FIN550 Group 6

Big Data Group Project: Property Assessment Case

Shammy Ho, Hyun Ji Lee, Ruilin Ni, Vishwas Rao, Zhijie Jin, Akshaya Suresh, Purva Vaswani

Preprocessing

- Dropped variables that were missing 50% or more data
  - Removed the following columns specifically : "char_apts", "char_tp_dsgn", "char_attic_fnsh", "char_porch" from the historic property data
- Dropped variables that are not predictors according to the codebook
  - Cleaned the variables which were not being used to predict anything in the model such as "geo_fips", "geo_municipality", "geo_property_city", "geo_property_zip" from the historic property data and others from the predict and historic property data
- Removed negative housing prices

Preprocessing

- Winsorized data to limit the effect of outliers
  - Used a function to replace extreme values that fell outside 1-99 percentile
- Replaced missing values with non-missing values in the same location group
  - For categorical variables, the missing values were replaced with the mode
  - For numerical variables, the missing values were replaced with the mean

Selecting Predictors – Preventing Multicollinearity

- Dropped categorical variables temporarily in order to generate correlation matrix
- Checked variable pairs having >0.75 correlation
- Dropped variables "char_beds","char_fbath" as they are relatively less important predictors
- Added back categorical variables

```
Correlation between meta_certified_est_bldg and sale_price : 0.8225603
Correlation between sale_price and meta_certified_est_bldg : 0.8225603
Correlation between char_beds and char_rooms : 0.9003237
Correlation between char_fbath and char_rooms : 0.7577657
Correlation between char_bldg_sf and char_rooms : 0.7856339
Correlation between char_rooms and char_beds : 0.9003237
Correlation between char_rooms and char_fbath : 0.7577657
Correlation between char_bldg_sf and char_fbath : 0.809186
Correlation between char_rooms and char_bldg_sf : 0.7856339
Correlation between char_fbath and char_bldg_sf : 0.809186
```

Selecting Predictors - Challenges

Exhaustive Search

- Got errors -> 'Warning: 19 linear dependencies found. Reordering variables and trying again.'

Backward Elimination, Forward Selection, and Stepwise Regression

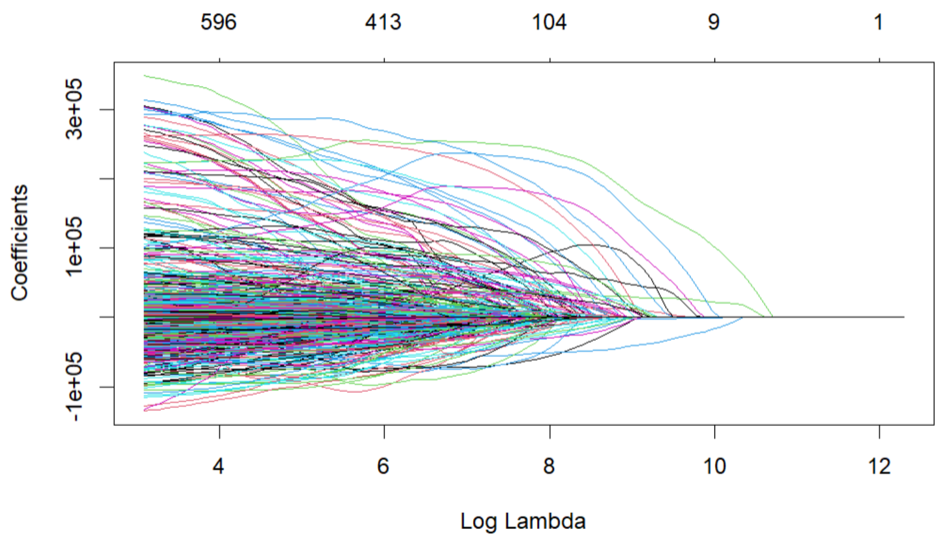
- Took too long to execute -> cause: too many dummy variables

