Final_Beijng_Housing_Prediction

Lian Chen

12/18/2020

Introduction:

Real Estate investment has always been a heated topic. The housing price has been increasing rapidly in major cities in China in the past several years, which makes the housing market a good investment choice. Each house/apartment is in a different condition and can be listed at various prices. How do we know that the price listed is a fair value? How do we know if this is a good investment?

I will take Beijing's real estate market as an example and construct a predictive model for the housing price, which will help to make an investment decision. I will use the historical data as the training da We can compare the real price with this model's prediction to see if the house is over-valued (may not be a good investment) or under-valued (maybe a good investment).

The dataset I use is downloaded from Kaggle, https://www.kaggle.com/ruiqurm/lianjia (https://www.kaggle.com/ruiqurm/lianjia), which is from a leading franchise real estate medium company in China. The dataset includes housing market information from 2010 to 2018.
I glanced through all the notebooks on Kaggle, people used various models. Some people compared the error rates among different strategies, but only use one model at a time. I would like to try something different, which I want to combine two models to increase the accuracy. After data cleansing, I will first fit a linear model, which captures the linear predictive information. Then I will use a Random Forest model, which is less sensitive to over-fitting than a decision tree model, to predict the residuals from the linear model. The residuals have linear information removed.
In order to evaluate if the combined model is better than using each algorithm along, I will randomly select a group of data from the historical dataset to train a model and test on a randomly selected new dataset (of more recent data). Repeat this process several times and calculate a mean. This method is similar to k-folds but accommodates the purpose of using historical data to predict a future price. For convenience of this project, I will train one model for each method and calculate out of sample MSE for each.

Some information about the dataset:
What I will predict is the unite price which unit is RMB(Chinese Currency)/m^2, the unit for square(later renamed as space) is m^2.
This dataset has 318851 rows of observations with 26 different variables.

Pre-test

I believe housing price is a time sensitive data. To test this scenario, I use Anova to check if there is difference in price among different years of listed on the market.

```
# Check the number of missing data
apply(housing_beijing,2,function(x) sum(is.na(x)))
```

```
##                 url                  id                 Lng                 Lat
##                   0                   0                   0                   0
##                 Cid           tradeTime                 DOM           followers
##                   0                   0              157977                   0
##          totalPrice               price              square          livingRoom
##                   0                   0                   0                   0
##         drawingRoom             kitchen            bathRoom               floor
##                   0                   0                   0                   0
##        buildingType    constructionTime renovationCondition   buildingStructure
##                2021                   0                   0                   0
##         ladderRatio            elevator   fiveYearsProperty             subway
##                   0                  32                  32                  32
##            district    communityAverage
##                   0                 463
```

```
# Change date format and exclude 6 data points with very old listing year
housing_year <- housing_beijing %>%
  mutate(listyear = lubridate::year(tradeTime)) %>%
  filter(2011 < listyear & listyear <= 2018) %>%
  mutate(listyear = as.factor(listyear))

# Use anova to test if there is significant difference in price across years
fit_year <- lm(price~listyear, data = housing_year)
anova(fit_year)
```

```
## Analysis of Variance Table
##
## Response: price
##               Df     Sum Sq    Mean Sq F value    Pr(>F)
## listyear       6 4.5558e+13 7.5929e+12   23506 < 2.2e-16 ***
## Residuals 312639 1.0099e+14 3.2302e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The result shows that there is a difference in the price among different years of “tradeTime”.

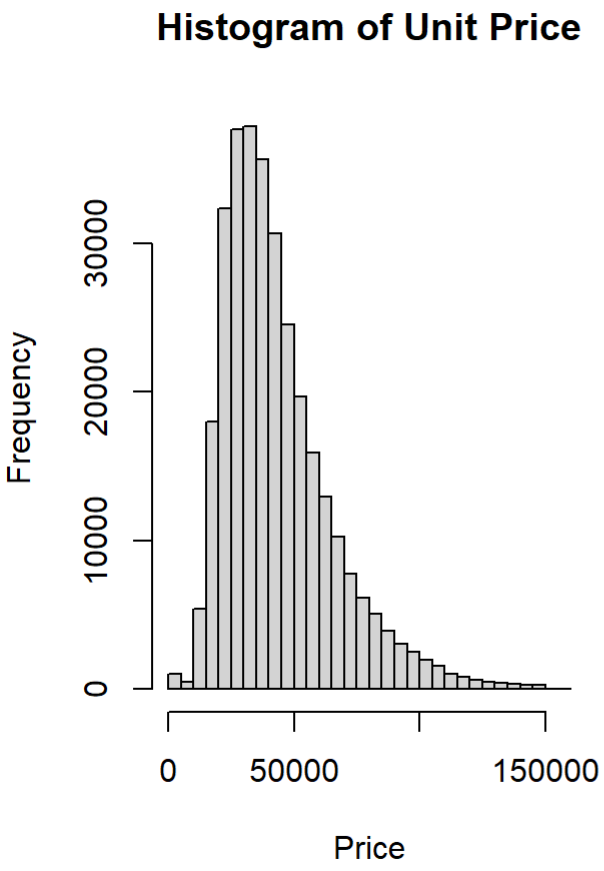
data cleansing

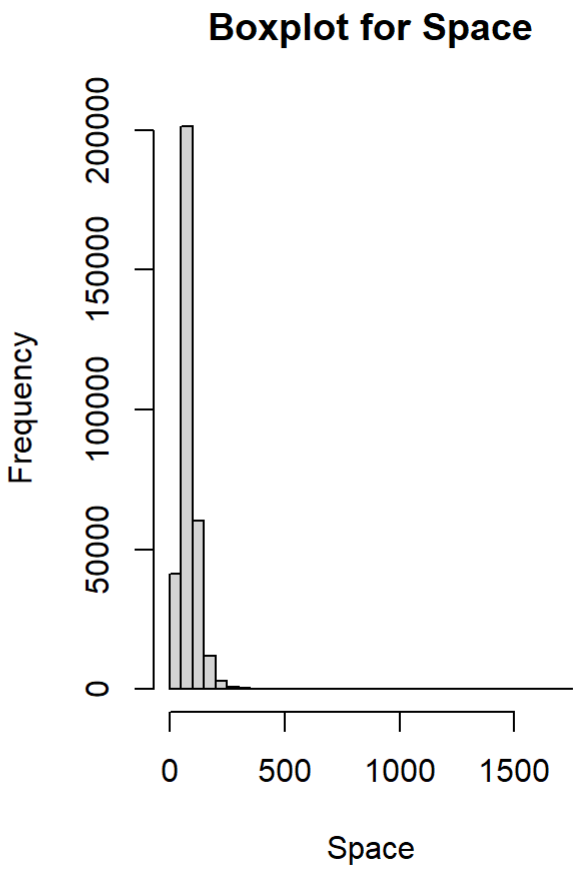
In order to get better knowledge of the data, I make histogram for the unit price and space.

