House Price Prediction

Final project for Math 342W Data Science at Queens College

5/25/2022

By Hengxin Cui

Abstract:

How much should we pay when we buy a new house? When we know what kind of house we need, we will know how much should we pay. Use the model to compare different choices of house.

Introduction

In this report, we would to modeling the house selling prices since 2016. In this dataset, It contains the different kinds of house (co-op or condo) with different features. Predict model is the model we used to predict the future. It help us to make any decisions. There are 2230 observations in the dataset. There are 56 responses for each observation. However there are some missing value or errors, we will dealing with that later. In this modeling, although there are many features in the dataset, but some are useless. They are not important when we predict sale price. We will choose some good features. In our class, we learned different model, linear model, random forest model, regression tree model, etc. In this paper, we will use these three models to predict the selling price.

The Data

In this dataset, there are 2230\*56 historical data frame. In the population, I think people are more like 3.2M. In the plot, we could see almost people buy house around 3.2M dollars. However, this dataset is incomplete, there are many missing values. We need to impute them.

Chart, histogram

Description automatically generated

Featurization

In my model, I only use some features. For example, "zip"," cats\_allowed"," common\_charges”, "coop\_condo","dining\_room\_type","dogs\_allowed"," fuel\_type",

“kitchen\_type","maintenance\_cost","num\_bedrooms","num\_floors\_in\_building","num\_full\_bathrooms","num\_half\_bathrooms","num\_total\_rooms","sale\_price","sq\_footage","total\_taxes","walk\_score”. These features are enough for our prediction.

Table

Description automatically generated

In this picture, we could see the details of these variables, their mean, standard deviation, number of observations are missing, and quantiles.

Errors and Missingness

In this dataset, it contains many missing values. Character value missing, numeric value missing, etc. We need to deal with these missing. We need set value for each missing. Because we could not use missing value to model selling price. Some of features contains half observations are missing values. That will result bias and inaccurate. We need to use some methods to impute these values. When we are dealing the numeric variables. I used mean or median to impute the missing values. The “sale\_price”,”maintaince\_cost”,”total\_taxes” and “commom\_charges” are character variables. But we need use is as numeric variables. First we should delete the $ sign and “,” in the format, then convert it to the numeric type. It will fix this error.

Regression Tree Modeling

The top 10 features are num\_floors\_in\_building, sq\_footage, num\_total\_rooms, num\_full\_bathrooms, fuel\_type, kitchen\_type, maintance\_cost, total\_taxes, num\_bedrooms, zip.

Linear Modeling

The R2 for this linear model is 19.55% and RMSE is 78754. This means these features can not explained sale price very well. This linear model is not good for this prediction, in the ols outcome table, we could see this model is overfit.

Table

Description automatically generated

Random Forest Modeling

Random forest modeling in good for this prediction because it has good performance at regression and classification tasks. We can understand it prediction easily. It can handle large dataset efficiently. The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm. It is a supervised learning algorithm. The “forest” it builds is an ensemble of decision trees, usually trained with the “bagging” method. Random forest builds multiple decision trees and merge them together to get a more accurate and stable prediction. It is non parametric models. In this model, I loose 7.53%. I think I got overfit in this model. I think it need to remove some features from random forest, it could do better.

Discussion

In this project, we try to predict the sale price in different setting. When I am modeling, I think I am short at random forest model. In the future, we should use this algorithm frequently. When we could do better in this algorithm, that will make our model ready for prediction. However, I did not believe my model could beat Zillow now.

Code Appendix

pacman::p\_load(tidyverse, magrittr, data.table, skimr, R.utils)

setwd("/Users/hiram")

apts = fread("housing\_data\_2016\_2017.csv")

#Missing & Error

substring <- function(x, n){

substr(x, nchar(x)-n+1, nchar(x))

}

apts$zip <- substring(apts$full\_address\_or\_zip\_code,5)

apt <- subset(apts, select = c("zip","cats\_allowed","common\_charges","coop\_condo","dining\_room\_type","dogs\_allowed","fuel\_type","kitchen\_type","maintenance\_cost","num\_bedrooms","num\_floors\_in\_building","num\_full\_bathrooms","num\_half\_bathrooms","num\_total\_rooms","sale\_price","sq\_footage","total\_taxes","walk\_score"))

apt[2,1]=11354

apt = data.frame(apt)

apt$num\_floors\_in\_building[is.na(apt$num\_floors\_in\_building)] <- median(apt$num\_floors\_in\_building, na.rm = T)

apt$num\_bedrooms[is.na(apt$num\_bedrooms)] <- median(apt$num\_bedrooms, na.rm = T)

apt$num\_half\_bathrooms[is.na(apt$num\_half\_bathrooms)] <- median(apt$num\_half\_bathrooms, na.rm = T)

apt$sq\_footage[is.na(apt$sq\_footage)] <- mean(apt$sq\_footage, na.rm = T)

apt$num\_total\_rooms[is.na(apt$num\_total\_rooms)] <- median(apt$num\_total\_rooms, na.rm = T)

#impute(apt$fuel\_type, "oil")

#impute(apt$kitchen\_type, "eat in")

