Random Forest

House Price Prediction in Sindian, China Taiwan

BY YUQING WANG, WENFEI YAN, JINGLIN ZHENG, JUSTIN ZHONG

Model Explanation – Context

Decision Trees

- Powerful machine learning tool for classification and prediction.
- Flow chart structure
  - Node: Attribute
  - Branch: Outcome
  - Leaf: Class label
- Ensemble Methods
  - Combination of multiple generated models to increase result accuracy.
  - Constructs set of classifiers from training data.
  - Able to predict class labels for missing values
  - Ensembles: Combination of decision trees .
    - Generate multiple data sets of the original data → Build multiple classifiers → Combine the classifiers.
    - Bagging
    - Boosting

Model Explanation – Context

Bagging

- Sampling with replacement.
- Bootstrapping data
  - Random selection of sample from original data.
  - Build classifier on each sample.
    - $p = (1 - n)^n$ of being selected.
- Random Forest
  - Construct multiple decision trees at training for classification and regression.
  - Create bootstrap datasets.
  - Create decision trees for bootstrap datasets.
  - New samples with missing values: Classify by running through random forest.
  - Evaluation of random forest: Run out-of-bag samples through different decision trees.

Model Explanation – Random Forest

Advantages/Disadvantages

- Advantages
  - Efficiently run through large datasets.
  - High accuracy.
  - Effectively estimate missing data with high accuracy when encountering large proportions of missing data.
- Disadvantages
  - May result overfitting.
  - Bias may be present when levels between each tree differ.

Key Assumptions

- Dataset have no formal distribution assumption.
- Inputted data contain multiple variables.
- No missing values(clean data if missing value exist).

Model Explanation – Random Forest

Training Random Forest

- Random forest consists of multiple decision trees. Alike decision trees, random forest behaves in the same manner with minimal difference.
- Split the data into training and test data.
- Random forest works with the training set(bootstrap sample from the original data).
- At each node of each training tree, *m* of *p* features are selected randomly and used as training candidates.
  - Classification: $m = sqrt(p)$
  - Regression: $m = p/3$
- To test the training error, out-of-bag data points are used.
- Analysis of out of bag data can contribute to variable weighing.

Model Explanation – Random Forest

Key Algorithm in Random Forest

$$\hat{H}_{\text{final}} = \sum_{k=1}^{\#\text{ models}} \alpha_k \hat{h}_k, \quad \alpha = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{\#\text{ data}} \| \sum_{k=1}^{\#\text{ models}} \alpha_k \hat{h}_k(x_i) - y_i \|_2^2$$

Where:

$\widehat{h_k}$ = ensemble of trees, where each ensemble is trained on a random subset of data and features

$$\alpha = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{\#\text{ data}} \| \sum_{k=1}^{\#\text{ models}} \alpha_k \hat{h}_k(x_i) - y_i \|_2^2$$

Random Subspace Method: ensemble method attempting to reduce correlation between bootstrap data instead of entire data.

Model Explanation – Random Forest

Information Gain

- Determine the variable that presents the most information about a class.
- Based on entropy(larger entropy, larger uncertainty).
- $H(x) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$

Gini Index

- Measures the degree of probability of a variable being incorrectly classified.
- $GINI(t) = 1 - \sum_j [p(j \mid t)]^2$
  - Where $p(j \mid t)$ is the relative frequency of class j at node t.

Project Outline

1. Explore the summary of the dataset.
2. Plot(boxplot) to check for normality and outliers.
3. Define outliers and perform capping.
4. Convert format of independent variable(transaction date).
5. Visualize number of transactions associated with the converted independent variable(transaction date → transaction year, transaction month, and transaction date).
6. Apply clustering to data base on latitude and longitude.
7. Display correlation matrix.
8. Split the data.
9. Perform simple linear regression and multiple linear regression.
10. Determine parameters for random forest model.
11. Process random forest model.
12. Examine contributions of independent variables.
13. Conclusion.

Introduction of Data

Real Estate Valuation Data Set

- Founded from “UCI Machine Learning Repository”.

Purpose

- Predict the housing prices in Sindian, Tapei via regression model based on market historical record.

Problem Statement

- To identify the variables affecting house prices, for example, house age, distance to the nearest MRT station, number of convenience store, etc.
- To create a random forest model that quantitatively associates house prices with independent variables.
- To compute the accuracy of the model, for example, how well these variables can predict house prices.
- To compare the performance of random forest model with the linear model and multiple regression model.

Libraries

MASS

- Analyze and visualize dataset.

tidyverse

- Transform and better present data.

ggplot2

- Create complex plot from data from data frame.

caret

- Fit large amounts of model.

magrittr

- Enable the use of the command “*%>%*”.

glmnet

- Fits generalized linear models by penalized maximum likelihood.

randomForest

- Generate random forest model.

