STAT 151A Project

Predicting Housing Resale Prices in Singapore

Michelle Vuong, Celina Mac, Lewis Chong

April 10th 2024

Introduction

Research Objectives

Our research objective is to gain understanding of the effect COVID-19 had on the Singaporean housing market. We aim to create an efficient and concise model that gives us an answer to the question: how do the resale prices differ between the years of 2017-2019 and 2020-2022? In addition, we will analyze the original set of housing metric indicators provided from the data set given to us from Kaggle and determine if we can refine the number of indicators to a minimum. Making use of statistical analysis techniques such as Ridge and Lasso regularization, we aim to extract the most important predictors that can still effectively provide us an understanding of the resale prices of Singaporean homes.

Data Collection

Our data collection process began with an open web research on the housing markets of Singapore. We landed on Kaggle, an open source hub of public data sets uploaded by public users, which can be used for data exploration, building predictive models, and general practice with real-world data. Specifically, our data from Kaggle was transcribed by a user from the Singapore Government Agency Website that studied Resale flat prices based on registration date from Jan-2017 onwards. Data was collected by the Housing and Development Board, commonly referred to as "HDB". It is a statutory board of the Ministry of National Development in Singapore and it seeks to provide support in homeownership and ease in rental processes for residents. Simultaneously as the HDB are providing aid, they are collecting data on what home are being bought, built, and sold for. As this government established board provides public housing for more than 80% of Singapore's population, this project will make the assumption that all data was collected as a random sample of Singapore's population and data quality is up to par with research standards.

EDA + Data Preprocessing

We will begin with Exploratory Data Analysis to understand the data that we are working with.

Data Inspection

```
cat("Dimension of data: ", dim(housing), "\n")
```

```
## Dimension of data:  134168 11
```

```r
#NA values in cols
na_counts <- colSums(is.na(housing))

#
any_na <- any(na_counts > 0)

cat("Datasets contains NA values : ",any_na)
```

```
## Datasets contains NA values :  FALSE
```

There are no null values in the dataset, so it is clean.

```r
head(housing)
```

```
##     month       town flat_type block       street_name storey_range
## 1 2017-01 ANG MO KIO    2 ROOM   406 ANG MO KIO AVE 10     10 TO 12
## 2 2017-01 ANG MO KIO    3 ROOM   108  ANG MO KIO AVE 4     01 TO 03
## 3 2017-01 ANG MO KIO    3 ROOM   602  ANG MO KIO AVE 5     01 TO 03
## 4 2017-01 ANG MO KIO    3 ROOM   465 ANG MO KIO AVE 10     04 TO 06
## 5 2017-01 ANG MO KIO    3 ROOM   601  ANG MO KIO AVE 5     01 TO 03
## 6 2017-01 ANG MO KIO    3 ROOM   150  ANG MO KIO AVE 5     01 TO 03
##   floor_area_sqm     flat_model lease_commence_date    remaining_lease
## 1             44       Improved                1979 61 years 04 months
## 2             67 New Generation                1978 60 years 07 months
## 3             67 New Generation                1980 62 years 05 months
## 4             68 New Generation                1980  62 years 01 month
## 5             67 New Generation                1980 62 years 05 months
## 6             68 New Generation                1981           63 years
##   resale_price
## 1       232000
## 2       250000
## 3       262000
## 4       265000
## 5       265000
## 6       275000
```

```r
summary(housing)
```

