Burhan A Hanif

Collaborators Vincent Miceli, Adriana Sham, Sakib Salim, Juan Diego Astudillo

title: “Term Project 390.4- 2019” output: word_document: default pdf_document: default Author: Juan D Astudillo, Vincent Miceli, Adriana Sham, Burhan Hanif, Sakib Salim —

R Markdown

```
pacman::p_load(dplyr, tidyr, ggplot2, magrittr, stringr, mlr)
housing_data = read.csv("housing_data_2016_2017.csv")
```

##Delete variables that we dont need

```
housing_data %<>%
  select(-c(HITId, HITTypeId, Title, Description, Keywords, Reward, CreationTime, MaxAssignments,  RequesterAnnotation,  AssignmentDurationInSeconds, AutoApprovalDelayInSeconds, Expiration, NumberOfSimilarHITs, LifetimeInSeconds, AssignmentId,  WorkerId,  AssignmentStatus,  AcceptTime, SubmitTime, AutoApprovalTime,  ApprovalTime,  RejectionTime,  RequesterFeedback,  WorkTimeInSeconds, LifetimeApprovalRate,  Last30DaysApprovalRate, Last7DaysApprovalRate, URL, url, date_of_sale))
```

Clean Data

```
housing_data %<>%
  mutate( zip_code = str_extract(full_address_or_zip_code, "[0-9]{5}"))
housing_data %<>%
  mutate(dogs_allowed = ifelse(substr(housing_data$dogs_allowed, 1, 3) == "yes", 1, 0)) %>%
  mutate(cats_allowed = ifelse(substr(housing_data$cats_allowed, 1, 3) == "yes", 1, 0)) %>%
  mutate( pets_allowed = ifelse( cats_allowed + dogs_allowed > 0, 1, 0)) %>%
  mutate(coop_condo = factor(tolower(coop_condo)))
housing_data %<>%
  select(-c(dogs_allowed,cats_allowed, fuel_type))
d = housing_data
d %<>%
  mutate(maintenance_cost = sjmisc::rec(maintenance_cost, rec = "NA = 0 ; else = copy")) %<>%
  mutate(common_charges = sjmisc::rec(common_charges, rec = "NA = 0 ; else = copy"))##recode from NA to 0.
# combine maintaince cost and common charges
d %<>%
  mutate( monthly_cost = common_charges + maintenance_cost)
d %<>%
  mutate(monthly_cost = sjmisc::rec(monthly_cost, rec = "0 = NA ; else = copy"))
## Garage exists conver it to binary
d %<>%
  mutate(garage_exists = sjmisc::rec(garage_exists, rec = "NA = 0 ; else = co
```

```
py")) ##recode from NA to 0.
d %<>%
  mutate(garage_exists = sjmisc::rec(garage_exists, rec = " eys = 1; UG = 1 ;
Underground = 1; yes = 1 ; Yes = 1 ; else = copy")) ##recode from NA to 0.
d %<>%
  select(-c(maintenance_cost , common_charges, model_type))
str(d)
```

```
## 'data.frame':    2230 obs. of  24 variables:
##  $ approx_year_built            : int  1955 1955 2004 2002 1949 1938 1950
1960 1960 2005 ...
##  $ community_district_num       : int  25 25 24 25 26 28 29 28 25 30 ...
##  $ coop_condo                   : Factor w/ 2 levels "co-op","condo": 1 1
2 2 1 1 1 1 1 2 ...
##  $ dining_room_type             : Factor w/ 5 levels "combo","dining area"
,..: 1 3 1 1 1 1 1 NA NA 5 ...
##  $ full_address_or_zip_code     : Factor w/ 1176 levels " Bayside NY, 1136
0",..: 1158 562 24 223 497 121 391 941 415 586 ...
##  $ garage_exists                : Factor w/ 2 levels "0","1": 1 1 1 1 1 1
1 1 1 1 ...
##  $ kitchen_type                 : Factor w/ 4 levels "combo","eat in",..:
2 2 3 2 2 2 3 3 2 2 ...
##  $ num_bedrooms                 : int  2 1 1 3 2 2 1 0 1 1 ...
##  $ num_floors_in_building       : int  6 7 1 NA 2 6 NA 2 NA 4 ...
##  $ num_full_bathrooms           : int  1 1 1 2 1 1 1 1 1 1 ...
##  $ num_half_bathrooms           : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ num_total_rooms              : int  5 4 3 5 4 4 3 2 4 3 ...
##  $ parking_charges              : Factor w/ 90 levels " NA ","100","105",.
.: 1 1 1 1 1 1 1 1 41 1 ...
##  $ pct_tax_deductibl            : int  NA NA NA NA 39 NA NA NA NA NA ...
##  $ sale_price                   : Factor w/ 316 levels " NA ","100000",..:
107 113 33 252 119 126 38 8 94 250 ...
##  $ sq_footage                   : int  NA 890 550 NA 675 1000 NA 375 NA 68
1 ...
##  $ total_taxes                  : Factor w/ 294 levels " NA ","100","1024"
,..: 1 1 255 68 1 1 1 1 1 19 ...
##  $ walk_score                   : int  82 89 90 94 71 90 72 93 70 98 ...
##  $ listing_price_to_nearest_1000: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ lat                          : num  40.7 40.8 40.7 40.8 40.7 ...
##  $ lon                          : num  -73.8 -73.8 -73.9 -73.8 -73.7 ...
##  $ zip_code                     : chr  "11355" "11354" "11368" "11354" ...
##  $ pets_allowed                 : num  0 0 0 0 1 1 0 0 0 0 ...
##  $ monthly_cost                 : num  767 604 167 275 660 932 660 514 781
NA ...
```

