모델 적합

다중선형회귀 모델

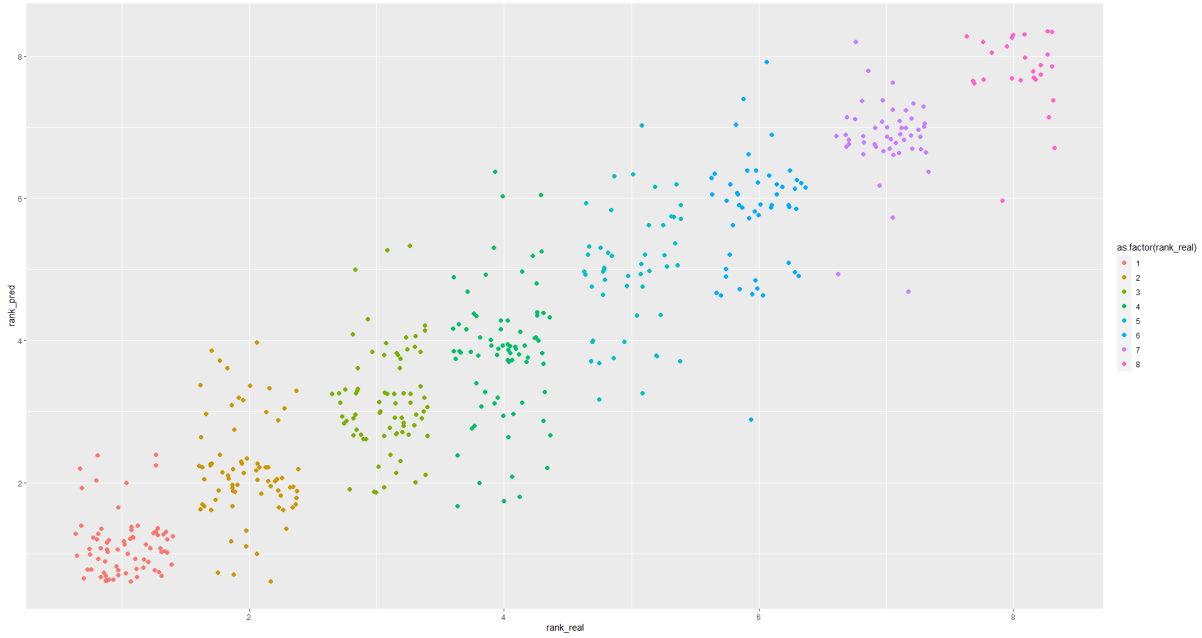
```
#입력변수 설정
#최종 입력변수
#확정 : 매출_월화수목,매출_금토일,매출_0614,매출_1421,대분류
#후보 : 매출_2106, 중분류,년도, 분기,행정구역,년분기
colnames(trainset)
vars_full <-     c(6,7,8,9,10,4,11,5,1,2,3,24) # 확정 + 후보 입력 변수
vars_selected <- c(6,7,8,9,10,4) # 확정 입력 변수
trainset_full <- trainset[,vars_full]
trainset_selected <- trainset[,vars_selected]

#다중선형회귀분석 모델
#Multiple Linear Regression Model

#모델 적합
fit_full <- lm(formula = 매출총액~.,data = trainset_full) #목표변수와 상관관계 있는 입력변수와 보류인 입력변수
trainset_full_new <- trainset_full %>% select(-c(년분기,중분류))

fit_full_new <- lm(formula = 매출총액~.,data = trainset_full_new)
fit_selected <- lm(formula = 매출총액~.,data = trainset_selected) #목표변수와 상관관계가 있는 입력변수
null <- lm(formula = 매출총액~1.,data = trainset_full)
full <- lm(formula = 매출총액~.,data = trainset_full)
fit_stepwise <- step(object = null,
             scope = list(lower = null, upper = full),
             direction = "both") #stepwise를 통한 단계적 변수 선택

summary(fit_full)
#년분기는 NA 생성, 중분류 중 종합소매 및 주점업 NA 생성
#매출_1421, 행정구 25구 중 8구, 분기2,3은 p-value = 0.05이상
#모델에서 제외

summary(fit_full_new) # 회귀계수가 NA인 변수 제거(년분기, 중분류)
summary(fit_selected) # 확정 입력변수로 모델 적합
summary(fit_stepwise) # 변수소거법으로 모델 적합

#다중공산성 문제 확인
library(car)
vif(mod = fit_full_new) # GVIF^(1/(2*Df)) > 2 => 매출_월화수목/금토일/0614/1421
vif(mod = fit_selected) # GVIF^(1/(2*Df)) > 2 => 매출_월화수목/금토일/0614/1421
vif(mod = fit_stepwise)# GVIF^(1/(2*Df)) > 2 => 매출_월화수목/금토일/0614/1421

#다중공산성를 발생시키는 인자 중 가장 영향이 큰 매출_월화수목 제외
trainset_full_new <- trainset_full_new %>% select(-c(매출_1421,매출_월화수목))
trainset_selected <-  trainset_selected %>% select(-c(매출_1421,매출_월화수목))
trainset_full <- trainset_full %>% select(-c(매출_1421,매출_0614,매출_금토일))

fit_full_new_af <- lm(formula = 매출총액~., data = trainset_full_new)
fit_selected_af <- lm(formula = 매출총액~., data = trainset_selected)
null <- lm(formula = 매출총액~1, data = trainset_full)
full <- lm(formula = 매출총액~., data = trainset_full)
fit_stepwise_af <- step(object = null,scope = list(lower = null, upper = full),direction = "both")

#결과재확인
summary(object = fit_full_new_af)
summary(object = fit_selected_af)
summary(object = fit_stepwise_af)