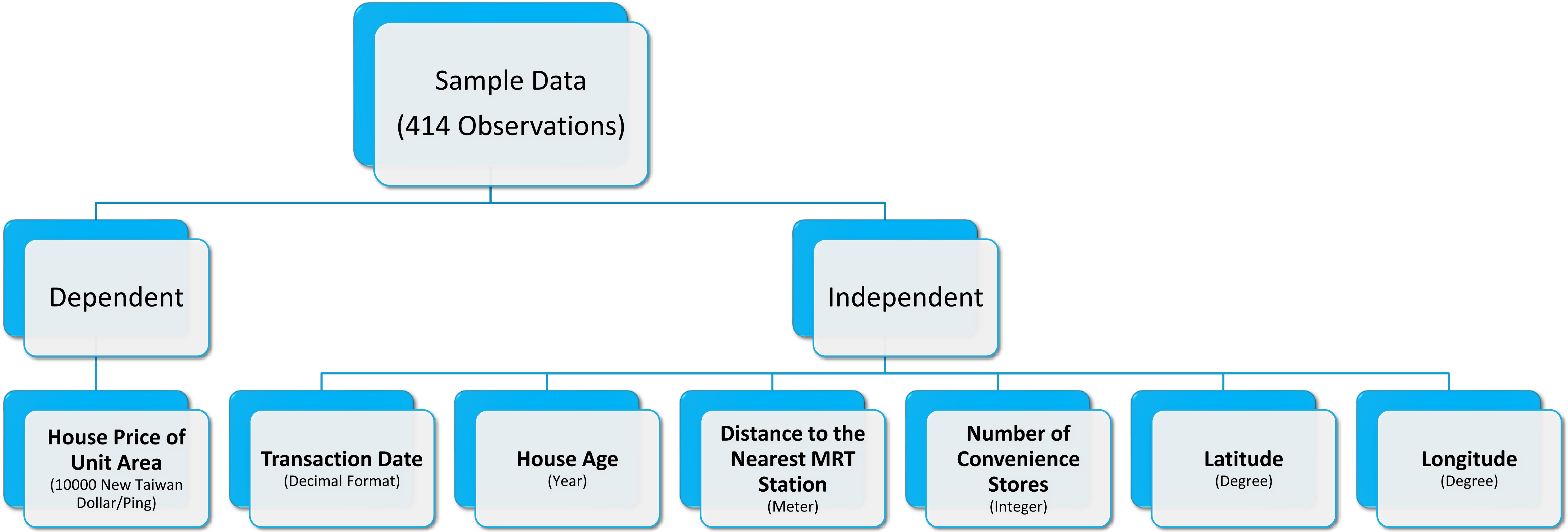
fpc

- computes several cluster validity statistics.

lubridate

- Convert date format.

Data Exploration



Data Exploration

Display of First *n* Rows of Data

	No	X1.transaction.date	X2.house.age	X3.distance.to.the.nearest.MRT.station
	<int>	<dbl>	<dbl>	<dbl>
1	1	2012.917	32.0	84.87882
2	2	2012.917	19.5	306.59470
3	3	2013.583	13.3	561.98450
4	4	2013.500	13.3	561.98450
5	5	2012.833	5.0	390.56840
6	6	2012.667	7.1	2175.03000

X4.number.of.convenience.stores	X5.latitude	X6.longitude	Y.house.price.of.unit.area
<int>	<dbl>	<dbl>	<dbl>
10	24.98298	121.5402	37.9
9	24.98034	121.5395	42.2
5	24.98746	121.5439	47.3
5	24.98746	121.5439	54.8
5	24.97937	121.5425	43.1
3	24.96305	121.5125	32.1

Integers

Floats

Features of Data

- Integers
- Floats

Data Exploration

Resulted Summaries of Data

```
       No        X1.transaction.date  X2.house.age
 Min.   :  1.0   Min.   :2013         Min.   : 0.000
 1st Qu.:104.2   1st Qu.:2013         1st Qu.: 9.025
 Median :207.5   Median :2013         Median :16.100
 Mean   :207.5   Mean   :2013         Mean   :17.713
 3rd Qu.:310.8   3rd Qu.:2013         3rd Qu.:28.150
 Max.   :414.0   Max.   :2014         Max.   :43.800

 X3.distance.to.the.nearest.MRT.station X4.number.of.convenience.stores
 Min.   :  23.38                        Min.   : 0.000
 1st Qu.: 289.32                        1st Qu.: 1.000
 Median : 492.23                        Median : 4.000
 Mean   :1083.89                        Mean   : 4.094
 3rd Qu.:1454.28                        3rd Qu.: 6.000
 Max.   :6488.02                        Max.   :10.000

  X5.latitude     X6.longitude    Y.house.price.of.unit.area
 Min.   :24.93   Min.   :121.5   Min.   :  7.60
 1st Qu.:24.96   1st Qu.:121.5   1st Qu.: 27.70
 Median :24.97   Median :121.5   Median : 38.45
 Mean   :24.97   Mean   :121.5   Mean   : 37.98
 3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.: 46.60
 Max.   :25.01   Max.   :121.6   Max.   :117.50
```

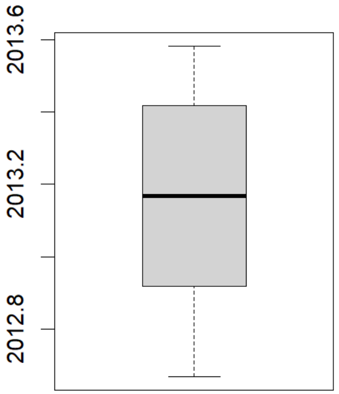
Observations:

- No missing values.
- Feature scaling is needed for X_3 due to
  - Large range for X_3.
- Clustering needed for X_5 , and X_6 due to
  - Smaller range for X_5 and X_6.

