3주차 과제 예시 답안

```
# Week_3 Quiz

# Description
# 문제 바로 아래에 답안을 작성하여, <<r script 형태>>로 저장 후 제출해주시면 됩니다.


library(dplyr)
library(caret)

## 3주차 과제는 kc_house와 titanic_train 데이터를 활용하여 과제를 진행해주세요.
## kc_house : 집과 관련된 정보를 바탕으로 가격을 예측하는 데이터 입니다. (수치형 데이터 예측)
## titanic_train : 타이타닉 호에 탑승한 승객 정보를 바탕으로 생존여부를 예측하는 데이터입니다. (범주형 데이터 예측)


## 데이터 불러오기
kc_house <- read.csv(file = "변경필요", header= TRUE)
titanic <-  read.csv(file = "변경필요", header= TRUE)

[외부 데이터 불러오기]
Script를 통해 가져오는 방법과, GUI를 통해 가져오는 방법 2가지가 존재합니다.

1) 파일의 위치를 변수로 정의
kc_house_path = "../Desktop/archive/kc_house_data.csv"
kc_house <- read.csv(file = kc_house_path, header= TRUE)

## 경로를 kc_house_path라는 변수로 지정하여, read.csv함수에서는 변수를 호출하고 있습니다.
##kc_house_path에서 ..은 이전 폴더, .은 현재 폴더를 의미하니 참고해주세요.

2) read.csv함수에 전체 경로 입력
titanic <- read.csv(file = '../Desktop/titanic/titanic_train.csv', header= TRUE)

3) GUI = Script가 어려우신 분들은 R Studio 에서 지원하는 GUI 기능을 활용해보시기 바랍니다.
  1. [Environment] -> [Import Dataset] -> [From Excel] 클릭
  2. Import Excel Data 창에서 원하는 엑셀불러오기
  3. Import 옵션 설정


## 1. kc_house 데이터를 활용하여 다음 조건을 충족하는 훈련, 테스트 데이터를 생성해주세요. (10점)
## 조건1) Random seed number : 2021
## 조건2) 8:2의 비율의 Train과 Test 데이터셋 구분

df <- kc_house
set.seed(2021)

train_index <- sample(1:nrow(df), size=0.8*nrow(df))
test_index <- (-train_index)

df_train <- df[train_index, ]
df_test <- df[test_index, ]

[문제 해설]
sort는 결과값을 쉽게 확인하기 위함이며, 필수적인 부분은 아닙니다.



## 2. price를 표준화(standardization)한 데이터를 kc_house에 추가해주세요. (변수명 무관) (10점)
## 힌트 : mean(평균), sd(표준편차) 함수 활용

kc_house = kc_house %>%
  mutate(price_st_1 = (price - mean(price)) / sd(price))

kc_house = kchouse %>% mutate(price_st_2 = scale(price))


## 3. sqft_living를 min-max 스케일링한 데이터를 kc_house에 추가해주세요. (변수명 무관) (10점)
## 힌트
## 1) sqft_living에 대한 min과 max값을 변수로 생성
## 2) 신규 생성 된 min, max 변수와 mutate 함수를 사용하여 신규 컬럼을 생성

min_sqft_living <- min(df_train$sqft_living)
max_sqft_living <- max(df_train$sqft_living)

df_train <- df_train %>%
  mutate( minmax_sqft_living = (sqft_living - min_sqft_living) / (max_sqft_living - min_sqft_living))
```

```
## 4~7번 : titanic_train 데이터를 활용하여 문제를 풀어주세요.
## titanic은 train데이터이기에 추가적으로 train / test를 분리하지 않으셔도 됩니다.


## 4. Survived 변수를 범주형 데이터로 변경해주세요. (10점)
titanic$Survived <- as.factor(titanic$Survived)




## 5. 다음 조건을 바탕으로 생존여부(Survived)를 측정하는 Rogistic Regression 모델을 생성해주세요. (10점)
### 1) Cross Validation : 5-fold
### 2) Cross Validation Repeat : 5
### 3) Boosting 기법이 적용된 Regression Method 사용

## 힌트
## 1) trainControl Help 문서
## 2) 정상 실행 되거나, "missing value 오류"가 출력되는 2가지 케이스 모두 정답입니다.
##    (missing value 오류 여부는 Console에 출력되는 Error 메세지를 통해 확인 가능)

ctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 5)
logitFit <- train(Survived ~ .,
                  data = df,
                  method = "LogitBoost",
                  na.action = na.omit,
                  trControl = ctrl,
                  metric = "Accuracy" )

[문제해설]
5-kold로 정의된 cross validation을 5번 반복하는 모델을 생성하는 문제입니다.

