ECON573 Final Project

Nicholas Wong

November 08, 2023

R set-up

```
library(tidyverse)
library(ISLR)
library(leaps)
library(glmnet)
library(pls)
library(MASS)
library(caret)
library(corrplot)
library(ggplot2)
library(sf)
library(RColorBrewer)
library(gridExtra)
library(modelr)
library(knitr)


airbnb <- read_csv("airbnb.csv",
                  col_select = -c("id", "amenities", "description", "name", "thumbnail_url", "neighbour
```

Data cleaning/pre-processing

```
set.seed(123)

# Mutate original price variable
airbnb <- airbnb |>
  mutate(price = exp(log_price))

# Drop NAs
airbnb <- na.omit(airbnb)

# host_response_rate to numeric for easier interpretation
airbnb <- airbnb |>
  mutate(host_response_rate = str_replace_all(host_response_rate, pattern = "%", replacement = "")) |>
  mutate_at(13, as.numeric)

# 80/20 train/test split
training_indices <- sample(1:nrow(airbnb), .8*nrow(airbnb))

# Split data into train and test sets
```

```r
train <- airbnb[training_indices, ]
test <- airbnb[-training_indices, ] # true unseen data for model testing

totalData <- rbind(train, test)
for (f in 1:length(names(totalData))) {
  levels(train[, f]) <- levels(totalData[, f])
  levels(test[,f]) <- levels(totalData[, f])
}
```

Method 1 - Forward Selection

Here, we use a validation set approach due to the heavy computational expense of using stepwise methods with K-fold CV.