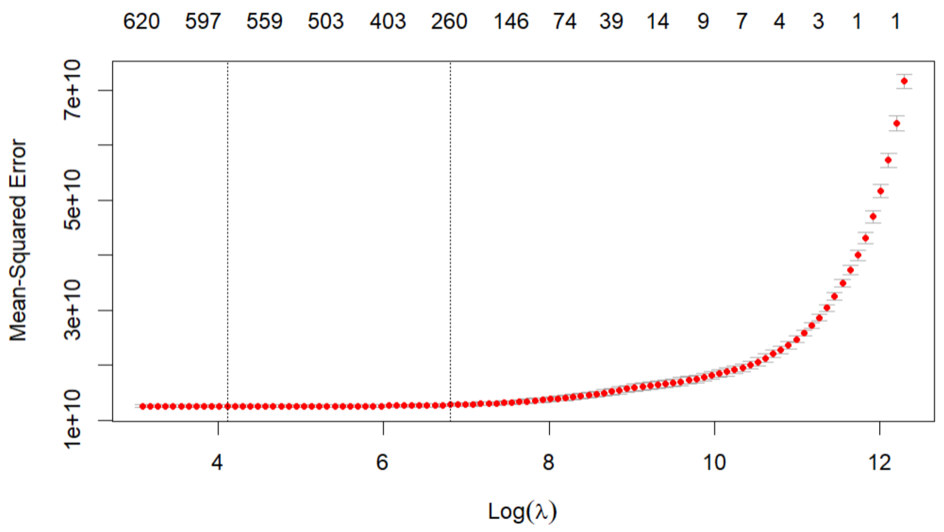
Hence, we used **Lasso Regression** to select the predictors

Selecting Predictors - Lasso Regression

MSE

```
mean((y.test - pred.lambda.best)^2)
```
```

[1] 11423335058
```





Fitting Predictive Algorithms - Preparation

- Selected predictors from Lasso Regression
- Converted categorical and logical variables into factors
- Removed "geo_other_perc" from models that were unable to run due to linear correlation between ethnic population percentages
  - $ geo_white_perc + $ geo_black_perc + $ geo_asian_perc + $ geo_his_perc + $ geo_other_perc = 1
- Dropped "geo_school_elem_district" and "geo_school_hs_district" despite its high significance due to hardware limitations
  - $ geo_school_elem_district: Factor w/ 474 levels
  - $ geo_school_hs_district: Factor w/ 79 levels