##Change variable type

```
d %<>%
  mutate( dining_room_type = as.factor(dining_room_type)) %>%
  mutate(garage_exists = as.character(garage_exists)) %>%
```

```
  mutate(garage_exists = as.numeric(garage_exists)) %>%
  mutate( parking_charges = as.character(parking_charges)) %>%
  mutate( parking_charges = as.numeric(parking_charges)) %>%
  mutate(sale_price = as.character(sale_price)) %>%
  mutate(sale_price = as.numeric(sale_price)) %>%
  mutate(total_taxes = as.character(total_taxes)) %>%
  mutate(total_taxes = as.numeric(total_taxes)) %>%
  mutate(price_persqft = listing_price_to_nearest_1000 / sq_footage)
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

#Added latitude and longitude features using ggmap

```
#Already run and included in the data
#pacman::p_load(ggmap)
#d %<>%
#  mutate(lat = geocode(full_address_or_zip_code)$lat, lon = #geocode(full_ad
dress_or_zip_code)$lon )
#geocoordinates for relevant LIRR stations
lirr_coord = coord
```

```
## Error in eval(expr, envir, enclos): object 'coord' not found
```

```
RAD_EARTH = 3958.8
degrees_to_radians = function(angle_degrees){
  for(i in 1:length(angle_degrees))
    angle_degrees[i] = angle_degrees[i]*pi/180
  return(angle_degrees)
}
compute_globe_distance = function(destination, origin){
  destination_rad = degrees_to_radians(destination)
  origin_rad = degrees_to_radians(origin)
  delta_lat = destination_rad[1] - origin_rad[1]
  delta_lon = destination_rad[2] - origin_rad[2]
  h = (sin(delta_lat/2))^2 + cos(origin_rad[1]) * cos(destination_rad[1]) * (
sin(delta_lon/2))^2
  central_angle = 2 * asin(sqrt(h))
  return(RAD_EARTH * central_angle)
}
#find the closest LIRR station and compute distance
shortest_lirr_distance = function(all_lirr_coords, house_coords){
  shortest_dist = Inf
  for (i in 1: nrow(all_lirr_coords)){
    ith_lirr = c(all_lirr_coords$lat[i], all_lirr_coords$lon[i])
    new_dist = compute_globe_distance(ith_lirr, house_coords)
    if( new_dist < shortest_dist){
      shortest_dist = new_dist
```

```
    }
  }
  return(shortest_dist)
}
d %<>%
  rowwise() %>%
  mutate(shortest_dist = shortest_lirr_distance(lirr_coord, c(lat, lon)) )

## Error in nrow(all_lirr_coords): object 'lirr_coord' not found

#makes any other addresses redundant
d %<>%
  select(-c(zip_code, full_address_or_zip_code, listing_price_to_nearest_1000
))
```

We are trying to predict `sale_price`. So let's section our dataset:

```
####CREATE A COLUMN ID
d %<>%
  ungroup(d) %>%
  mutate(id = 1 : 2230)
d %<>%
  mutate(total_taxes = ifelse(d$total_taxes < 1000, NA, total_taxes))
real_y = data.frame(d$id, d$sale_price)
real_d = subset(d, (!is.na(d$sale_price)))
fake_d = subset(d, (is.na(d$sale_price)))
real_d$sale_price = NULL
fake_d$sale_price = NULL
```

#Split the data that has y into train and test sets

```
train_indices = sample(1 : nrow(real_d), nrow(real_d)*4/5)
training_data = real_d[train_indices, ]
testing_data = real_d[-train_indices, ]
X = rbind(training_data, testing_data, fake_d)
```

#Let's first create a matrix with p columns that represents missingness

```
M = tbl_df(apply(is.na(X), 2, as.numeric))
colnames(M) = paste("is_missing_", colnames(X), sep = "")
```

#Some of these missing indicators are collinear because they share all the rows they are missing on. Let's filter those out:

```
M = tbl_df(t(unique(t(M))))
```

#Some featuers did not have missingness so let's remove them:

```
M %<>% select_if(function(x){sum(x) > 0})
```

Now let's impute using the package. we cannot fit RF models to the entire dataset (it's 26,000! observations) so we will sample 5 for X1 and for each of the trees and then average. That will be good enough.

```
pacman::p_load(missForest)
Ximp = missForest(data.frame(X), sampsize = rep(172, ncol(X)))$ximp
```

```
##   missForest iteration 1 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?
```

```
## done!
##   missForest iteration 2 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?
```

```
## done!
##   missForest iteration 3 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?
```

```
## done!
##   missForest iteration 4 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?
```

```
## done!
##   missForest iteration 5 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?
```

```
## done!
##   missForest iteration 6 in progress...
```

```
## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?
```

```
## done!
##   missForest iteration 7 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?

## done!
##   missForest iteration 8 in progress...

## Warning in randomForest.default(x = obsX, y = obsY, ntree = ntree, mtry =
## mtry, : The response has five or fewer unique values. Are you sure you wan
t
## to do regression?

## done!
```

```
Ximp %<>%
  arrange(id)
Xnew = data.frame(cbind(Ximp, M, real_y))
Xnew %<>%
  mutate(price = d.sale_price) %>%
  select(-c(id, d.id, d.sale_price))

linear_mod_impute_and_missing_dummies = lm(price ~ ., data = Xnew)
summary(linear_mod_impute_and_missing_dummies)
```

```
##
## Call:
## lm(formula = price ~ ., data = Xnew)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -332100  -38713    -528   39033  335196
##
## Coefficients: (3 not defined because of singularities)
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -4.239e+07  9.620e+06  -4.407 1.29e-05
## approx_year_built                -2.058e+02  2.619e+02  -0.786 0.432354
## community_district_num            3.651e+03  1.191e+03   3.067 0.002281
## coop_condocondo                   1.790e+05  1.692e+04  10.581  < 2e-16
## dining_room_typedining area       2.195e+04  5.444e+04   0.403 0.686959
## dining_room_typeformal            2.459e+04  8.835e+03   2.783 0.005592
## dining_room_typeother             1.970e+04  1.154e+04   1.708 0.088308
## garage_exists                     9.918e+03  9.308e+03   1.066 0.287128
## kitchen_typeeat in               -7.176e+03  1.053e+04  -0.682 0.495763
## kitchen_typeefficiency           -2.602e+04  1.023e+04  -2.545 0.011235
## num_bedrooms                      4.510e+04  8.170e+03   5.521 5.47e-08
## num_floors_in_building            3.409e+03  7.319e+02   4.658 4.11e-06
## num_full_bathrooms                3.638e+04  5.520e+04   0.659 0.510198
```