#다중공산성 문제 확인
vif(mod = fit_full_new_af)
vif(mod = fit_selected_af)
```

```
vif(mod = fit_stepwise_af)

#변수 소거 전후 모델 비교평가
anova(fit_full_new, fit_full_new_af) # p-value 0.05 이하로 변수소거 전후 성능 차이 있음을 확인
anova(fit_selected, fit_selected_af) # p-value 0.05 이하로 변수소거 전후 성능 차이 있음을 확인
anova(fit_stepwise, fit_stepwise_af) # p-value 0.05 이하로 변수소거 전후 성능 차이 있음을 확인

#잔차가정 검정 bonferroni p 0.05이하 제거
outliers <- function(model, dataset,stepwise){
  repeat{
    outliers <- outlierTest(model = model)
    outliers <- as.integer(names(outliers$bonf.p[outliers$bonf.p<0.05]))
    if(length(outliers)==0) break
    dataset <- dataset %>% slice(-outliers)
    if(stepwise == 0){
      model <- lm(formula = 매출총액~., data = dataset)
    } else {
      null <- lm(formula = 매출총액~1, data = dataset)
      full <- lm(formula = 매출총액~., data = dataset)
      model <- step(object = null,scope = list(lower = null, upper = full),direction = "both")
    }
  }
  return(model)
}
fit_full_new_af <- outliers(model = fit_full_new_af,dataset = trainset_full_new, stepwise = 0)
fit_selected_af <- outliers(model = fit_selected_af,dataset = trainset_selected, stepwise = 0)
fit_stepwise_af <- outliers(model = fit_stepwise_af,dataset = trainset_full, stepwise = 1)

#잔차 패턴 확인
windows()
par(mfrow = c(2,2))
plot(x = fit_stepwise_af)
par(mfrow = c(1,1))

#잔차가정 검정
library(car)
ncvTest(model = fit_stepwise_af)
durbinWatsonTest(model = fit_stepwise_af)
crPlots(model = fit_selected_af)
influencePlot(model = fit_stepwise_af)
# fit_full_new_af - 애매하거나 확실한 입력변수 기준
# fit_selected_af - 확실한 입력변수 기준
# fit_stepwise_af - 전체 데이터(확정변수+비교용변수)에서 변수소거법 적용

# 다중공산성 문제를 일으키는 인자 제거
# 이상치 제거 => bonfferoni p 0.05 이하인 index 제거

# 최종 확인결과, 세 경우 모두 잔차의 목표변수가 정규성을 위배하여 이후 순서 진행불가

#######
library(olsrr)
olsrr::ols_plot_cooksd_bar(model = fit_t) #cook 거리 바플랏
                                          #해당 관측값이 전체 최소제곱추정량에 미치는 영향력을 보여주는 지표
ols_plot_dfbetas(model = fit_t) #해당 관측치의 개별 베타 값에 대한 영향력 지표
ols_plot_dffits(model = fit_t) #베타값의 분상 공분상 행렬의 Cov(b^) 추정값에 대한 해당 관측치에 대한 영향력

#5000개 이상 데이터 정규성 확인 => 앤더슨 달링 테스트
library(nortest)
ad.test(fit_t2$residuals)
```

