SCHOOL OF COMPUTER SCIENCES

UNIVERSITI SAINS MALAYSIA

CDS503 MACHINE LEARNING

SEMESTER 2 2021/2022

FINAL REPORT

GROUP 9

SINGAPORE HDB FLAT RESALE PRICE PREDICTION MODEL

GROUP MEMBERS

NAME	MATRIC NO.	USM EMAIL ADDRESS	EXPERIMENT SET
KAM SAY HONG	P-COM0293/21	kamsayhong@student.usm.my	1: COMPARING ML ALGORITHMS
ZHOU XIUKANG	P-COM0028/22	zhouxiukang@student.usm.my	2. FEATURE SELECTION
TAN YI XUAN	P-COM0018/22	yixuan51@student.usm.my	3. ENSEMBLE LEARNING
TEO KENG BOON	P-COM0022/22	kbteo@student.usm.my	4: VARYING TRAINING SAMPLE SIZE

SUBMISSION DATE

16TH JULY 2022

Content

DATASET SELECTION

The dataset is publicly available on the Singapore Government's public data website (Data.gov.sg). The dataset contains the resale prices of the public housing of Singapore (HDB Flat) since January 2017 to April 2022.

The attributes of the dataset are as follows:

Table 1 Attributes of the dataset

Attribute	Categories	Description
month	Ordinal categorical	Starts from 2017-01 to 2017-04
town	Nominal categorical	Different towns are listed
flat_type	Ordinal categorical	Ranges from 1 Room to 5 Room types, executives and multigeneration
block	Discrete Numerical	Represents block no of the flat
street_name	Nominal categorical	Represents the street name
storey_range	Ordinal categorical	It is classified based on a group of 3 level. Starts from Level 1 to Level 3 till Level 49 to Level 51
floor_area_sqm	Continuous Numerical	Ranges from 39sqm to 249 sqm with mean value of 97.81sqm
flat_model	Nominal ordinal	Divided into 20 types of flat model based on luxurious
lease_commence_date	Discrete Numerical	Ranges from the year 1966 to the year 2019
remaining_lease	Discrete Numerical	Ranges from 43 years and 9 months to 97 years and 9 months
resale_price	Continuous Numerical	Ranges from $140K to $1.39M with mean of $464K

The resale_price is identified as the target variables of the machine learning problem.

PROBLEM STATEMENT

Housing prices are always an issue for Singaporeans. The rise in property prices over the years has benefited Singapore in many ways, such as increasing investment in property, raising asset quality in the banking sector and increasing consumer spending. However, they have also had negative impacts, such as increasing banking risk, economic instability and making homes unaffordable for first-time buyers.

Therefore, property price forecasting is necessary for the banks, investors, and authorities to alleviate the problems. The banking sector relies on the forecast to increase mortgage loans; investors decide whether to increase their investments in real estate based on the forecast; authorities decide whether to suppress or encourage the real estate market based on the forecast.

In this project, we aim to build a predicting model for the HDB resale price in Singapore with the dataset of resale prices of Singapore public housing (HDB flats) from January 2017 to April 2022. The ideal output would be the suggested HDB resale price while the input would be the location, date of transactions, size and model of the flat and remaining lease. It is a supervised learning task.

DATA PREPARATION

Preliminary data cleansing including duplicated instance removal is performed on the dataset.

To reduce the scale of the machine learning to manageable levels, the data is resampled to 30,000 instances.

Feature engineering has been performed to transform the raw data into features to be used in the supervised machine learning task. The description of features after transformation is as follows:

Table 2 Descriptions an transformation done on feature variables

Attributes	Analysis and Transformation
month	The month is transformed into integer that represent the number of months since Jan 2017, which is the earlier month in the dataset.
flat_type	The flat type is encoded to integer based on the order of functionality, size, and number of rooms. The information is available at the HDB Flat Portal. The order of encoding is {0: '1 ROOM', 1: '2 ROOM', 2: '3 ROOM', 3: '4 ROOM', 4: '5 ROOM', 5: 'EXECUTIVE', 6: 'MULTI-GENERATION'.
block	The block number is parsed as numbers.
storey_range	The storey levels are grouped into ranges of 3 levels at each group. The values are encoded to integer. For example: {0: '04 to 06', 1: '07 to 09', …}
floor_area_sqm	The floor areas are already numerical values. No additional transformation is performed.
flat_model	Flat model is encoded to integer based on the implicit order of luxuriousness. The information is available at the HDB Flat Portal.
lease_commence_date	The lease commence year was given. Therefore, the number of months since the house is commenced is calculated. For example, a house commenced in 2005, the months since commenced by the month Jan 2017 will be 144 (12 years * 12 months).
remaining_lease	The text is processed to transform the values into number of months for remaining lease.
region_central	The location (town and street) of the houses is mapped to five urban planning subdivisions as demarcated by the Urban Redevelopment Authority (URA) of Singapore - Central Region, East Region, North Region, North-East Region, and West Region. The feature is one-hot encoded into five binary features.
region_east	
region_north	
region_north_east	
region_west	

Various methods to perform further feature scaling have been tested in Experiment 1. The best improvement of performance is observed when each feature is scaled to range between 0 and 1, by using MinMaxScaler.

Therefore, the same method of feature scaling will be applied in Experiment 2 to 4.

TARGET VARIABLE DISTRIBUTION ANALYSIS

The distribution of the target variable (resale_price) is analyzed. The data is randomly resampled to 30,000 instances as mentioned above. After the resampling, the original data distribution trend is generally retained.

