모델 적합

선형회귀

```
library(tidyverse)
#파일 불러오기
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
load("dataset_set.rda")
# load("dataset_set_outlier.rda")

#라벨인코딩
function_path = "C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/코드/"
source(file = paste0(function_path,"function.r"))
# vars <- c(3,4,5)
# for(i in vars){
#   trainset <- label_encoding(dataframe = trainset,column_num = i)
#   testset <- label_encoding(dataframe = testset,column_num = i)
# }

#다중선형회귀분석 모델
#Multiple Linear Regression Model
#모델 적합
fit1 <- lm(formula = 매출총액~.,data = trainset)
summary(fit1)

vars <- c("년분기","중분류")
loc <- which(colnames(trainset) %in% vars)

#P-value가 NA 컬럼 제거 후 재적합
fit1_2 <- lm(formula = 매출총액~., data = trainset[,-loc])
summary(fit1_2)

#변수소거법을 통한 모델 적합
null <- lm(formula = 매출총액~1.,data = trainset)
full <- lm(formula = 매출총액~.,data = trainset)
fit2 <- step(object = null,
             scope = list(lower = null, upper = full),
             direction = "both") #stepwise를 통한 단계적 변수 선택
summary(fit2)


#다중공산성 인자 확인
library(car)
vif(mod = fit1_2) # GVIF^(1/(2*Df)) > 2 => 매출_월화수목/금토일/0614/1421
vif(mod = fit2) # GVIF^(1/(2*Df)) > 2 => 매출_월화수목/금토일/0614/1421

#다중공산성 인자 제거
vif_test <- function(dataset,stepwise_uages){
  library(car)
  name_list <- c()
  name_list_backup <- c()
  repeat{
    if(stepwise_uages == 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")
    }
    if(length(vif(mod = model))>4){
      vif_list <- vif(mod = model)[,3]
    } else {
      vif_list = vif(mod = model)
    }

    name <- names(which.max(vif_list[vif_list>2]))
    name_list <- c(name_list,name)
    if(length(name_list) == length(name_list_backup)) break
    name_list_backup <- name_list
    dataset <- dataset %>% select(-c(name))
  }
  data_list = list(model,name_list)
  return(data_list)
}
fit1_3_list <- vif_test(dataset = trainset[,-loc],stepwise_uages = 0)
fit2_2_list <- vif_test(dataset = trainset,stepwise_uages = 1)

#다중공산성 인자 확인
fit1_3_list[2]
fit2_2_list[2] #매출비율_0614, 총매출건수, 매출비율_1724 제거
```

```r
#다중공산성 컬럼을 제거한 후 적합한 모델
fit1_3 <- fit1_3_list[[1]]
fit2_2 <- fit2_2_list[[1]]

#결과재확인
summary(object = fit1_3)
summary(object = fit2_2)

#다중공산성 문제 확인
vif(mod = fit1_3)
vif(mod = fit2_2)

#변수 소거 전후 모델 비교평가
anova(fit1_2, fit1_3) # 소거된 변수 없음
anova(fit2, fit2_2) # 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)
}
fit1_3 <- outliers(model = fit1_3,dataset = trainset[,-loc], stepwise = 0)
fit2_2 <- outliers(model = fit2_2,dataset = trainset, stepwise = 0)


#잔차 패턴 확인
windows()
par(mfrow = c(2,2))
plot(x = fit1_3)
plot(x = fit2_2)
par(mfrow = c(1,1))

#잔차가정 검정
library(car)
ncvTest(model = fit1_3)
durbinWatsonTest(model = fit1_3)
crPlots(model = fit1_3)
influencePlot(model = fit1_3)

ncvTest(model = fit2_2)
durbinWatsonTest(model = fit2_2)
crPlots(model = fit2_2)
influencePlot(model = fit2_2)
```

모델 적합

- fit1 : 회귀계수가 NA로 출력되는 변수(년분기, 중분류) 제거 후 모델 적합
- fit2 : 변수소거법을 통해 변수(년도,분기,대분류) 제거 후 모델 적합