```
par(mfrow=c(1,2))

hist(housing_beijing$price,
     breaks=50,
     xlab="Price",
     main="Histogram of Unit Price")

hist(housing_beijing$square,
        breaks=50,
        xlab="Space",
        main="Boxplot for Space")
```





I choose 13 variables out of all 26 variables based on their relevance to unite price and how common the information can be collected. Use the recent 2017 data as the training data set, and predict the price for 2018.

```
Housing <- housing_year %>%
  filter(listyear %in% c(2017,2018)) %>%
  select(price, listyear, square, livingRoom, drawingRoom, bathRoom, kitchen, buildingStructure,
         constructionTime, renovationCondition, elevator, subway, district) %>%
  rename(space=square,  bedroom = livingRoom, livingroom=drawingRoom, bathroom=bathRoom) %>%
  # Rename the variables to make them easier to understand
  # change categorical vectors type to factor format
  mutate(constructionTime = as.numeric(as.character(constructionTime)),
         kitchen = as.factor(kitchen),
         district = as.factor(district),
         renovationCondition = as.factor(renovationCondition),
         buildingStructure = as.factor(buildingStructure),
         elevator = as.factor(elevator),
         subway = as.factor(subway),
         price=as.numeric(price)) %>%
  # Use construction time to calculate how long the house has been used
  mutate(year_used = 2017-constructionTime,
         major_district = as.factor(ifelse(district!=1 & district!=7 & district!=8 & district!=1
0, 2, district)),
         logspace=log10(space)) %>%
  # Since the space is skewed, calculate the log of space for further analysis
  select(-c(constructionTime)) %>%
  # Base on the above distribution and other variable analysis I exclude a very few amount of ou
tliers
  filter(price>9000, space<700,  bedroom %in% seq(1,7,1), livingroom %in% seq(0,4,1),
         bathroom %in% seq(0,5,1), kitchen %in% c(0,1,2), buildingStructure %in% c(2,4,6))%>%
  drop_na()

# Choose some main districts of Beijing and combine the other smaller ones
levels(Housing$major_district) <- c( "Dongcheng",  "Others", "Chaoyang", "Haidian",  "Xicheng")
Housing$major_district <- relevel(Housing$major_district, ref = "Others")

# rename the levels
levels(Housing$buildingStructure) <- c("missing","unknow","mixed","brick and wood",
                                       "brick and concrete","steel", "composite")
levels(Housing$renovationCondition) <- c("missing", "other", "rough", "simplicity", "fine furnis
hed")
```

split data into training and testing dataset

Try to use historical data (2017)to predict a future unit price (2018).

