이름 : 정희철

학번 : 2016314895

library(mice)

library(rms)

library(finalfit)

library(data.table)

library(dplyr)

library(magrittr)

library(ggplot2)

library(corrplot)

library(gam)

library(astsa)

^ 사용한 패키지들

Q1 )

training.data <- read.csv("Google Drive/My Drive/skku class files/2021년도 2학기/통계적 모델링과 머신러닝 실습/11주차/train.csv", header = TRUE)

test.data <- read.csv("Google Drive/My Drive/skku class files/2021년도 2학기/통계적 모델링과 머신러닝 실습/11주차/test.csv", header = TRUE)

describe(training.data)

md.pattern(training.data)

missing\_pairs(training.data,'Y',c('X1','X2','X3','X4') )

md.pairs(training.data)

# 먼저 데이터를 불러들이고 결측치들이 있는지 확인하였고, 결측치 패턴에 대해 간략하게 보았다.

# 다른 변수들과 달리 X3변수에서 결측치에 패턴이 있는 것으로 보인다.

# 결측된 데이터들의 MNAR 여부를 확인하기 위해 Y변수를 제외한 나머지 변수들과 X3변수를 binary 변수로 변환하여 유의한 값이 나오는지 확인한다.

fit = glm(is.na(X3) ~ X1 + X2 + X4, data = training.data, family=binomial )

summary(fit)

# X1과 X2의 p-values가 유의하게 나왔다. X4 또한 유의하진 않지만 0.05에 매우 근사하게 나왔으므로 결측치에 패턴이 존재하는 것이 분명해 보인다.

#위에서 X3를 제외한 나머지 변수들은 결측치에 패턴이 없는 것으로 확인하였으므로 imputation method에 큰 제약은 없다. X3는 predictive mean matching으로 하였고, 나머지 변수들 또한 큰 차이가 없을 것으로 판단되어 변수 전부 predictive mean matching을 적용하였다.

set.seed(0)

imp1 = mice(training.data, m=10, method=c('pmm','pmm','pmm','pmm',''), print=F)

imp1$predictorMatrix

pred = imp1$predictorMatrix

pred[,'Y'] = 0

imp2 = mice(training.data, m=10, method=c('pmm','pmm','pmm','pmm',''), predictorMatrix = pred, print=F)

stripplot(imp2, pch=20, cex=1.2)

densityplot(imp2, scales=list(relation='free'),layout=c(2,1))

#위 2개의 그래프로 실제 데이터와 imputed 데이터가 유의한 차이는 없다는 것으로 추정한다.

# 10개의 Imputed set들로 regression model set을 생성한다.

fit = with(imp2, lm(Y ~ X1 + X2 + X3 + X4))

summary(pool(fit))

#10개의 model set으로 예측을 하고, 예측치들의 평균을 낸다. 구해진 평균들을 final estimates로 하고 test mse를 구한다.

M = imp2$m

imp2.dat = vector(mode='list',length=M)

for (m in 1:M) imp2.dat[[m]] = complete(imp2,m)

p.model = function(training.data) lm(Y ~ X1 + X2 + X3 + X4, data=training.data)

fit.imp2 = lapply(imp2.dat, p.model)

test.obs = data.frame(X1 = test.data[,1], X2 = test.data[,2], X3 = test.data[,3], X4 = test.data[,4])

yhat = lapply(fit.imp2, predict, newdata=test.obs)

yhat = matrix(unlist(yhat),nrow(test.obs),M)

predicted.mean <- apply(yhat,1,mean)

test.mse <- sum((test.data[,5] - predicted.mean)^2)/length(predicted.mean)

# test.mse = 20.54701

Q2 )

training.2 <- read.csv("Google Drive/My Drive/skku class files/2021년도 2학기/통계적 모델링과 머신러닝 실습/11주차/pm25\_tr.csv", header = TRUE)

test.2 <- read.csv("Google Drive/My Drive/skku class files/2021년도 2학기/통계적 모델링과 머신러닝 실습/11주차/pm25\_te.csv", header = TRUE)

### Y variable

#일단 Y 변수의 분포를 알아보고 transformation이 필요한지 판단한다.

plot(density(training.2$pm25)) # right-skewed되어 있는 것을 확인할 수 있다.

plot(density(log(training.2$pm25))) # log-transformation이 적절해 보인다.

### time-related variables

# year, month, day, hour변수가 있는데 이 중 year변수는 모두 동일하게 출력이 되어있기 때문에 의미가 없다. 또한, month와 day 데이터 또한 만약 패턴이 존재한다면 hour변수의 시계열 분석을 통해 알아낼 수 있다고 생각하여 제외하였다.