```
## num_half_bathrooms                  5.999e+03  2.705e+04   0.222 0.824579
## num_total_rooms                     1.957e+04  5.411e+03   3.617 0.000329
## parking_charges                     3.521e+02  1.007e+02   3.497 0.000512
## pct_tax_deductibl                  -1.364e+02  1.044e+03  -0.131 0.896075
## sq_footage                          2.696e+01  1.329e+01   2.029 0.043005
## total_taxes                        -2.290e+00  6.383e+00  -0.359 0.719860
## walk_score                         -5.691e+02  3.546e+02  -1.605 0.109111
## lat                                 6.560e+05  1.425e+05   4.602 5.32e-06
## lon                                -2.131e+05  8.739e+04  -2.438 0.015121
## pets_allowed                        1.480e+04  7.134e+03   2.075 0.038514
## monthly_cost                        1.340e+02  1.453e+01   9.221  < 2e-16
## price_persqft                       4.211e+05  6.843e+04   6.153 1.57e-09
## is_missing_approx_year_built       -5.540e+04  3.502e+04  -1.582 0.114289
## is_missing_community_district_num  -1.727e+05  7.735e+04  -2.233 0.026030
## is_missing_dining_room_type         1.177e+04  8.035e+03   1.465 0.143684
## is_missing_kitchen_type             2.762e+04  2.973e+04   0.929 0.353315
## is_missing_num_bedrooms                    NA         NA      NA       NA
## is_missing_num_floors_in_building   8.964e+01  8.686e+03   0.010 0.991770
## is_missing_num_half_bathrooms       5.708e+03  1.479e+04   0.386 0.699705
## is_missing_num_total_rooms                 NA         NA      NA       NA
## is_missing_parking_charges         -7.411e+03  8.050e+03  -0.921 0.357701
## is_missing_pct_tax_deductibl       -8.716e+03  8.935e+03  -0.976 0.329786
## is_missing_sq_footage              -1.730e+02  6.971e+03  -0.025 0.980206
## is_missing_total_taxes             -1.056e+03  9.450e+03  -0.112 0.911089
## is_missing_monthly_cost             6.488e+03  2.076e+04   0.312 0.754823
## is_missing_price_persqft                   NA         NA      NA       NA
##
## (Intercept)                          ***
## approx_year_built
## community_district_num               **
## coop_condocondo                      ***
## dining_room_typedining area
## dining_room_typeformal               **
## dining_room_typeother                .
## garage_exists
## kitchen_typeeat in
## kitchen_typeefficiency               *
## num_bedrooms                         ***
## num_floors_in_building               ***
## num_full_bathrooms
## num_half_bathrooms
## num_total_rooms                      ***
## parking_charges                      ***
## pct_tax_deductibl
## sq_footage                           *
## total_taxes
## walk_score
## lat                                  ***
## lon                                  *
## pets_allowed                         *
```

```
## monthly_cost                          ***
## price_persqft                         ***
## is_missing_approx_year_built
## is_missing_community_district_num *
## is_missing_dining_room_type
## is_missing_kitchen_type
## is_missing_num_bedrooms
## is_missing_num_floors_in_building
## is_missing_num_half_bathrooms
## is_missing_num_total_rooms
## is_missing_parking_charges
## is_missing_pct_tax_deductibl
## is_missing_sq_footage
## is_missing_total_taxes
## is_missing_monthly_cost
## is_missing_price_persqft
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74950 on 492 degrees of freedom
##   (1702 observations deleted due to missingness)
## Multiple R-squared:  0.8373, Adjusted R-squared:  0.8257
## F-statistic: 72.32 on 35 and 492 DF,  p-value: < 2.2e-16
```

REMOVING MISSING Y SECTION

```
Data = Xnew
### sale price is our imputed Y
Y = Data$price
Data %<>%
  filter(!is.na(price)) %>%
  select(-price)
Xtrain = Data[1:422, ]
Xtest = Data[423:528, ]
Ytrain = Y[1:422]
Ytest = Y[423:528]
dtrain = cbind(Xtrain, Ytrain) ## combine x train with y train, x test with y
test
dtest = cbind(Xtest, Ytest)
```

Dropping colinear features

```
Xtrain %<>%
  select(-c(is_missing_num_total_rooms, is_missing_num_bedrooms, is_missing_p
rice_persqft))
```

Linear Regression

```
linear = lm(Ytrain ~ ., data = Xtrain)## simple linear model
summary(linear)
```