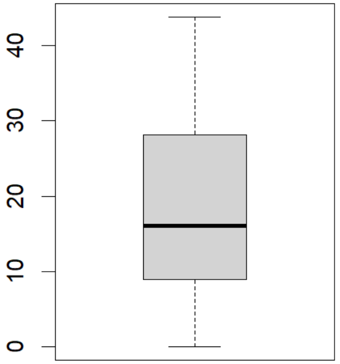
Data Exploration – Outlier Preview

Boxplot of Independent Variables

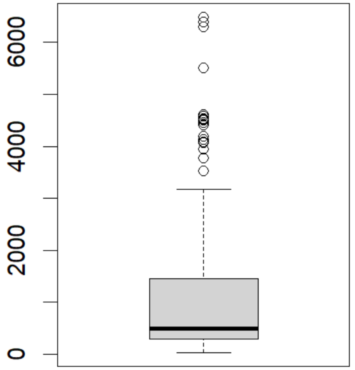
Transaction Date



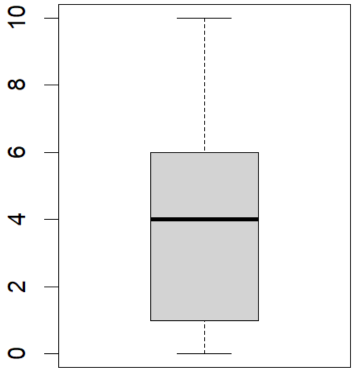
House Age



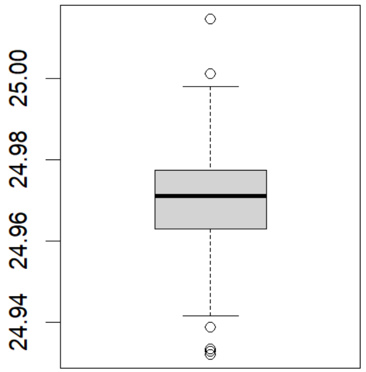
Distance to the Nearest MRT Station



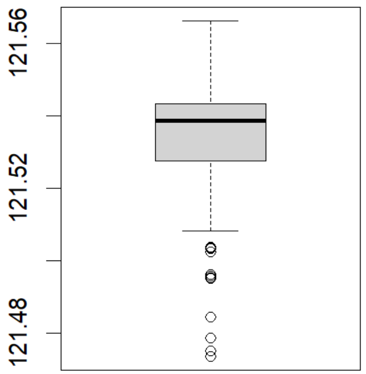
Number of Convenience Stores



Latitude

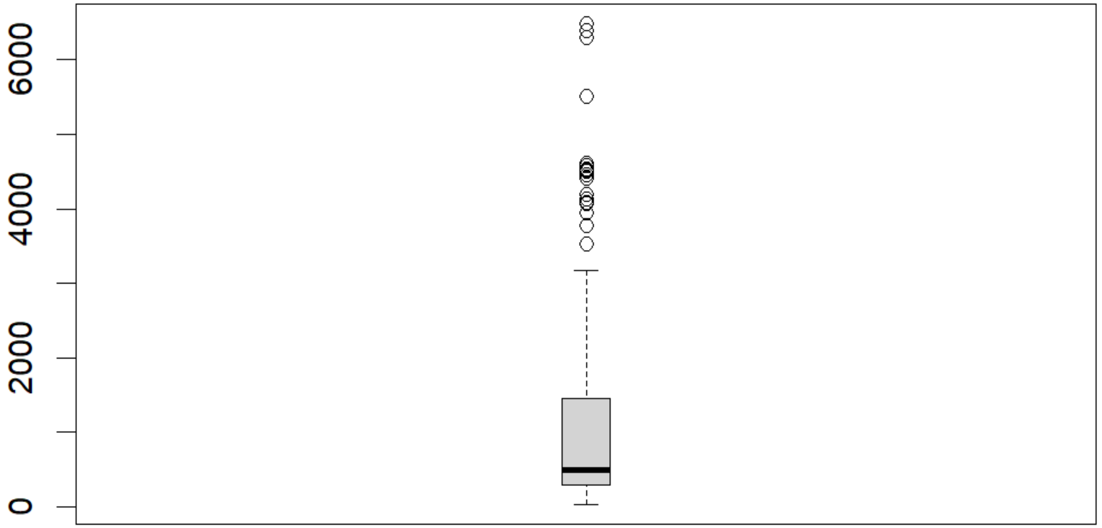


Longitude

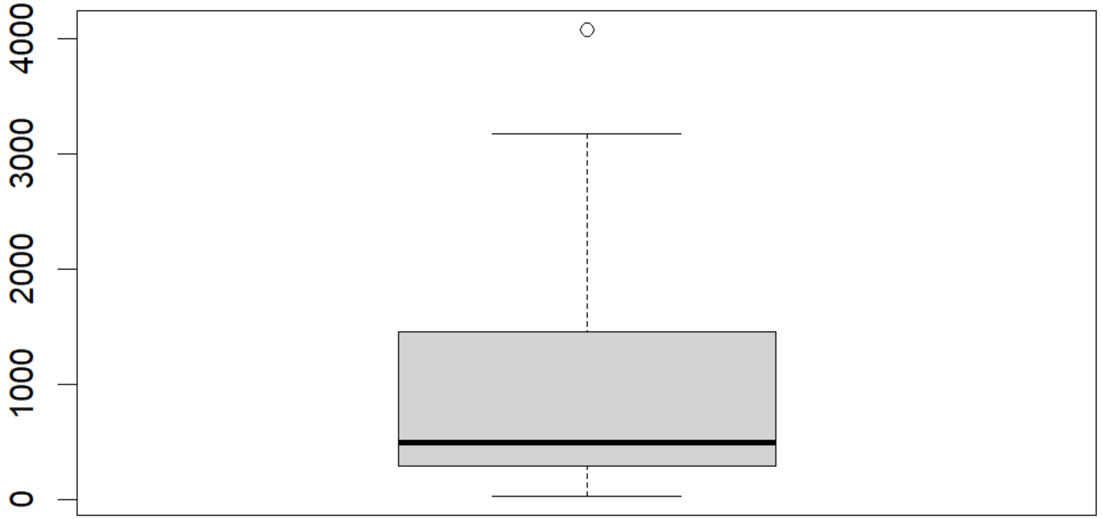


Data Exploration – Handling Outliers

Before Handling Outliers



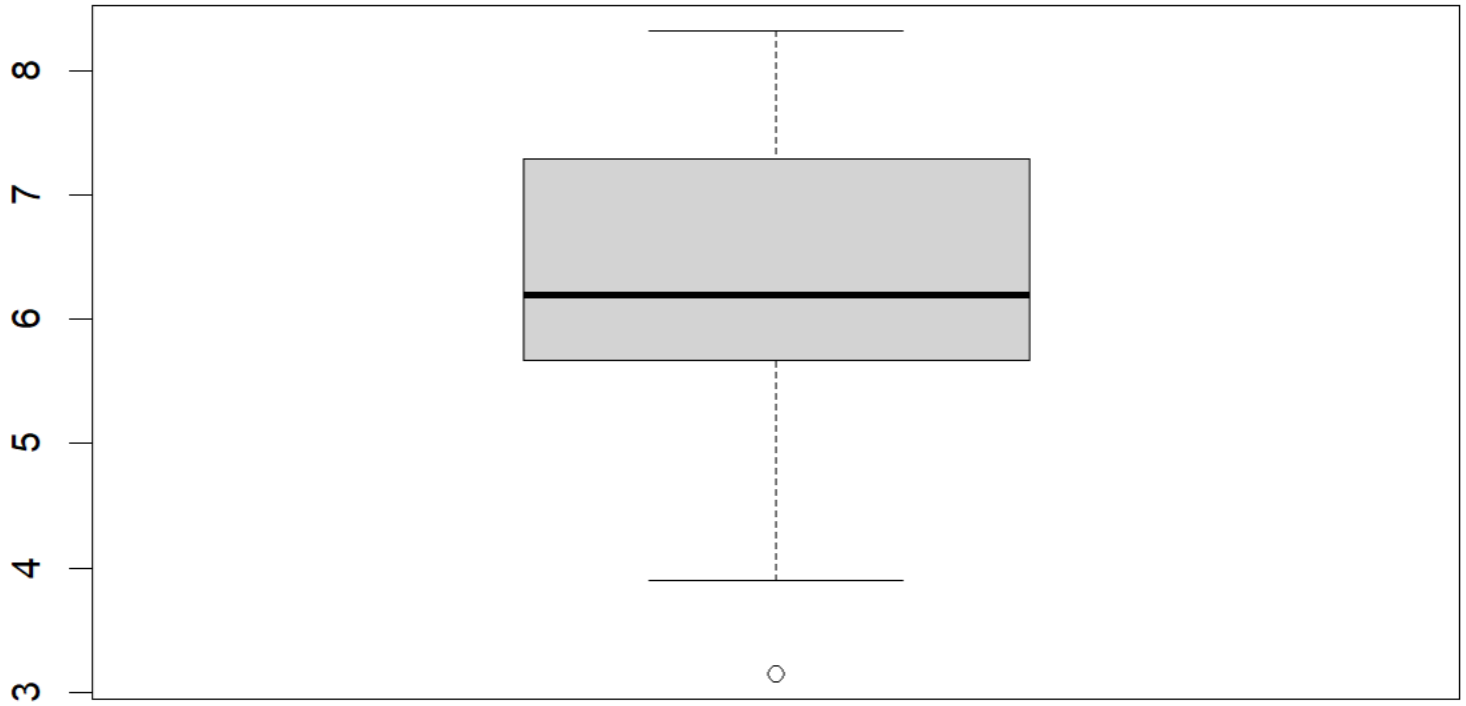
After Handling Outliers



Based on Outlier Preview, feature scaling is applied to X_3.
