NUS

National University
of Singapore

EC4308:

2023/2024 Semester 1

Andrew Michael Walker (A0218138E)

Gao Yike (A0221548H)

Goh Meng Huang (A0218003X)

Liew Jun Yang (A0236555Y)

SECTION 1: Introduction

The Singapore Housing and Development Board (HDB) plays a pivotal role in mitigating housing challenges in Singapore, primarily driven by the constraints of limited land availability. In 2008, approximately 80% of Singapore's residents resided in HDB flats, and this percentage is projected to increase in the coming years, (Yuen, 2021). This underscores the critical role HDB flats play in enhancing the overall welfare of Singaporean society, as a substantial portion of the population depends on these flats to secure stable and affordable housing options. The prices of flats, thus, hold significant importance as they capture the attention of both investors and households. These price trends not only influence investment decisions but also have a direct impact on Singapore's overall welfare and development.

In this project, our objective is to develop an efficient model for predicting flat prices. To achieve this, we will explore a range of machine learning techniques, including penalized regression, principal component regression, partial least squares, decision trees, and other ensemble methods. We will rigorously assess the outcomes of each model and compare the predictive performance of each model.

Our dataset contains a wealth of indicator variables that provide valuable insights into the characteristics and configuration of flats. This information empowers our Machine Learning model to identify the most significant factors that contribute to the resale price of a flat. Consequently, investors can make more informed predictions about flat prices based on the flat's configuration. The utility of our models extends beyond buyers; it also assists sellers in setting fair and reasonable prices for their flats, enhancing the overall transparency and efficiency of the real estate market.

SECTION 2: Data

2.1 Data Overview

The dataset under analysis pertains to HDB resale prices in Singapore for the period from 2002 to 2012. It includes essential variables such as "year," "town," "flat_type," "storey_range," "floor_area_sqm," "flat_model," "lease_commence_date," and "resale_price." Our main goal is to build a predictive model for HDB resale prices using this dataset.

2.2 Independent Variables Data Preprocessing - Variable creation

In dealing with categorical variables like "town," "flat_type," "storey_range," and "flat_model," we've opted to manually create separate dummy variables for each category to ensure compatibility with different programming languages. These dummy variables are set to 1 when an observation belongs to a specific category and 0 otherwise.

The "year" column initially combines both the year and month into a single string format (e.g., "2001-10"). To capture the potential influence of the month on resale prices, we've split the "year" column into a new variable called "month."

Additionally, we've introduced an "age" variable, calculated by subtracting the "year" from the "lease_commence_date." This numerical variable is more computationally efficient and interpretable than creating numerous dummy variables for individual years. To address potential multicollinearity issues between dummy columns and the "age" variable, we omit one column when necessary to ensure model robustness and reliability.

We modified one of our categorical variables “storey_range” to change it to numerical values by taking the median of the ranges. For example, a storey_range value of “10 to 12” would be modified to 11. We decided to do this as we have a hypothesis that the price of apartments tends to be higher if they are located on a higher floor. The regression coefficient of our new variable “storey_range_median” would allow us to test that hypothesis. This comes with the added benefit of simplifying our regression equation as we can reduce the number of dummy variables required.

2.3 Independent Variables Data Preprocessing - Standardization

We decided to standardize the data because data standardization is a crucial preprocessing step that ensures the fairness and effectiveness of the model. Standardization involves scaling the features in the dataset to have a mean of zero and a standard deviation of one. This process helps bring all the features to a common scale, which is especially important when dealing with models that are not scale invariant, such as Ridge and Lasso Regressions. We decided to use this standardized dataset for the rest of our models to ensure equivalence and a fair comparison of our model results.

Standardization formula:

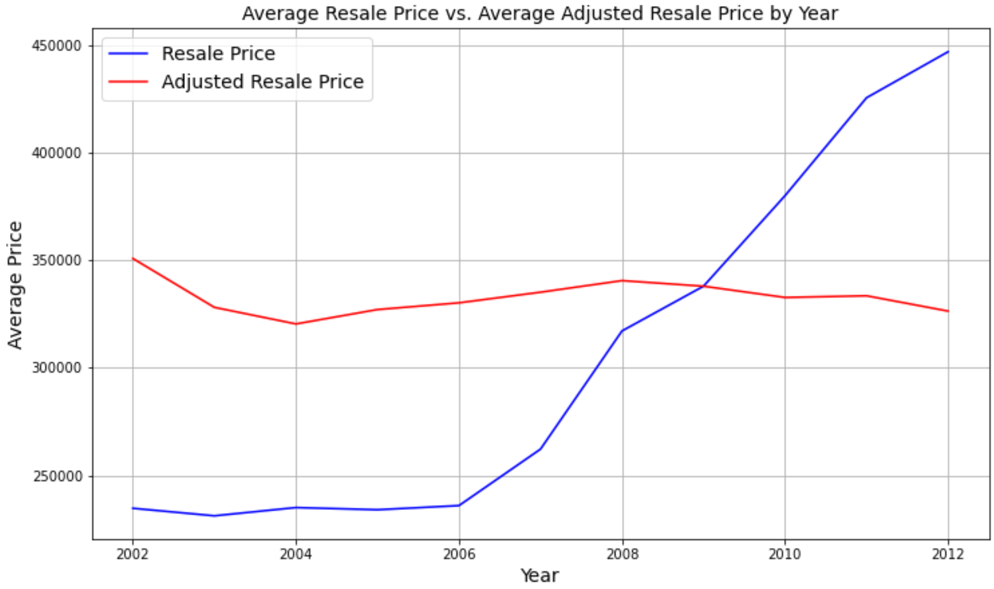
$$Z = \frac{x \; - \; \mu}{\sigma}$$

Where **Z** is our standardized independent value, **x** is our unstandardized independent variable, **μ** is the mean of the distribution of the unstandardized independent variables and **σ** is the standard deviation of the distribution of the unstandardized independent variables.

We also looked at anomalies that needed to be removed from our dataset before using it to train and test our models. Firstly, we examined our Dependent Variable, the Resale Price. We found the maximum resale price to be $1045595 and the minimum to be $43317 which are both reasonable bounds for HDB prices. The other variable that we needed to check the boundaries was the Floor Area. Before standardization, the largest Floor Area in our data set was 297m^2 , while the smallest was 28m^2 . These are all realistic bounds and thus we do not need to remove them from the dataset.

