MATH 242 Midterm Project

Jiaxing and Gage

3/02/2024

```r
# install.packages("leaps")
# install.packages("glmnet")
```

```r
library(dplyr)
library(readr)
library(ggplot2)
library(glmnet)
library(leaps)
```

```r
nyc_condos <- read.csv("data/nyc-condos_s24.csv")
nyc_condos_full <- read.csv("data/full_data.csv")
```

```r
# Set the seed
set.seed(123)
# Create an index to randomly sample 70% of the data for training
train_index <- sample(1:nrow(nyc_condos_full), 0.7 * nrow(nyc_condos_full))
# Create the training set
train_data <- nyc_condos_full[train_index, ]
# Create the testing set
test_data <- nyc_condos_full[-train_index, ]
```

```r
# summary of dataset
str(nyc_condos)
```

```
## 'data.frame':    200 obs. of  16 variables:
##  $ Boro.Block.Lot         : chr  "1-01613-7501" "1-01171-7501" "3-02237-7519" "4-04955-7512" ...
##  $ Condo.Section          : chr  "0267-R1" "1058-R1" "3457-R1" "0278-R1" ...
##  $ Address                : chr  "1255 5 AVENUE" "200 RIVERSIDE BOULEVARD" "135 MIDDLETON STREET" "13
##  $ Neighborhood           : chr  "UPPER EAST SIDE (96-110)" "UPPER WEST SIDE (59-79)" "WILLIAMSBURG-C
##  $ Building.Classification: chr  "R4-CONDOMINIUM" "R4  -ELEVATOR" "R4-ELEVATOR" "R2-CONDOMINIUM" ...
##  $ Total.Units            : int  59 358 14 4 198 10 60 6 10 20 ...
##  $ Year.Built             : int  1925 1997 1942 1987 1963 1983 1928 1959 2005 2004 ...
##  $ Gross.SqFt             : int  63284 512280 26964 4010 206278 10962 61084 4497 9082 22295 ...
##  $ Estimated.Gross.Income : int  1613742 29871047 579187 60391 6266726 392220 742781 92683 242853 72
##  $ Gross.Income.per.SqFt  : num  25.5 58.3 21.5 15.1 30.4 ...
##  $ Estimated.Expense      : int  726500 5665817 205466 24782 2044215 162457 417204 37100 73837 218714
##  $ Expense.per.SqFt       : num  11.48 11.06 7.62 6.18 9.91 ...
##  $ Net.Operating.Income   : int  887242 24205230 373721 35609 4222511 229763 325577 55583 169016 5058
##  $ Full.Market.Value      : int  6857996 196582995 2914000 239000 32481000 1826000 2048000 437001 135
##  $ Market.Value.per.SqFt  : num  108.4 383.7 108.1 59.6 157.5 ...
##  $ Report.Year            : int  2015 2019 2016 2012 2012 2015 2014 2018 2019 2012 ...
```

```r
# # calculate average market value for each year
# nyc_condos <- nyc_condos %>%
#   group_by(Report.Year) %>%
#   mutate(average_market_value = mean(Full.Market.Value, na.rm = TRUE))
#
#
# # Log transform Gross SqFt
# nyc_condos$log_GrossSqFt <- log(nyc_condos$Gross.SqFt)
# # Log transform Estimated Gross Income
# nyc_condos$log_EstimatedGrossIncome <- log(nyc_condos$Estimated.Gross.Income)
# # Log transform Estimated Expense
# nyc_condos$log_EstimatedExpense <- log(nyc_condos$Estimated.Expense)
# Log transform Net Operating Income
nyc_condos$log_NetOperatingIncome <- log(nyc_condos$Net.Operating.Income)
nyc_condos_full$log_NetOperatingIncome <- log(nyc_condos_full$Net.Operating.Income)
train_data$log_NetOperatingIncome <- log(train_data$Net.Operating.Income)
test_data$log_NetOperatingIncome <- log(test_data$Net.Operating.Income)




# Log transform Full Market Value
# nyc_condos$log_FullMarketValue <- log(nyc_condos$Full.Market.Value)
#
# nyc_condos$log_average_market_value <- log(nyc_condos$average_market_value)
```

```r
set.seed(250)
## model.matrix() creates our design matrix of predictors
x <- model.matrix(Full.Market.Value  ~ Year.Built + Total.Units + Estimated.Gross.Income + Gross.Income
## select our outcome and convert it into a vector
## instead of a dataframe
y <- nyc_condos %>% select(Full.Market.Value) %>% unlist() %>% as.numeric()
## fit lasso for a range of lambda values (lambda is the tuning parameter
## that controls shrinkage)
cv.out <- cv.glmnet(x, y, alpha = 1)
## pick out the optimal lambda
bestlam <- cv.out$lambda.min
## get coefficients from the Lasso model
lasso <- as.matrix(coef(cv.out, s = bestlam))
t(lasso)
```

```
##    (Intercept) Year.Built Total.Units Estimated.Gross.Income
## s1    -1836394    963.828   -5292.705               7.701503
##    Gross.Income.per.SqFt Estimated.Expense Expense.per.SqFt
## s1               10508.7         -6.664841                0
##    log_NetOperatingIncome Gross.SqFt
## s1              -43867.68 -0.8515914
```

```r
lm <- lm(Full.Market.Value ~ Year.Built + Total.Units + Gross.SqFt+ Gross.Income.per.SqFt + Expense.per
summary(lm)
```

```
##
```

```
## Call:
## lm(formula = Full.Market.Value ~ Year.Built + Total.Units + Gross.SqFt +
##     Gross.Income.per.SqFt + Expense.per.SqFt + log_NetOperatingIncome,
##     data = train_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -279198850   -6244013   -1629018    3749614  239342766
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.039e+08  2.626e+06 -39.567  < 2e-16 ***
## Year.Built              8.193e+03  1.149e+03   7.132 1.02e-12 ***
## Total.Units            -4.845e+03  4.705e+02 -10.297  < 2e-16 ***
## Gross.SqFt              1.074e+02  9.815e-01 109.380  < 2e-16 ***
## Gross.Income.per.SqFt   5.532e+05  1.342e+04  41.231  < 2e-16 ***
## Expense.per.SqFt       -5.158e+05  3.844e+04 -13.419  < 2e-16 ***
## log_NetOperatingIncome  6.079e+06  1.160e+05  52.412  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13790000 on 22039 degrees of freedom
##   (56 observations deleted due to missingness)
## Multiple R-squared:  0.7336, Adjusted R-squared:  0.7336
## F-statistic: 1.012e+04 on 6 and 22039 DF,  p-value: < 2.2e-16
```