Outliers are capped. Outliers are replaced by 0.25 and 0.75 to rid existing outliers.

Feature Scaling

Feature Scaling of X_3: Values of X_3 are scaled to $log(X_3)$



Format Conversion

Transaction Type Conversion

- Original Format: Decimal Date
- Converted Format: YYYYMMDD(Integer Form)
  - Divide year into Year Blocks.
  - First Half of Year: January to June(Included)
  - Second Half of Year: July to December(Included)
  - Total Year Blocks Created: 4
    - 2012 First Half
    - 2012 Second Half
    - 2013 First Half
    - 2013 Second Half

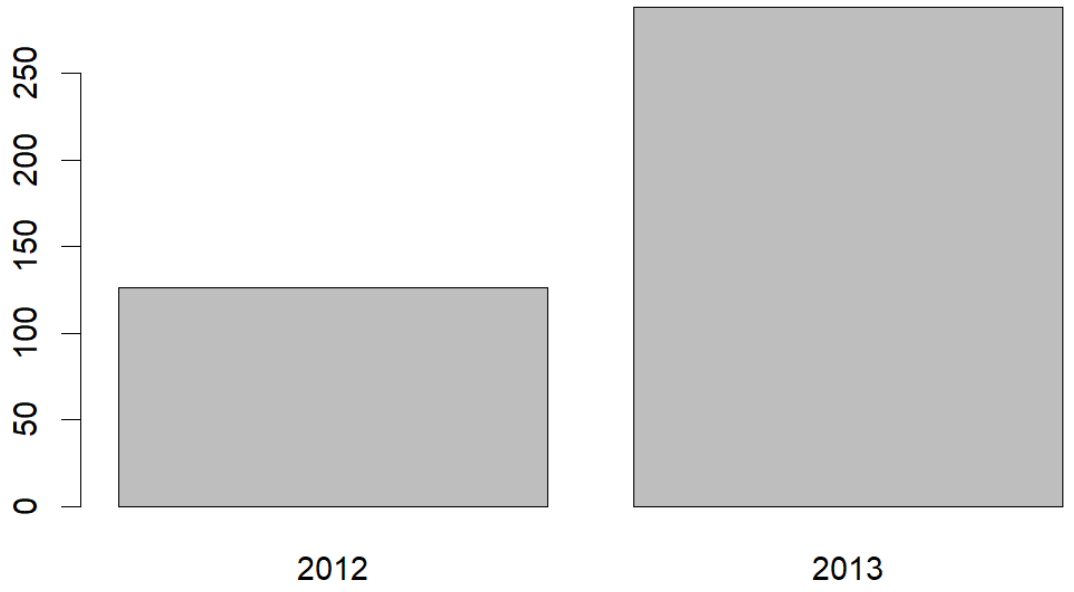
Original Format

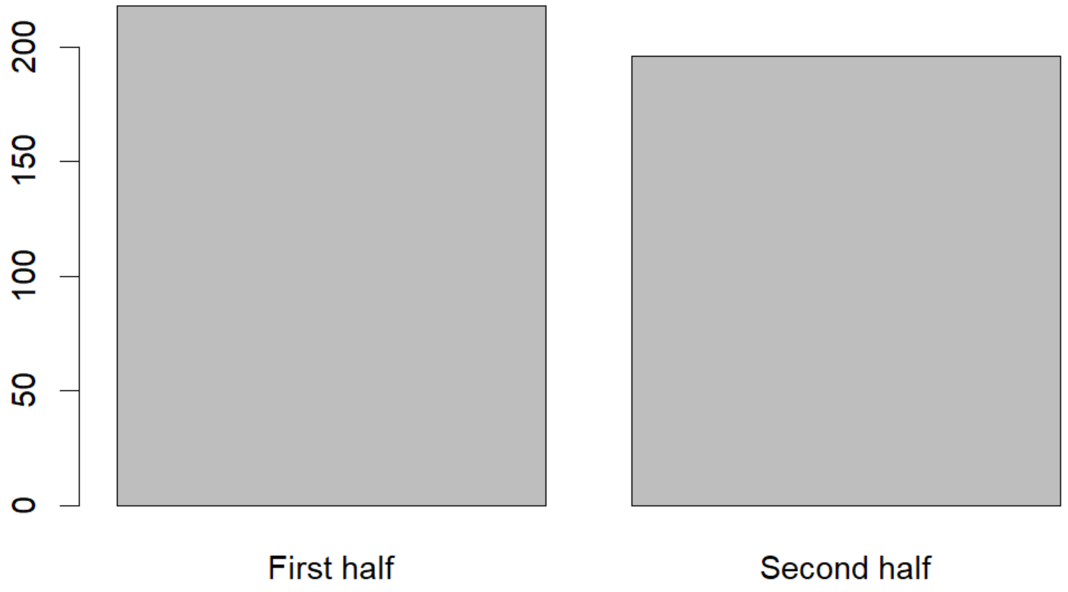
X1.transaction.date <dbl>
2012.917
2012.917
2013.583
2013.500
2012.833
2012.667