```
##
## Call:
## lm(formula = Ytrain ~ ., data = Xtrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -343443  -34486    1798   35988  322090
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -4.258e+07  1.072e+07  -3.974 8.44e-05
## approx_year_built              -1.846e+02  2.835e+02  -0.651 0.515460
## community_district_num          3.273e+03  1.257e+03   2.604 0.009569
## coop_condocondo                 2.122e+05  1.912e+04  11.100  < 2e-16
## dining_room_typedining area     2.452e+04  5.335e+04   0.460 0.646033
## dining_room_typeformal          3.038e+04  9.790e+03   3.103 0.002057
## dining_room_typeother           1.632e+04  1.265e+04   1.289 0.198023
## garage_exists                   1.059e+04  1.067e+04   0.993 0.321527
## kitchen_typeeat in             -3.539e+01  1.161e+04  -0.003 0.997570
## kitchen_typeefficiency         -2.447e+04  1.124e+04  -2.177 0.030078
## num_bedrooms                    3.686e+04  9.047e+03   4.074 5.61e-05
## num_floors_in_building          3.117e+03  8.129e+02   3.835 0.000147
## num_full_bathrooms              2.812e+04  5.442e+04   0.517 0.605584
## num_half_bathrooms              6.014e+03  2.934e+04   0.205 0.837715
## num_total_rooms                 1.941e+04  6.012e+03   3.228 0.001353
## parking_charges                 4.459e+02  1.068e+02   4.175 3.68e-05
## pct_tax_deductibl              -1.559e+02  1.351e+03  -0.115 0.908236
## sq_footage                      2.630e+01  1.389e+01   1.893 0.059087
## total_taxes                    -3.057e+00  7.469e+00  -0.409 0.682571
## walk_score                     -5.029e+02  3.890e+02  -1.293 0.196843
## lat                             6.821e+05  1.540e+05   4.428 1.24e-05
## lon                            -2.012e+05  9.838e+04  -2.045 0.041525
## pets_allowed                    1.012e+04  7.938e+03   1.275 0.203162
## monthly_cost                    1.589e+02  1.873e+01   8.480 4.81e-16
## price_persqft                   3.043e+05  7.808e+04   3.897 0.000115
## is_missing_approx_year_built   -5.270e+04  3.457e+04  -1.524 0.128235
## is_missing_community_district_num         NA         NA      NA       NA
## is_missing_dining_room_type     5.254e+03  8.669e+03   0.606 0.544828
## is_missing_kitchen_type         3.872e+04  3.173e+04   1.220 0.223058
## is_missing_num_floors_in_building  5.339e+03  9.584e+03   0.557 0.577768
## is_missing_num_half_bathrooms   1.278e+04  1.577e+04   0.811 0.418073
## is_missing_parking_charges     -8.300e+03  8.710e+03  -0.953 0.341185
## is_missing_pct_tax_deductibl   -1.088e+04  9.420e+03  -1.155 0.248873
## is_missing_sq_footage          -8.243e+02  7.656e+03  -0.108 0.914321
## is_missing_total_taxes          2.840e+03  1.029e+04   0.276 0.782694
## is_missing_monthly_cost         1.060e+04  2.282e+04   0.465 0.642492
##
## (Intercept)                    ***
## approx_year_built
## community_district_num          **
```

```
## coop_condocondo                      ***
## dining_room_typedining area
## dining_room_typeformal               **
## dining_room_typeother
## garage_exists
## kitchen_typeeat in
## kitchen_typeefficiency               *
## num_bedrooms                         ***
## num_floors_in_building               ***
## num_full_bathrooms
## num_half_bathrooms
## num_total_rooms                      **
## parking_charges                      ***
## pct_tax_deductibl
## sq_footage                           .
## total_taxes
## walk_score
## lat                                  ***
## lon                                  *
## pets_allowed
## monthly_cost                         ***
## price_persqft                        ***
## is_missing_approx_year_built
## is_missing_community_district_num
## is_missing_dining_room_type
## is_missing_kitchen_type
## is_missing_num_floors_in_building
## is_missing_num_half_bathrooms
## is_missing_parking_charges
## is_missing_pct_tax_deductibl
## is_missing_sq_footage
## is_missing_total_taxes
## is_missing_monthly_cost
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73070 on 387 degrees of freedom
## Multiple R-squared:  0.8431, Adjusted R-squared:  0.8293
## F-statistic: 61.16 on 34 and 387 DF,  p-value: < 2.2e-16
```

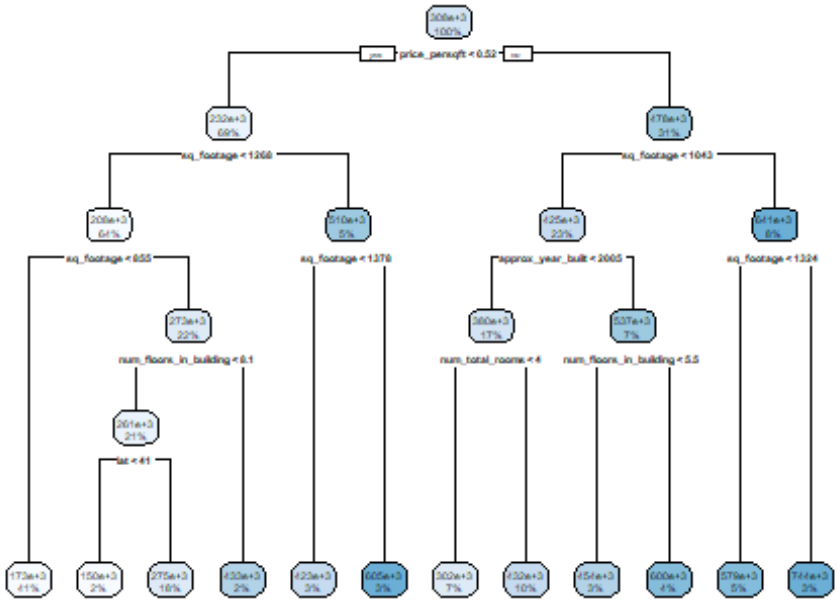
```
yhat = predict(linear, Xtest)
```

```
## Warning in predict.lm(linear, Xtest): prediction from a rank-deficient fit
## may be misleading
```

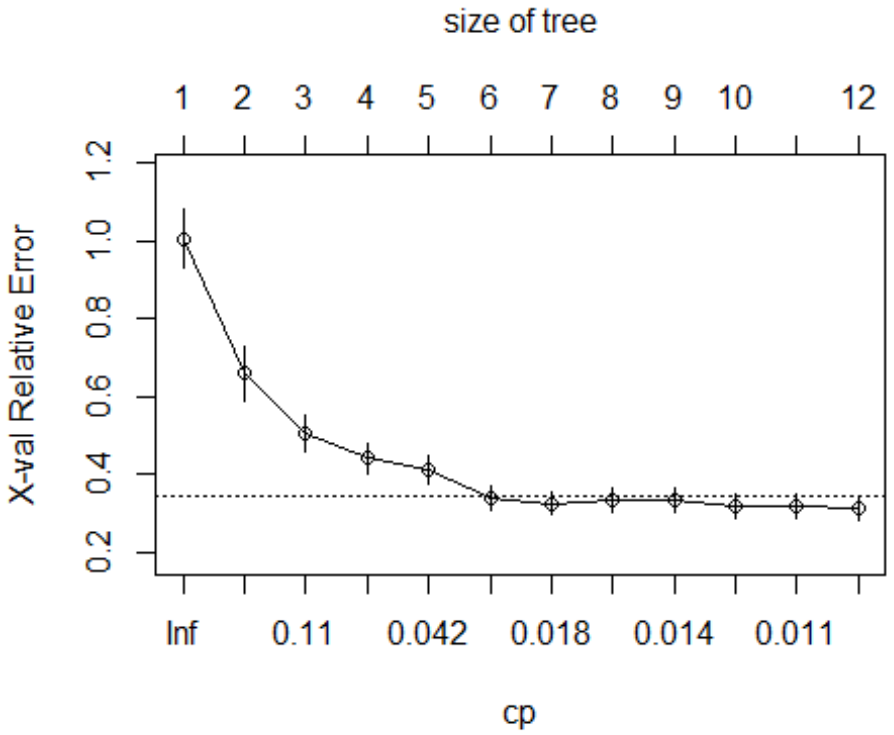
```
e = yhat - Ytest
sqrt(sum(e^2) / nrow(Xtest))
```