Fitting Predictive Algorithms - Preparation

```
tibble [50,000 × 36] (S3: tbl_df/tbl/data.frame)
 $ sale_price              : num [1:50000] 291000 1035000 235000 280500 369000 ...
 $ meta_certified_est_bldg : num [1:50000] 276700 602590 116690 207290 250800 ...
 $ meta_certified_est_land : num [1:50000] 35880 239580 44500 42610 48050 ...
 $ char_hd_sf              : num [1:50000] 6525 38801 4945 11364 3844 ...
 $ char_age                : num [1:50000] 37 65 65 46 32 11 60 96 26 67 ...
 $ char_ext_wall           : Factor w/ 4 levels "1","2","3","4": 3 3 2 1 3 1 3 2 2 2 ...
 $ char_roof_cnst          : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ char_rooms              : num [1:50000] 8 8 7 6 7 9 6 4 9 7 ...
 $ char_frpl               : num [1:50000] 1 3 1 0 0 1 0 0 1 0 ...
 $ char_attic_type         : Factor w/ 3 levels "1","2","3": 3 2 1 3 3 1 3 2 3 3 ...
 $ char_hbath              : num [1:50000] 1 1 1 1 1 1 0 1 1 1 ...
 $ char_tp_plan            : Factor w/ 2 levels "1","2": 2 1 2 2 2 2 2 2 2 2 ...
 $ char_bldg_sf            : num [1:50000] 2480 3666 1794 1251 1724 ...
 $ char_use                : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ char_type_resd          : Factor w/ 9 levels "1","2","3","4",..: 2 1 5 4 2 2 1 1 2 2 ...
 $ geo_white_perc          : num [1:50000] 0.4373 0.865 0.0165 0.6379 0.6439 ...
 $ geo_black_perc          : num [1:50000] 0.26532 0 0.96186 0.00371 0 ...
 $ geo_asian_perc          : num [1:50000] 0.0419 0.125 0 0.1579 0.0827 ...
 $ geo_his_perc            : num [1:50000] 0.24592 0.00731 0.01809 0.15864 0.24825 ...
 $ geo_withinmr100         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ geo_withinmr101300      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
 $ econ_tax_rate           : num [1:50000] 9.24 8.08 6.8 10.86 6.79 ...
 $ econ_midincome          : num [1:50000] 76042 140789 50426 107174 81014 ...
 $ ind_garage              : Factor w/ 2 levels "FALSE","TRUE": 2 2 2 2 1 2 2 2 2 2 ...
 $ ind_arms_length         : logi [1:50000] TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ geo_school_elem_district : Factor w/ 474 levels "ADDAMS","AGASSIZ",..: 399 377 171 415 167 72 414 9 73 403 ...
 $ geo_school_hs_district  : Factor w/ 79 levels "AMUNDSEN HS",..: 60 54 7 58 74 46 56 33 19 4 ...
 $ meta_town_code          : Factor w/ 38 levels "10","11","12",..: 22 16 31 26 32 10 13 31 20 2 ...
 $ basement_combined       : chr [1:50000] "1_3" "2_3" "1_3" "3_1" ...
 $ climate_control         : chr [1:50000] "1_5_1" "1_5_1" "1_5_2" "1_5_2" ...
 $ char_gar1_size          : Factor w/ 8 levels "1","2","3","4",..: 3 3 3 3 7 3 5 3 5 3 ...
 $ char_gar1_cnst          : Factor w/ 4 levels "1","2","3","4": 2 2 1 1 1 1 1 1 2 1 ...
 $ char_gar1_att           : Factor w/ 2 levels "1","2": 1 1 2 1 2 2 2 2 1 2 ...
 $ geo_floodplain          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ geo_fs_flood_factor     : num [1:50000] 1 1 3 1 1 1 6 1 1 1 ...
 $ geo_fs_flood_risk_direction: num [1:50000] 0 0 1 0 0 0 1 0 0 0 ...
```

Fitting Predictive Algorithms - Linear Regression

```
columns_to_convert <- intersect(columns_to_convert, names(data))
columns_to_convert <- intersect(columns_to_convert, columns_to_keep)

# Convert specified columns to factors
data[columns_to_convert] <- lapply(data[columns_to_convert], as.factor)
str(data)
```

```
##Data Partition
# set seed for reproducing the partition
set.seed(1)

# row numbers of the training set
train.index <- sample(c(1:dim(data)[1]), dim(data)[1]*0.6)
head(train.index)
```

```
# training set
train.df <- data[train.index, ]
head(train.df)
```

```
# test set
test.df <- data[-train.index, ]
head(test.df)
```

MSE: 11,423,649,406

```
lm <- lm(sale_price ~ ., data = train.df)
summary_lm <- summary(lm)
print(summary_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ ., data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2651638   -50599    -2807    45261  1018447
##
## Coefficients: (24 not defined because of singularities)
##                                                     Estimate
## (Intercept)                                       -8.940e+04
## meta_certified_est_bldg                            2.660e-01
## meta_certified_est_land                            3.863e-01
## char_hd_sf                                         1.296e+00
## char_age                                          -4.580e+02
## char_ext_wall2                                    -8.801e+03
```

```
predictions <- predict(lm, test.df, type="response")
```