2.4 Dependent Variables Data Preprocessing - Adjusting for inflation

A reasonable assumption about our Dependent Variable "Resale Price" is that the price changes every year due to inflation. We mitigated this by scaling the prices of each year appropriately, taking 2009 as the benchmark year. This means that the prices post-adjustment are 2009 level prices. We got the data for resale flat quarterly price index from HDB, who also used 2009 as the benchmark year for prices. We then took the average price index of the 4 quarters in each year to get our index for the year. This calculated average yearly price index was used to scale our Resale Price variable to account for inflation.



The figure above compares Resale Price (Dependent Variable) before (blue) and after (red) adjusting for inflation.

2.5 Dependent Variables Data Preprocessing - Log Transformation

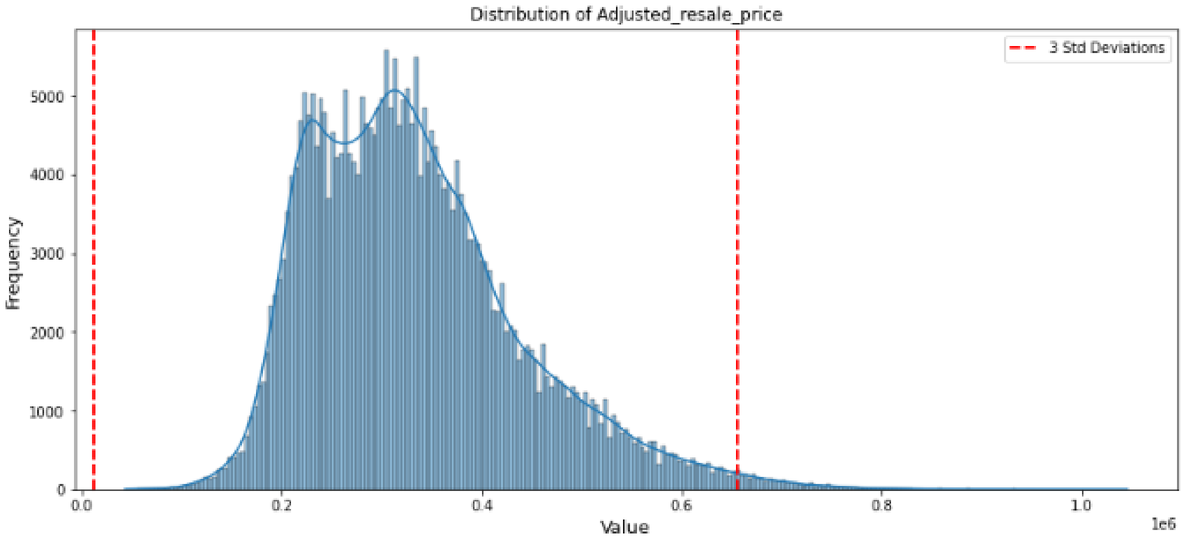
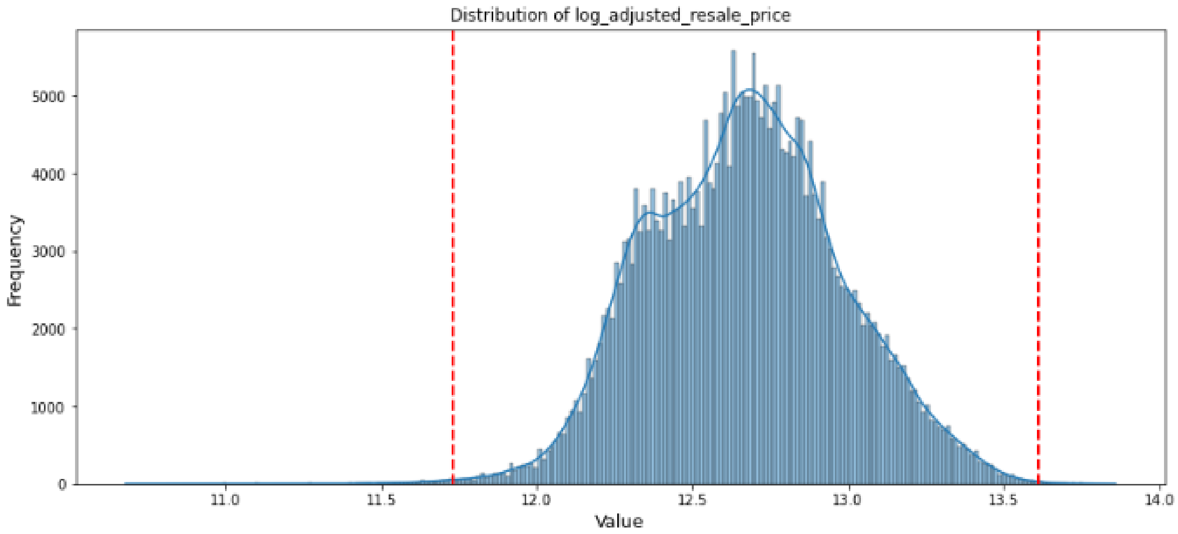


Fig 2. Histogram of Resale Price

From the plot of the histogram of "Resale Price", shown in the figure above, it can be clearly observed that there is a right skew, and that there are many data points above the 3 Standard Deviation red dotted line on the right side, with no points lower than the 3SD line on the left. This skewness can be due to various factors, such as most people are only able to afford, or are only willing to spend an amount that is closer to the minimum value HDB resale prices. People who are buying HDB resales instead of other housing options, such as Condominiums, are likely to have a lower purchasing power on average. This results in a huge clump of data towards the cheapest HDB prices, causing a right skew as there is essentially no upper limit to these resale prices.

Skewness is an issue as it could lead to biased results in our machine-learning models. We opted to deal with this skewness by taking the log of our Dependent Variable.



After applying a logarithmic transformation on “Resale Price”, shown in the figure above, we found that the histogram looks more like a normal distribution than a bimodal right skew distribution that was seen previously.

Dependent Variable used	3 Standard Deviation boundary	Number of data points (% of total points)
Resale Price	Upper Bound	2898 (0.97%)
	Lower Bound	0 (0%)
Log of Resale Price	Upper Bound	136 (0.045%)

	Lower Bound	652 (0.22%)

As seen from the table above, the number of points outside the 3 Standard Deviation boundaries was dramatically reduced via logarithmic transformation. Furthermore, the skewness has been greatly limited, as the difference in the number of points outside of the 3 Standard Deviation upper and lower bounds respectively has been greatly reduced.