다중공산성 변수 제거

- fit1 : X
- fit2 : 매출비율_0614, 총매출건수, 매출비율_1724

#잔차 확인

Aa 이름	생성일	태그
fit1	@2021년 1월 25일 오전 12:14	
fit2	@2021년 1월 25일 오전 12:14	

목표변수가 정규성을 위배하고, 잔차도 정규성을 위배하며, 잔차의 특정 패턴이 보이므로 선형회귀모델은 진행 불가

회귀나무

- 다회 튜닝 진행

```
rm(list = ls())
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
load("dataset_set.rda")

#함수 불러오기
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)
library(MLmetrics)

#입력변수 설정
vars <- colnames(trainset) #1차
vars <- c("매출총액","중분류","총매출건수","대분류","행정구역","매출비율_0614")#2차


#회귀나무 모델 튜닝을 위한 정지요인 설정
grid <- expand.grid(
  minsplit = seq(from = 2, to = 20, by = 1),
  cp = seq(from = 0.0001, to = 0.001, length.out = 10),
  seed = 1234,
  RMSE = NA,
  F1 = NA,
  R2 = NA) #1차 튜닝 = 모든 변수 / 2차 튜닝 = 선택 변수

grid <- expand.grid(
  minsplit = c(5,20),
  cp = 0.0001,
  seed = sample(x = 1:9999,size = 1000, replace = F),
  RMSE = NA,
  F1 = NA,
  R2 = NA) #3차,  minsplit 5,20 선택, cp 0.0001, setseed 1000개 설정
           #모든 변수

grid <- expand.grid(
  minsplit = 5:6,
  cp = 0.0001,
  seed = sample(x = 1:9999,size = 1000, replace = F),
  RMSE = NA,
  F1 = NA,
  R2 = NA) #3차,  minsplit 5,6 선택, cp 0.0001, setseed 1000개 설정
           #선택 변수
           #결과 차이 없음


#모델 튜닝 진행
for(i in 1:nrow(grid)){
  sentence <- str_glue('{i}번째 행 실행 중 {round((i-1)*100/nrow(grid),2)}% 완료 [minsplit : {grid$minsplit[i]}, cp = {grid$cp[i]}')
  print(sentence,"\n")
  #정지규칙 설정
  ctrl <- rpart.control(minsplit = grid$minsplit[i],
                        cp = grid$cp[i],
                        maxdepth = 30L)

  #모델적합
  set.seed(seed = grid$seed[i])
    fit <- rpart(formula = 매출총액~.,
                 data = trainset[,vars],
                 control = ctrl)
    #가지치기 여부 확인 후 적합
    num1 <- nrow(fit$cptable)
    num2 <- which.min(fit$cptable[,4])
    if(num1 != num2){
      fit2 <- prune.rpart(tree = fit, cp = grid$cp[i])
    } else {
      fit2 = fit
    }
    #성능 분석
    real <- testset$매출총액
    pred1 <- predict(object = fit, newdata = testset, type = "vector")
    pred2 <- predict(object = fit2, newdata = testset, type = "vector")

    #RMSE 계산
    reg1 <- MLmetrics::RMSE(y_pred = pred1, y_true = real)
```

```
    reg2 <- MLmetrics::RMSE(y_pred = pred2, y_true = real)

    R21 <- MLmetrics::R2_Score(y_pred = pred1, y_true = real)
    R22 <- MLmetrics::R2_Score(y_pred = pred1, y_true = real)

    #실측값 및 예측값 Rank
    testset$매출총액_pred1 <- pred1
    testset$매출총액_pred2 <- pred2
    result <- testset %>% select(행정구역,대분류,중분류,매출총액,매출총액_pred1,매출총액_pred2) %>%
      group_by(행정구역,대분류) %>%
      mutate(rank_real = row_number(desc(매출총액)),
             rank_pred1 = row_number(desc(매출총액_pred1)),
             rank_pred2 = row_number(desc(매출총액_pred2)),
             top_real = ifelse(rank_real <=3,"1","0"),
             top_pred1 = ifelse(rank_pred1 <=3,"1","0"),
             top_pred2 = ifelse(rank_pred2 <=3,"1","0"))

    #Top3 예측 성능
    F1_1 <- F1_Score(y_true = result$top_real, y_pred = result$top_pred1, positive = "1")
    F1_2 <- F1_Score(y_true = result$top_real, y_pred = result$top_pred2, positive = "1")

    #grid라는 dataframe에 RMSE 및 F1_Score, R2_score 저장
    grid$RMSE[i] <-ifelse(reg1>=reg2,reg1,reg2)
    grid$F1[i] <- ifelse(F1_1>=F1_2,F1_1,F1_2)
    grid$R2[i] <- ifelse(R21>=R22,R21,R22)
}

# R2, F1, CP 선 그래프 그리기
windows()
text <- data.frame(x = rep(nrow(grid)+10,3),
                   y = as.numeric(grid[nrow(grid),4:6]),
                   label = colnames(grid)[4:6])
grid %>% mutate(order = row_number()) %>%
  ggplot(aes(x = order, y = RMSE))+geom_line(col = "blue")+geom_point(col = "blue")+ylab("")+
  geom_line(aes(y = F1), col = "red")+geom_point(aes(y = F1),col = "red")+
  geom_line(aes(y = R2), col = "black")+geom_point(aes(y = R2), col = "black")+
  scale_y_continuous(sec.axis = dup_axis(), breaks = seq(0,1,0.05))+
  geom_text(data = text, mapping = aes(x = text$x, y = text$y, label = text$label),col = c("blue","red","black"), size = 10)
#RMSE가 가장 낮은 경우, F1이 가장 높은 경우, R2가 가장 높은 경우 세 가지를 선택하고, random set.seed로 가장 성능 좋은 모형 찾기

RMSE <- which.min(grid$RMSE)
F1 <- which.max(grid$F1)
R2 <- which.max(grid$R2)
cat(RMSE,F1,R2)
#1차 => minsplit 5 or 20 / cp = 0.0001 random set.seed 3차 fitting
#2차 => minsplit 5 or 6 / cp = 0.0001 random set.seed 4차 fitting

grid4 <- grid
#튜닝값 및 모델 저장
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
save(grid4,file = "RegressionTree4.rda")
```