회귀나무모델

```
#회귀나무로 모델 만들기
rm(list = ls())
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
load("dataset_set.rda")
function_path = "C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/코드/"
source(file = paste0(function_path,"function.r"))
library(tidyverse)
library(rpart)
library(rpart.plot)

#최종 입력변수
#확정 : 매출_월화수목,매출_금토일,매출_0614,매출_1421,매출_2106,대분류
#비교 : 중분류,년도, 분기,행정구역,년분기
vars_full <- c(6,7,8,9,10,11,4,5,1,2,3,24)
vars_selected <- c(6,7,8,9,10,11,5)

#회귀나무 정지요인 설정
ctrl <- rpart.control(minsplit = 10L,
                      cp = 0.001,
                      maxdepth = 30L)

#회귀나무모델 적합
set.seed(seed = 1234)
fit1 <- rpart(formula = 매출총액~.,
              data = trainset[,vars_selected],
              control = ctrl)

#결과 확인 및 가지치기 여부 확인
summary(object = fit1)
plotcp(x = fit1)

#가지치기 불필요 확인
which.min(fit1$cptable[,4])

#성능 분석
real <- testset$매출총액
pred1 <- predict(object = fit1, newdata = testset, type = "vector")

testset$매출총액_pred <- pred1
result <- testset %>% select(행정구역,대분류,중분류,매출총액,매출총액_pred) %>%
  group_by(행정구역,대분류) %>%
  mutate(rank_real = row_number(desc(매출총액)),
         rank_pred = row_number(desc(매출총액_pred)))

result %>%
  ggplot(aes(x = rank_real, y = rank_pred))+
  geom_point(position = position_jitter())
```

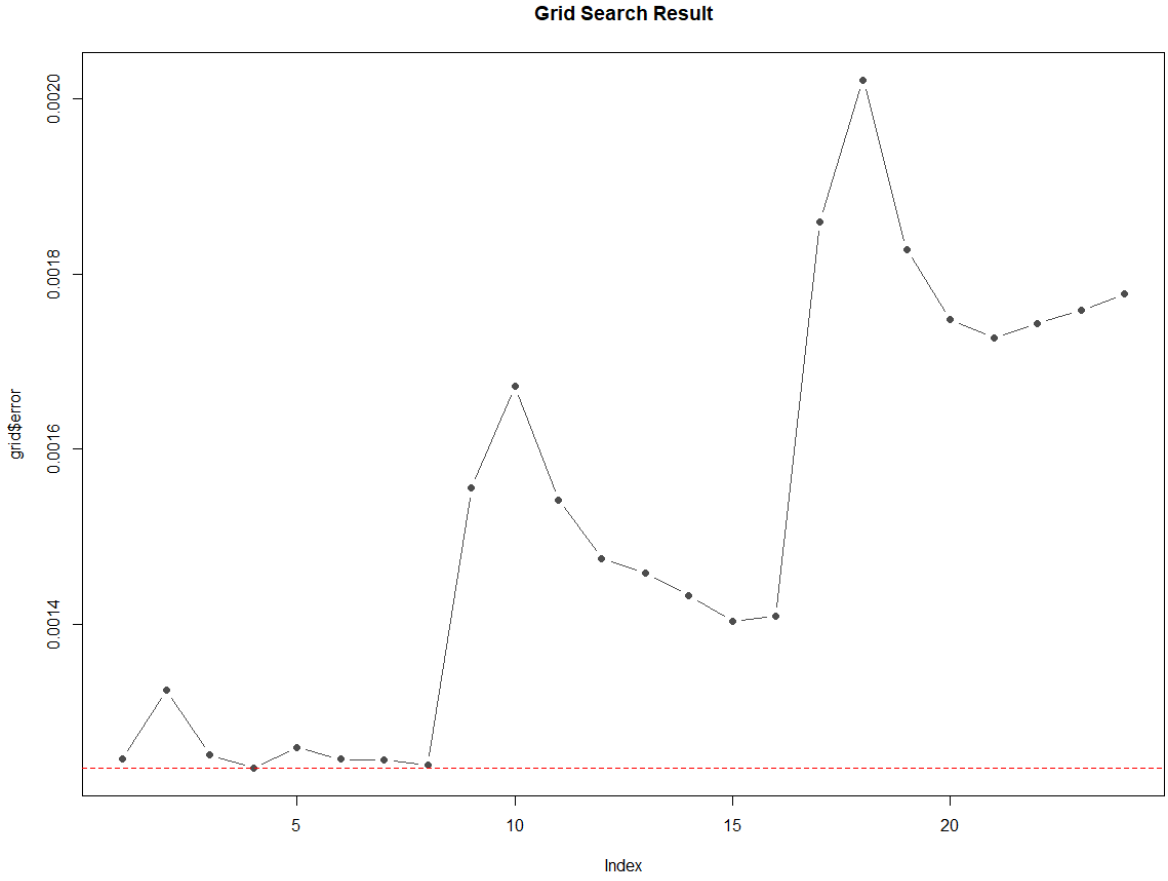
```
     MSE    RMSE    MAE  MAPE
0.0193 0.1391 0.103 0.006
```