Section 3: Baseline Results

3.1 Best Subset / Forward Stepwise / Backward Stepwise Selection

In this section, we will explore three feature selection methods: Best Subset Selection, Forward Stepwise Selection, and Backward Selection. Our objective is to identify the model that yields the lowest Out-Of-Sample Mean Squared Error (OOS-MSE) among these three methods, helping us determine the most effective feature selection approach for our analysis.

By setting the maximum number of features to 10, the model with 11 features has the lowest AIC and BIC value. Hence, we decided to keep it as one of our models to be considered later. We also use 10-fold Cross-Validation to select a model that has the lowest CV-MSE. This time the model with 8 features is selected, which is different from the model chosen by AIC/BIC. However, the 10-features-model performs better than the 8-features-model in Out-Of-Sample testing by achieving a lower OOS-MSE.

We repeat the same procedures for forward and backward stepwise selection. Both methods chose different numbers of features depending on the method we implemented (AIC/BIC and 10-fold CV). The number of features they have selected and their respective OOS-MSE can be summarised as follows:

Methods	Number of Features	OOS-MSE

Best Subset based on AIC/BIC	11	0.08923711
Best Subset based on 10-fold CV	8	0.08944854
FSS based on AIC/BIC	**10**	**0.08921166**
FSS based on 10-fold CV	9	0.08942443
BSS based on AIC/BIC	10	0.09281105
BSS based on 10-fold CV	9	0.09324931

From the table above, the model chosen by FSS using AIC/BIC value is the best out of all other models. The features selected are

- *bukit.merah* : = 1 if located in Bukit Merah
- *bukit.panjang* : = 1 if located in Bukit Panjang
- *choa.chu.kang* : = 1 if located in Choa Chu Kang
- *jurong.west* : = 1 if located in Jurong West
- *pasir.ris* : = 1 if located in Pasir Ris
- *sembawang* : = 1 if located in Sembawang
- *X1room* : = 1 if it is a 1-room flat type
- *X2room* : = 1 if it is a 2-room flat type
- *multi_gen* : = 1 if it is a multi-generation flat type
- *terrace* : = 1 if it is a terrace flat model

Variables	**Coefficients**
(Intercept)	12.662288699
bukit.merah	0.115082575
bukit.panjang	0.008699986
choa.chu.kang	0.082156085

jurong.west	-0.055446455
pasir.ris	0.241341605
sembawang	0.099885689
X1room	-1.267503312
X2room	-0.692943253
multi_gen	0.554223889
terrace	0.619249660

These are the 10 most useful features that are useful in predicting the HDB resale prices.

3.2 Ridge and LASSO Regression

We used ridge and LASSO regression to keep all the regressors but shrink the coefficients of not-so-relevant ones. LASSO, compared with ridge, can lead to some of the coefficient estimates being precisely zero when the regularisation parameter (λ) is set to a high value. Then we used 10-fold cross-validation to find a proper λ.

The results are as follows:

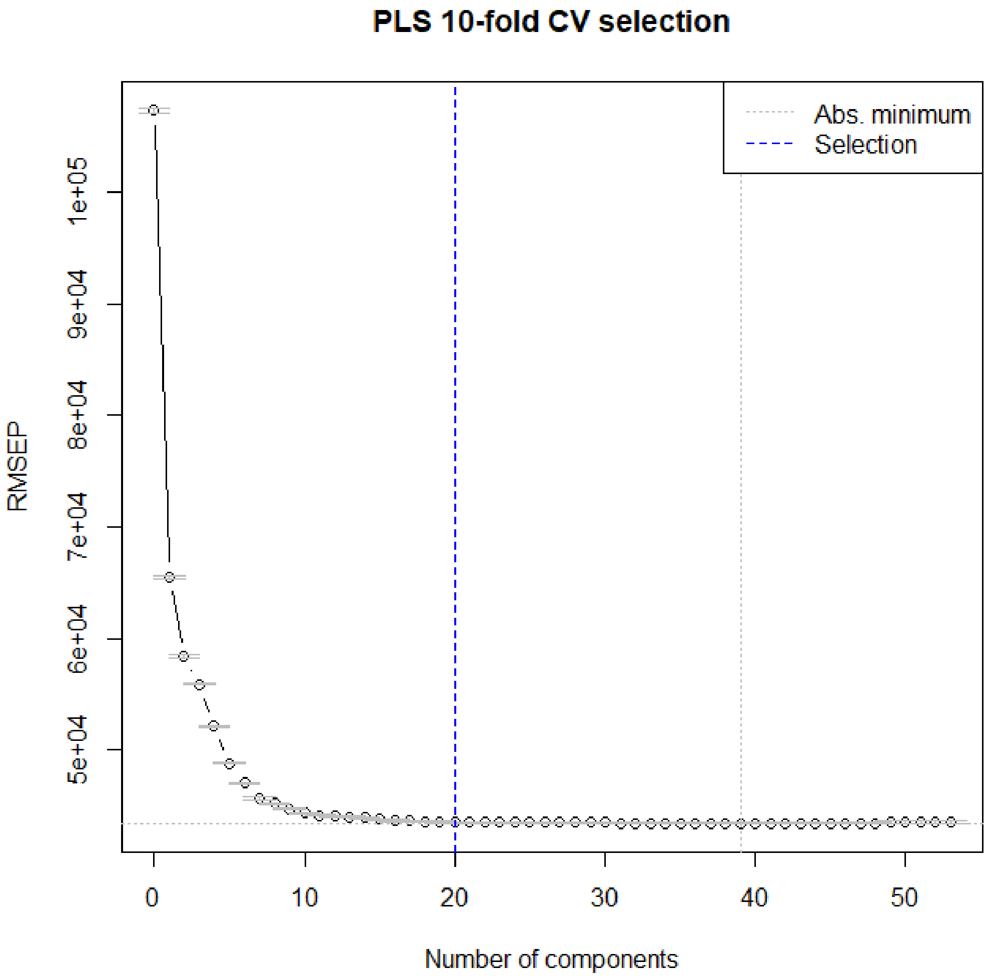
Criterion	λ	Test MSE
Ridge 10-fold CV	0.01	0.014993279
LASSO 10-fold CV	0.01	0.018807543
Post-LASSO 10-fold CV	0.01	0.015901595

It can be seen that the lambda values are extremely small in all cases, this indicates that little regularisation is required for this dataset.