Fitting forward selection on training set

```r
full = lm(price ~., data=train)
none = lm(price ~., data = train)
MSE = (summary(full)$sigma)^2
forward_selection_mod <- step(none, scope = list(upper = full), scale = MSE, direction = 'forward', trac
```

```
## Start:  AIC=64
## price ~ log_price + property_type + room_type + accommodates +
##     bathrooms + bed_type + cancellation_policy + cleaning_fee +
##     city + first_review + host_has_profile_pic + host_identity_verified +
##     host_response_rate + host_since + instant_bookable + last_review +
##     latitude + longitude + number_of_reviews + review_scores_rating +
##     bedrooms + beds
```

```r
summary(forward_selection_mod)
```

```
##
## Call:
## lm(formula = price ~ log_price + property_type + room_type +
##     accommodates + bathrooms + bed_type + cancellation_policy +
##     cleaning_fee + city + first_review + host_has_profile_pic +
##     host_identity_verified + host_response_rate + host_since +
##     instant_bookable + last_review + latitude + longitude + number_of_reviews +
##     review_scores_rating + bedrooms + beds, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -381.98  -25.38   -5.95   16.20 1241.87
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -8.594e+02  3.078e+02  -2.792 0.005243 **
## log_price                         1.723e+02  7.753e-01 222.260  < 2e-16 ***
## property_typeBed & Breakfast     -1.469e+01  3.761e+00  -3.905 9.42e-05 ***
## property_typeBoat                 1.595e+01  9.983e+00   1.598 0.110079
## property_typeBoutique hotel      -7.397e+00  1.151e+01  -0.643 0.520301
```

```
## property_typeBungalow               6.537e+00  4.119e+00   1.587 0.112556    
## property_typeCabin                  1.520e+01  9.622e+00   1.580 0.114109    
## property_typeCamper/RV              2.890e+01  8.605e+00   3.359 0.000784 ***
## property_typeCastle                -1.095e+01  1.828e+01  -0.599 0.549220    
## property_typeCave                  -4.731e+01  6.057e+01  -0.781 0.434709    
## property_typeChalet                -2.148e+01  2.712e+01  -0.792 0.428236    
## property_typeCondominium           -4.514e+00  1.688e+00  -2.674 0.007508 ** 
## property_typeDorm                   1.558e+01  6.901e+00   2.258 0.023970 *  
## property_typeEarth House            8.239e+00  4.282e+01   0.192 0.847435    
## property_typeGuest suite            4.388e+00  7.065e+00   0.621 0.534501    
## property_typeGuesthouse             8.190e+00  3.422e+00   2.393 0.016707 *  
## property_typeHostel                 1.318e+01  9.149e+00   1.441 0.149618    
## property_typeHouse                  8.282e+00  8.460e-01   9.790  < 2e-16 ***
## property_typeHut                    3.517e+01  2.710e+01   1.298 0.194338    
## property_typeIn-law                 5.948e+00  8.294e+00   0.717 0.473323    
## property_typeIsland                -4.598e+01  6.065e+01  -0.758 0.448370    
## property_typeLoft                   1.129e+01  2.329e+00   4.846 1.27e-06 ***
## property_typeOther                  8.455e+00  3.479e+00   2.430 0.015100 *  
## property_typeServiced apartment     3.073e+01  1.619e+01   1.897 0.057785 .  
## property_typeTent                   3.944e+01  1.832e+01   2.153 0.031346 *  
## property_typeTimeshare              2.910e+01  1.239e+01   2.349 0.018845 *  
## property_typeTipi                   1.756e+01  3.500e+01   0.502 0.615838    
## property_typeTownhouse             -6.730e+00  2.035e+00  -3.306 0.000946 ***
## property_typeTrain                 -2.631e+01  4.284e+01  -0.614 0.539096    
## property_typeTreehouse              1.844e+02  3.498e+01   5.272 1.36e-07 ***
## property_typeVacation home         -1.959e+01  3.029e+01  -0.647 0.517895    
## property_typeVilla                  1.017e+02  6.155e+00  16.514  < 2e-16 ***
## property_typeYurt                  -5.717e+00  2.475e+01  -0.231 0.817354    
## room_typePrivate room               3.481e+01  9.130e-01  38.126  < 2e-16 ***
## room_typeShared room                8.464e+01  2.294e+00  36.899  < 2e-16 ***
## accommodates                        1.469e+00  2.949e-01   4.983 6.29e-07 ***
## bathrooms                           3.768e+01  6.976e-01  54.021  < 2e-16 ***
## bed_typeCouch                       5.243e+00  7.379e+00   0.711 0.477342    
## bed_typeFuton                      -5.375e+00  5.260e+00  -1.022 0.306811    
## bed_typePull-out Sofa              -1.512e+01  5.442e+00  -2.779 0.005463 ** 
## bed_typeReal Bed                   -1.921e+01  4.317e+00  -4.451 8.56e-06 ***
## cancellation_policymoderate        -7.924e-01  9.402e-01  -0.843 0.399317    
## cancellation_policystrict           1.572e+00  8.878e-01   1.770 0.076666 .  
## cancellation_policysuper_strict_30  1.663e+00  7.530e+00   0.221 0.825181    
## cancellation_policysuper_strict_60  3.200e+02  2.300e+01  13.915  < 2e-16 ***
## cleaning_feeTRUE                   -6.852e+00  8.595e-01  -7.972 1.60e-15 ***
## cityChicago                        -3.954e+01  5.817e+01  -0.680 0.496682    
## cityDC                             -1.563e+01  2.662e+01  -0.587 0.557080    
## cityLA                             -1.141e+02  1.713e+02  -0.666 0.505247    
## cityNYC                             1.764e-01  1.279e+01   0.014 0.988995    
## citySF                             -1.339e+02  1.817e+02  -0.737 0.461233    
## first_review                        6.103e-04  9.027e-04   0.676 0.498962    
## host_has_profile_picTRUE           -2.267e+01  7.848e+00  -2.888 0.003874 ** 
## host_identity_verifiedTRUE          9.615e-01  7.492e-01   1.283 0.199408    
## host_response_rate                  1.282e-02  2.427e-02   0.528 0.597456    
## host_since                          3.697e-03  5.801e-04   6.373 1.88e-10 ***
## instant_bookableTRUE               -8.934e-01  7.219e-01  -1.238 0.215862    
## last_review                        -9.173e-03  2.263e-03  -4.054 5.04e-05 ***
## latitude                            1.161e+00  4.528e+00   0.256 0.797689    
```

```
## longitude                           -2.702e+00  3.495e+00  -0.773 0.439390
## number_of_reviews                   -6.129e-02  9.167e-03  -6.686 2.33e-11 ***
## review_scores_rating                -1.456e-02  4.423e-02  -0.329 0.741982
## bedrooms                             9.466e+00  6.044e-01  15.661  < 2e-16 ***
## beds                                -9.208e-01  4.472e-01  -2.059 0.039521 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.55 on 38164 degrees of freedom
## Multiple R-squared:  0.8001, Adjusted R-squared:  0.7998
## F-statistic:  2425 on 63 and 38164 DF,  p-value: < 2.2e-16
```

```
#plot(forward_selection_mod)
```

Get predictions and residuals for forward selection, compute MSE

```
test_forwardselection <- test |> add_predictions(forward_selection_mod, var = "forward_pred")
test_forwardselection <- test_forwardselection |> add_residuals(forward_selection_mod, var = "forward_re

# Args: vector of residuals
# Return: MSE
MSE_func <- function(resid){
  return(mean(resid^2))
}

MSE_func(test_forwardselection$forward_resid)
```

```
## [1] 4395.526
```

MSE for forward selection: 4395.526

Method 2 - Lasso

Cluster Analysis

Top 3 priced listings

```
#top_500

head(airbnb |> arrange(desc(log_price)) |>  dplyr::select(log_price, city), 3) |> kable()
```

log_price	city
7.600402	NYC
7.598399	LA
7.588324	NYC

2 out of 3 top listings are from NYC. Perform cluster analysis in NYC