library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.6 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.4 ✓ stringr 1.4.0  
## ✓ readr 2.1.2 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

# Survived is available in train  
testData=read\_csv("test.csv")

## Rows: 418 Columns: 11

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Name, Sex, Ticket, Cabin, Embarked  
## dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

train=read\_csv("train.csv")

## Rows: 891 Columns: 12

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Name, Sex, Ticket, Cabin, Embarked  
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

ex\_sub=read\_csv("gender\_submission.csv")

## Rows: 418 Columns: 2

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (2): PassengerId, Survived

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

table(train$Survived)

##   
## 0 1   
## 549 342

######################### 1 #########################   
### GLM Logistic Regression  
summary(train)

## PassengerId Survived Pclass Name   
## Min. : 1.0 Min. :0.0000 Min. :1.000 Length:891   
## 1st Qu.:223.5 1st Qu.:0.0000 1st Qu.:2.000 Class :character   
## Median :446.0 Median :0.0000 Median :3.000 Mode :character   
## Mean :446.0 Mean :0.3838 Mean :2.309   
## 3rd Qu.:668.5 3rd Qu.:1.0000 3rd Qu.:3.000   
## Max. :891.0 Max. :1.0000 Max. :3.000   
##   
## Sex Age SibSp Parch   
## Length:891 Min. : 0.42 Min. :0.000 Min. :0.0000   
## Class :character 1st Qu.:20.12 1st Qu.:0.000 1st Qu.:0.0000   
## Mode :character Median :28.00 Median :0.000 Median :0.0000   
## Mean :29.70 Mean :0.523 Mean :0.3816   
## 3rd Qu.:38.00 3rd Qu.:1.000 3rd Qu.:0.0000   
## Max. :80.00 Max. :8.000 Max. :6.0000   
## NA's :177   
## Ticket Fare Cabin Embarked   
## Length:891 Min. : 0.00 Length:891 Length:891   
## Class :character 1st Qu.: 7.91 Class :character Class :character   
## Mode :character Median : 14.45 Mode :character Mode :character   
## Mean : 32.20   
## 3rd Qu.: 31.00   
## Max. :512.33   
##

### Survived, Sex, Age, Pclass, Embarked -> categorical, factor can be used?  
train$Sex <- as.factor(train$Sex)  
train$Embarked <- as.factor(train$Embarked)  
  
### handle NA's   
# Check NA's or empty strings, then remove  
colSums(is.na(train) | train == "")

## PassengerId Survived Pclass Name Sex Age   
## 0 0 0 0 0 177   
## SibSp Parch Ticket Fare Cabin Embarked   
## 0 0 0 0 687 2

train <- train %>% drop\_na(Age)  
  
  
# Split 70/30:   
set.seed(31)  
train\_size\_70 <- floor(0.70 \* nrow(train))  
train\_split <- sample(seq\_len(nrow(train)), size = train\_size\_70)  
train\_splitted\_data <- train[train\_split, ]  
  
titanic\_glm <- glm(Survived ~ Sex + Age + Pclass + Embarked + Fare + Parch, data = train\_splitted\_data, family = 'binomial')  
summary(titanic\_glm)

##   
## Call:  
## glm(formula = Survived ~ Sex + Age + Pclass + Embarked + Fare +   
## Parch, family = "binomial", data = train\_splitted\_data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2804 -0.6985 -0.4315 0.6611 2.3402   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.092440 0.717790 7.095 1.3e-12 \*\*\*  
## Sexmale -2.698662 0.265208 -10.176 < 2e-16 \*\*\*  
## Age -0.026418 0.009187 -2.876 0.00403 \*\*   
## Pclass -1.134367 0.198484 -5.715 1.1e-08 \*\*\*  
## EmbarkedQ -1.017384 0.673807 -1.510 0.13107   
## EmbarkedS -0.441302 0.307956 -1.433 0.15186   
## Fare -0.003974 0.003163 -1.257 0.20886   
## Parch -0.185864 0.139592 -1.331 0.18303   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 670.16 on 497 degrees of freedom  
## Residual deviance: 462.73 on 490 degrees of freedom  
## (1 observation deleted due to missingness)  
## AIC: 478.73  
##   
## Number of Fisher Scoring iterations: 4