RMSE, F2, R2

Aa 이름	생성일	태그
17.regtree1	@2021년 1월 25일 오전 12:38	
17.regtree2	@2021년 1월 25일 오전 12:38	
17.regtree3	@2021년 1월 25일 오전 12:38	
17.regtree4	@2021년 1월 25일 오전 12:38	

	1차		2차	3차	4차
MinSplit	5,6	20	5	1차 튜닝 결과와 동일	1차 튜닝 결과와 동일
CP	0.0001	0.0001	0.0001		
RMSE	0.389	0.424	0.394		
F1 Score	0.827	0.858	0.840		
R2 Score	0.856	0.830	0.853		

랜덤포레스트

```
library(tidyverse)
library(randomForest)
setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/데이터")
load("dataset_set.rda")

#입력변수를 변경하며 모델 생성을 위해 데이터셋 더미 만들어놓기기
trainset_dummy <- trainset
testset_dummy <- testset


#반복문을 사용한 모형 튜닝
grid <- expand.grid(ntree = seq(from = 300, to = 500, by = 100),
                    mtry = 3:9,
                    error = NA)
grid_tot <- data.frame(grid, tuning = 1)
grid <- expand.grid(ntree = seq(from = 350, to = 550, by = 50), #2차 튜닝 : 고정 mtry, ntree 범위 지정 to 600까지
                    mtry = 7,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 2))
grid <- expand.grid(ntree = seq(from = 600, to = 900, by = 100), #3차 튜닝 : 고정 mtry, ntree 범위 지정
                    mtry = 7,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 3))
grid <- expand.grid(ntree = seq(from = 300, to = 600, by = 100), #4차 튜닝 : 변수중요도가 높은 변수를 추린 후 재튜닝
                    mtry = 3:6,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 4))
grid <- expand.grid(ntree = seq(from = 800, to = 1000, by = 100),#5차 튜닝 : 고정 mtry, ntree 범위 지정 from = 700
                    mtry = 6,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 5))
grid <- expand.grid(ntree = seq(from = 300, to = 1000, by = 100), #6차 튜닝 : 변수중요도가 높은 상위 3개를 선택 후 재 튜닝
                    mtry = 2:3,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 6))
grid <- expand.grid(ntree = seq(from = 1100, to = 1400, by = 100), #7차 튜닝 :고정 mtry, ntree 범위 지정 from = 1000 to 1500
                    mtry = 3,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 7))
grid <- expand.grid(ntree = seq(from = 1500, to = 2000, by = 100), #8차 튜닝 :고정 mtry, ntree 범위 지정
                    mtry = 3,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 8))
grid <- expand.grid(ntree = seq(from = 2000, to = 2500, by = 100), #9차 튜닝 :고정 mtry, ntree 범위 지정
                    mtry = 3,
                    error = NA)
grid_tot <- rbind(grid_tot,data.frame(grid, tuning = 9))
grid <- grid_tot

grid$RMSE <- NA
grid$F1 <- NA
grid$R2 <- NA
#입력변수 지정
vars1 <- colnames(trainset) #1~3차 튜닝
vars2 <- c("행정구역","중분류","총매출건수","소득분위","매출비율_1724","년도","매출총액") #4~5차 튜닝(변수 소거)
vars3 <- c("행정구역","중분류","총매출건수","매출총액") #6~9차 튜닝(변수 소거)

n <-  nrow(x = grid)
pred_list = c()

for(i in 52:n){
  disp <- str_glue('현재 {i}행 실행 중! [ntree: {grid$ntree[i]}, mtry: {grid$mtry[i]}]  {Sys.time()}')
  cat(disp,"\n")
  if(grid$tuning[i] %in% 1:3){
    vars_loc <- which(colnames(trainset_dummy) %in% vars1)
  } else if(grid$tuning[i] %in% 4:5){
    vars_loc <- which(colnames(trainset_dummy) %in% vars2)
  } else {
    vars_loc <- which(colnames(trainset_dummy) %in% vars3)
  }
  trainset <- trainset_dummy[,vars_loc]
  testset <- testset_dummy[,vars_loc]

  set.seed(seed = 1234)
  fit <- randomForest(formula = 매출총액~.,
                      data = trainset,