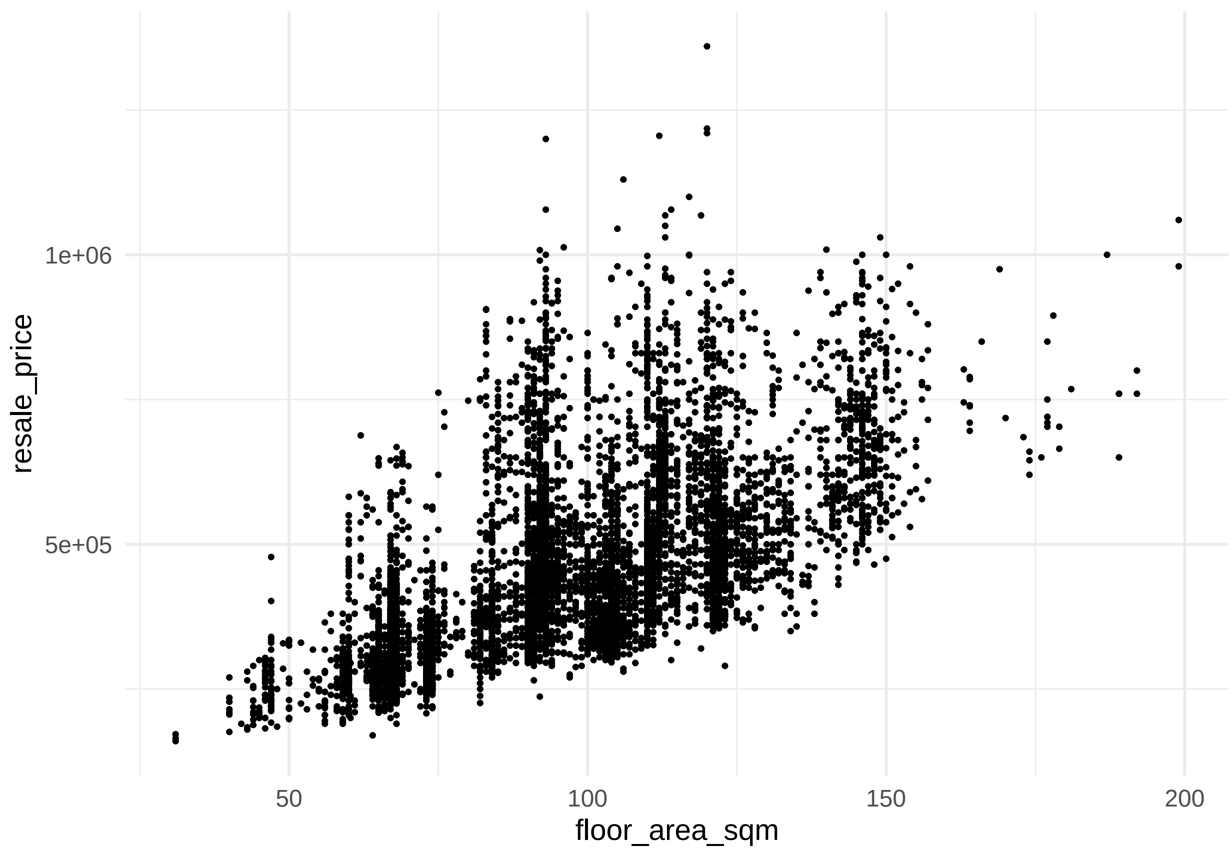
```
##     month               town             flat_type            block          
##  Length:134168      Length:134168      Length:134168      Length:134168     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##  street_name        storey_range       floor_area_sqm    flat_model       
##  Length:134168      Length:134168      Min.   : 31.00   Length:134168     
##  Class :character   Class :character   1st Qu.: 82.00   Class :character  
##  Mode  :character   Mode  :character   Median : 94.00   Mode  :character  
##                                        Mean   : 97.77                     
##                                        3rd Qu.:113.00                     
```

```
##                                               Max.   :249.00
##  lease_commence_date remaining_lease      resale_price
##  Min.   :1966        Length:134168      Min.   : 140000
##  1st Qu.:1985        Class :character   1st Qu.: 350000
##  Median :1996        Mode  :character   Median : 440000
##  Mean   :1995                           Mean   : 470669
##  3rd Qu.:2006                           3rd Qu.: 555000
##  Max.   :2019                           Max.   :1418000
```