3.3 Partial Least Squares (PLS)

We applied PLS to our prediction of the log of flat resale price. We deployed the pls_mod command to perform PLS.

We note that the pls_mod command included a ‘scale’ option which would automatically standardise the predictors to prevent variables with relatively larger values from affecting the analysis. As this process is automated, the ‘scale’ option would standardize every numerical column that was in our data set. We tried using scale = TRUE with our non-standardized data set and obtained many NaN results after running summary(pls_mod).
Hence, we used our manually standardized data set and obtained the following cross-validation plot that indicates the 1SE number of principal components to be used in predicting the resale prices.

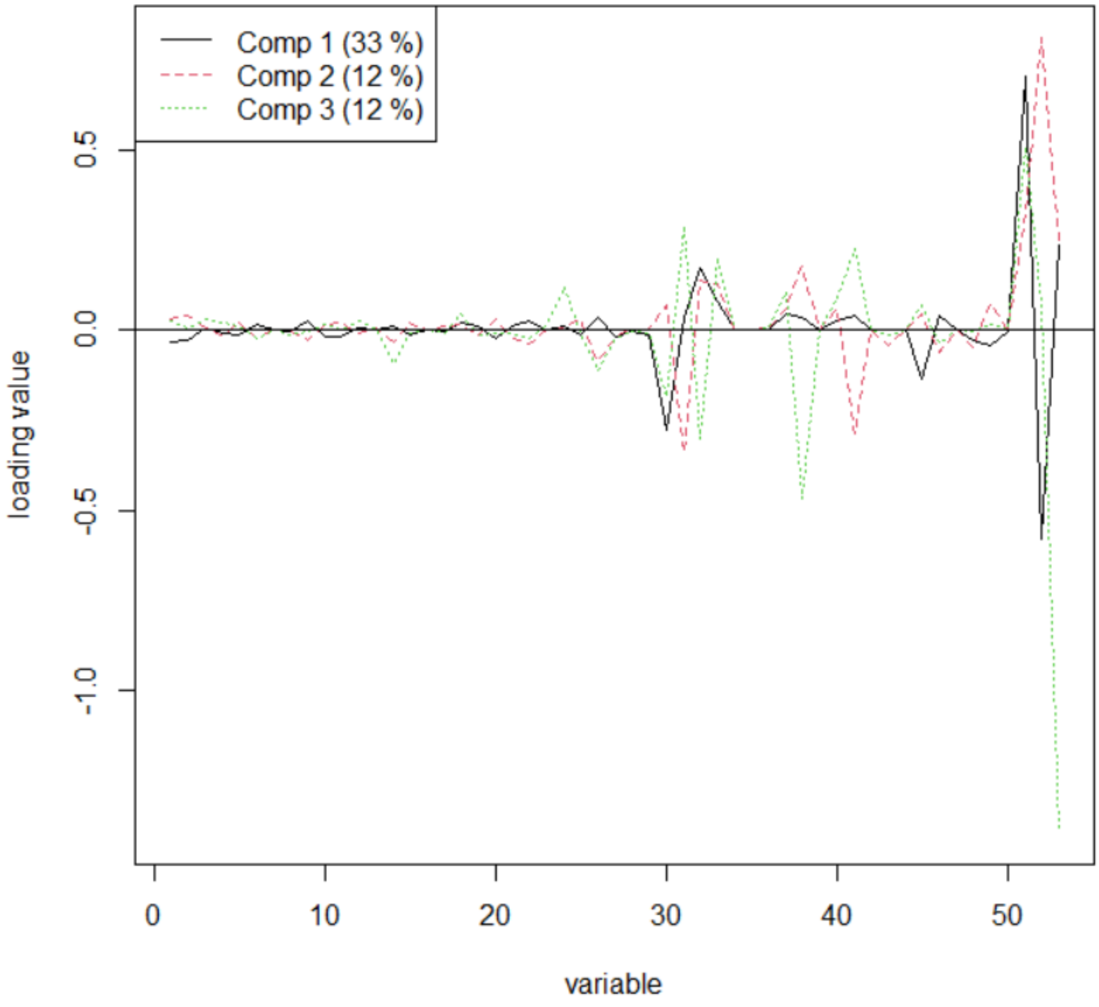


In this figure, the 1SE number of components is 20.

Using 20 principle components in predicting the log of resale prices, we obtained an OOS MSE: 0.0144

Loadings for Principal Components

We also performed an analysis to seek out the most important predictors in predicting the log of resale prices.