                      ntree = grid$ntree[i],
                      mtry = grid$mtry[i])
  grid$error[i] <- tail(x = fit$mse, n = 1)
```

```r
  #변수중요도 플랏 저장
  setwd("C:/Users/ChangYong/Desktop/나노디그리/1.정규강의 학습자료/1차 프로젝트/소상공인/eda/1501_2009/랜덤포레스트")
  png(filename = paste0("변수중요도_",i,"_",grid$tuning[i],".png"), width = 8000, height = 4000, res = 500)
  varImpPlot(x = fit, main = 'variable importance')
  dev.off()

  #시험셋으로 목표변수 추정값 생성
  pred1 <- predict(object = fit, newdata = testset, type = 'response')
  pred_list <- cbind(pred_list,pred1)
  #실제 관측치 벡터 생성
  real <- testset$매출총액

  #실측값과 비교하기 위해 testset 조작
  testset <- testset_dummy
  testset$매출총액_pred <- pred1
  testset <- testset %>%
    group_by(행정구역,대분류) %>%
    mutate(rank_real = row_number(desc(매출총액)),
           rank_pred = row_number(desc(매출총액_pred)),
           top3_real = ifelse(rank_real <=3,"1","0"),
           top3_pred = ifelse(rank_pred <=3,"1","0"))

  #rank를 factor형으로 변경
  testset[,13:16] <- map_df(.x = testset[,13:16],.f = as.factor)

  #real_rank와 pred_rank 산점도 그리기
  testset %>%
    ggplot(aes(x = rank_real, y = rank_pred, color = as.factor(rank_real)))+
    geom_point(position = position_jitter(),size = 2)+
    ggsave(filename = paste0("rank산점도_",i,"_",grid$tuning[i],".png"), width = 24, height = 12, units = "cm")


  #회귀값 예측 결과
  grid$RMSE[i] <- MLmetrics::RMSE(y_pred = pred1, y_true = real)

  #Top3 범주값 예측 결과
  grid$F1[i] <- MLmetrics::F1_Score(y_true = testset$top3_real, y_pred = testset$top3_pred, positive = "1")
  grid$R2[i] <- MLmetrics::R2_Score(y_true = real, y_pred = pred1)

  disp <- str_glue('현재 {i}행 완료! [{round((i)/n,2)*100}% 완료]')
  Sys.sleep(3)
  save(grid,pred_list,file = "RandomForest_fitting.rda")
  Sys.sleep(1)
  cat(disp, "\n")
}


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

#RMSE,F1,R2 플랏
text <- data.frame(x = rep(nrow(grid)+3,3),
                   y = as.numeric(grid[nrow(grid),5:7]),
                   label = colnames(grid)[5:7])

#Top3 실제 값과 예측 값 및 매출 실제 값 및 예측값 결과 비교
#RMSE, F1, R2,
grid %>% mutate(order = row_number()) %>%
  ggplot(aes(x = order, y = RMSE))+geom_line(col = "blue")+geom_point(col = "blue")+ylab("")+
  geom_vline(xintercept = which.min(grid$RMSE), col = "blue", lty = 1, lwd = 2, alpha = 0.7)+
  geom_line(aes(y = F1), col = "red")+geom_point(aes(y = F1),col = "red")+
  geom_vline(xintercept = which.max(grid$F1), col = "red", lty = 6, lwd = 1.75)+
  geom_line(aes(y = R2), col = "orange")+geom_point(aes(y = R2), col = "orange")+
  geom_vline(xintercept = which.max(grid$R2), col = "orange", lty = 2, lwd = 1.2)+
  scale_y_continuous(sec.axis = dup_axis(), breaks = seq(0,1,0.05))+
  geom_text(data = text, mapping = aes(x = text$x, y = text$y, label = text$label),col = c("blue","red","orange"), size = 10)+
  theme_classic()
```

Aa 이름	생성일	태그
18.rd	@2021년 1월 25일 오후 12:22	
rank산점도_35_4	@2021년 1월 25일 오후 12:35	
rank산점도_38_4	@2021년 1월 25일 오후 12:35	
rank산점도_47_5	@2021년 1월 25일 오후 12:35	
rank산점도_72_8	@2021년 1월 25일 오후 12:35	

ntree	mtry	error	RMSE	F1	R2
300	4	0.0287	0.3390	0.9111	0.8910
600	4	0.0285	0.3387	0.9111	0.8913
800	6	0.0274	0.3226	0.8933	0.9016
1700	3	0.0288	0.3216	0.8800	0.9020