#impute(apt$dining\_room\_type, "condo")

apt$sale\_price =gsub("\\$", "", apt$sale\_price)

apt$maintenance\_cost =gsub("\\$", "", apt$maintenance\_cost)

apt$total\_taxes =gsub("\\$", "", apt$total\_taxes)

apt$common\_charges =gsub("\\$", "", apt$common\_charges)

apt$sale\_price =gsub("\\,", "", apt$sale\_price)

apt$maintenance\_cost =gsub("\\,", "", apt$maintenance\_cost)

apt$total\_taxes =gsub("\\,", "", apt$total\_taxes)

apt$common\_charges =gsub("\\,", "", apt$common\_charges)

apt$sale\_price <- as.numeric(apt$sale\_price)

apt$maintenance\_cost <- as.numeric(apt$maintenance\_cost)

apt$total\_taxes <- as.numeric(apt$total\_taxes)

apt$common\_charges <- as.numeric(apt$common\_charges)

apt$sale\_price[is.na(apt$sale\_price)] <- mean(apt$sale\_price, na.rm = T)

apt$maintenance\_cost[is.na(apt$maintenance\_cost)] <- mean(apt$maintenance\_cost, na.rm = T)

apt$total\_taxes[is.na(apt$total\_taxes)] <- mean(apt$total\_taxes, na.rm = T)

apt$common\_charges[is.na(apt$common\_charges)] <- mean(apt$common\_charges, na.rm = T)

skim(apt)

view(apt)

#Regression Tree

if (!pacman::p\_isinstalled(YARF)){

pacman::p\_install\_gh("kapelner/YARF/YARFJARs", ref = "dev")

pacman::p\_install\_gh("kapelner/YARF/YARF", ref = "dev", force = TRUE)

}

options(java.parameters = "-Xmx4000m")

pacman::p\_load(YARF)

test\_prop = 0.2

train\_indices = sample(1 : nrow(apt), round((1 - test\_prop) \* nrow(apt)))

apt\_train = apt[train\_indices, ]

y\_train = apt\_train$sale\_price

X\_train = apt\_train

X\_train$medv = NULL

n\_train = nrow(X\_train)

tree\_mod = YARFCART(X\_train, y\_train, calculate\_oob\_error = TRUE)

get\_tree\_num\_nodes\_leaves\_max\_depths(tree\_mod)

nrow(apt\_train) / get\_tree\_num\_nodes\_leaves\_max\_depths(tree\_mod)$num\_leaves

illustrate\_trees(tree\_mod, max\_depth = 5, length\_in\_px\_per\_half\_split = 30, open\_file = TRUE)

#Linear Model

pacman::p\_load(ggplot2)

simple\_linear\_model = lm(sale\_price ~ num\_bedrooms + num\_half\_bathrooms + num\_full\_bathrooms + maintenance\_cost + total\_taxes + sq\_footage + cats\_allowed + dogs\_allowed + coop\_condo + common\_charges, data = apt)

coef(simple\_linear\_model)

summary(simple\_linear\_model)$r.squared

summary(simple\_linear\_model)$sigma

simple\_linear\_model1 = lm(sale\_price ~ num\_bedrooms + num\_half\_bathrooms + num\_full\_bathrooms + maintenance\_cost + total\_taxes + sq\_footage + cats\_allowed + dogs\_allowed + coop\_condo + common\_charges + num\_floors\_in\_building, data = apt)

coef(simple\_linear\_model1)

summary(simple\_linear\_model1)$r.squared

summary(simple\_linear\_model1)$sigma

simple\_linear\_model2 = lm(sale\_price ~ num\_bedrooms + num\_half\_bathrooms + num\_full\_bathrooms + maintenance\_cost + total\_taxes + sq\_footage + cats\_allowed + dogs\_allowed + coop\_condo + common\_charges + num\_floors\_in\_building + fuel\_type, data = apt)

coef(simple\_linear\_model2)

summary(simple\_linear\_model2)$r.squared

summary(simple\_linear\_model2)$sigma

#Random Forest

num\_trees = 500

train\_size = 2000

training\_indices = sample(1 : nrow(apt), train\_size)

apt\_train = apt[training\_indices, ]

y\_train = apt\_train$sale\_price

X\_train = apt\_train

X\_train$sale\_price = NULL

mod\_bag = YARFBAG(X\_train, y\_train, num\_trees = num\_trees, calculate\_oob\_error = TRUE)

mod\_rf = YARF(X\_train, y\_train, num\_trees = num\_trees, calculate\_oob\_error = TRUE)

tree\_mod = YARFCART(X\_train, y\_train)

y\_hat\_train = predict(tree\_mod, X\_train)

y\_hat\_test = predict(tree\_mod, X\_test)

table(y\_train, y\_hat\_train)

oos\_confusion\_table = table(y\_test, y\_hat\_test)

oos\_confusion\_table

test\_indices = sample(setdiff(1 : nrow(apt), training\_indices), 100)

apt\_test = apt[test\_indices, ]

y\_test = apt\_test$sale\_price

X\_test = apt\_test

X\_test$sale\_price = NULL

y\_hat\_test\_bag = predict(mod\_bag, X\_test)

y\_hat\_test\_rf = predict(mod\_rf, X\_test)

oos\_conf\_table\_bag = table(y\_test, y\_hat\_test\_bag)

oos\_conf\_table\_rf = table(y\_test, y\_hat\_test\_rf)

oos\_conf\_table\_bag

oos\_conf\_table\_rf

miscl\_err\_bag = mean(y\_test != y\_hat\_test\_bag)

miscl\_err\_rf = mean(y\_test != y\_hat\_test\_rf)

miscl\_err\_bag

miscl\_err\_rf

cat("gain: ", (miscl\_err\_bag - miscl\_err\_rf) / miscl\_err\_bag \* 100, "%\n")

#Note

ggplot(apt, aes(x = sale\_price, colour = num\_bedrooms))+

geom\_freqpoly(binwidth = 50)