New Format

	Transaction.Year	Transaction.Month
1	0	1
2	0	1
3	1	1
4	1	1
5	0	1
6	0	1
7	0	1
8	1	0
9	1	1
10	1	0

Visualization – Converted

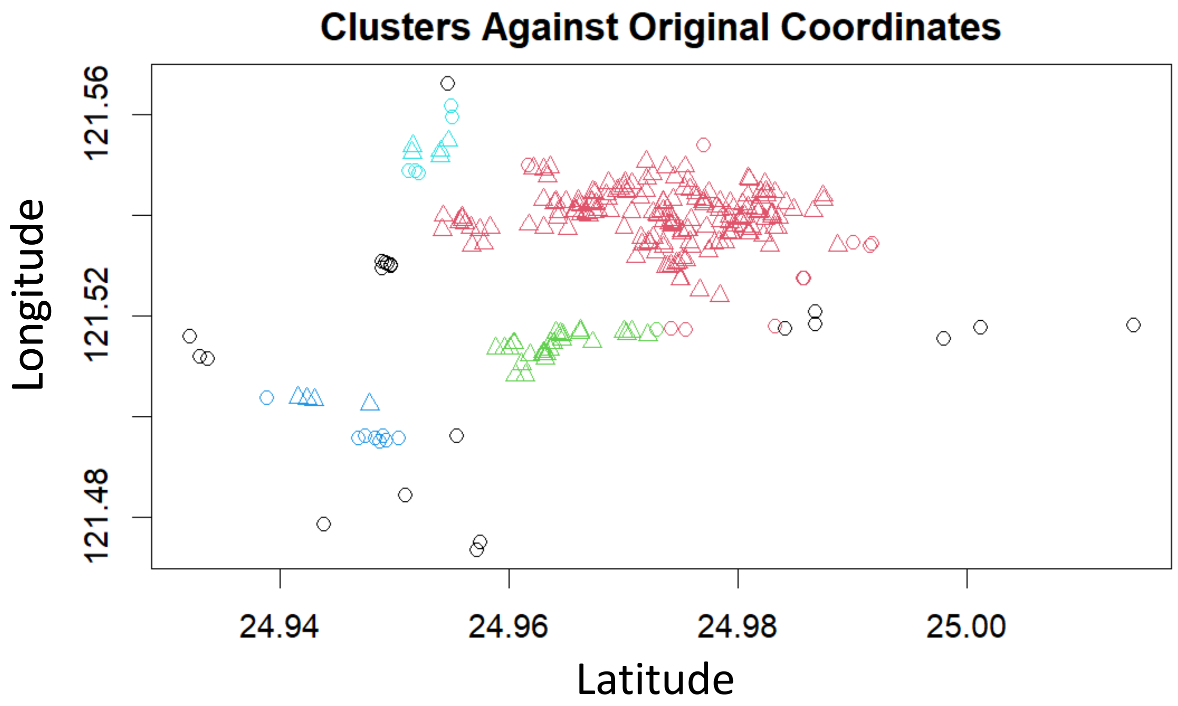




According to the above boxplot, there are more completed transactions in the year of 2013.

According to the above boxplot, the number of transactions in the first half and second half of the year are approximately the same.

Clusters



Classify Clusters

Set a default radius.

If the density is higher than the threshold, they will be considered as a cluster.

This method is used gather non-distributed clusters.

Correlation Matrix

```
                                      Transaction.Year Transaction.Month
Transaction.Year                           1.000000000       -0.69757232
Transaction.Month                         -0.697572321        1.00000000
Transaction.Day                           -0.077326689       -0.08739504
X2.house.age                               0.049171281       -0.03603718
X3.distance.to.the.nearest.MRT.station     0.062874202        0.03969007
X4.number.of.convenience.stores           -0.005585733       -0.03529447
Y.house.price.of.unit.area                 0.081544519       -0.05952584
clusters                                  -0.026441951        0.03728911
                                      Transaction.Day X2.house.age
Transaction.Year                         -0.077326689  0.049171281
Transaction.Month                        -0.087395039 -0.036037177
Transaction.Day                           1.000000000  0.000868411
X2.house.age                              0.000868411  1.000000000
X3.distance.to.the.nearest.MRT.station   -0.046056076  0.064005482
X4.number.of.convenience.stores          -0.062867382  0.049592513
Y.house.price.of.unit.area                0.024575778 -0.210567046
clusters                                 -0.059166116 -0.146548774
                                      X3.distance.to.the.nearest.MRT.station
Transaction.Year                                                   0.06287420
Transaction.Month                                                  0.03969007
Transaction.Day                                                   -0.04605608
X2.house.age                                                       0.06400548
X3.distance.to.the.nearest.MRT.station                             1.00000000
X4.number.of.convenience.stores                                   -0.68663404
Y.house.price.of.unit.area                                        -0.73182669
clusters                                                           0.47632977
```

```
                                      X4.number.of.convenience.stores
Transaction.Year                                          -0.005585733
Transaction.Month                                         -0.035294472
Transaction.Day                                           -0.062867382
X2.house.age                                               0.049592513
X3.distance.to.the.nearest.MRT.station                    -0.686634042
X4.number.of.convenience.stores                            1.000000000
Y.house.price.of.unit.area                                 0.571004911
clusters                                                  -0.356160984
                                      Y.house.price.of.unit.area
Transaction.Year                                       0.08154452
Transaction.Month                                     -0.05952584
Transaction.Day                                        0.02457578
X2.house.age                                          -0.21056705
X3.distance.to.the.nearest.MRT.station                -0.73182669
X4.number.of.convenience.stores                        0.57100491
Y.house.price.of.unit.area                             1.00000000
clusters                                              -0.42567247
                                         clusters
Transaction.Year                      -0.02644195
Transaction.Month                      0.03728911
Transaction.Day                       -0.05916612
X2.house.age                          -0.14654877
X3.distance.to.the.nearest.MRT.station  0.47632977
X4.number.of.convenience.stores       -0.35616098
Y.house.price.of.unit.area            -0.42567247
clusters                               1.00000000
```

From the observation of the correlation matrix, we can conclude that X_3 affects Y the most.
The relation between X_3 and Y presents a strong correlation.

