TITANIC

ABS

2023-02-07

```
library(tidyverse)
library(caret)
library(rpart)
library(kableExtra)
```

Explore the dataset

```
dataset <- read.csv(file.choose())
dim(dataset)
```

```
[1] 891  12
```

```
table(is.na(dataset))
```

```
FALSE  TRUE
10515   177
```

```
for(i in 1:ncol(dataset)){
  cat(names(dataset)[i],sum(is.na(dataset[,i])),"\n")
}
```

```
PassengerId 0
Survived 0
Pclass 0
Name 0
Sex 0
Age 177
SibSp 0
Parch 0
Ticket 0
Fare 0
Cabin 0
Embarked 0
```

```
length(unique(dataset$Cabin))
```

```
[1] 148
```

```
data_main <- dataset %>%
  select(PassengerId,Survived,Pclass,Sex,Age,SibSp, Parch,  Fare, Embarked)
age_test <- data_main %>%
  filter(is.na(Age))
age_train <- data_main %>%
```

```
  filter(!is.na(Age))
agena <- na.omit(age_train)
md_age <- lm(Age ~ ., data = agena[,!names(agena) %in% c("PassengerId")])
age_test$Age <- predict(md_age, age_test[,!names(age_test) %in% c("Age","PassengerId")]) %>%
  ceiling()

data_imp <- age_train %>%
  full_join(age_test)
```

Split the dataset

```
t <- createDataPartition(data_imp$PassengerId, p =.8, list = F)
tran <- data_imp[t,]
test <- data_imp[-t,]
```

Fit a logistic model

```
log_m <- tran %>%
  select(-PassengerId) %>%
  glm(Survived ~ ., family = binomial(link = "logit"), data = .)
```

Random forest

```
rf_m <- tran %>%
  select(-PassengerId) %>%
  train( Survived ~ . , data = ., method  = "rf", trControl = trainControl("cv"))
```

Xgboost model

```
xg <- tran %>%
  select(-PassengerId) %>%
  train( Survived ~ . , data = ., method  = "xgbTree", trControl = trainControl("cv"))
```

Tree model

```
DT <- tran %>%
  select(-PassengerId) %>%
  rpart( Survived ~ . , data = .)
```

```
Accuracy <- function(model){

  cl <- predict(model, test %>% select(-Survived))
  P <- ifelse(cl<.5,0,1)
  acc <- mean(P==test$Survived)
  return(acc)
}
```

Accuracy checking

```r
AccuTable <- tibble(
  Model = c("Logisticc", "Random Forest", "Xgboost", "Decision Tree"),
  Model_Accuracy = c(Accuracy(log_m), Accuracy(rf_m), Accuracy(xg), Accuracy(DT))
)
kbl(AccuTable, format= "latex")
```

Model	Model_Accuracy
Logisticc	0.8352273
Random Forest	0.8238636
Xgboost	0.8636364
Decision Tree	0.8011364

Omitting missing value

```r
############  now fit the logistoic regression  model model
data_im = dataset
data_im <- data_im %>%
  select(PassengerId,Survived,Pclass,Sex,Age,SibSp, Parch,  Fare, Embarked)

data_im <- na.omit(data_im)
t <- createDataPartition(data_im$PassengerId, p =.8, list = F)
trana <- data_im[t,]
testa <- data_im[-t,]
loga_m <- trana %>%
  select(-PassengerId) %>%
  glm(Survived ~ ., family = binomial(link = "logit"), data = .)


########################  Random forest

rfa_m <- trana %>%
  select(-PassengerId) %>%
  train( Survived ~ . , data = ., method  = "rf", trControl = trainControl("cv"))

Accuracy <- function(model){

  cl <- predict(model, testa %>% select(-Survived))
  P <- ifelse(cl<.5,0,1)
  acc <- mean(P==testa$Survived)
  return(acc)
}
Accuracy(rf_m)
############ Xgboost model

xga <- trana %>%
  select(-PassengerId) %>%
  train( Survived ~ . , data = ., method  = "xgbTree", trControl = trainControl("cv"))

Accuracy(xg)

######   Tree model
```

```
DTa <- trana %>%
  select(-PassengerId) %>%
  rpart( Survived ~ . , data = .)
Accuracy(DTa)
```

Accuracy checking

```
AccuTable <- tibble(
  Model = c("Logisticc", "Random Forest", "Xgboost", "Decision Tree"),
  Model_Accuracy = c(Accuracy(loga_m), Accuracy(rfa_m), Accuracy(xga), Accuracy(DTa))
)
AccuTable
```

```
# A tibble: 4 x 2
  Model         Model_Accuracy
  <chr>                  <dbl>
1 Logisticc              0.821
2 Random Forest          0.843
3 Xgboost                0.857
4 Decision Tree          0.829
```