Appendix

Faisal

2023-04-15

library(ggplot2)  
library(cowplot)  
library(tidyverse)  
library(caret)  
library(dplyr)  
library(tidyverse)  
library(glmnet)  
library(neuralnet)  
library(MASS)

## CODE BLOCK A

DF = read.csv('./Data/insurance.csv')  
  
#### Observe dataset and handle missing values ####  
  
str(DF)   
plot(DF)  
  
#check for missing values  
sum(is.na(DF))  
  
#Check the outliers first  
p1 <- ggplot(data=DF,aes(x=age))+geom\_histogram()  
p2 <- ggplot(data=DF, aes(x=bmi))+geom\_histogram()  
p3 <- ggplot(data=DF, aes(x=children))+geom\_histogram()  
p <- plot\_grid(p1,p2,p3, ncol=2, labels="auto")  
p  
  
  
  
#boxplot of bmi  
ggplot(data=DF, aes(y=bmi))+geom\_boxplot()  
  
#handle the outliers of bmi  
# First Quantile:  
Q1 = quantile(DF[,3])[2]   
# Third Quantile:  
Q2 = quantile(DF[,3])[4]  
# Inner Quantile distance:  
IQR = Q2-Q1  
  
# Lower bound Quantile Range:  
lo = Q1-1.5\*IQR  
# Upper bound Quantile Range:  
up = Q2+1.5\*IQR  
# Put outliers greater than upper quantile equal to upper quantile range:   
if(sum(DF[3]>up)){  
 indexu <- which(DF[3]>up)  
 DF[[3]][indexu] = up  
} else if(sum(DF[3]<lo)){   
 indexl <- which(DF[3]<lo)  
 DF[[3]][indexl] = lo}  
  
  
#Checking BMI outlier again  
ggplot(data=DF, aes(y=bmi))+geom\_boxplot()  
  
#### ####

## CODE BLOCK B

#### EDA of the dataset and Scaling ####  
#EDA for smoker  
ggplot(data=DF, mapping = aes(x=smoker,y=charges,fill=smoker))+geom\_boxplot()  
  
#Charges and children  
ggplot(data=DF, mapping = aes(x=as.factor(children), y=charges,))+geom\_boxplot()  
  
#Charges and sex  
ggplot(data=DF, mapping = aes(x=sex,y=charges,fill=sex))+geom\_boxplot()  
  
#Charges and region  
ggplot(data=DF, mapping = aes(x=region,y=charges))+geom\_boxplot()  
  
#Charges with sex and status of smoker  
ggplot(data=DF, mapping = aes(x=sex,y=charges,fill=smoker))+geom\_boxplot()  
  
#Charges and BMI  
ggplot(data=DF, mapping = aes(x=bmi,y=charges,color=bmi))+geom\_point(size=5)  
  
#### ####  
  
##Scale the numeric variables ####  
SDF <- DF  
SDF$age <- (DF$age - mean(DF$age) ) / sd(DF$age)  
SDF$bmi <- (DF$bmi - mean(DF$bmi) ) / sd(DF$bmi)  
SDF$children <- (DF$children - mean(DF$children) ) / sd(DF$children)  
SDF$charges <- (DF$charges - mean(DF$charges) ) / sd(DF$charges)  
  
##Check for right scaling  
max(SDF[,7])  
min(SDF[,7])  
mean(SDF[,7])  
#### ####

## CODE BLOCK C

#### Linear Model ####  
set.seed(7)  
  
random\_sample <- createDataPartition(SDF$charges,  
 p=0.8,list=FALSE)  
train.set <- SDF[random\_sample,]  
test.set <- SDF[-random\_sample,]  
  
  
LinearM <- lm(charges ~.,data = train.set)  
summary(LinearM)  
  
  
plot(LinearM)  
  
predictions <- predict(LinearM, test.set)  
predictiona = (predictions\*sd(DF[,7])+mean(DF[,7]))  
test.a = (test.set$charges\*sd(DF[,7])+mean(DF[,7]))  
data.frame(R2 = R2(predictions,test.set$charges),  
 RMSE = RMSE(predictiona,test.a))

