ams580\_hw4

Mustafa Yigit Isik

2023-02-28

#**Problem 1-2**

#loading packages and data  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(rpart)  
library(rattle)

## Loading required package: bitops  
## Rattle: A free graphical interface for data science with R.  
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(xlsx)  
library(glmnet)

## Loading required package: Matrix  
##   
## Attaching package: 'Matrix'  
##   
## The following object is masked from 'package:bitops':  
##   
## %&%  
##   
## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack  
##   
## Loaded glmnet 4.1-6

library(caTools)

#read the data  
data <- read.csv('/Users/mustafayigitisik/Desktop/stuff/semesters/spring 2023/ams 580/hw4\_pen\_reg\_models/Ames\_Housing\_Data.csv')  
data <- na.omit(data)  
cat('There are', nrow(data), 'observations left.')

## There are 1460 observations left.

#manipulate "CentralAir : chr" column to 0-1 boolean to prevent problems later on  
data$CentralAir <- ifelse(data$CentralAir=="Yes",1,0)

str(data) # the response variable is 'SalePrice' and the others are used to fit the model

## 'data.frame': 1460 obs. of 20 variables:  
## $ Id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ OverallQual : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ OverallCond : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...  
## $ YearRemodAdd: int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...  
## $ CentralAir : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ BedroomAbvGr: int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr: int 1 1 1 1 1 1 1 1 2 2 ...  
## $ TotRmsAbvGrd: int 8 6 6 7 9 5 7 7 8 5 ...  
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...  
## $ GarageCars : int 2 2 2 3 3 2 2 2 2 1 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

set.seed(123)  
training.samples <- data$SalePrice %>% createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]  
str(train.data) #1097 obs

## 'data.frame': 1097 obs. of 20 variables:  
## $ Id : int 2 3 5 6 7 9 10 12 14 15 ...  
## $ LotArea : int 9600 11250 14260 14115 10084 6120 7420 11924 10652 10920 ...  
## $ OverallQual : int 6 7 8 5 8 7 5 9 7 6 ...  
## $ OverallCond : int 8 5 5 5 5 5 6 5 5 5 ...  
## $ YearBuilt : int 1976 2001 2000 1993 2004 1931 1939 2005 2006 1960 ...  
## $ YearRemodAdd: int 1976 2002 2000 1995 2005 1950 1950 2006 2007 1960 ...  
## $ CentralAir : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ X1stFlrSF : int 1262 920 1145 796 1694 1022 1077 1182 1494 1253 ...  
## $ X2ndFlrSF : int 0 866 1053 566 0 752 0 1142 0 0 ...  
## $ GrLivArea : int 1262 1786 2198 1362 1694 1774 1077 2324 1494 1253 ...  
## $ FullBath : int 2 2 2 1 2 2 1 3 2 1 ...  
## $ HalfBath : int 0 1 1 1 0 0 0 0 0 1 ...  
## $ BedroomAbvGr: int 3 3 4 1 3 2 2 4 3 2 ...  
## $ KitchenAbvGr: int 1 1 1 1 1 2 2 1 1 1 ...  
## $ TotRmsAbvGrd: int 6 6 9 5 7 8 5 11 7 5 ...  
## $ Fireplaces : int 1 1 1 0 1 2 2 2 1 1 ...  
## $ GarageCars : int 2 2 3 2 2 2 1 3 3 1 ...  
## $ GarageArea : int 460 608 836 480 636 468 205 736 840 352 ...  
## $ YrSold : int 2007 2008 2008 2009 2007 2008 2008 2006 2007 2008 ...  
## $ SalePrice : int 181500 223500 250000 143000 307000 129900 118000 345000 279500 157000 ...

str(test.data) #363 obs

## 'data.frame': 363 obs. of 20 variables:  
## $ Id : int 1 4 8 11 13 16 22 30 34 39 ...  
## $ LotArea : int 8450 9550 10382 11200 12968 6120 7449 6324 10552 7922 ...  
## $ OverallQual : int 7 7 7 5 5 7 7 4 5 5 ...  
## $ OverallCond : int 5 5 6 5 6 8 7 6 5 7 ...  
## $ YearBuilt : int 2003 1915 1973 1965 1962 1929 1930 1927 1959 1953 ...  
## $ YearRemodAdd: int 2003 1970 1973 1965 1962 2001 1950 1950 1959 2007 ...  
## $ CentralAir : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ X1stFlrSF : int 856 961 1107 1040 912 854 1108 520 1700 1057 ...  
## $ X2ndFlrSF : int 854 756 983 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 1710 1717 2090 1040 912 854 1108 520 1700 1057 ...  
## $ FullBath : int 2 1 2 1 1 1 1 1 1 1 ...  
## $ HalfBath : int 1 0 1 0 0 0 0 0 1 0 ...  
## $ BedroomAbvGr: int 3 3 3 3 2 2 3 1 4 3 ...  
## $ KitchenAbvGr: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ TotRmsAbvGrd: int 8 7 7 5 4 5 6 4 6 5 ...  
## $ Fireplaces : int 0 1 2 0 0 0 1 0 1 0 ...  
## $ GarageCars : int 2 3 2 1 1 2 1 1 2 1 ...  
## $ GarageArea : int 548 642 484 384 352 576 280 240 447 246 ...  
## $ YrSold : int 2008 2006 2009 2008 2008 2007 2007 2008 2010 2010 ...  
## $ SalePrice : int 208500 140000 200000 129500 144000 132000 139400 68500 165500 109000 ...