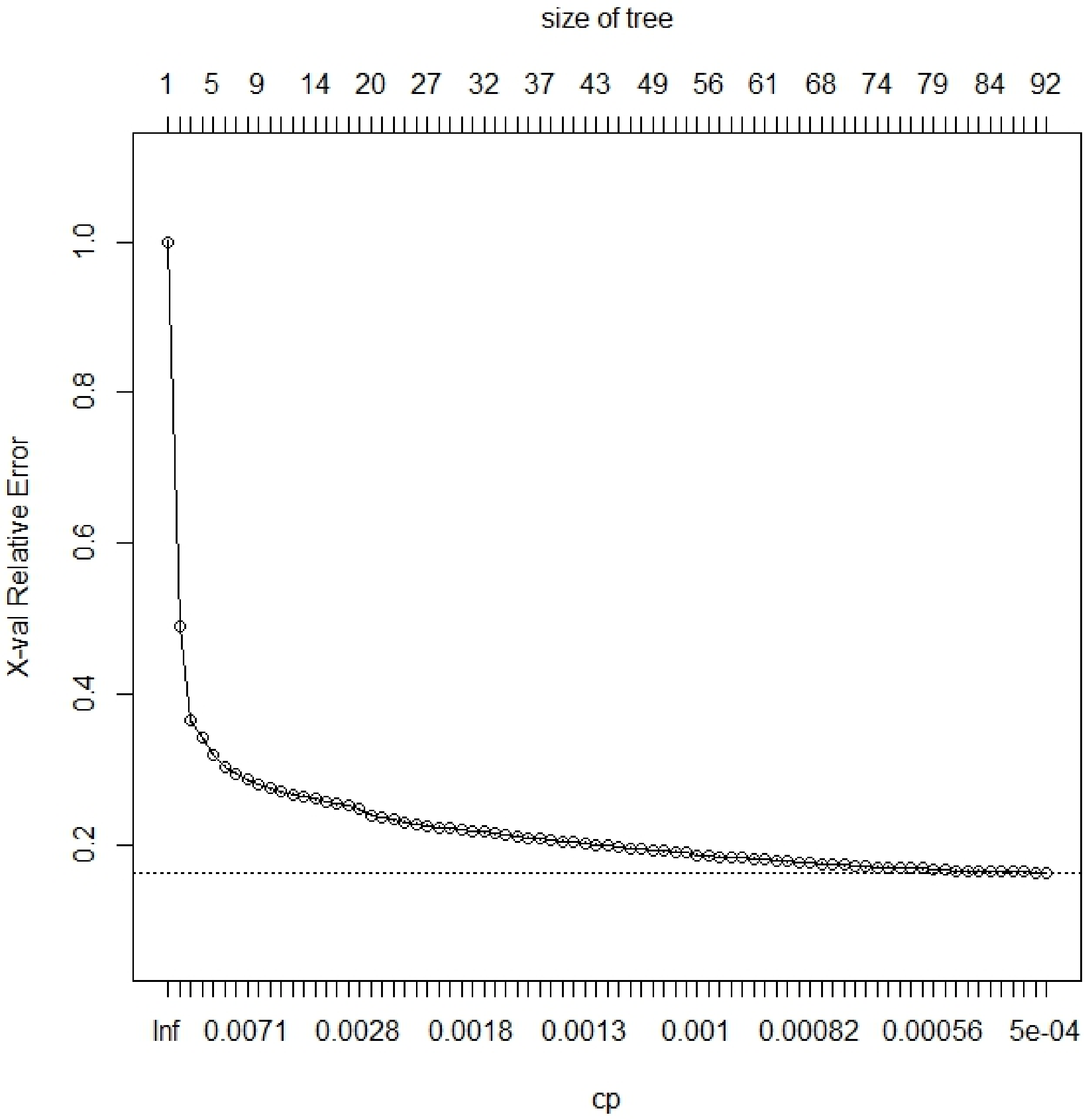
In this figure, the top 3 principal components and the loadings of different predictors on each principal component are illustrated.

Interpretation of PLS result: We see that variables 51 (floor_area_sqm_std) and 52 (age_std) have the highest loadings, which means that they help explain the dependent variable (log of resale prices) the most.

3.4 Regression Tree

Growing a big tree

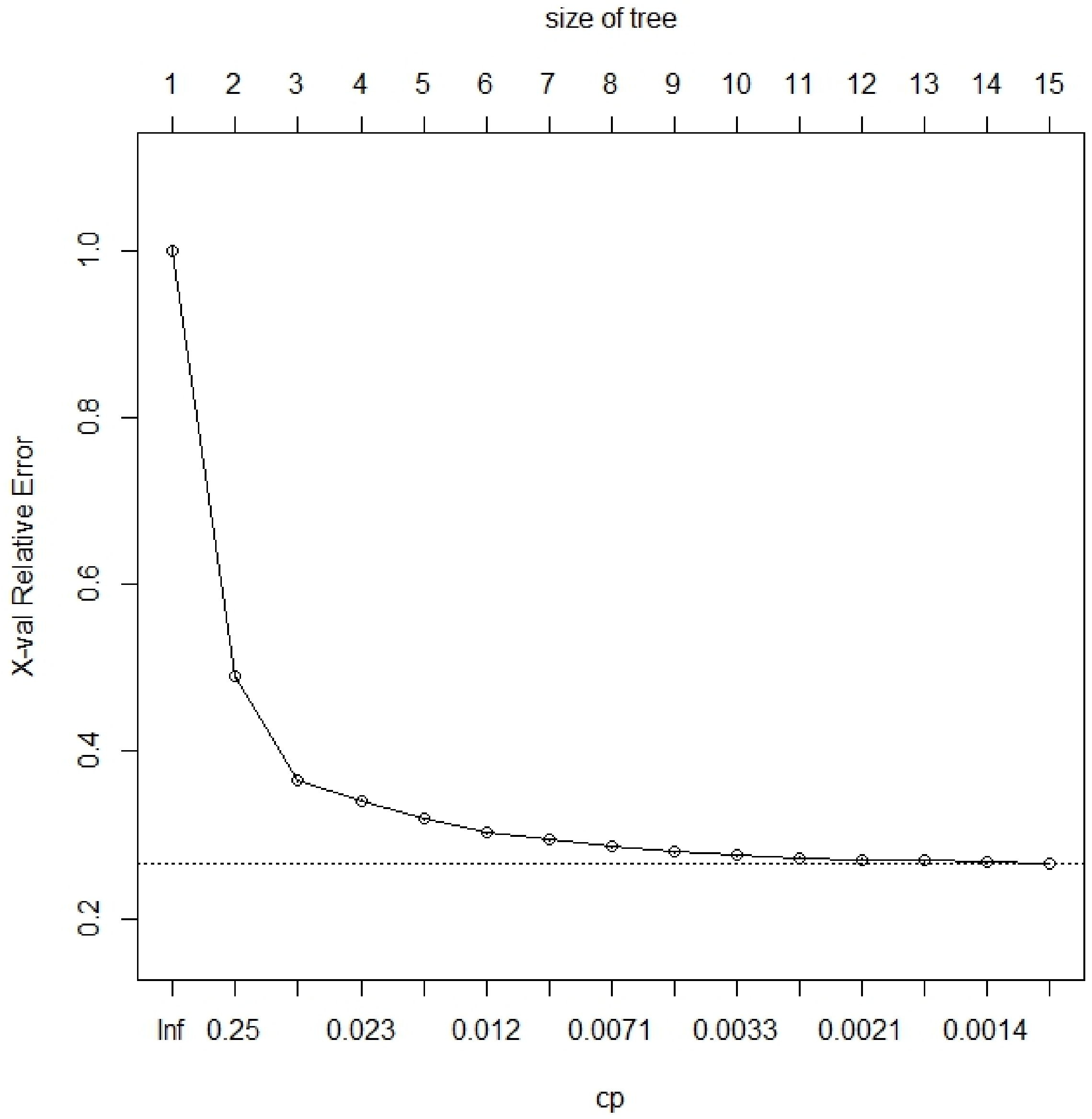
We first set the penalty parameter (alpha) (cp in rpart) to a small number to allow a big tree to be grown. We conducted cross-validation to assess the value of cp that will give the minimum X-val relative error (ratio of MSE of subtree to the MSE of sample mean of flat prices).