```
## [1] 87255.3
```

```r
#REGRESSION TREE
pacman::p_load(rsample)#data spliting
pacman::p_load(rpart) #performing reg tree
pacman::p_load(rpart.plot) #ploting reg tree
pacman::p_load(ipred) #bagging
pacman::p_load(caret) #bagging
m1 = rpart(
  formula = Ytrain ~ .,
  data    = Xtrain,
  method  = "anova"
  )
rpart.plot(m1)
```




```r
plotcp(m1)
```




```
summary(m1)
```

```
## Call:
## rpart(formula = Ytrain ~ ., data = Xtrain, method = "anova")
##   n= 422
##
##            CP nsplit rel error    xerror       xstd
## 1  0.41393401      0 1.0000000 1.0039469 0.07571490
## 2  0.14647965      1 0.5860660 0.6592117 0.06921072
## 3  0.08579395      2 0.4395863 0.5056281 0.04612952
## 4  0.04682539      3 0.3537924 0.4412382 0.03862777
## 5  0.03751948      4 0.3069670 0.4126056 0.03692895
## 6  0.02169174      5 0.2694475 0.3374549 0.03105146
## 7  0.01547974      6 0.2477558 0.3254956 0.02903562
## 8  0.01461144      7 0.2322760 0.3331287 0.03111290
## 9  0.01434865      8 0.2176646 0.3320980 0.03103616
## 10 0.01116708      9 0.2033159 0.3190301 0.03111571
## 11 0.01039146     10 0.1921489 0.3163690 0.03100559
## 12 0.01000000     11 0.1817574 0.3137754 0.03021857
##
## Variable importance
##          price_persqft             sq_footage      approx_year_built
##                     19                     14                     12
##           monthly_cost             coop_condo           total_taxes
##                     12                     12                     10
##        parking_charges        num_total_rooms           num_bedrooms
##                      8                      5                      3
```

```
##     num_half_bathrooms num_floors_in_building       dining_room_type
##                      2                       1                      1
##
## Node number 1: 422 observations,    complexity param=0.413934
##   mean=308191.7, MSE=3.121006e+10
##   left son=2 (291 obs) right son=3 (131 obs)
##   Primary splits:
##       price_persqft     < 0.5247497 to the left,  improve=0.4139340, (0 mi
ssing)
##       coop_condo        splits as  LR, improve=0.3754617, (0 missing)
##       approx_year_built < 1970.5    to the left,  improve=0.3463094, (0 mi
ssing)
##       total_taxes       < 3977.52   to the left,  improve=0.2924978, (0 mi
ssing)
##       sq_footage        < 853.97    to the left,  improve=0.2878910, (0 mi
ssing)
##   Surrogate splits:
##       coop_condo        splits as  LR, agree=0.874, adj=0.595, (0 split)
##       approx_year_built < 1970.5    to the left,  agree=0.865, adj=0.565,
(0 split)
##       parking_charges   < 141.1692  to the left,  agree=0.813, adj=0.397,
(0 split)
##       monthly_cost      < 408.5     to the right, agree=0.813, adj=0.397,
(0 split)
##       total_taxes       < 4058.812  to the left,  agree=0.773, adj=0.267,
(0 split)
##
## Node number 2: 291 observations,    complexity param=0.1464796
##   mean=231930.8, MSE=1.479049e+10
##   left son=4 (268 obs) right son=5 (23 obs)
##   Primary splits:
##       sq_footage      < 1267.97   to the left,  improve=0.4482381, (0 miss
ing)
##       num_total_rooms < 4.5       to the left,  improve=0.3504908, (0 miss
ing)
##       monthly_cost    < 1019      to the left,  improve=0.3183507, (0 miss
ing)
##       num_bedrooms    < 1.5       to the left,  improve=0.2807438, (0 miss
ing)
##       total_taxes     < 4050.542  to the left,  improve=0.2368172, (0 miss
ing)
##   Surrogate splits:
##       total_taxes       < 4217.965  to the left,  agree=0.962, adj=0.522,
(0 split)
##       num_total_rooms   < 6.5       to the left,  agree=0.945, adj=0.304,
(0 split)
##       monthly_cost      < 1461.5    to the left,  agree=0.945, adj=0.304,
(0 split)
##       coop_condo        splits as  LR, agree=0.931, adj=0.130, (0 split)
##       approx_year_built < 1979.5    to the left,  agree=0.928, adj=0.087,
```