# 위 4개의 시간 관련 변수 중 hour만 남겨둔다.

training.2 <- training.2[,4:10]

test.2 <- test.2[,4:10]

### continuous variables

# 연속형 데이터로는 DEWP, TEMP, PRES, Iws가 있다. 먼저 4개 변수를 모두 활용한 모델을 간단히 만들어 Y값에 기여하지 못하는 변수를 제외한다.

# X변수들의 대한 아무 정보도 없기 때문에 GAM을 사용하여 모델을 만든다.

library(gam)

training.2[,8] <- log(training.2[,2])

test.2[,8] <- log(test.2[,2])

colnames(training.2) <- c('hour', 'pm25', 'DEWP', 'TEMP', 'PRES', 'cbwd', 'Iws', 'log.pm25')

colnames(test.2) <- c('hour', 'pm25', 'DEWP', 'TEMP', 'PRES', 'cbwd', 'Iws', 'log.pm25')

fit.gam <- gam(log.pm25 ~ s(DEWP,5) + s(TEMP,5) + s(PRES,5) + s(Iws,5), data = training.2)

summary(fit.gam)

# 4개의 변수 모두 유의한 p-value를 갖는다. Multicollinearity가 있는지 확인해보는 것이 필요하다.

cor(training.pm25[,c(3,4,5,7)])

#TEMP변수와 PRES변수가 상당히 높은 상관관계를 가지고 있는 것으로 보인다. 때문에 두 변수 중 하나는 제외하는 것이 적절해 보인다.

cor(training.2[,c(2,4,5)])

cor(training.2[,c(2,4,5)],method='spearman')

cor(training.2[,c(2,4,5)],method='kendall')

# 그 어떤 correlation method로도 TEMP-pm25의 상관계수가 PRES-pm25의 상관계수보다 낮기 때문에 둘 중 PRES변수를 선택한다.

fit.gam.2 <- gam(log.pm25 ~ s(DEWP,5) + s(PRES,5) + s(Iws,5), data = training.2)

par(mfrow=c(3,1))

plot(fit.gam.2)

par(mfrow=c(1,1))

# DEWP : 선형 또는 3차식 중 택1

# PRES : 2차식 또는 4차식 중 택1

# Iws : 2차식 또는 3차식 중 택1

# ANOVA test를 통해 어떤 모델이 더 적합할지 판단한다.

dewp.gam1 <- gam(log.pm25 ~ DEWP + s(PRES,5) + s(Iws,5), data = training.2) #linear

dewp.gam2 <- gam(log.pm25 ~ s(DEWP,3) + s(PRES,5) + s(Iws,5), data = training.2) # cubic

anova(dewp.gam1, dewp.gam2)

# DEWP는 3차식이 적절하다.

pres.gam1 <- gam(log.pm25 ~ s(DEWP,3) + s(PRES,2) + s(Iws,5), data = training.2) #quadratic

pres.gam2 <- gam(log.pm25 ~ s(DEWP,3) + s(PRES,4) + s(Iws,5), data = training.2) #4th degree

anova(pres.gam1, pres.gam2)

# PRES는 4차식이 적절하다.

iws.gam1 <- gam(log.pm25 ~ s(DEWP,3) + s(PRES,4) + s(Iws,2), data = training.2) #quadratic

iws.gam2 <- gam(log.pm25 ~ s(DEWP,3) + s(PRES,4) + s(Iws,3), data = training.2) #cubic

anova(iws.gam1, iws.gam2)

# Iws는 3차식이 적절하다.

#지금까지의 model

# log.pm25 ~ DEWP + I(DEWP^2) + I(DEWP^3) + PRES + I(PRES^2) + I(PRES^3) + I(PRES^4) + Iws + I(Iws^2) + I(Iws^3) + cbwd

## 주어진 데이터는 hourly data이기 때문에 time-correlation이 존재할 가능성이 다분하여, 확인된다면 적절한 조치가 필요하다.

acf2(fit.gam$residuals)

#ACF는 점진적으로 감소하고 PACF는 1 이후 없어지는 것으로 보아 lag-1 time correlation이 존재한다. 때문에 lag-1 term을 만들어준다.

data.with.lag = training.2 %>% mutate(Iws.lag1 = lag(Iws),DEWP.lag1 = lag(DEWP),PRES.lag1=lag(PRES))

data.with.lag.test = test.2 %>% mutate(Iws.lag1 = lag(Iws),DEWP.lag1 = lag(DEWP),PRES.lag1=lag(PRES))

# 각 변수가 독립적으로 Y variable에 관여할 수 도 있지만, 변수간Interaction이 있을 가능성이 있기 때문에 possible interaction term을 찾아본다.

without.inter.model1 <- lm(log.pm25 ~ DEWP + PRES + Iws + cbwd, data = training.2)

with.inter.model1 <- lm(log.pm25 ~ DEWP + PRES + Iws + cbwd + PRES\*DEWP, data = training.2)

anova(without.inter.model1, with.inter.model1)

without.inter.model3 <- lm(log.pm25 ~ DEWP + PRES + Iws + cbwd, data = training.2)

with.inter.model3 <- lm(log.pm25 ~ DEWP + PRES + Iws + cbwd + cbwd\*Iws, data = training.2)

anova(without.inter.model3, with.inter.model3)

# PRES\*DEWP와 cbwd\*Iws의 interaction term은 적절하다.

# Final Model

lmod <- lm(log.pm25 ~ . + I(DEWP^2) + I(DEWP^3) + I(PRES^2) + I(PRES^3) + I(PRES^4) + I(Iws^2) + I(Iws^3) + PRES\*DEWP + cbwd\*Iws, data = data.with.lag)

y\_hat <- predict(lmod, data.with.lag.test)

mean(((data.with.lag.test$log.pm25)[2:120] - y\_hat[2:120])^2)

mean(((data.with.lag.test$pm25)[2:120] - exp(y\_hat[2:120]))^2)

# test.mse = 408.9258