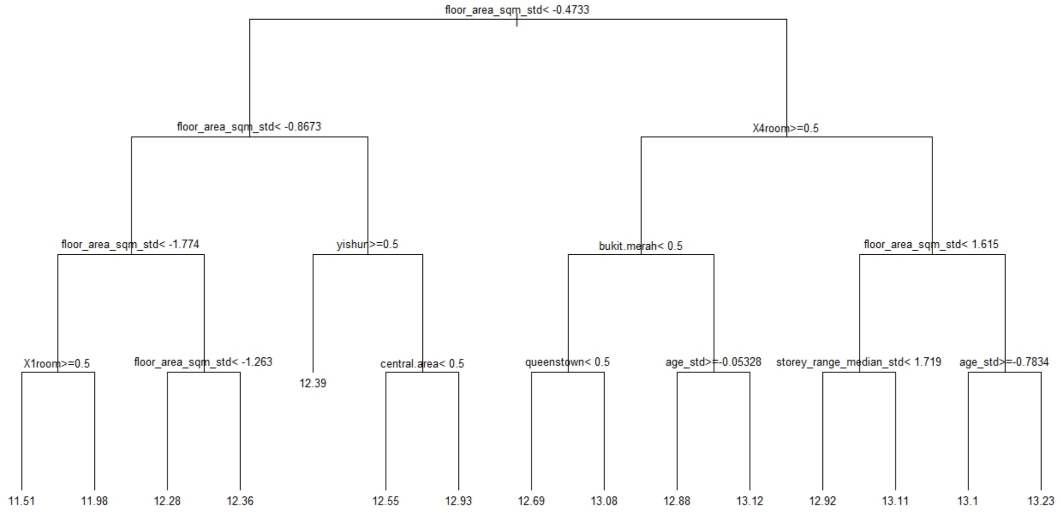
However, as the X-val relative error does not have a turning point, the number of splits indicated by the cross-validation is large. This would likely lead to overfitting.

Hence, to prevent overfitting we restricted the max depth to 4, while keeping the cp small. This will ensure that the tree does not get too big.

We then applied cross-validation again to determine the value of cp.



We retrieved the minimum cp generated (15 terminal nodes) and produced the regression tree.



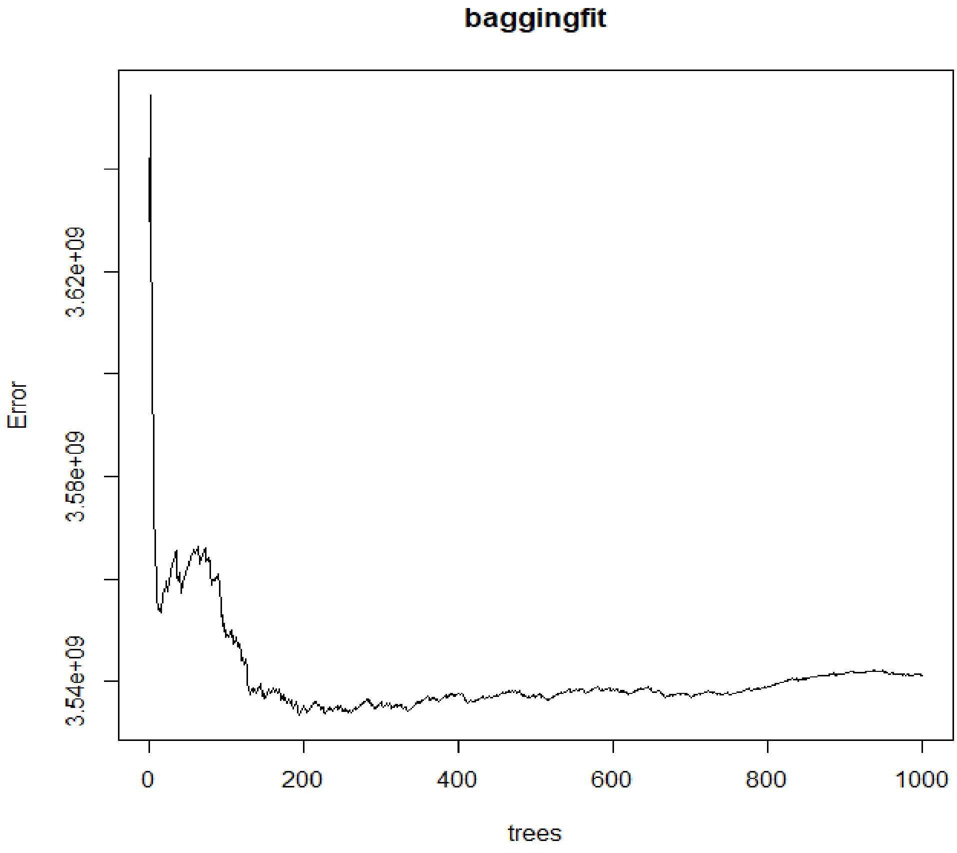
Interpretation of Regression Tree result: we see that the floor area is the most significant variable in determining the price of HDB flats, followed by the location of the flat, storey and age. We see that the bulk of the nodes comprise these variables (with location appearing the most), indicating that the other variables may not be helpful in determining the price of HDB flats.

After testing the predictions of log resale price on the test set, we obtain an
OOS MSE: 0.026

Section 4: Ensemble Learning Results

4.1 Bagging

To see whether ensemble learning will provide a better prediction, we deployed the bagging algorithm.