Split the Data

```{r}
training.samples <- df$Y.house.price.of.unit.area %>%
  createDataPartition(p = 0.75, list = FALSE)
train.data <- df[training.samples, ]
test.data <- df[-training.samples, ]
```

Data was partitioned into training and test data.

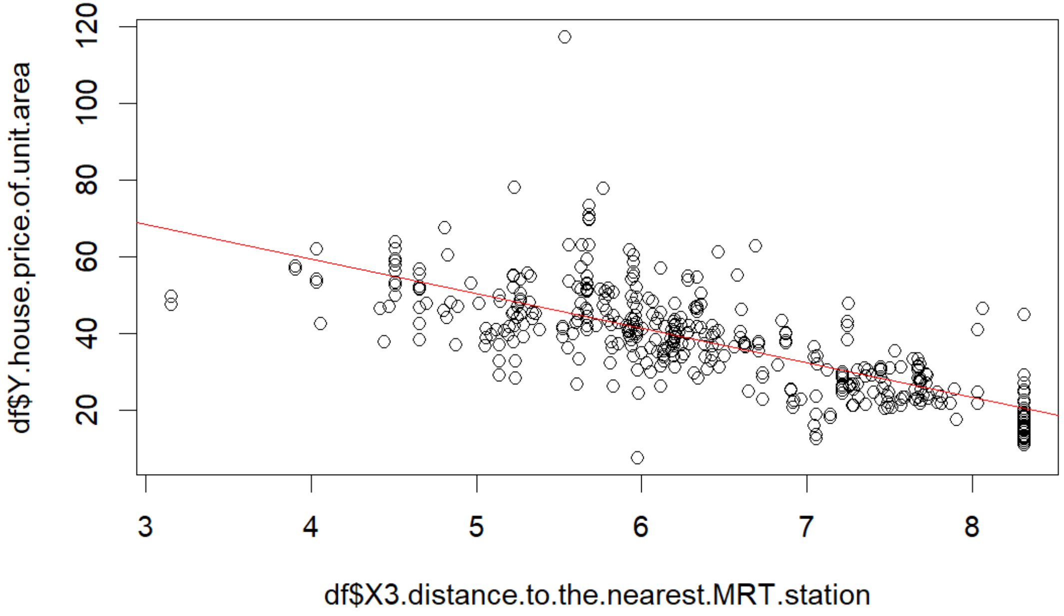
- Training: 75%
- Test: 25%

Result Exploration

A simple linear regression, multiple regression, and random forest model was then performed based on the partition of 75% training and 25% test data.

The purpose of the comparison is to see whether random forest is a suitable method to use.

Result Exploration - Simple Linear Model

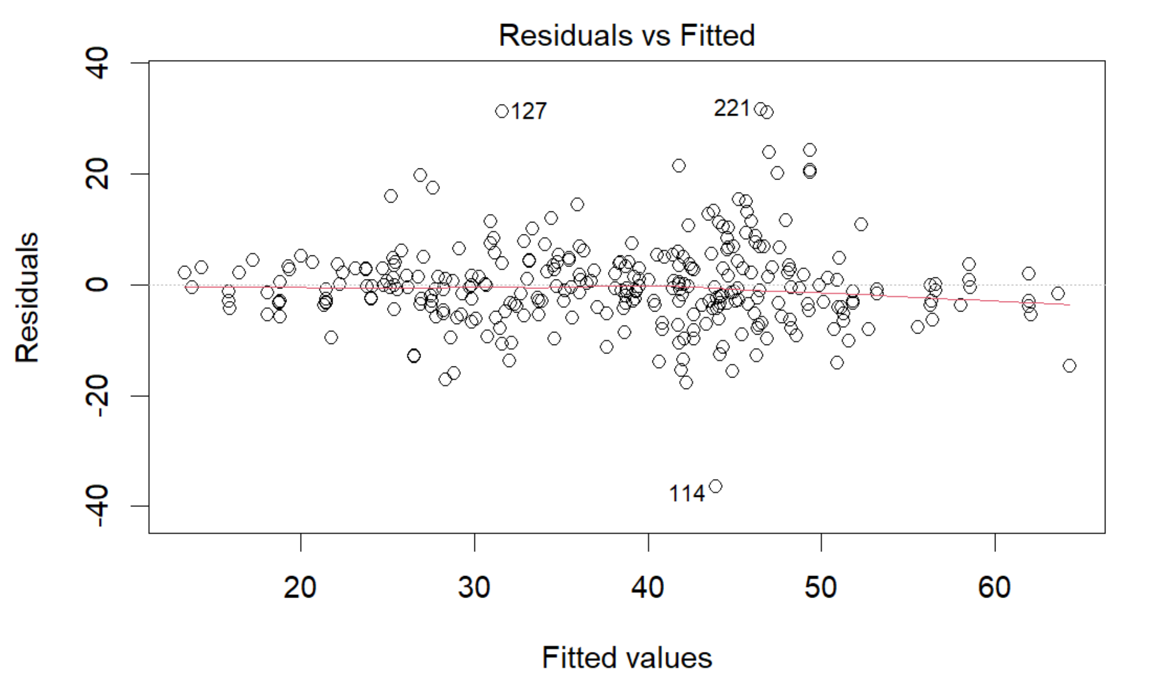



```
Call:
lm(formula = Y.house.price.of.unit.area ~ X3.distance.to.the.nearest.MRT.station,
    data = train.data)

Coefficients:
                           (Intercept)
                                95.644
X3.distance.to.the.nearest.MRT.station
                                -9.042
```