From the histogram below, the target variable is not normally distributed. The data skewed to the left, i.e., more instances having lower resale prices than having high resale prices. Most instances have resale prices of around SGD $ 400,000.

The outliers are determined by calculating the quartiles and interquartile range (IQR) of the data distribution. In the dataset, there are 838 outliers found, which amounts to 2.79% of the total data. All the outliers are above the upper bound, I.e., with resale prices higher than SGD$ 845,000. Hence, it will make more sense to refer to RMSE to measure the error of the model.

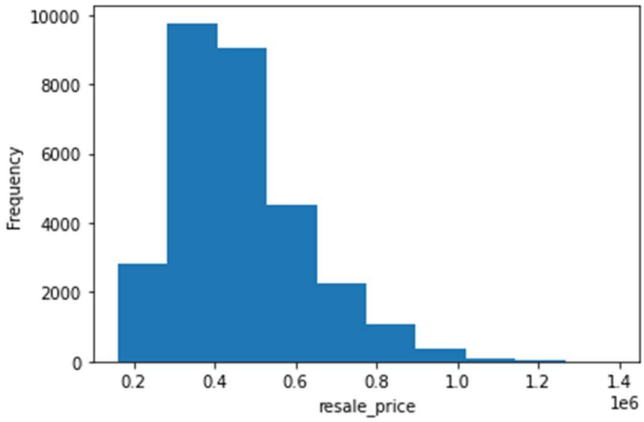


Figure 1 The data distribution on target variable

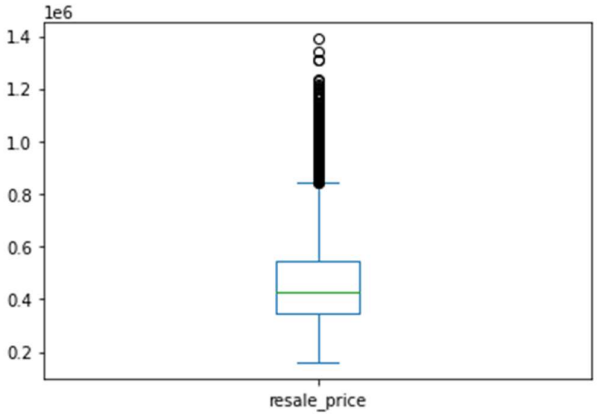


Figure 2 Box plot on the distribution of target variable

Table 3 Calculations of quartile to determine outliers

First Quartile (Q1)	345,000
Third Quartile (Q3)	545,000
Interquartile range (IQR) = Q3 - Q1	200,000
Lower bound = Q1 – (IQR * 1.5)	45,000
Upper bound = Q3 + (IQR * 1.5)	845,000
Outlier count	838
Outlier %	2.79333%

EXPERIMENT SETUP

EXPERIMENT 1: COMPARING MACHINE LEARNING ALGORITHMS

The goal of this experiment is to find the most suitable machine learning algorithm. The machine learning algorithm chosen to be explored in this section are KNN Regression, Decision Tree Regression and Linear Regression.

The experiment set up of experiment 1 is summarized as follows:

Table 4 Process of Experiment 1

Steps	Description
1	Load the processed data
2	Represents the data in three forms: raw data, normalized data, and standardized data. (Feature scaling only applied at features)
3	Set up the target and feature variables
4	Separate the data into training set, development set and test set
5	Create baseline regressor as a point of comparison with another algorithm (highlighted in blue)
6	Run the algorithm and tune it with different parameters
7	Record the result
8	Repeat Steps 6 and 7 with different algorithms

Based on the result in table 5, all the algorithms perform better than the baseline algorithm. Best performed algorithm is KNN Regression with parameter Manhattan (k=3). The difference between test set and validation set is not significant, indicate the model does not suffer from overfitting. The linear regression achieves the least accuracy among three of the algorithms. From the figure 3, it shows that it tends to be unpredictable at higher end.

Figure 3 Performance of Linear Regression

From the table 5, it can observe that feature scaling has made improvement to the model. This is due to feature scaling has reduced the difference in values between features, avoid the algorithm puts more weight on the features with larger value.

It is also observed that feature scaling does not offer decision tree and linear regression significant improvement. In the case of decision tree, feature scaling does not work on it since it is insensitive to variance in data. The split of nodes depends on the homogeneity of features rather than scale of data. In case of linear regression, feature scaling does not change the way it fits to the model, thus the result remains. On the other hand, feature scaling improves the performance of KNN algorithm because it utilizes distance to determine data similarity.

Table 5 Result of algorithm performance

Feature Scaling	Machine Learning Algorithm	Parameters	Validation Option	Validation R2	Validation MAE	Validation RMSE	Test R2	Test MAE	Test RMSE
None	DummyRegressor	strategy = mean	Percentage Split (64% Train, 16% Validation, 20% Test)	0.00	123,457.84	158,876.45	0.00	124,458.07	160,908.80
None	KNN Regression	Manhattan (k = 2)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.85	39,246.26	62,423.71	0.83	40,921.50	65,516.62
		Euclidean (k = 2)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.83	40,436.28	64,595.72	0.82	42,023.17	67,936.10
	Decision Tree	Mean Square Error (D = 16)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.88	35,471.68	56,148.72	0.87	36,778.45	58,929.90
		Absolute error (D = 17)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.88	33,457.20	54,601.34	0.87	34,679.94	57,503.35
	Linear Regression	NA	Percentage Split (64% Train, 16% