01 라이브러리 불러오기

02 데이터 불러오기

03 데이터 전처리

범주형으로 변환

파생변수 생성

03 데이터 전처리

결측치 처리

04 데이터 나누기

05 데이터 모델 만들기

05 데이터 모델 학습 후, 예측

05 데이터 모델 학습 후, 예측, 모델 평가

06 데이터 모델 학습 후, 예측, 모델 평가 - 앙상블 모델

06 데이터 모델 학습 후, 예측, 모델 평가 - 앙상블 모델

예측

06 데이터 모델 학습 후, 예측, 모델 평가 - 앙상블 모델

07 가장 좋은 모델로 최종 예측

예측

제출

Rproject02B_Titanic

01 라이브러리 불러오기

- dplyr : 데이터 처리
- caret : 모델 평가
- rpart : 의사결정트리
- randomForest : 랜덤 포레스트(앙상블)

```
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(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
set.seed(1004)
```

02 데이터 불러오기

```
train <-  read.csv("./R_Data/titanic_train.csv", stringsAsFactors=F, na.strings = c(
"", "NA"))
test <- read.csv("./R_Data/titanic_test.csv", stringsAsFactors=F, na.strings = c("",
"NA"))
sub <- read.csv("./R_Data/sample_submission.csv", stringsAsFactors=F)
```

03 데이터 전처리

- 학습용 데이터, 테스트 데이터를 하나로 만들어 처리.

```
test$Survived <- NA
all <- rbind(train, test)
colSums(is.na(all))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1        1014           2
```

범주형으로 변환

- 성별(Sex)
- 생존 유무(Survived)
- 등급(Pclass)

```
all$Sex <- as.factor(all$Sex)
all$Survived <- as.factor(all$Survived)
all$Pclass <- as.ordered(all$Pclass)
str(all)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Floren
ce Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May P
eel)" ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

파생변수 생성

- Pclass와 Sex를 이용한 변수 생성

```
all$PclassSex[all$Pclass=='1' & all$Sex=='male'] <- 'P1Male'
all$PclassSex[all$Pclass=='2' & all$Sex=='male'] <- 'P2Male'
all$PclassSex[all$Pclass=='3' & all$Sex=='male'] <- 'P3Male'
all$PclassSex[all$Pclass=='1' & all$Sex=='female'] <- 'P1Female'
all$PclassSex[all$Pclass=='2' & all$Sex=='female'] <- 'P2Female'
all$PclassSex[all$Pclass=='3' & all$Sex=='female'] <- 'P3Female'
all$PclassSex <- as.factor(all$PclassSex)
names(all); table(all$PclassSex)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"    "PclassSex"
```

```
##
## P1Female   P1Male P2Female   P2Male P3Female   P3Male
##      144      179      106      171      216      493
```

03 데이터 전처리

결측치 확인

- Pclass와 Sex를 이용한 변수 생성

```
all[is.na(all$Fare), ]
```

```
##      PassengerId Survived Pclass               Name  Sex  Age SibSp Parch
## 1044        1044     <NA>      3 Storey, Mr. Thomas male 60.5     0     0
##      Ticket Fare Cabin Embarked PclassSex
## 1044   3701   NA  <NA>        S    P3Male
```

```
all[is.na(all$Embarked), ]
```

```
##     PassengerId Survived Pclass                                      Name
## 62           62        1      1                       Icard, Miss. Amelie
## 830         830        1      1 Stone, Mrs. George Nelson (Martha Evelyn)
##        Sex Age SibSp Parch Ticket Fare Cabin Embarked PclassSex
## 62  female  38     0     0 113572   80   B28     <NA>  P1Female
## 830 female  62     0     0 113572   80   B28     <NA>  P1Female
```

```
names(all)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"        
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"       
## [11] "Cabin"       "Embarked"    "PclassSex"
```

```
str(all)
```

```
## 'data.frame':    1309 obs. of  13 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Floren
ce Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May P
eel)" ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
##  $ PclassSex  : Factor w/ 6 levels "P1Female","P1Male",..: 6 1 5 1 6 6 2 6 5 3 ...
```

```
all %>% group_by(PclassSex) %>% summarise(n=n(),
                                          mean_age=mean(Age, na.rm=T),
                                          median_age=median(Age, na.rm=T))
```

```
## # A tibble: 6 x 4
##   PclassSex     n mean_age median_age
##   <fct>     <int>    <dbl>      <dbl>
## 1 P1Female    144     37.0       36
## 2 P1Male      179     41.0       42
## 3 P2Female    106     27.5       28
## 4 P2Male      171     30.8       29.5
## 5 P3Female    216     22.2       22
## 6 P3Male      493     26.0       25
```

결측치 처리

- 정박항은 다수의 값으로
- 나이는 등급별/성별 중앙값으로

```
all[ is.na(all$Embarked), 'Embarked'] = 'S'
all[ is.na(all$Fare), 'Fare'] = median(all$Fare,na.rm=T)

all[ is.na(all$Age) & all$PclassSex=="P1Female", 'Age'] = 36
all[ is.na(all$Age) & all$PclassSex=="P1Male", 'Age'] = 42

all[ is.na(all$Age) & all$PclassSex=="P2Female", 'Age'] = 28
all[ is.na(all$Age) & all$PclassSex=="P2Male", 'Age'] = 29.5

all[ is.na(all$Age) & all$PclassSex=="P3Female", 'Age'] = 22
all[ is.na(all$Age) & all$PclassSex=="P3Male", 'Age'] = 25

colSums(is.na(all))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age 
##           0         418           0           0           0           0 
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked 
##           0           0           0           0        1014           0 
##   PclassSex 
##           0
```

04 데이터 나누기

- 학습용
- 테스트용(제출)

```
all$Embarked <- as.factor(all$Embarked)
trainClean <- all[!is.na(all$Survived),]
nrow(trainClean);
```

```
## [1] 891
```

```
# 학습용(모델학습, 모델평가)
idx <- sample(1:nrow(trainClean), size=nrow(trainClean)*0.7, replace=F)
train_tr <- trainClean[idx, ]
train_test <- trainClean[-idx, ]

# 제출용(테스트용)
testClean <- all[is.na(all$Survived),]
nrow(testClean);
```

```
## [1] 418
```

05 데이터 모델 만들기

- 로지스틱 회귀 모델

```
m <- glm(Survived ~ Pclass + Sex + Age + SibSp + Embarked + PclassSex, family=binomia
l, data=train_tr)
summary(m)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Embarked +
##     PclassSex, family = binomial, data = train_tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0274  -0.6052  -0.4462   0.3616   2.6610
##
## Coefficients: (3 not defined because of singularities)
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.84101    0.48561   5.850 4.90e-09 ***
## Pclass.L           -2.73467    0.48455  -5.644 1.66e-08 ***
## Pclass.Q            1.44228    0.42491   3.394 0.000688 ***
## Sexmale            -1.77680    0.29907  -5.941 2.83e-09 ***
## Age                -0.04938    0.01061  -4.652 3.29e-06 ***
## SibSp              -0.35612    0.12639  -2.818 0.004837 **
## EmbarkedQ           0.08404    0.44383   0.189 0.849813
## EmbarkedS          -0.33632    0.29662  -1.134 0.256865
## PclassSexP1Male    -1.94436    0.71090  -2.735 0.006237 **
## PclassSexP2Female   2.95269    0.70263   4.202 2.64e-05 ***
## PclassSexP2Male          NA         NA      NA       NA
## PclassSexP3Female        NA         NA      NA       NA
## PclassSexP3Male          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 824.69  on 622  degrees of freedom
## Residual deviance: 511.13  on 613  degrees of freedom
## AIC: 531.13
##
## Number of Fisher Scoring iterations: 5
```

05 데이터 모델 학습 후, 예측

```
pred <- predict(m, newdata=train_test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
pred[0:15]
```

```
##          1         14         16         17         18         23
## 0.11310787 0.05221481 0.82694520 0.15188648 0.12939787 0.69836518
##         25         27         31         40         41         44
## 0.42468460 0.18018887 0.41759707 0.61033893 0.23661317 0.98388632
##         47         48         50
## 0.14342270 0.62102425 0.47874246
```

```
pred <- as.integer(pred > 0.5)
pred[0:15]
```

```
##  [1] 0 0 1 0 0 1 0 0 0 1 0 1 0 1 0
```

```
length(pred)
```

```
## [1] 268
```

05 데이터 모델 학습 후, 예측, 모델 평가

```
actual <- train_test[ ,"Survived"]
xt = xtabs(~ pred + actual)
xt
```

```
##     actual
## pred   0   1
##    0 149  42
##    1  11  66
```

```
# library(caret)
pred <- as.factor(pred)
actual <- as.factor(actual)
confusionMatrix(pred, actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 149  42
##          1  11  66
##
##                Accuracy : 0.8022
##                  95% CI : (0.7494, 0.8482)
##     No Information Rate : 0.597
##     P-Value [Acc > NIR] : 5.961e-13
##
##                   Kappa : 0.5689
##
##  Mcnemar's Test P-Value : 3.775e-05
##
##             Sensitivity : 0.9313
##             Specificity : 0.6111
##          Pos Pred Value : 0.7801
##          Neg Pred Value : 0.8571
##              Prevalence : 0.5970
##          Detection Rate : 0.5560
##    Detection Prevalence : 0.7127
##       Balanced Accuracy : 0.7712
##
##        'Positive' Class : 0
##
```

```
str(train_tr)
```

```
## 'data.frame':    623 obs. of  13 variables:
##  $ PassengerId: int  395 760 854 673 845 517 272 52 355 298 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 2 1 1 1 ...
##  $ Pclass     : Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 1 2 3 2 3 3 3 1 ...
##  $ Name       : chr  "Sandstrom, Mrs. Hjalmar (Agnes Charlotta Bengtsson)" "Rothe
s, the Countess. of (Lucy Noel Martha Dyer-Edwards)" "Lines, Miss. Mary Conover" "Mit
chell, Mr. Henry Michael" ...
##  $ Sex        : Factor w/ 2 levels "female","male": 1 1 1 2 2 1 2 2 2 1 ...
##  $ Age        : num  24 33 16 70 17 34 25 21 25 2 ...
##  $ SibSp      : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Parch      : int  2 0 1 0 0 0 0 0 0 2 ...
##  $ Ticket     : chr  "PP 9549" "110152" "PC 17592" "C.A. 24580" ...
##  $ Fare       : num  16.7 86.5 39.4 10.5 8.66 ...
##  $ Cabin      : chr  "G6" "B77" "D28" NA ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 3 3 3 3 3 3 3 1 3 ...
##  $ PclassSex  : Factor w/ 6 levels "P1Female","P1Male",..: 5 1 1 4 6 3 6 6 6 1 ...
```

06 데이터 모델 학습 후, 예측, 모델 평가 - 앙상블 모델

```
# library(randomForest)
m2 <- randomForest(Survived ~ Pclass + Sex + PclassSex + SibSp + Age + Fare + Embarke
d, data=train_tr)
summary(m2)
```

```
##                 Length Class  Mode
## call               3   -none- call
## type               1   -none- character
## predicted        623   factor numeric
## err.rate        1500   -none- numeric
## confusion          6   -none- numeric
## votes           1246   matrix numeric
## oob.times        623   -none- numeric
## classes            2   -none- character
## importance         7   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y                623   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

06 데이터 모델 학습 후, 예측, 모델 평가 - 앙상블 모델

예측

```
rf_pred <- predict(m2, newdata=train_test, type=c("prob"))
rf_pred[0:15]
```

```
##  [1] 0.926 0.970 0.174 0.852 1.000 0.212 0.802 0.974 0.746 0.492 0.694
## [12] 0.096 0.866 0.082 0.504
```

```
rf_pred <- predict(m2, newdata=train_test, type=c("class"))
rf_pred[0:15]
```

```
##  1 14 16 17 18 23 25 27 31 40 41 44 47 48 50
##  0  0  1  0  0  1  0  0  0  1  0  1  0  1  0
## Levels: 0 1
```

```
length(pred)
```

```
## [1] 268
```

06 데이터 모델 학습 후, 예측, 모델 평가 - 앙상블 모델

```
# library(caret)
actual <- train_test[ ,"Survived"]
pred <- as.factor(rf_pred)
actual <- as.factor(actual)
confusionMatrix(pred, actual)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 143  36
##          1  17  72
## 
##                Accuracy : 0.8022
##                  95% CI : (0.7494, 0.8482)
##     No Information Rate : 0.597
##     P-Value [Acc > NIR] : 5.961e-13
## 
##                   Kappa : 0.5769
## 
##  Mcnemar's Test P-Value : 0.01342
## 
##             Sensitivity : 0.8938
##             Specificity : 0.6667
##          Pos Pred Value : 0.7989
##          Neg Pred Value : 0.8090
##              Prevalence : 0.5970
##          Detection Rate : 0.5336
##    Detection Prevalence : 0.6679
##       Balanced Accuracy : 0.7802
## 
##        'Positive' Class : 0
## 
```

07 가장 좋은 모델로 최종 예측

예측

```
nrow(testClean)
```

```
## [1] 418
```

```
pred <- predict(m2, newdata=testClean, type="prob")
pred[0:15,2]
```

```
##   892   893   894   895   896   897   898   899   900   901   902   903 
## 0.002 0.312 0.138 0.020 0.542 0.156 0.540 0.098 0.782 0.058 0.000 0.084 
##   904   905   906 
## 0.940 0.080 0.998
```

```
length(pred[,2])
```

```
## [1] 418
```

```
pred <- as.integer(pred[,2] > 0.5)
pred[0:15]
```

```
##  [1] 0 0 0 0 1 0 1 0 1 0 0 0 1 0 1
```

```
length(pred)
```

```
## [1] 418
```

제출

```
sub[ ,'Survived'] = pred
sub[0:15,]
```

```
##    PassengerId Survived
## 1          892        0
## 2          893        0
## 3          894        0
## 4          895        0
## 5          896        1
## 6          897        0
## 7          898        1
## 8          899        0
## 9          900        1
## 10         901        0
## 11         902        0
## 12         903        0
## 13         904        1
## 14         905        0
## 15         906        1
```

```
write.csv(sub, file="SecondSub.csv", row.names = F)
list.files(path=".", pattern=NULL)
```

```
##  [1] "df_score.csv"
##  [2] "df_score.rda"
##  [3] "firstSub.csv"
##  [4] "img"
##  [5] "pdf"
##  [6] "R_Data"
##  [7] "R_STAT_ANALYSIS"
##  [8] "RBasic_Source"
##  [9] "README.md"
## [10] "RLevelUp_Source"
## [11] "RProject_practice_withdoit.ipynb"
## [12] "RProject01A_dplyr_withdoit_v11.ipynb"
## [13] "RProject01B_dplyr_ggplot_withdoit.ipynb"
## [14] "RProject01C_dplyr_ggplot_withdoit.ipynb"
## [15] "RProject02A_Titanic.html"
## [16] "RProject02A_Titanic.md"
## [17] "RProject02A_Titanic.rmd"
## [18] "RProject02A_Titanic_files"
## [19] "RProject02B_Titanic.html"
## [20] "RProject02B_Titanic.rmd"
## [21] "SecondSub.csv"
```