Titanic Survival Prediction by Logistic Regression

```
# install.packages("titanic")
library(titanic)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# clean all row have some value "NA"
# drop NA (missing values)
titanic_train = na.omit(titanic_train)
glimpse(titanic_train)
```

```
## Rows: 714
## Columns: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19~
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1~
## $ Pclass      <int> 3, 1, 3, 1, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 3, 2, 2, 3~
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl~
## $ Sex         <chr> "male", "female", "female", "female", "male", "male", "mal~
## $ Age         <dbl> 22, 38, 26, 35, 35, 54, 2, 27, 14, 4, 58, 20, 39, 14, 55, ~
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 1, 0, 0, 0~
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0, 0~
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37~
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 51.8625, 21.0750~
## $ Cabin       <chr> "", "C85", "", "C123", "", "E46", "", "", "", "G6", "C103"~
## $ Embarked    <chr> "S", "C", "S", "S", "S", "S", "S", "S", "C", "S", "S", "S"~
```

```
# Split Data
set.seed(42)
n = nrow(titanic_train)
id = sample(1:n , size = n*0.7) #70% train 30%test

train_data = titanic_train[id,]
nrow(train_data)
```

```
## [1] 499
```

```
test_data = titanic_train[-id,]
nrow(test_data)
```

```
## [1] 215
```

```r
## Train Model
train = glm(Survived ~ Pclass, data = train_data, family = "binomial")
## Predict
train_data$predict = ifelse(predict(train, type = "response") >= 0.5,1,0)
train
```

```
##
## Call:  glm(formula = Survived ~ Pclass, family = "binomial", data = train_data)
##
## Coefficients:
## (Intercept)       Pclass
##      1.6673      -0.9378
##
## Degrees of Freedom: 498 Total (i.e. Null);  497 Residual
## Null Deviance:       673.6
## Residual Deviance: 604   AIC: 608
```

```r
## Test Model
test = glm(Survived ~ Pclass, data = test_data, family = "binomial")
## Predict
test_data$predict = ifelse(predict(test, type = "response") >= 0.5,1,0)

## Confusion Matrix
conTable = table(test_data$predict, test_data$Survived, dnn = c("Predict","Actual"))
conTable
```

```
##        Actual
## Predict   0   1
##       0 107  54
##       1  20  34
```

```r
## accuracy
test_accuracy =
(( conTable[1,1] + conTable[2,2] ) / nrow(test_data))
test_accuracy
```

```
## [1] 0.655814
```

```r
## precision
test_precision =
(conTable[2,2] / ( conTable[2,2] + conTable[2,1] ))
test_precision
```

```
## [1] 0.6296296
```

```r
## recall
test_recall =
(conTable[2,2] / ( conTable[2,2] + conTable[1,2] ))
test_recall
```

```
## [1] 0.3863636
## F1 Score
test_f1s =
((conTable[2,2] / ( conTable[2,2] + conTable[2,1] )) *
    (conTable[2,2] / ( conTable[2,2] + conTable[1,2] )) /
    (conTable[2,2] / ( conTable[2,2] + conTable[2,1] )) +
    (conTable[2,2] / ( conTable[2,2] + conTable[1,2] )))
test_f1s
```

```
## [1] 0.7727273
```

```
# build  DataFrame
test_df = data.frame(
  Model = "Test",
  accuracy = test_accuracy,
  precision = test_precision,
  recall = test_recall,
  F1_Score = test_f1s
)

# Summary_train model
summary(train)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass, family = "binomial", data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4987  -0.7430  -0.7430   0.8871   1.6865
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.6673     0.2730   6.107 1.02e-09 ***
## Pclass       -0.9378     0.1179  -7.951 1.85e-15 ***
## ---
## 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: 603.97  on 497  degrees of freedom
## AIC: 607.97
##
## Number of Fisher Scoring iterations: 4
```

```
# Summary_test model
tibble(test_df)
```

```
## # A tibble: 1 x 5
##   Model accuracy precision recall F1_Score
##   <chr>    <dbl>     <dbl>  <dbl>    <dbl>
## 1 Test     0.656     0.630  0.386    0.773
```