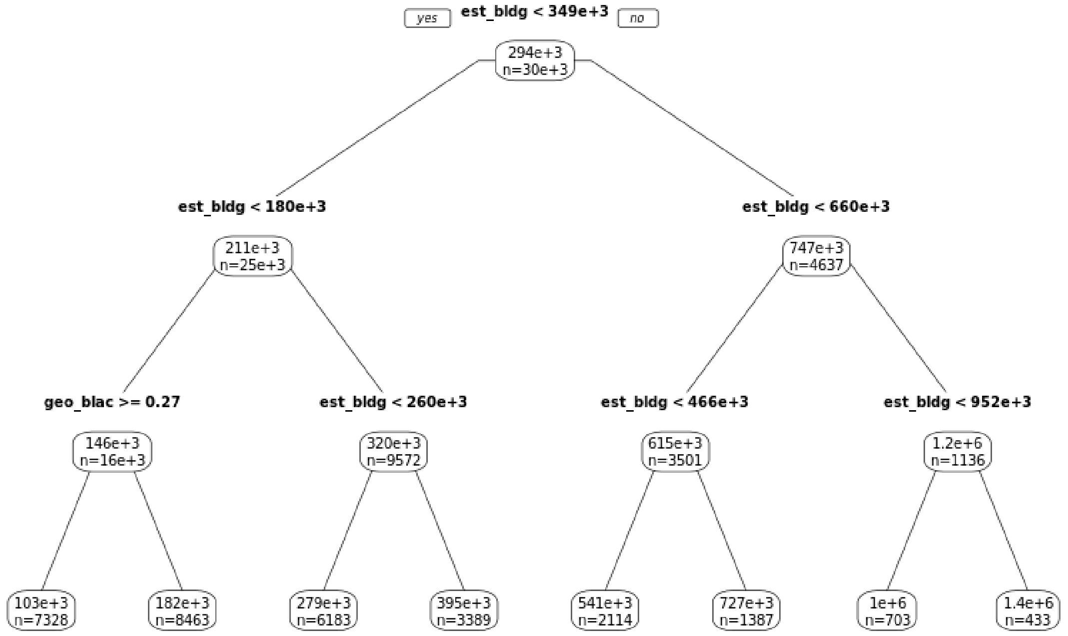
```
head(predictions)
```

```
##        1        2        3        4        5        6
## 944304.3 167001.0 522701.1 114734.4 114366.5 199682.8
```

```
mean((test.df$sale_price - predictions)^2)
```