Our Model is:

$$
\text{Full.Market.Value} = -103900000 + 8193 \times \text{Year.Built} - 4845 \times \text{Total.Units} + 107.4 \times \text{Gross.SqFt} + 553200 \times \text{Gross.Income.per.SqFt}
$$

Title:

Abstract

Introduction

Explanation of our variables:

1. **CondoSection**: Identification information for the condominium.

2. **Address**: Street address of the property.
3. **Neighborhood**: Name of the neighborhood where the property is located.
4. **BldgClassification**: Building classification code and description indicating the property's use.
5. **TotalUnits**: Total number of units in the building.
6. **YearBuilt**: Year the building was constructed.
7. **GrossSqFt**: Gross square footage of the building.
8. **EstGrossIncome**: Estimated gross income, calculated as income per square foot multiplied by gross square footage.
9. **GrossIncomePerSqFt**: Estimated gross income per square foot.
10. **EstimatedExpense**: Estimated expense, calculated as expense per square foot multiplied by gross square footage.
11. **ExpensePerSqFt**: Estimated expense per square foot.
12. **NetOperatingIncome**: Net operating income, calculated as estimated gross income minus estimated expense.
13. **FullMarketValue**: Current year's total market value of the property (land and building).
14. **MarketValuePerSqFt**: Market value per square foot, calculated as full market value divided by gross square footage.
15. **ReportYear**: Year of the report.
16. **Boro-Block-Lot**: Borough-Block-Lot location identifier for the property.

Methods

Results

Discussion

Conclusion

References