## CODE BLOCK D

## Step model ####  
  
step.model = train(charges~.,data=train.set,  
 method="leapBackward",  
 tuneGrid=data.frame(nvmax=1:6),  
 trControl=train\_control)  
  
step.model$results  
step.model$bestTune  
  
summary(step.model$finalModel)  
  
predictions = predict(step.model,test.set)  
predictiona = (predictions\*sd(DF[,7])+mean(DF[,7]))  
test.a = (test.set$charges\*sd(DF[,7])+mean(DF[,7]))  
data.frame(R2 = R2(predictiona,test.set$charges),  
 RMSE = RMSE(predictiona,test.b))

## CODE BLOCK E

## Ridge ####  
set.seed(7)  
  
x <- model.matrix(charges~.,train.set)[,-1]  
y <- train.set$charges  
  
#Best lambda using cv  
cv <- cv.glmnet(x,y,alpha = 0)  
cv$lambda.min  
  
#Fit the final model on the train set  
ridge <- glmnet(x=as.data.frame(x),y,alpha = 0,lambda = cv$lambda.min)  
coef(ridge)  
  
#Make predictions on test set  
x.test <- model.matrix(charges~.,test.set)[,-1]  
predictions <- ridge %>% predict(x.test) %>% as.vector()  
predictionsa <- (predictions\*sd(DF[,7])+mean(DF[,7]))  
test.a = (test.set$charges\*sd(DF[,7])+mean(DF[,7]))  
data.frame(  
 R2 = R2(predictionsa,test.set$charges),  
 RMSE = RMSE(predictionsd,test.a)  
)

## CODE BLOCK F

## Lasso ####  
set.seed(7)  
cv <- cv.glmnet(x,y,alpha=1)  
cv$lambda.min  
  
lasso <- glmnet(x,y,alpha=1,lambda=cv$lambda.min)  
coef(lasso)  
plot(cv)  
#Make predictions on the test data  
x.test <- model.matrix(charges~.,test.set)[,-1]  
predictions <- lasso %>% predict(x.test) %>% as.vector()  
predictionsa <- (predictions\*sd(DF[,7])+mean(DF[,7]))  
test.a = (test.set$charges\*sd(DF[,7])+mean(DF[,7]))  
data.frame(  
 R2 = R2(predictionsa,test.set$charges),  
 RMSE = RMSE(predictionsd,test.a)  
)  
  
## ####

## CODE BLOCK G

## ElasticNet ####  
set.seed(7)  
cv <- cv.glmnet(x,y,alpha=0.5)  
cv$lambda.min  
  
elastic <- glmnet(x,y,alpha=0.5,lambda=cv$lambda.min)  
coef(elastic)  
  
#Make predictions on the test data  
x.test <- model.matrix(charges~.,test.set)[,-1]  
predictions <- elastic %>% predict(x.test) %>% as.vector()  
predictionsa <- (predictions\*sd(DF[,7])+mean(DF[,7]))  
test.a = (test.set$charges\*sd(DF[,7])+mean(DF[,7]))  
data.frame(  
 R2 = R2(predictionsa,test.set$charges),  
 RMSE = RMSE(predictionsd,test.a)  
)  
  
## ####

## CODE BLOCK H

## Neural Network ####  
dataset1 <- model.matrix(  
 ~age+sex+bmi+children+smoker+region+charges,  
 data=DF  
)  
dataset <- model.matrix(  
 ~age+sex+bmi+children+smoker+region+charges,  
 data=SDF  
)  
  
train1 <- model.matrix(  
 ~age+sex+bmi+children+smoker+region+charges,  
 data=train.set  
)  
test1 <-model.matrix(  
 ~age+sex+bmi+children+smoker+region+charges,  
 data=test.set  
)  
  
#Train NN  
nn <- neuralnet(charges~age+sexmale+bmi+children+smokeryes+regionnorthwest+regionsoutheast+regionsouthwest,  
 data=train1,hidden = c(6),  
 linear.output = TRUE, threshold = 0.02,stepmax = 100000)  
  
plot(nn)  
#predict on test data  
predict.nn <- compute(nn,test1[,-10])  
  
#Caculate MSE  
predict.nn2 <- predict.nn$net.result \* sd(dataset1[,10]) +mean(dataset1[,10])  
test.r <- (test1[,10])\* sd(dataset1[,10]) +mean(dataset1[,10])  
MSE.nn <- sum((test.r - predict.nn2)^2)/nrow(test1)  
print(MSE.nn)  
RMSE <- sqrt(MSE.nn)  
print(RMSE)  
  
## ####