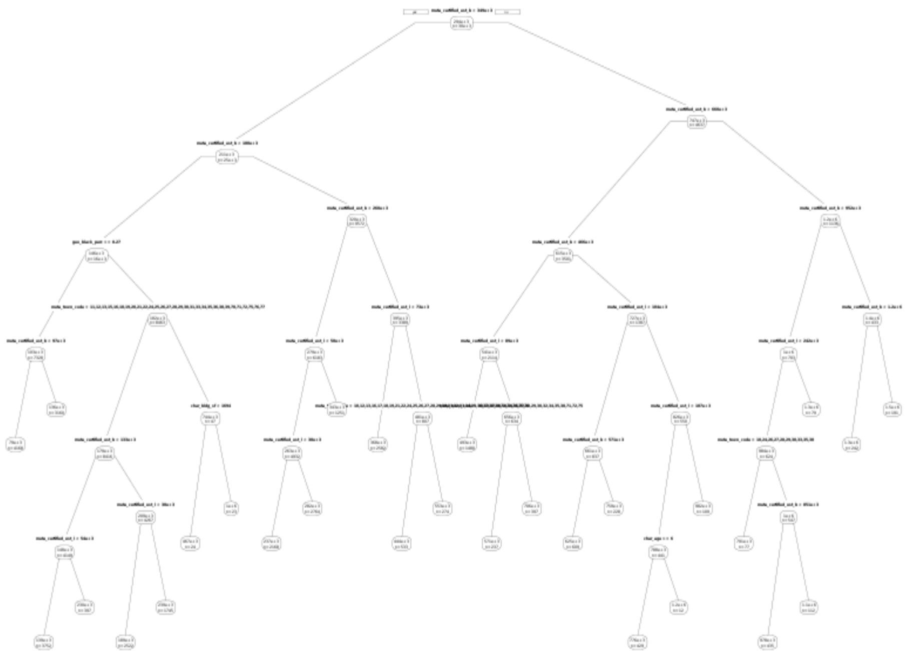
```
## [1] 11423649406
```

Fitting Predictive Algorithms – Regression Tree



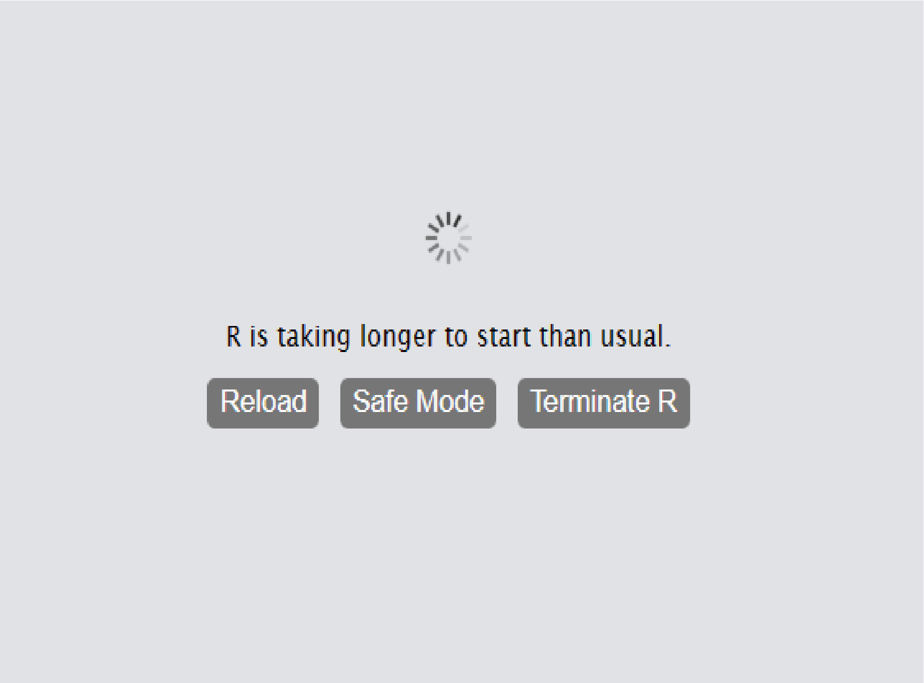
- Data partition: set.seed(70)
- rt <- rpart(sale_price ~ ., data = train.df, method = "anova", cp = 0.01)
- Variables used in decision tree:
  - est_bldg (Assessor Certified Estimate (Building))
  - geo_black_perc (Tract Percent Pop. Black)
- MSE: 15,503,818,679

Fitting Predictive Algorithms - Pruned Regression Tree



- Data partition: set.seed(70)
- set.seed(7)
  cv.rt <- rpart(sale_price ~ ., data = train.df, method = "anova", cp = 0.001, xval = 5)
- Variables used in decision tree:
  - char_age (Age)
  - char_bldg_sf (Building Square Feet)
  - est_bldg (Assessor Certified Estimate (Building))
  - est_land (Assessor Certified Estimate (Land))
  - geo_black_perc (Tract Percent Pop. Black)
  - meta_town_code (Township Code)
- MSE: 12,504,983,754

Fitting Predictive Algorithms - Bagging & Boosting



- Multiple attempts were made:
  - sale_price → factor by quantiles of 5
  - Top 10 highest coefficient variables from Lasso Regression (excluding school districts)
  - Changing instance type to t2.medium
  - sale_price → factor by quantiles of 4
  - Only one other variable was run with sale_price_bin
- Each attempt was 30 min. ↑, with most 1hr. ↑ and the last 10 hr. ↑