#**Problem 3 - Ridge Regression model**

x <- model.matrix(SalePrice~., train.data)[,-1]  
y <- train.data$SalePrice  
cv <- cv.glmnet(x, y, alpha = 0)  
cv$lambda.min

## [1] 6452.855

Best lambda for ridge regression is 6452.8554828

model <- glmnet(x, y, alpha = 0, lambda = cv$lambda.min) # alpha=0: ridge  
coef(model)

## 20 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) -4.921730e+03  
## Id -2.370607e+00  
## LotArea 6.800815e-01  
## OverallQual 1.550492e+04  
## OverallCond 4.255424e+03  
## YearBuilt 4.061514e+02  
## YearRemodAdd 1.996065e+02  
## CentralAir .   
## X1stFlrSF 5.123985e+01  
## X2ndFlrSF 1.745124e+01  
## GrLivArea 3.878434e+01  
## FullBath -2.175481e+03  
## HalfBath 2.205009e+02  
## BedroomAbvGr -1.096476e+04  
## KitchenAbvGr -2.905523e+04  
## TotRmsAbvGrd 4.295209e+03  
## Fireplaces 4.907615e+03  
## GarageCars 2.398330e+03  
## GarageArea 3.596972e+01  
## YrSold -6.202902e+02

x.test <- model.matrix(SalePrice ~., test.data)[,-1]  
predictions <- model %>% predict(x.test) %>% as.vector()  
data.frame(  
 RMSE = RMSE(predictions, test.data$SalePrice),  
 Rsquare = R2(predictions, test.data$SalePrice)  
)

## RMSE Rsquare  
## 1 46610.31 0.6700558

RMSE is 46610.31 and coefficient of determination is 0.6700558

#**Problem 4 - LASSO model**

cv <- cv.glmnet(x, y, alpha = 1)  
cv$lambda.min

## [1] 352.4729

Best lambda for LASSO is 352.4728549

model <- glmnet(x, y, alpha = 1, lambda = cv$lambda.min) # alpha=1: lasso  
coef(model)

## 20 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) -5.423586e+05  
## Id -1.319502e+00  
## LotArea 6.861566e-01  
## OverallQual 1.677154e+04  
## OverallCond 4.980216e+03  
## YearBuilt 4.750727e+02  
## YearRemodAdd 1.107893e+02  
## CentralAir .   
## X1stFlrSF 3.573514e+01  
## X2ndFlrSF .   
## GrLivArea 6.477753e+01  
## FullBath -4.240515e+03  
## HalfBath -2.265113e+02  
## BedroomAbvGr -1.218997e+04  
## KitchenAbvGr -2.736537e+04  
## TotRmsAbvGrd 3.106596e+03  
## Fireplaces 2.636584e+03  
## GarageCars .   
## GarageArea 3.628796e+01  
## YrSold -3.371908e+02

x.test <- model.matrix(SalePrice ~., test.data)[,-1]  
predictions <- model %>% predict(x.test) %>% as.vector()  
data.frame(  
 RMSE = RMSE(predictions, test.data$SalePrice),  
 Rsquare = R2(predictions, test.data$SalePrice)  
)

## RMSE Rsquare  
## 1 48332.64 0.6615918

RMSE is 48332.64 and coefficient of determination is 0.6615918

#**Problem 5 - Elastic Net model model**

model <- train(  
 SalePrice ~., data = train.data, method = "glmnet",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10  
)  
model$bestTune

## alpha lambda  
## 5 0.1 849.1096

coef(model$finalModel, model$bestTune$lambda)

## 20 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) -2.302994e+05  
## Id -2.021318e+00  
## LotArea 6.966544e-01  
## OverallQual 1.644734e+04  
## OverallCond 5.158666e+03  
## YearBuilt 4.948139e+02  
## YearRemodAdd 1.232650e+02  
## CentralAir .   
## X1stFlrSF 5.496184e+01  
## X2ndFlrSF 2.125715e+01  
## GrLivArea 4.540506e+01  
## FullBath -6.196033e+03  
## HalfBath -2.290343e+03  
## BedroomAbvGr -1.293127e+04  
## KitchenAbvGr -2.862276e+04  
## TotRmsAbvGrd 3.853760e+03  
## Fireplaces 2.957892e+03  
## GarageCars .   
## GarageArea 3.688081e+01  
## YrSold -5.228556e+02

x.test <- model.matrix(SalePrice ~., test.data)[,-1]  
predictions <- model %>% predict(x.test)  
data.frame(  
 RMSE = RMSE(predictions, test.data$SalePrice),  
 Rsquare = R2(predictions, test.data$SalePrice)  
)

## RMSE Rsquare  
## 1 48525.83 0.6606771

RMSE is 48525.83 and coefficient of determination is 0.6606771

#**Problem 6 - choose which model**

#Proposed: Choice should be based on coeffiecient of determination. The coef of determination that is closest to 1 is 0.6700558 of Ridge Reg (compared to 0.6615918 & 0.6606771 for lasso and elastic net model), which leads me to believe that it's the best option of these 3 whereas the other 2 underfits during shrinkage process.