```
(0 split)
##
## Node number 3: 131 observations,    complexity param=0.08579395
##   mean=477595.7, MSE=2.606744e+10
##   left son=6 (99 obs) right son=7 (32 obs)
##   Primary splits:
##       sq_footage      < 1043.37   to the left,  improve=0.3308979, (0 miss
ing)
##       num_total_rooms < 4.5       to the left,  improve=0.2905401, (0 miss
ing)
##       num_bedrooms    < 1.5       to the left,  improve=0.2309136, (0 miss
ing)
##       total_taxes     < 2947.48   to the left,  improve=0.1727705, (0 miss
ing)
##       monthly_cost    < 1555      to the left,  improve=0.1588476, (0 miss
ing)
##   Surrogate splits:
##       num_bedrooms     < 2.5       to the left,  agree=0.817, adj=0.250, (
0 split)
##       monthly_cost     < 1006.5    to the left,  agree=0.817, adj=0.250, (
0 split)
##       total_taxes      < 4765.825  to the left,  agree=0.809, adj=0.219, (
0 split)
##       dining_room_type splits as  L-R-L, agree=0.802, adj=0.187, (0 split)
##       num_total_rooms  < 5.5       to the left,  agree=0.802, adj=0.187, (
0 split)
##
## Node number 4: 268 observations,    complexity param=0.04682539
##   mean=208077.8, MSE=6.9692e+09
##   left son=8 (174 obs) right son=9 (94 obs)
##   Primary splits:
##       sq_footage      < 854.63    to the left,  improve=0.3301953, (0 miss
ing)
##       num_bedrooms    < 1.5       to the left,  improve=0.2735225, (0 miss
ing)
##       monthly_cost    < 966.72    to the left,  improve=0.2516223, (0 miss
ing)
##       num_total_rooms < 4.5       to the left,  improve=0.2416782, (0 miss
ing)
##       total_taxes     < 2304.195  to the left,  improve=0.2017775, (0 miss
ing)
##   Surrogate splits:
##       num_bedrooms       < 1.5       to the left,  agree=0.877, adj=0.649,
(0 split)
##       num_half_bathrooms < 0.975     to the left,  agree=0.851, adj=0.574,
(0 split)
##       num_total_rooms    < 4.5       to the left,  agree=0.851, adj=0.574,
(0 split)
##       monthly_cost       < 761.5     to the left,  agree=0.817, adj=0.479,
(0 split)
```

```
##       total_taxes        < 2441.115  to the left,  agree=0.709, adj=0.170,
(0 split)
##
## Node number 5: 23 observations,    complexity param=0.01434865
##   mean=509869.6, MSE=2.204592e+10
##   left son=10 (12 obs) right son=11 (11 obs)
##   Primary splits:
##       sq_footage                   < 1378.438  to the left,  improve=0.372
7023, (0 missing)
##       is_missing_pct_tax_deductibl < 0.5       to the right, improve=0.310
3049, (0 missing)
##       price_persqft                < 0.4478212 to the left,  improve=0.232
1375, (0 missing)
##       num_bedrooms                 < 2.5       to the left,  improve=0.211
5748, (0 missing)
##       monthly_cost                 < 1439      to the left,  improve=0.210
4842, (0 missing)
##   Surrogate splits:
##       monthly_cost       < 1439      to the left,  agree=0.826, adj=0.636,
(0 split)
##       num_bedrooms       < 2.5       to the left,  agree=0.739, adj=0.455,
(0 split)
##       num_half_bathrooms < 1.005     to the right, agree=0.739, adj=0.455,
(0 split)
##       num_total_rooms    < 6.5       to the left,  agree=0.739, adj=0.455,
(0 split)
##       total_taxes        < 4363.26   to the left,  agree=0.739, adj=0.455,
(0 split)
##
## Node number 6: 99 observations,    complexity param=0.03751948
##   mean=424793.3, MSE=1.854507e+10
##   left son=12 (71 obs) right son=13 (28 obs)
##   Primary splits:
##       approx_year_built < 2004.5    to the left,  improve=0.2691537, (0 mi
ssing)
##       coop_condo        splits as  LR, improve=0.2334085, (0 missing)
##       price_persqft     < 0.65656   to the left,  improve=0.2127935, (0 mi
ssing)
##       pct_tax_deductibl < 48.405    to the right, improve=0.1892664, (0 mi
ssing)
##       num_total_rooms   < 3.5       to the left,  improve=0.1505040, (0 mi
ssing)
##   Surrogate splits:
##       price_persqft   < 0.6855313 to the left,  agree=0.818, adj=0.357, (0
split)
##       total_taxes     < 4401.84   to the left,  agree=0.798, adj=0.286, (0
split)
##       lon             < -73.93462 to the right, agree=0.768, adj=0.179, (0
split)
##       parking_charges < 173.1725  to the left,  agree=0.747, adj=0.107, (0
```

```
split)
##       sq_footage      < 541       to the right, agree=0.747, adj=0.107, (0
split)
##
## Node number 7: 32 observations,    complexity param=0.01547974
##   mean=640953.1, MSE=1.402847e+10
##   left son=14 (20 obs) right son=15 (12 obs)
##   Primary splits:
##       sq_footage             < 1323.5    to the left,  improve=0.4541617,
(0 missing)
##       monthly_cost           < 1517.5    to the left,  improve=0.2305505,
(0 missing)
##       num_floors_in_building < 13.375    to the left,  improve=0.1985273,
(0 missing)
##       num_bedrooms           < 2.5       to the left,  improve=0.1901020,
(0 missing)
##       kitchen_type           splits as  LRL-, improve=0.1898224, (0 missin
g)
##   Surrogate splits:
##       monthly_cost           < 816       to the left,  agree=0.844, adj=0.
583, (0 split)
##       num_floors_in_building < 13.375    to the left,  agree=0.812, adj=0.
500, (0 split)
##       total_taxes            < 4483.285  to the left,  agree=0.812, adj=0.
500, (0 split)
##       coop_condo             splits as  RL, agree=0.750, adj=0.333, (0 spl
it)
##       num_half_bathrooms     < 0.945     to the right, agree=0.719, adj=0.
250, (0 split)
##
## Node number 8: 174 observations
##   mean=172819.1, MSE=2.745539e+09
##
## Node number 9: 94 observations,    complexity param=0.01461144
##   mean=273343.9, MSE=8.226607e+09
##   left son=18 (87 obs) right son=19 (7 obs)
##   Primary splits:
##       num_floors_in_building < 8.105     to the left,  improve=0.2488580,
(0 missing)
##       parking_charges        < 87.44     to the left,  improve=0.2384759,
(0 missing)
##       lat                    < 40.69952  to the left,  improve=0.2220240,
(0 missing)
##       price_persqft          < 0.4380646 to the left,  improve=0.2098772,
(0 missing)
##       monthly_cost           < 1026      to the left,  improve=0.1666749,
(0 missing)
##
## Node number 10: 12 observations
##   mean=423083.3, MSE=9.583535e+09
```