Fitting Predictive Algorithms – Random Forest

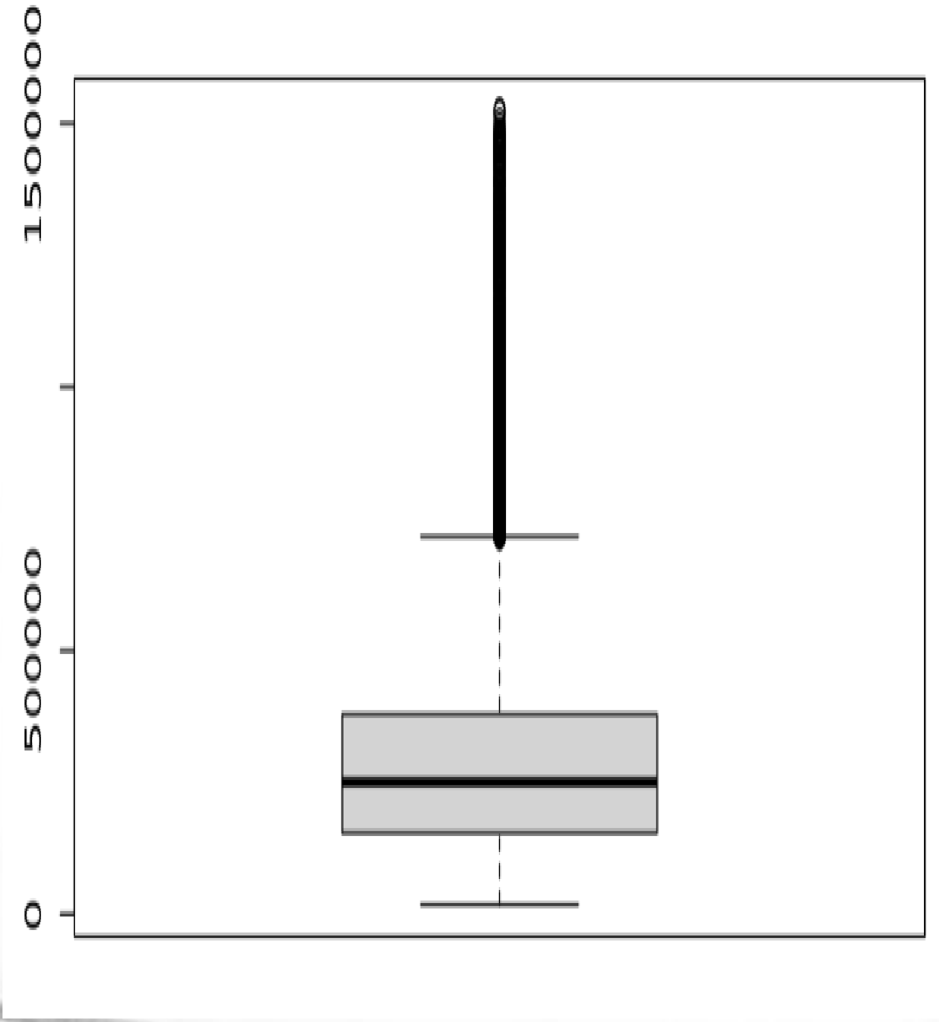
```
Call:
 randomForest(formula = sale_price ~ ., data = train.df, mtry = 4)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 9293780067
                    % Var explained: 87.11
```

```{r}
# Make predictions on the test data
predictions <- predict(rf, newdata = test.df)

# Calculate the Mean Squared Error
mse <- mean((test.df$sale_price - predictions)^2)

# Print the MSE
print(mse)
```

```
[1] 8716355447
```

- Random Forest model explained 87% of the variability
- Model generated the least MSE among all the models we tried to fit on the data
- MSE = 8,716,355,447
- Parameters used in the code:
  - Response variable = '*sale_price*'
  - mtry = 4
- Created Property Assessment Prediction CSV file with PID and assessed_value columns

Final Result – Assessed Values!




```
assessed_value
Min.   :  18320
1st Qu.: 154312
Median : 250671
Mean   : 315411
3rd Qu.: 379366
Max.   :1524386
```

Work Done by Members

- Shammy Ho – Converting variables, modeling Bagging + Boosting + Regression Tree, consolidating R Script
- Hyun Ji Lee – Using Variable Selection methods to select predictors, analyzing assessed values of property prices
- Ruilin Ni – Modeling Linear Regression using variables selected by Lasso Regression
- Vishwas Rao – Preprocessing data, removing non predictor variables, cleaning data
- Zhijie Jin – Running Lasso Regression to select predictors, generating Assessment File
- Akshaya Suresh – Preventing multicollinearity before selecting predictors, consolidating Executive Summary, consolidating presentation slides
- Purva Vaswani – Modeling Random Forest, using the model for prediction, consolidating R Script, consolidating presentation slides