$$Y.house.price.of.unit.area = X3.distance.to.the.nearest.MRT.station$$

RMSE <dbl>	MSE <dbl>	Rsquare <dbl>
10.5387	111.0643	0.4797385

Result Exploration - Multiple Linear Model



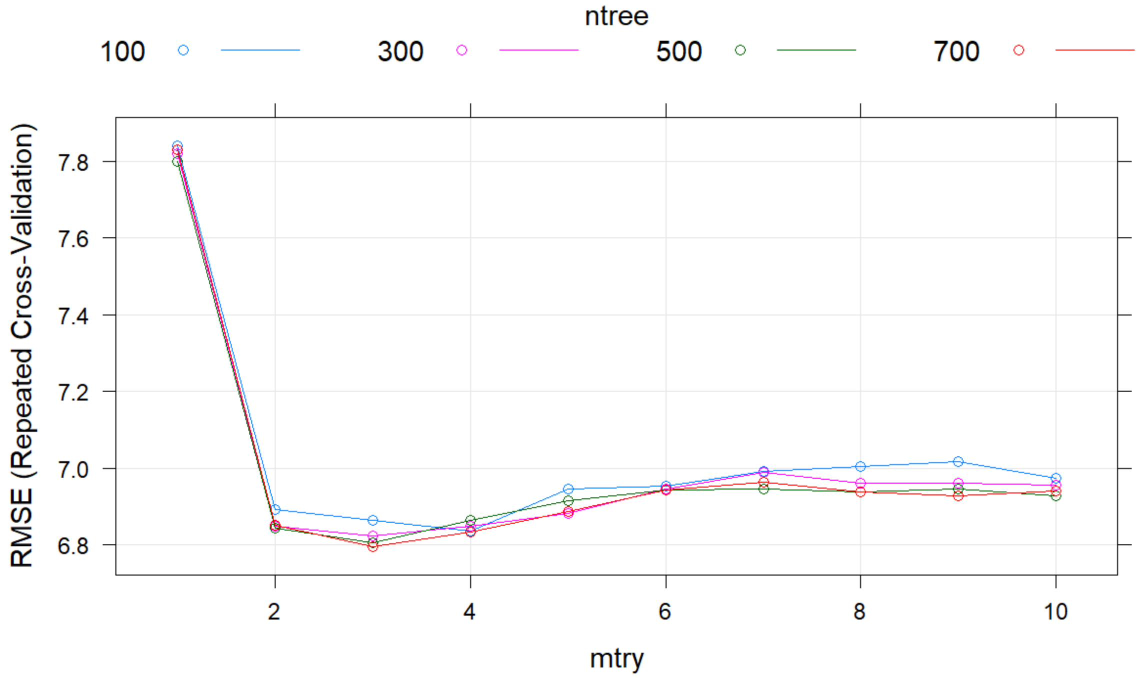
Residuals plot shows no clear patterns.

```
Call:
lm(formula = Y.house.price.of.unit.area ~ Transaction.Year +
    Transaction.Month + X2.house.age + X3.distance.to.the.nearest.MRT.station +
    X4.number.of.convenience.stores + clusters, data = train.data)

Coefficients:
                           (Intercept)
                               77.1982
                      Transaction.Year
                                5.7926
                     Transaction.Month
                                3.4502
                          X2.house.age
                               -0.2352
X3.distance.to.the.nearest.MRT.station
                               -6.6161
       X4.number.of.convenience.stores
                                0.9842
                              clusters
                               -2.0142
```

$$Y.house.price.of.unit.area = Transaction.Year + Transaction.Month + X2.house.age + X3.distance.to.the.nearest.MRT.station + X4.number.of.convenience.stores + clusters$$

RMSE <dbl>	MSE <dbl>	Rsquare <dbl>
10.28328	105.7459	0.5099844

Result Exploration - Random Forest

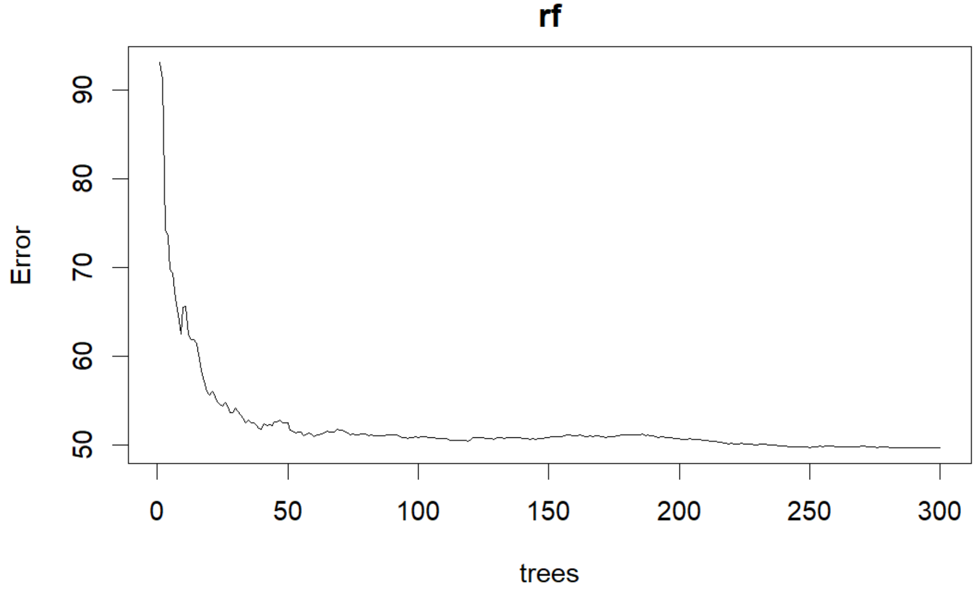


By plotting the custom, it can finalize the parameters for the final random forest model.

According to the plot, it can claim that there are much variations between setting *ntree = 300*, *ntree =* 500, or *ntree = 700*.

ntree = 300 was set to continue with the process.