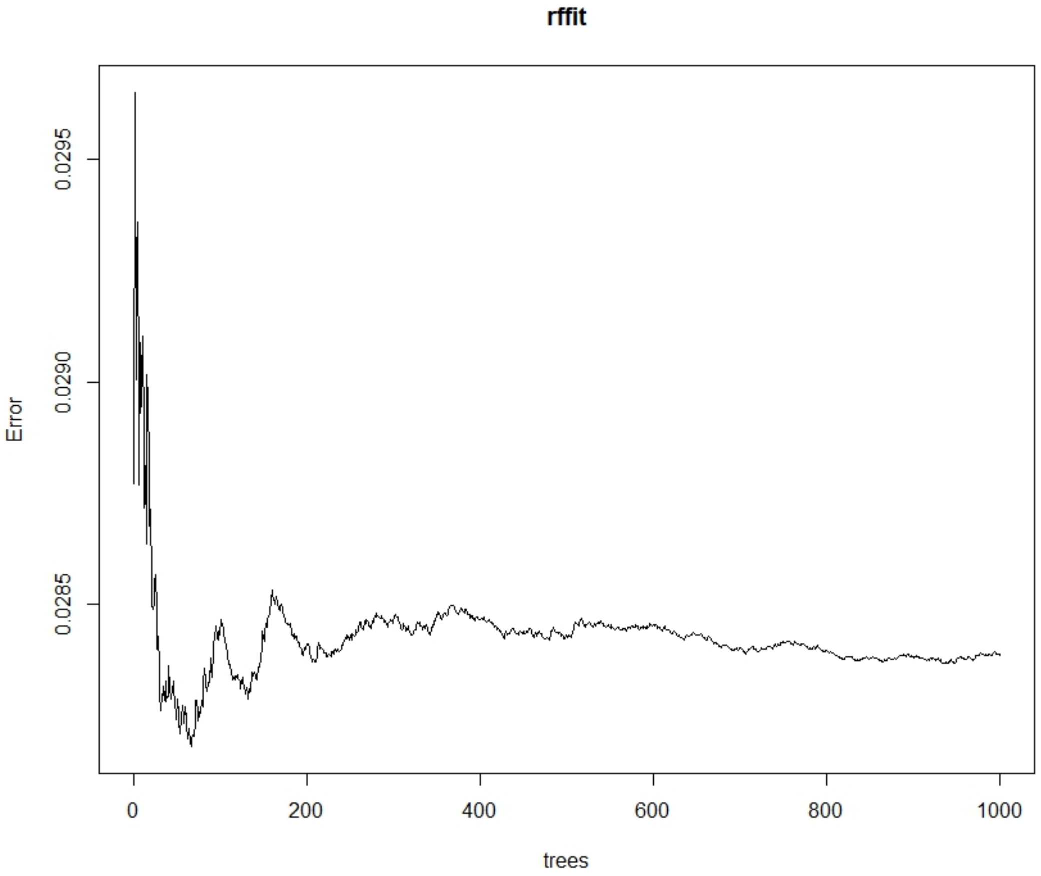
Using bagging with out-of-bag (OOB) validation, the training error tends to stabilise around 200 trees as seen in the figure.

After testing the predictions of log resale price on the test set, we obtain an

OOS MSE: 0.028

4.2 Random Forest

As a standing issue with bagging is that the trees it produced are correlated due to the algorithm using every predictor in its construction of trees, we tried using random forest as well. We used the recommended number of predictors ⅓*P = 18 for the number of predictors to be used under the random forests algorithm.



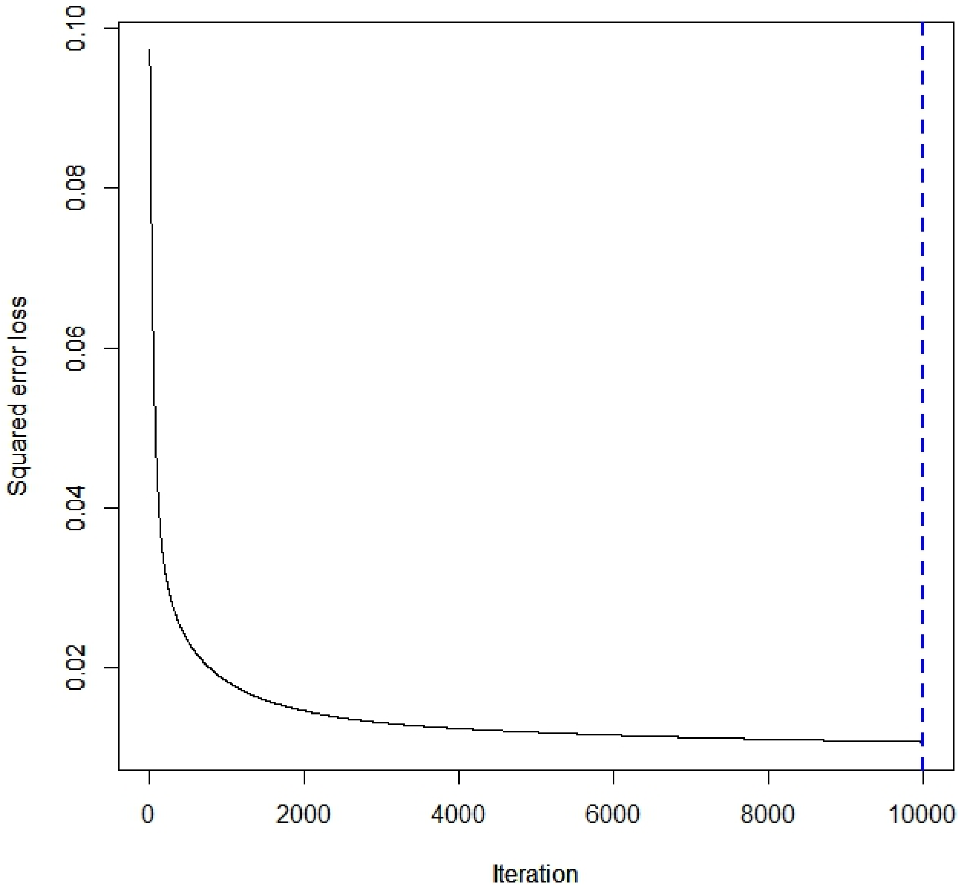
Using out-of-bag (OOB) validation, we see that the training error tends to stabilise around 200 trees as well.
After testing the predictions of log resale price on the test set, we obtain an

OOS MSE: 0.028

Interpretation of bagging versus random forest: We see that the training error stabilizes around the same number of trees, and both algorithms have the same OOS MSE. Since random forest tries to de-correlate the trees by reducing the number of predictors used, we can say that there are quite a few highly correlated and moderately good predictors in predicting the log of resale flat prices, which leads to a similar OOS MSE between the two algorithms.

4.3 Boosting

To see whether another form of ensemble learning will help with predicting, we used boosting as well. We configured the algorithm to compute the OOB squared error loss and the optimal number of iterations (number of weak learners). We set the maximum number of iterations to be 10000.