## Best predictors are Age, Fare and then Parch  
predict\_survived\_7 <- predict(titanic\_glm ,newdata = testData,type = 'response')   
# Above 0.51 will be accepted as 1  
predict\_survived\_7 <- ifelse(predict\_survived\_7 > 0.51, 1, 0)  
  
testData$Survived = predict\_survived\_7  
# testData <- na.omit(testData)   
  
View(testData)  
# Replace NA's  
testData$Survived[is.na(testData$Survived)] <- 0  
  
write.csv(testData[,c("PassengerId","Survived")],  
 "glm\_submission.csv",  
 row.names=F)  
  
######################### 2 #########################   
library(randomForest)

## randomForest 4.7-1

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

##   
## Attaching package: 'randomForest'

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

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

library(xgboost)

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

## Random Forest  
test <- read\_csv("test.csv")

## Rows: 418 Columns: 11

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Name, Sex, Ticket, Cabin, Embarked  
## dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

train <- read\_csv("train.csv")

## Rows: 891 Columns: 12

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Name, Sex, Ticket, Cabin, Embarked  
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# train <- train %>% drop\_na()  
train <- train %>% drop\_na(Sex, Age, Pclass, Embarked, Fare, Parch)  
  
  
rfmodel <- randomForest(train[,c("Sex" , "Age" , "Pclass" , "Embarked" , "Fare" , "Parch")],  
 train$Survived,  
 n.trees = 1000)

## Warning in randomForest.default(train[, c("Sex", "Age", "Pclass", "Embarked", :  
## The response has five or fewer unique values. Are you sure you want to do  
## regression?

importance(rfmodel)

## IncNodePurity  
## Sex 42.438009  
## Age 24.051401  
## Pclass 16.030248  
## Embarked 3.856641  
## Fare 25.976638  
## Parch 4.938443

titanic\_shuffle = train[sample(nrow(train),nrow(train),F),]  
titanic\_train=train[1:500,]  
titanic\_test=train[1:418,]  
  
# train on training daa  
rfmodel <- randomForest(titanic\_train[,c("Sex" , "Age" , "Pclass" , "Embarked" , "Fare" , "Parch")],  
 titanic\_train$Survived,  
 n.trees=10000,  
 nodesize=20)

## Warning in randomForest.default(titanic\_train[, c("Sex", "Age", "Pclass", :  
## The response has five or fewer unique values. Are you sure you want to do  
## regression?

predict\_rf <- predict(rfmodel, titanic\_test[,c("Sex" , "Age" , "Pclass" , "Embarked" , "Fare" , "Parch")])  
predict\_rf <- ifelse(predict\_rf > 0.51, 1, 0)  
  
test$Survived = predict\_rf  
  
write.csv(test[,c("PassengerId","Survived")],  
 "glm\_submission\_rf.csv",  
 row.names=F)  
## GBM  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

test <- read\_csv("test.csv")

## Rows: 418 Columns: 11

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Name, Sex, Ticket, Cabin, Embarked  
## dbl (6): PassengerId, Pclass, Age, SibSp, Parch, Fare

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

train <- read\_csv("train.csv")

## Rows: 891 Columns: 12

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Name, Sex, Ticket, Cabin, Embarked  
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare

##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

train\_x = data.matrix(train[, -12])  
train\_y = train$Embarked  
train\_y <- na.omit(train\_y)  
train\_x = na.omit(train\_x)  
View(train\_y)  
  
test\_x = data.matrix(test[, -11])  
test\_y = test[, 11]  
  
# xgb\_train = xgb.DMatrix(data = train\_x, label = train\_y)  
# xgb\_test = xgb.DMatrix(data = test\_x, label = test\_y)  
# length(train\_y)  
  
# gmb\_model <- xgboost(data = train, label = train$Survived, nrounds = 2, objective = "binary:logistic")  
  
  
  
  
  
### Accuracy : GLM > RF