Result Exploration – Random Forest



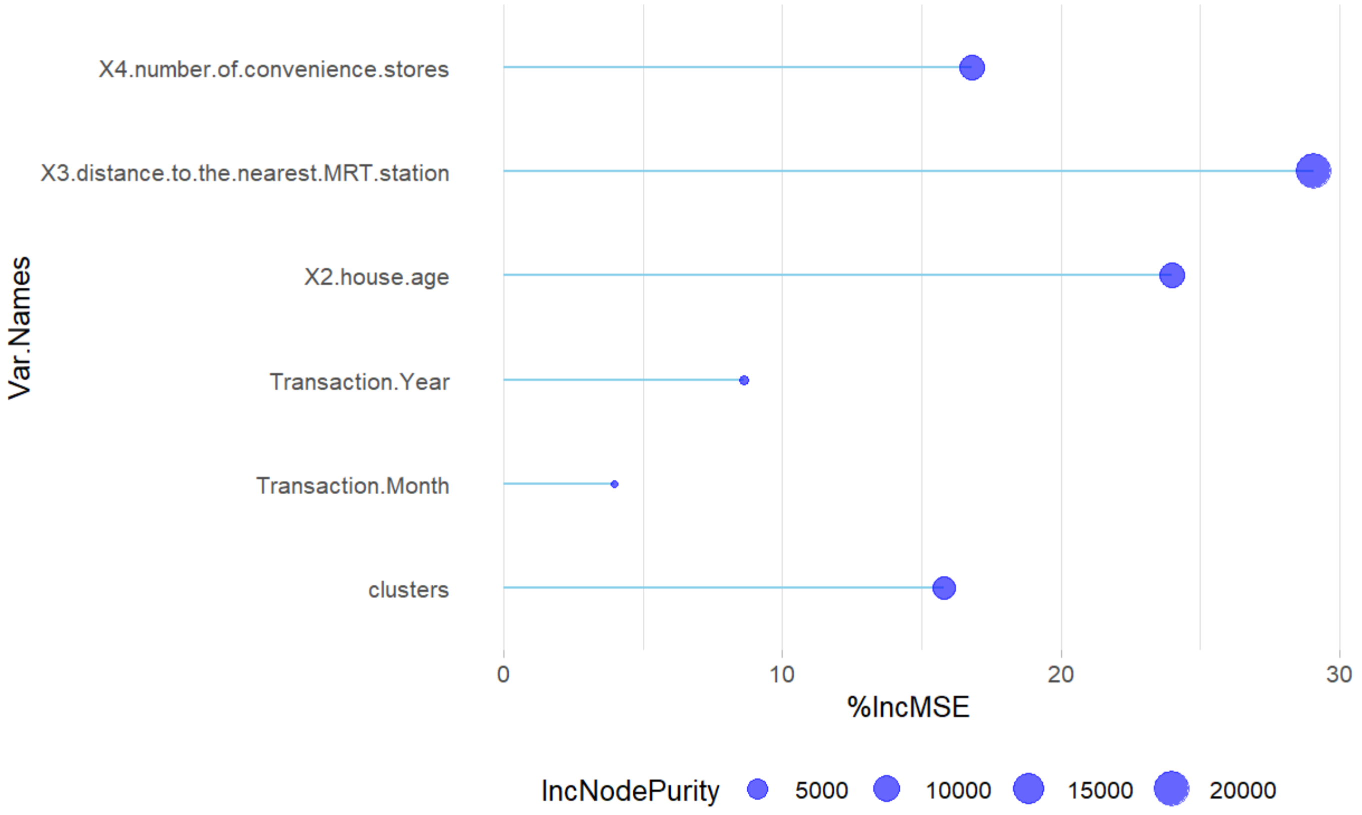

```
Call:
 randomForest(formula = Y.house.price.of.unit.area ~ Transaction.Year +
Transaction.Month + X2.house.age + X3.distance.to.the.nearest.MRT.station +
X4.number.of.convenience.stores + clusters, data = train.data,      ntree = 300,
keep.forest = TRUE, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 300
No. of variables tried at each split: 2

          Mean of squared residuals: 50.24379
                    % Var explained: 71.35
```

The error of the random forest model significantly decrease as the number of trees increases.

RMSE <dbl>	MSE <dbl>	Rsquare <dbl>
9.635758	92.84783	0.5698098

R^2(SLR): 0.4797385
R^2(MLR): 0.5099844
R^2(RF): 0.5698098

Contributions



The Variable Importance Plot displays the variable importance to the data.

According to the plot, X_3 plays an important measure to derive at the result.

Conclusion

Comparing to SLR and MLR, random forest is a more fitted method for the dataset.

House Price(Dependent Variable) was affected by other unknown measures.

Example:

	No	X1.transaction.date	X2.house.age	X3.distance.to.the.nearest.MRT.station	X4.number.of.convenience.stores	X5.latitude	X6.longitude	Y.house.price.of.unit.area	clusters
307	307	2013.500	14.4	169.98030	1	24.97369	121.5298	50.2	1
400	400	2012.917	12.7	170.12890	1	24.97371	121.5298	37.3	1
403	403	2012.833	12.7	187.48230	1	24.97388	121.5298	28.5	1

This set of data display similar measures for each independent variable but house price differ drastically.

These data distracts and lead to inaccurate results during the model training process.

References

UCI Machine Learning Repository: Real estate valuation data set Data Set. (n.d.). Archive.ics.uci.edu. https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set

Introduction to Random Forest in R. (n.d.). Simplilearn.com. https://www.simplilearn.com/tutorials/data-science-tutorial/random-forest-in-r

Donges, N. (2021, July 22). *A Complete Guide to the Random Forest Algorithm*. Built In. https://builtin.com/data-science/random-forest-algorithm

Decision Tree - GeeksforGeeks. (2017, October 16). GeeksforGeeks. https://www.geeksforgeeks.org/decision-tree/

Raj, A. (2021, June 11). *A Quick and Dirty Guide to Random Forest Regression*. Medium. https://towardsdatascience.com/a-quick-and-dirty-guide-to-random-forest-regression-52ca0af157f8

E R, S. (2021, June 17). *Random Forest | Introduction to Random Forest Algorithm*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Sun, Yifan, CSE 353(FALL2022): Bagging and Boosting

Maliar, Serguei ECO 352(FALL2022): Decision Trees, Ensembles, and Nearest Neighbor Methods