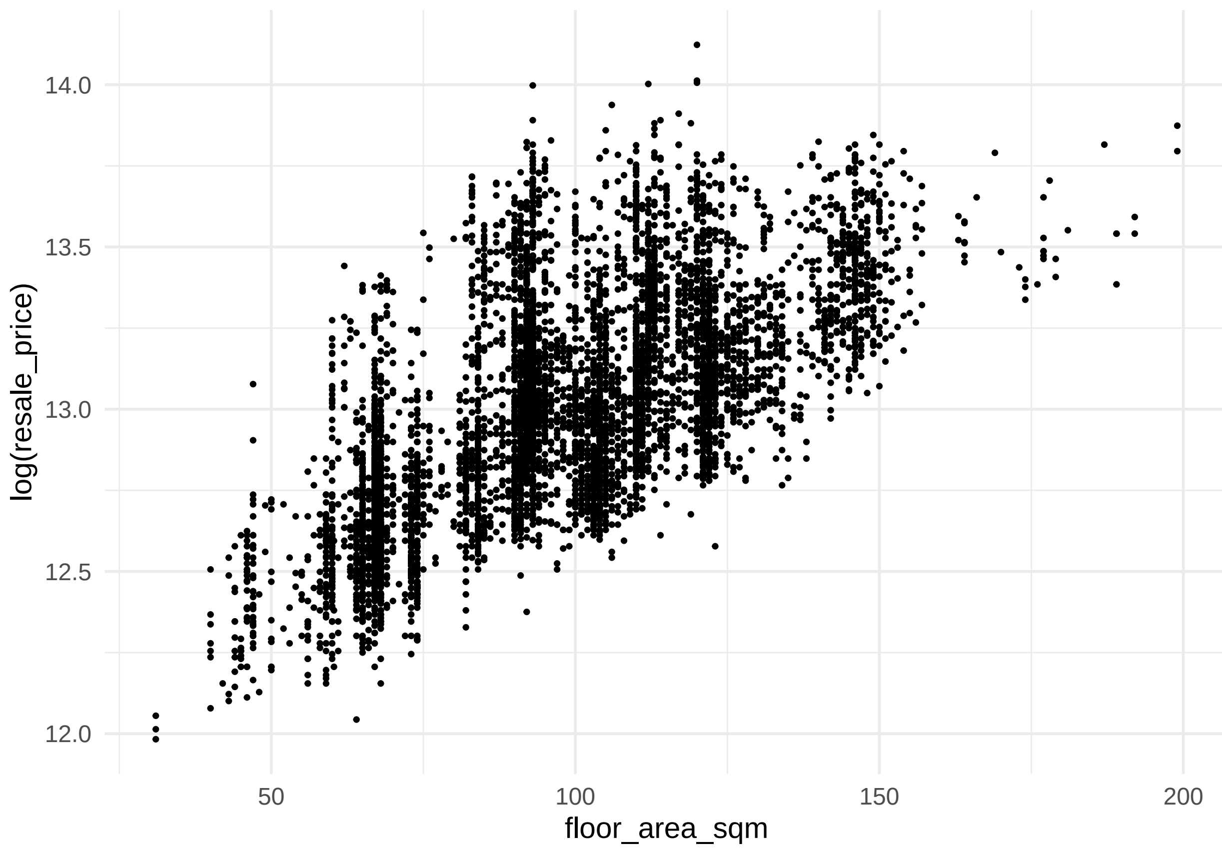
1. Log transform

```r
id <- sample(nrow(housing),7000)
sample_housing <- housing[id,]

## histogram
ggplot(sample_housing) +
  geom_point(aes(x=floor_area_sqm,y=resale_price),size=0.5,position="identity") +
  theme_minimal()
```




```r
ggplot(sample_housing) +
  geom_point(aes(x=floor_area_sqm,y=log(resale_price)),size=0.5,position="identity") +
  theme_minimal()
```



We do a sample of 7000 on the original dataset, to argue that the increase of a small amount of floor area(sqm) doesn't result it a linear amount of resale price being added, but instead some non-linear increase in the price. This is equivalent to adding to a log of the resale prices. So we conclude that it results in better prediction if we do a regression on the log(resale price).

2. One hot encoding for Flat Type

We will be apply one hot encoding for the `flat_type` column to regress on the categorical values

```
housing <- housing %>%
  mutate(
    is_2_room = ifelse(flat_type == "2 ROOM", 1, 0),
    is_3_room = ifelse(flat_type == "3 ROOM", 1, 0),
    is_4_room = ifelse(flat_type == "4 ROOM", 1, 0),
    is_5_room = ifelse(flat_type == "5 ROOM", 1, 0),
    is_executive = ifelse(flat_type == "EXECUTIVE", 1, 0),
    is_1_room = ifelse(flat_type == "1 ROOM", 1, 0),
    is_multi_generation = ifelse(flat_type == "MULTI-GENERATION", 1, 0)
  )
```

