library(correlation)

library(ggplot2)

library(dplyr)

library(broom)

installed.packages("ggpubr")

installed.packages("Metrics")

installed.packages("PerformanceAnalytics")

read.csv(file.choose(BostonHousing))

housing.df<-data.frame(BostonHousing)

# (1) Tell me why should the data be partitioned into training and validation sets? What will the training setbe used for?

# What will the validation set be used for?

# We need to partition the data into the training and validation set for the purpose

# of assessing the generalization of the model; also, it is useful to identify and

# evaluate the relationships between the predictor and predicted variables.

# The training data is used for the purpose of model fitting. And, the validation data

# is used for empirical validation and measures of errors.

#(2) Fit a multiple linear regression model to the median house price (MEDV)

# as a function of CRIM,CHAS, and RM. Write the equation for

# predicting the median house price from the predictors in the model.

# We will split the data 70% for training, and 30% for validation

reg <-lm(MEDV~CRIM+CHAS+RM, data=housing.df)

summary(reg)

# The linear regression model is MEDV= -28.81 + (-.261\*CRIM)

# + (3.76\*CHAS) + (8.28\*RM)

#(3) Using the estimated regression model, what median house price is predicted for a tract in the

# Boston area that does not bound the Charles River, has a crime rate of 0.1, and where the average number of rooms per house is 6?

# MEDV= -28.81 + (-.261\*.1) + (3.76\*0) + (8.28\*6)

reg$coef%\*%c(1,0.1,0,6)

# The median house price is $20,832.32

# (4)(a) Reduce the number of predictors: Which predictors are likely to be measuring the same thing among the 13 predictors?

# (b) Discuss the relationships among INDUS, NOX, and TAX.

# (a) Some of the predictors are likely to measure the same thing, but in different ways.

# Some of the predictors are : ZN, INDUS, Tax.

# All this provides a proportion related to the area of land, house.

indus=housing.df$INDUS

nox=housing.df$NOX

tax=housing.df$TAX

d=data.frame(indus,nox,tax)

cor(d)

# correlation between indus and nox is .7636; correlation between indus and tax is .7208, and

# correlation between nox and tax is .6680

# (b) There is a high correlation between INDUS, NOX, and TAX as they include

# a higher percentage of non-retail businesses that translate to higher pollution and taxes.

# (c) Compute the correlation table for the 12 numerical predictors

# and search for highly correlated pairs. These have potential redundancy and can cause multi-collinearity.

# Choose which ones to remove based on the above table

crim=housing.df$CRIM

zn=housing.df$ZN

indus=housing.df$INDUS

chas=housing.df$CHAS

nox=housing.df$NOX

rm=housing.df$RM

age=housing.df$AGE

dis=housing.df$DIS

rad=housing.df$RAD

tax=housing.df$TAX

ptratio=housing.df$PTRATIO

lstat=housing.df$LSTAT

data=data.frame(cr,zn,indus,ch,nx,rm,ag,di,rd,tx,pt,ls)

cor(data)

# There is a high positive correlation between nox and indus = 0.7637

# There is a high positive correlation between rad and tax = 0.91022

# There is a high negative correlation between dis and nox = -0.7692

# We might remove the nox predictor according to the given matrix

# (d) Use stepwise regression with the three options (backward, forward, both)

# to reduce the remaining predictors as follows: Run stepwise on the training set.

# Choose the top model from each stepwise run. Then use each of these models separately to predict the validation set.

# Compare RMSE,MAPE, and mean error, as well as lift charts. Finally, describe the best model.

# Stepwise regression

#The models with minimum AIC are:

# Backward: medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + lstat

# Formard: medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + lstat

# Both: medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + lstat

spec = c(train = .7, validate = .3)

df=data.frame(cr,zn,indus,ch,nx,rm,ag,di,rd,tx,pt,ls,medv)

df=data.frame(housing.df)

g = sample(cut(

seq(nrow(df)),

nrow(df)\*cumsum(c(0,spec)),

labels = names(spec)

))

res = split(df, g)

train=res$train

validate=res$validate

model\_backward=lm(medv~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + lstat,data=validate)

summary(model\_backward)

# rmse(validate$medv,model\_backward$fitted.values)

# mape(validate$medv,model\_backward$fitted.values)

step(model\_backward,direction = "backward")

model\_forward=lm(medv~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + lstat,data=validate)

summary(model\_forward)

# rmse(validate$medv,model\_forward$fitted.values)

# mape(validate$medv,model\_forward$fitted.values)

step(model\_forward, direction = "forward")