?trainControl 를 통해 help문서를 보시면,

number = ifelse(grepl("cv", method), 10, 25),
-> method가 "cv" 일 경우에는 10, 그 외 경우는 25를 기본값으로 가집니다.
-> 저희는 "repeatedcv"가 적절한 정답이며, number를 선언하지 않을 경우 25-fold가 수행된다고 보시면 됩니다.

number : Either the number of folds or number of resampling iterations
repeats : For repeated k-fold cross-validation only: the number of complete sets of folds to compute
-> repeats는 "repeatedcv"에서만 사용되는 파라미터라는 점도 체크하시기 바랍니다.

관련 강의
1) k-fold에 대한 소개 : Ch 01. 지도학습 개요 - 05. 지도학습에 필요한 개념 - 교차검증 (06:45 ~ )
2) trainControl 함수의 "repeatedcv"에 파라미터 소개 : Ch 02. k-Nearest Neighbor - 02. R code로 구현하는 k-Nearest Neighbor (04:11 ~ )



## 6. 데이터 전처리
## 5번 문제에서 null인 데이터가 존재할 경우 모델링이 정상적으로 수행하지 않음을 확인하였습니다.
## 전체 컬럼 중 null인 데이터가 1개라도 있으면 모두 삭제하여 5번 모델링을 재실행해주세요. (10점)

## 힌트
## 1) View 함수를 통해 null로 표시된 데이터는 어떤식으로 보이는지 확인해보시기 바랍니다.
## 2) 1안 : 참고 링크(https://m.blog.naver.com/PostView.nhn?blogId=liberty264&logNo=220992831831&proxyReferer=https:%2F%2Fwww.google.com%
## 3) 2안 : train()함수에서 지원하는 파라미터 사용하기하여 5번 모델링을 재실행해주세요.

df_notnull <- na.omit(df)
ctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 5)
logitFit <- train(Survived ~ .,
                  data = df_notnull,
                  method = "LogitBoost",
                  na.action = na.omit,
                  trControl = ctrl,
                  metric = "Accuracy" )

[null에 대한 에러]
- 1안 : na.omit 함수를 사용하여 null인 모든 데이터를 삭제하는 것입니다.
- 2안 : train() 함수에서는 na.action 파라미터를 통해 null인 데이터를 어떻게 할 것인지 정의할 수 있습니다.

[null을 처리하는 방법]
null 데이터는 머신러닝 모델링을 수행함에 있어 상당히 중요한 역할을 수행합니다.
불필요한 데이터라 판단될 경우 과감히 날리는 것이 가장 간단한 방법이지만, 필요한 경우라면 null인 값을 채워줄 필요가 있습니다.
1) 평균이나 중위수로 채우기
2) null이 많은 변수와 연계된 변수를 활용한 회귀분석 모델을 만들어서, 예측값을 채우기
3) 999라는 임의의 큰 수를 입력하기
등 모델 특성에 맞게 다양한 방법이 존재하니, 이러한 내용은 구글링을 통해 한번 공부해보시는 것을 권장드립니다.

[null 데이터 확인하기]
- is.na(data) : null이 포함되어있을 경우 TRUE, 아닐경우 FALSE를 반환해줍니다.
- colSums(is.na(titanic)) : 어떤 컬럼에서 null이 존재하는지 확인할 수 있습니다.
- table(is.na(titanic)) : 전체 데이터에서 null이 존재하는지 확인할 수 있습니다.


## (10점) 7. 6번에서 생성된 모델에 대한 해석을 작성해주세요.

## 해석 : 11 개의 Feature를 사용하여 생존 여부를 예측하였음
```

```
## 해석 : 11 번의 반복이 가장 높은 정확도를 보이고 있음

모델 결과 해석에 대한 내용은 아래 강의에서 다루고 있습니다.
- Ch 03. Logistic Regression - 04. Logistic Regression 예제 실습 (05:40 ~ )
```