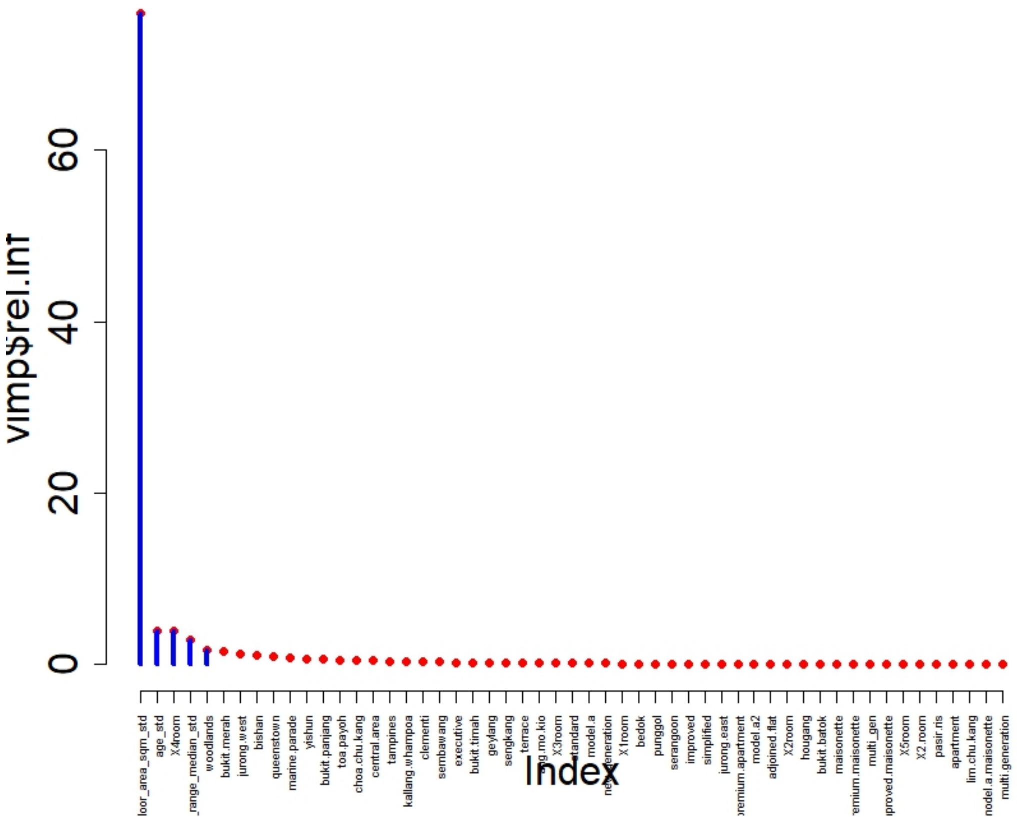


As OOB MSE tends to underestimate the number of iterations, we initially considered doing a cross-validation (CV) exercise as well. However, as seen from the graph, the optimal number of iterations is the maximum number of iterations. Hence, there is no incentive to perform a computationally expensive CV to obtain the optimal iterations since it will likely be the maximum number of iterations as well.

After testing the predictions of log resale price on the test set, we obtain an
OOS MSE: 0.0107

Relative importance of predictors

Using the OOB MSE optimal number of iterations (10000), we obtain the following relative importance of the predictors in predicting the log resale prices.



We see from this figure that floor area and age of the housing unit are the leading two predictors of log of resale flat prices. However, the importance of floor area is much higher (at about 76% of total contribution compared to other predictors). This potentially means that floor area will be able to predict the flat prices fairly well.

Section 5: Conclusion

After evaluating our different models, we concluded that our best model is boosting based on test MSE.

Model	Test MSE
Best Subset based on AIC/BIC	0.0892
Best Subset based on 10-fold CV	0.0894
FSS based on AIC/BIC	0.0892
FSS based on 10-fold CV	0.0894
BSS based on AIC/BIC	0.0928
BSS based on 10-fold CV	0.0932
Ridge 10-fold CV	0.0149
LASSO 10-fold CV	0.0188
Post-LASSO	0.0159
Partial Least Squares	0.0144
Regression Tree	0.026
Random Forest	0.028
Boosting	0.0107

In summary, our exploration into machine learning algorithms for predicting resale prices led us through various methods, including best subset, forward stepwise, and backward stepwise selection. These approaches resulted in relatively high Test Mean Squared Error (MSE). One potential explanation for this outcome is that, despite their selection through 10-fold cross-validation, these models may not be adequately penalized, leading to overfitting of the training data.

Surprisingly, more complex methods did not necessarily outshine penalized regression techniques like Ridge and LASSO. This could be attributed to the careful selection of a penalized term ($\boldsymbol{\lambda}$) based on 10-fold cross-validation, effectively mitigating the risk of overfitting and producing a model capable of robust Out-Of-Sample predictions. Ultimately, our boosting model emerged as the frontrunner, boasting the lowest Test MSE among all other models.

However, it is essential to acknowledge certain limitations in our project. Due to computational constraints, we could only select 10 regressors in the best subset/forward stepwise/backward stepwise procedure. It is plausible that allowing more features could potentially enhance the models' performance beyond our current results. Another limitation lies in the absence of crucial predictors for resale price, such as proximity to the MRT station. While many similar projects online incorporate this feature by tapping into online APIs, our lack of expertise in other domains prevented its inclusion in our models. Furthermore, we were constrained by computational limits for the method of boosting - a more powerful machine would have enabled us to perform CV and obtain the number of optimal iterations (small trees) by locating where the test MSE starts to increase. With OOB error, we cannot be sure that the optimal number of iterations is truly “optimal”. Likewise, random forests can determine the optimal number of predictors through the tuneRF() command. However, this is also computationally expensive and would require more computational power.

References

Yuen, S. (2021). *Singapore resident population in HDB flats falls to 3.04m, with smaller households spread over more flats*. The Straits Times. https://www.straitstimes.com/singapore/singapore-resident-population-in-hdb-flats-falls-to-304m-with-smaller-households-spread#:~:text=Home-,Singapore%20resident%20population%20in%20HDB%20flats%20falls%20to%203.04m,households%20spread%20over%20more%20flats&text=SINGAPORE%20%2D%20The%20number%20of%20Singapore,HDB%20households%20continued%20to%20climb.

Wang, Yijia & Zhao, Qiaotong. (2022). House Price Prediction Based on Machine Learning: A Case of King County. 10.2991/aebmr.k.220307.253.