```
##
## Node number 11: 11 observations
##   mean=604545.5, MSE=1.846116e+10
##
## Node number 12: 71 observations,    complexity param=0.02169174
##   mean=380425.9, MSE=1.428335e+10
##   left son=24 (28 obs) right son=25 (43 obs)
##   Primary splits:
##       num_total_rooms   < 3.5       to the left,  improve=0.2817169, (0 mi
ssing)
##       sq_footage        < 677.2617  to the left,  improve=0.1917659, (0 mi
ssing)
##       lon               < -73.83396 to the left,  improve=0.1858177, (0 mi
ssing)
##       pct_tax_deductibl < 48.405    to the right, improve=0.1748105, (0 mi
ssing)
##       total_taxes       < 2417.442  to the left,  improve=0.1684632, (0 mi
ssing)
##   Surrogate splits:
##       sq_footage         < 794.195   to the left,  agree=0.845, adj=0.607,
(0 split)
##       num_bedrooms       < 1.5       to the left,  agree=0.817, adj=0.536,
(0 split)
##       parking_charges    < 144.68    to the right, agree=0.732, adj=0.321,
(0 split)
##       num_half_bathrooms < 0.835     to the left,  agree=0.704, adj=0.250,
(0 split)
##       walk_score         < 96.5      to the right, agree=0.704, adj=0.250,
(0 split)
##
## Node number 13: 28 observations,    complexity param=0.01116708
##   mean=537296.4, MSE=1.170314e+10
##   left son=26 (12 obs) right son=27 (16 obs)
##   Primary splits:
##       num_floors_in_building < 5.5       to the left,  improve=0.4488346,
(0 missing)
##       parking_charges        < 188.32    to the left,  improve=0.3987746,
(0 missing)
##       community_district_num < 29        to the left,  improve=0.3027133,
(0 missing)
##       monthly_cost           < 459       to the left,  improve=0.2343286,
(0 missing)
##       lon                    < -73.89867 to the right, improve=0.1869287,
(0 missing)
##   Surrogate splits:
##       approx_year_built < 2007.5    to the left,  agree=0.75, adj=0.417, (
0 split)
##       parking_charges   < 141.1275  to the left,  agree=0.75, adj=0.417, (
0 split)
##       pct_tax_deductibl < 40.92667  to the right, agree=0.75, adj=0.417, (
```

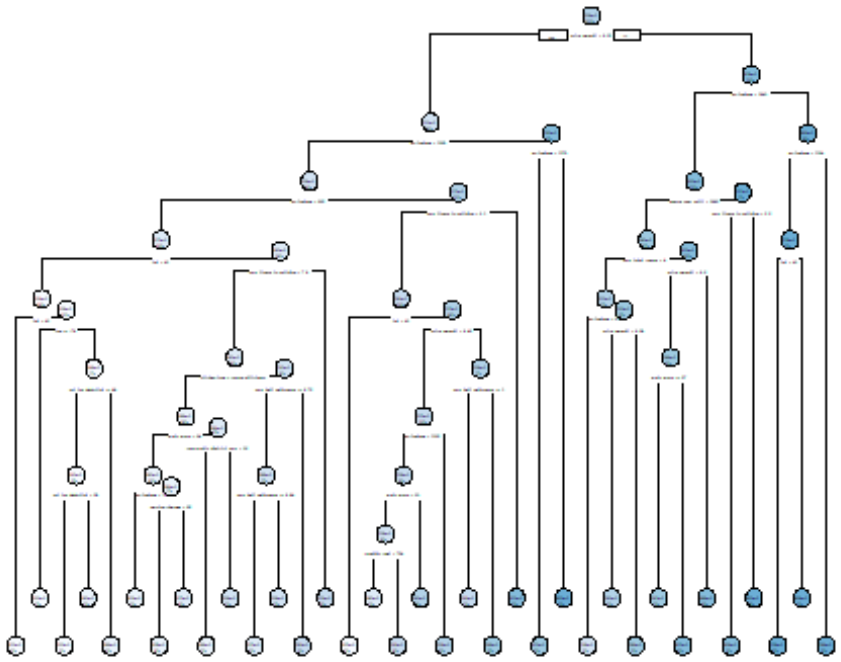
```
0 split)
##       total_taxes       < 3486.505  to the left,  agree=0.75, adj=0.417, (
0 split)
##       monthly_cost      < 304.5     to the left,  agree=0.75, adj=0.417, (
0 split)
##
## Node number 14: 20 observations
##   mean=579125, MSE=3.170647e+09
##
## Node number 15: 12 observations
##   mean=744000, MSE=1.5135e+10
##
## Node number 18: 87 observations,    complexity param=0.01039146
##   mean=260509.5, MSE=6.47784e+09
##   left son=36 (10 obs) right son=37 (77 obs)
##   Primary splits:
##       lat               < 40.66729  to the left,  improve=0.24284780, (0 m
issing)
##       price_persqft     < 0.3895313 to the left,  improve=0.17111720, (0 m
issing)
##       parking_charges   < 80.9625   to the left,  improve=0.16514750, (0 m
issing)
##       walk_score        < 91.5      to the left,  improve=0.11611160, (0 m
issing)
##       pct_tax_deductibl < 50.085    to the right, improve=0.09714846, (0 m
issing)
##   Surrogate splits:
##       price_persqft < 0.3444474 to the left,  agree=0.92, adj=0.3, (0 spli
t)
##
## Node number 19: 7 observations
##   mean=432857.1, MSE=2.469551e+09
##
## Node number 24: 28 observations
##   mean=301816.1, MSE=9.284512e+09
##
## Node number 25: 43 observations
##   mean=431613.7, MSE=1.089436e+10
##
## Node number 26: 12 observations
##   mean=453608.3, MSE=5.001271e+09
##
## Node number 27: 16 observations
##   mean=600062.5, MSE=7.537184e+09
##
## Node number 36: 10 observations
##   mean=150450, MSE=1.048723e+09
##
## Node number 37: 77 observations
##   mean=274802.9, MSE=5.405488e+09
```

