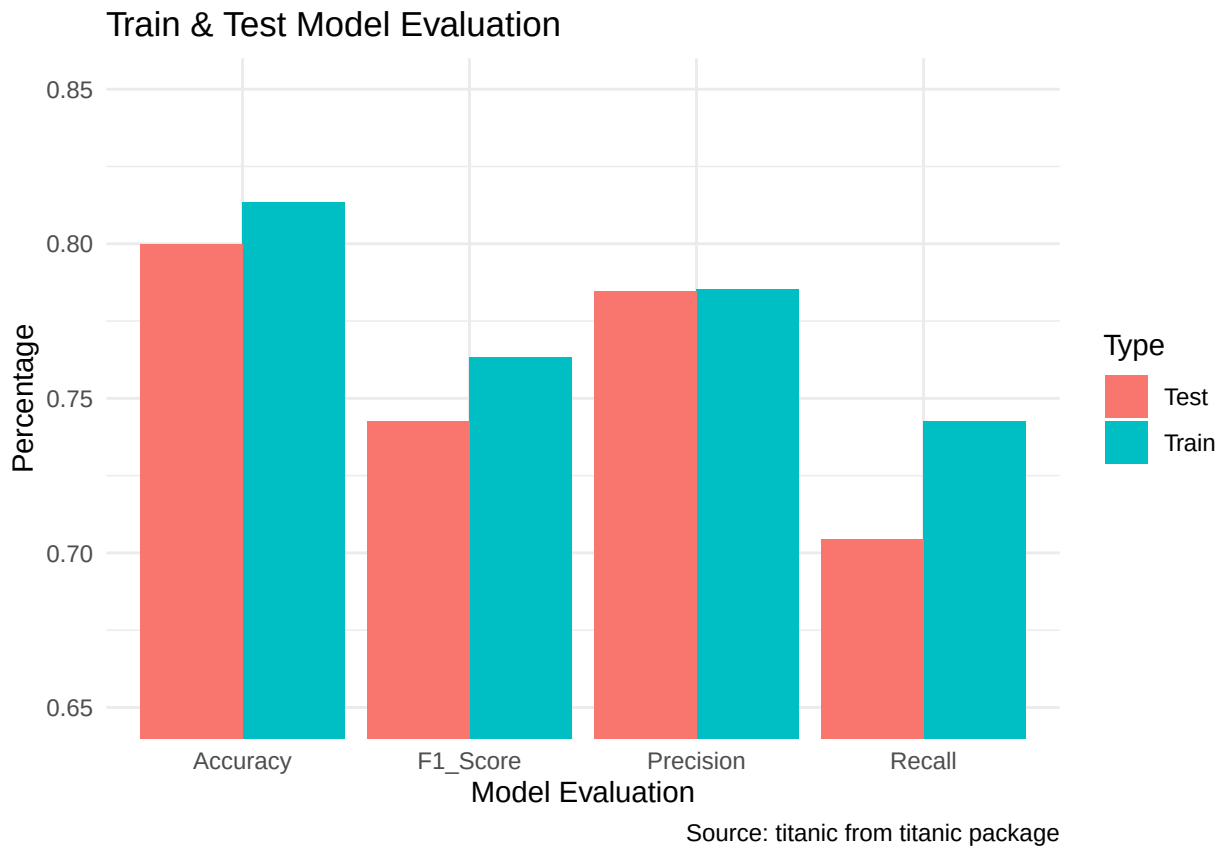
Plot model

```r
sum_model <- data.frame(
    Type = c('Train', 'Test'),
    Accuracy = c(train_a, test_a),
    Precision = c(train_p, test_p),
    Recall = c(train_r, test_r),
    F1_Score = c(Train_f1, Test_f1))

# Turn to data for plot
final <- sum_model %>%
    pivot_longer(-Type ,
                 names_to = "me_type",
                 values_to = "percent")
```

Plot data

```r
ggplot(final,aes(me_type, percent, fill = Type)) +
    geom_bar(stat='identity', position = 'dodge') +
    coord_cartesian(ylim = c(0.65, 0.85)) +
    theme_minimal() +
    labs(title = "Train & Test Model Evaluation",
         x ="Model Evaluation", y="Percentage",
         caption = "Source: titanic from titanic package")
```



The bar chart shows that the model performs better with train data. We can prove this theory based on the F1 score.