Sprint 06 - titanic_train model

In this project, I will use titanic_train data to build a model that will predict whether a passenger survived based on passenger class and age.

Titanic train dataset

```
library(titanic)
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

head(titanic_train)
```

Drop NA (missing value)

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

## [1] 714
```

Change class of 'Survived'

```
titanic_train$Survived <- as.factor(titanic_train$Survived)
```

Split data

```
set.seed(27)
n <- nrow(titanic_train)
id <- sample(1:n,size = n*0.7) # 70% train 30% test
train_data <- titanic_train[id, ]
test_data <- titanic_train[-id, ]
```

Train data

Train model

```
model_train <- glm(Survived ~ Pclass + Age, data = train_data,
                   family = "binomial")
summary(model_train)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Age, family = "binomial", data = train_d
ata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8865  -0.7979  -0.5827   0.9042   2.4137
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.699465   0.487406   7.590 3.20e-14 ***
## Pclass      -1.338449   0.143734  -9.312  < 2e-16 ***
## Age         -0.040338   0.008145  -4.952 7.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 670.35  on 498  degrees of freedom
## Residual deviance: 559.81  on 496  degrees of freedom
## AIC: 565.81
##
## Number of Fisher Scoring iterations: 4
```

Predict using train model

```
train_data$prob_survived <- predict(model_train, type = "response")
train_data$pred_survived <- ifelse(train_data$prob_survived >= 0.5, 1, 0)
```

Confusion matrix of train model

```
con_m_train <- table(train_data$pred_survived, train_data$Survived,
                     dnn = c("predicted","actual"))
con_m_train
```

```
##          actual
## predicted   0   1
##         0 247  89
##         1  54 109
```

Evaluate train model

```
acc_train <- (con_m_train[1,1]+con_m_train[2,2]) / sum(con_m_train)
prec_train <- con_m_train[2,2] / (con_m_train[2,2]+con_m_train[2,1])
rec_train <- con_m_train[2,2] / (con_m_train[1,2]+con_m_train[2,2])
f1_train <- 2*(prec_train*rec_train)/(prec_train+rec_train)

# Print results
cat("Accuracy =", acc_train, "\nPrecistion =", prec_train,
    "\nRecall =", rec_train, "\nF1 Score =", f1_train)
```

```
## Accuracy = 0.7134269
## Precistion = 0.6687117
## Recall = 0.5505051
## F1 Score = 0.6038781
```

Test data

Test model

```r
model_test <- glm(Survived ~ Pclass + Age, data = test_data,
                  family = "binomial")
summary(model_test)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Age, family = "binomial", data = test_da
ta)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1246  -0.9480  -0.6862   1.1163   1.9606
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.27012    0.74488   4.390 1.13e-05 ***
## Pclass      -1.03612    0.21496  -4.820 1.44e-06 ***
## Age         -0.04376    0.01208  -3.622 0.000292 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.57  on 214  degrees of freedom
## Residual deviance: 264.66  on 212  degrees of freedom
## AIC: 270.66
##
## Number of Fisher Scoring iterations: 4
```

Predicting using test model

```r
test_data$prob_survived <- predict(model_test, type = "response")
test_data$pred_survived <- ifelse(test_data$prob_survived >= 0.5, 1, 0)
```

Confusion matrix of test model

```r
con_m_test <- table(test_data$pred_survived, test_data$Survived,
                    dnn = c("predicted","actual"))
con_m_test
```

```
##          actual
## predicted  0  1
##          0 95 42
##          1 28 50
```

Evaluate test model

```
acc_test <- (con_m_test[1,1]+con_m_test[2,2]) / sum(con_m_test)
prec_test <- con_m_test[2,2] / (con_m_test[2,2]+con_m_test[2,1])
rec_test <- con_m_test[2,2] / (con_m_test[1,2]+con_m_test[2,2])
f1_test <- 2*(prec_test*rec_test)/(prec_test+rec_test)

# Print results
cat("Accuracy =", acc_test, "\nPrecistion =", prec_test,
    "\nRecall =", rec_test, "\nF1 Score =", f1_test)
```

```
## Accuracy = 0.6744186
## Precistion = 0.6410256
## Recall = 0.5434783
## F1 Score = 0.5882353
```

Evaluation metrics for train and test model

```
# Create a data frame to store the evaluation metrics
evaluation_metrics <- data.frame(
  Accuracy = c(acc_test,acc_train),
  Precision = c(prec_test,prec_train),
  Recall = c(rec_test,rec_train),
  F1_score = c(f1_test,f1_train),
  row.names = c("Test", "Train")
)

print(evaluation_metrics)
```

```
##        Accuracy Precision    Recall  F1_score
## Test  0.6744186 0.6410256 0.5434783 0.5882353
## Train 0.7134269 0.6687117 0.5505051 0.6038781
```