```r
yhat = predict(m1, Xtest)
e = yhat - Ytest
sqrt(sum(e^2)/106)
```

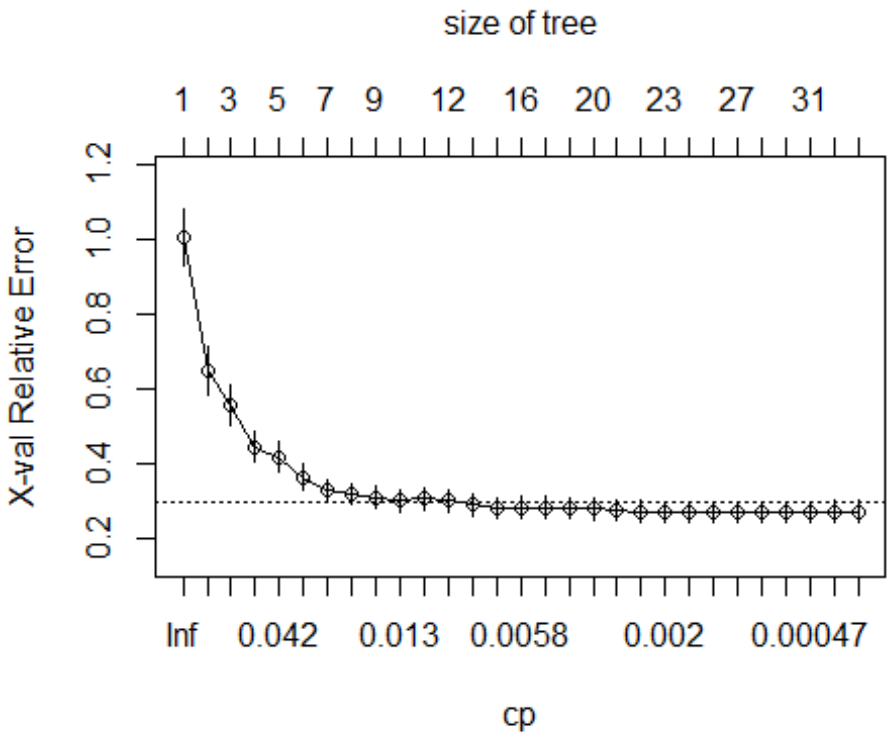
```
## [1] 112773.6
```

```r
m2 <- rpart(
    formula = Ytrain ~ .,
    data    = Xtrain,
    method  = "anova",
    control = list(cp = 0, xval = 10)
)
rpart.plot(m2)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



```r
plotcp(m2)
```




```
yhat = predict(m2, Xtest)
e = yhat - Ytest
sqrt(sum(e^2)/106)

## [1] 107881.6

jpeg(file = "save_m2.jpeg")

###Tuning
m3 <- rpart(
    formula = Ytrain ~ .,
    data    = Xtrain,
    method  = "anova",
    control = list(minsplit = 10, maxdepth = 12, xval = 10)
)
yhat = predict(m3, Xtest)
e = yhat - Ytest
sqrt(sum(e^2)/106)

## [1] 112773.6

m3$cptable

##           CP nsplit rel error    xerror       xstd
## 1  0.41393401      0 1.0000000 1.0030228 0.07576166
## 2  0.14647965      1 0.5860660 0.6524574 0.06438383
## 3  0.08579395      2 0.4395863 0.6227751 0.06254779
## 4  0.04682539      3 0.3537924 0.4442755 0.03962382
```

```
## 5  0.03751948      4 0.3069670 0.4103436 0.03871336
## 6  0.02169174      5 0.2694475 0.3690193 0.03412299
## 7  0.01547974      6 0.2477558 0.3102251 0.02733550
## 8  0.01461144      7 0.2322760 0.3127359 0.02831154
## 9  0.01434865      8 0.2176646 0.3092347 0.02825127
## 10 0.01116708      9 0.2033159 0.3050691 0.02798997
## 11 0.01039146     10 0.1921489 0.3084012 0.02861245
## 12 0.01000000     11 0.1817574 0.3084021 0.02861635
```

```
# function to get optimal cp
get_cp <- function(x) {
  min    <- which.min(x$cptable[, "xerror"])
  cp <- x$cptable[min, "CP"]
}
# function to get minimum error
get_min_error <- function(x) {
  min    <- which.min(x$cptable[, "xerror"])
  xerror <- x$cptable[min, "xerror"]
}

optimal_tree <- rpart(
    formula = Ytrain ~ .,
    data    = Xtrain,
    method  = "anova",
    control = list(minsplit = 11, maxdepth = 8, cp = 0.01)
    )
pred <- predict(optimal_tree, newdata = Xtrain)
RMSE(pred = pred, obs = Ytrain)
```

```
## [1] 75317.07
```

##RANDOM FORESTS

```
m1 <- randomForest(
  formula = Ytrain ~ .,
  data    = Xtrain
)
m1
```

```
##
## Call:
##  randomForest(formula = Ytrain ~ ., data = Xtrain)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 5071097324
##                     % Var explained: 83.75
```

```
which.min(m1$mse)
```

```
## [1] 305
```

```r
# RMSE of this optimal random forest
sqrt(m1$mse[which.min(m1$mse)])
```

```
## [1] 71179.25
```

```r
features <- setdiff(names(Xtrain), Ytrain)
set.seed(1989)
m2 <- tuneRF(
  x          = Xtrain,
  y          = Ytrain,
  ntreeTry   = 500,
  mtryStart  = 5,
  stepFactor = 1.5,
  improve    = 0.01,
  trace      = FALSE      # to not show real-time progress
)
```

```
## -0.03972194 0.01
## 0.04282308 0.01
## 0.005418261 0.01
```