###3. Data Manipulation We will be converting the `remaining_lease` column that contains how long the lease is to be of unit month instead of the current year+month.

```
##Function to convert from years+ months to months
extract_months <- function(duration_str) {
```

```r
    # Split into components
    components <- strsplit(duration_str, " ", perl = TRUE)[[1]]

    # Extract years and months (if available)
    years <- as.numeric(components[1])
    months <- ifelse(length(components) >= 3, as.numeric(components[length(components)-1]), 0)

    # Return total months
    return(years * 12 + months)
}

housing <- housing %>%
            rowwise() %>%
            mutate(remaining_lease_mth = extract_months(remaining_lease))
```

Then, we will categorize the different towns of Singapore into NSEW regions:

```r
# Function to categorize towns into NSEW regions
categorize_town <- function(town) {
  north <- c("ANG MO KIO", "SEMBAWANG", "SENGKANG", "WOODLANDS", "YISHUN","BISHAN")
  south <- c("BUKIT MERAH", "BUKIT TIMAH", "CENTRAL AREA", "QUEENSTOWN")
  east <- c("BEDOK", "MARINE PARADE", "PASIR RIS", "TAMPINES")
  west <- c("BUKIT BATOK", "BUKIT PANJANG", "CHOA CHU KANG", "CLEMENTI", "JURONG EAST", "JURONG WEST",

  if (town %in% north) {
    return("North")
  } else if (town %in% south) {
    return("South")
  } else if (town %in% east) {
    return("East")
  } else if (town %in% west) {
    return("West")
  } else {
    return("Other")
  }
}

# Add a new column for NSEW region
housing <- housing %>%
  rowwise() %>%
  mutate(region = categorize_town(toupper(town)))
```

Then, we apply one-hot encoding on the regions as well:

```r
housing <- housing %>%
  mutate(
    is_north = ifelse(region == "North", 1, 0),
    is_south = ifelse(region == "South", 1, 0),
    is_west = ifelse(region == "West", 1, 0),
    is_east = ifelse(region == "East", 1, 0)
  )
```

Note that for one-hot encoding on the columns `region` and `flat_type` , we can just drop one of the columns as it can be identified by the rest of the columns, i.e `(is_north,is_south,is_west) = (0,0,0)`corresponds to the house being in the East Region.

```
housing <- housing %>%
        dplyr::select(!c(is_east,is_multi_generation))

##
```

3. Standardization

Model Training and Evaluation

In order to evaluate the most statistically significant housing metrics and reduce the number of predictors we have, we will use regularization. Beginning with Ridge Regularization:

```
X <- housing %>% dplyr::select(floor_area_sqm,remaining_lease_mth,
                          is_2_room,is_3_room,is_4_room,is_5_room,is_executive
                          ,is_1_room
                          #,is_multi_generation
                          ,is_north,is_south,is_west
                          # ,is_east
                          ,resale_price) %>%
                as.data.frame %>%
                slice_sample(n=10000)

# ggplot(housing) +
#   geom_boxplot(aes(x=resale_price))
# scale_cols <- c("floor_area_sqm","remaining_lease_mth")
#
# X[,scale_cols] <- scale(X[, scale_cols])
#
#
# ans <- lm(log(resale_price) ~ ., data=X)
```

Beta values very big, so we do regularization

```
# # Create an X matrix that represents the data frame above, but in linear perspective

# Perform Ridge regression
# ridge_model <- glmnet(X[,-ncol(X)], X[,ncol(X)], alpha = 0)
# #predictions <- predict.glm(ridge_model, newdata = X[,-ncol(X)])  # Adjust s (lambda) as needed
#
# # Select the index with best lambda value.
# min_lambda <- ridge_model$lambda %>%
#                  as.vector %>%
#                      which.min
# best_lambda <- ridge_model$lambda[min_lambda]
# best_lambda
#
# ridge_best <-  glmnet(X, y, alpha = 0,s=best_lambda)
```

Subsetting our dataset

We will filter all dates before April 2020 as that was when the Singaporean government began enforcing preventive measures for the pandemic.

```
pre_covid <- housing %>%
  filter(month < "2020-04")

covid <- housing %>%
  filter(month >= "2020-04")
```

Limitations and Future Work