```
train <- Housing[which(Housing$listyear==2017), ]
test <- Housing[which(Housing$listyear==2018), ]
```

modeling

select the optimal list of variables in predicting the price

linear model (including all 12 variables in the linear model) and two models combined

```
# set.seed(123)
# selec_train <- sample(1:floor(nrow(train)*0.7))
# selec_test <- sample(1:floor(nrow(test)*0.7))
#
# train1 <- train[selec_train,]
# test1 <- test[selec_test,]

train1 <- train
test1 <- test

# Linear model
lm_model <- lm(price ~  logspace + bedroom + livingroom + bathroom + kitchen + buildingStructure
+ renovationCondition + elevator + subway + year_used + major_district, data=train1)
summary(lm_model)   # All variables seem to be significant.
```

```
##
## Call:
## lm(formula = price ~ logspace + bedroom + livingroom + bathroom +
##     kitchen + buildingStructure + renovationCondition + elevator +
##     subway + year_used + major_district, data = train1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -89576  -9015   -882   7917  82437
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     91412.96    2223.15  41.119  < 2e-16 ***
## logspace                       -36859.53     867.06 -42.511  < 2e-16 ***
## bedroom2                         1899.17     225.40   8.426  < 2e-16 ***
## bedroom3                         4586.86     326.03  14.069  < 2e-16 ***
## bedroom4                         5607.35     658.04   8.521  < 2e-16 ***
## bedroom5                        -1501.35    1287.70  -1.166 0.243654
## bedroom6                         -984.80    3055.58  -0.322 0.747231
## bedroom7                        -6672.43    5538.19  -1.205 0.228285
## livingroom1                      4865.78     340.07  14.308  < 2e-16 ***
## livingroom2                      7387.96     419.78  17.600  < 2e-16 ***
## livingroom3                      9200.02    1385.47   6.640 3.17e-11 ***
## livingroom4                      4310.36    4761.74   0.905 0.365360
## bathroom1                       -3413.29    1873.54  -1.822 0.068486 .
## bathroom2                         431.10    1884.13   0.229 0.819019
## bathroom3                        7433.58    2031.91   3.658 0.000254 ***
## bathroom4                       12726.27    2624.25   4.849 1.24e-06 ***
## bathroom5                       22369.57    5018.07   4.458 8.30e-06 ***
## kitchen1                        16093.82    1065.73  15.101  < 2e-16 ***
## kitchen2                        21072.13    1748.99  12.048  < 2e-16 ***
## buildingStructurebrick and wood  2440.48     408.40   5.976 2.31e-09 ***
## buildingStructuresteel           1944.56     270.90   7.178 7.19e-13 ***
## renovationConditionother        -2408.10     532.49  -4.522 6.13e-06 ***
## renovationConditionrough        -1889.86     236.38  -7.995 1.33e-15 ***
## renovationConditionsimplicity    1192.78     228.66   5.216 1.83e-07 ***
## elevator1                        4477.26     263.09  17.018  < 2e-16 ***
## subway1                          5110.21     153.18  33.361  < 2e-16 ***
## year_used                         163.79      11.22  14.597  < 2e-16 ***
## major_districtDongcheng         42009.98     341.01 123.191  < 2e-16 ***
## major_districtChaoyang          14054.24     177.49  79.184  < 2e-16 ***
## major_districtHaidian           32026.86     244.68 130.895  < 2e-16 ***
## major_districtXicheng           51864.23     279.12 185.815  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14380 on 42328 degrees of freedom
## Multiple R-squared:  0.6485, Adjusted R-squared:  0.6482
## F-statistic:  2603 on 30 and 42328 DF,  p-value: < 2.2e-16
```

```
# Random Forest model
# rf_model <- randomForest(price ~ logspace + bedroom + livingroom + bathroom + kitchen + buildi
ngStructure + renovationCondition  + elevator + subway + year_used + major_district, data=train
1)

# summary(rf_model)

# Combind two
train_pred <- predict(lm_model, newdata = train1)
res <- train1$price-train_pred

train_tree <- train1[,which(colnames(train1)!="price")]
train_tree$res <- res

tree_pred_res <- randomForest(res ~ logspace + bedroom + livingroom + bathroom + kitchen + build
ingStructure + renovationCondition + elevator + subway + year_used + major_district, data=train_
tree)


pred_lm <- predict(lm_model, newdata = test)
pred_res <- predict(tree_pred_res, newdata = test)
pred_comb <- pred_lm + pred_res
```

Calculate the MSE

```
## for 2017 data
mean((test$price - pred_lm)^2) # linear model
```

```
## [1] 198010029
```

```
mean((test$price - pred_comb)^2) # combined model
```

```
## [1] 171229581
```

About 25% of the price is undervalued using this model, which indicates that if the predicted price is lower than the actual listing price in that year, it is more likely that the house is undervalued.

```
sum((test$price - pred_lm)>0)/length(pred_lm)
```

```
## [1] 0.2488688
```

Conclusion and Discussion

In conclusion, the method which first fits a linear model and then uses the random forest to fit the residual performs better than a linear model alone. The model is more likely to make an undervalued prediction than an overvalued prediction.

I could use all previous years of data to build a model and include the list year as a variable, but it took a long time to use my laptop to compute the random forest algorithm.

To determine if this new method is better than another stand-alone method, I can also calculate the MSE of other algorithms and use other error measurements to confirm the result. Another limitation is that we do not have more recent data, which makes it uncertain if the model fits best for the recent trend, but we can perform further analysis once we collect new data. To slightly improve this model we can multiply the predicted value by a coefficient, which is the general percentage increase of the price of a selected year compared to the year that the model used.