랜덤포레스트 모델

```
rm(list = ls())
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
load("dataset_set.rda")
list.files()
function_path = "C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/코드/"
source(file = paste0(function_path,"function.r"))
library(tidyverse)
# library(tidymodels)
library(randomForest)
library(tictoc)

#입력변수 설정
#최종 입력변수
#확정 : 매출_월화수목,매출_금토일,매출_0614,매출_1421,대분류
#비교 : 매출_2106, 중분류,년도, 분기,행정구역,년분기
colnames(trainset)
vars_full <-     c(6,7,8,9,10,4,11,5,1,2,3,24)
vars_selected <- c(6,7,8,9,10,4)
trainset_full <- trainset[,vars_full]
trainset_selected <- trainset[,vars_selected]

#반복문을 사용한 모형 튜닝
grid <- expand.grid(ntree = seq(from = 100, to = 1500, by = 200),
                    mtry = 2:4,
                    error = NA)
n = nrow(x = grid)
for(i in 1:n){
  tic()
  start_time <- Sys.time()
  ntree = grid$ntree[i]
  mtry = grid$mtry[i]
  disp <- str_glue('현재 {i}행 실행 중! [ntree: {ntree}, mtry: {mtry}')
  cat(disp,"\n")
  set.seed(seed = 1234)
  fit <- randomForest(formula = 매출총액~.,
```

```
                      data = trainset[,vars_selected],
                      ntree = ntree,
                      mtry = mtry)
  grid$error[i] <- tail(x = fit$mse, n = 1)
  time <- round(Sys.time() - start_time,digits = 2L)
  attr
  disp <- str_glue('현재 {i}행 완료! [ntree: {ntree}, mtry: {mtry}')
  cat(disp,"\n")
  toc()
}

#튜닝 결과 확인
plot(x = grid$error, type = 'b', pch = 19, col = 'gray30', main = 'Grid Search Result')
abline(h = min(x = grid$error), col = 'red', lty = 2)
loc <- which.min(x = grid$error)
print(x = loc)

#OOB 오차가 최소인 하이퍼파라미터 지정
bestPara <- grid[loc,] #ntree 700, mtry 2, mse 0.0011497

#최적 하이퍼파라미터로 모형 적합
set.seed(seed = 1234)
fit1 <- randomForest(formula = 매출총액~.,
                     data = trainset[,vars_selected],
                     ntree = bestPara$ntree,
                     mtry = bestPara$mtry,
                     importance = T,
                     do.trace = T,
                     keep.forest = T)

#결과 확인
plot(x = fit1, main = "best Fit")
importance(x = fit1)
varImpPlot(x = fit1, main = 'variable importance', type = 1 )

#시험셋으로 목표변수 추정값 생성
pred1 <- predict(object = fit1, newdata = testset[,vars_selected], type = 'response')

#실제 관측치 벡터 생성
real <- testset$매출총액

#모형 성능 확인
regMeasure(real = real, pred = pred1)
plot(x = real, y = pred1, type = "p")

#trainset에 rank_real과 rank_pred 생성
testset$매출총액_pred <- pred1
result <- testset %>% select(행정구역,대분류,중분류,매출총액,매출총액_pred) %>%
  group_by(행정구역,대분류) %>%
  mutate(rank_real = row_number(desc(매출총액)),
         rank_pred = row_number(desc(매출총액_pred))) %>%
  arrange(행정구역,rank_real,rank_pred)
result %>%
  ggplot(aes(x = rank_real, y = rank_pred))+
  geom_point(position = position_jitter())

table(result$rank_real==result$rank_pred)
saveRDS(object = fit1, file = "randomforest.rds")
```



ntree = 700, mtry = 2

MSE	RMSE	MAE	MAPE
0.0006	0.024	0.0112	0.0007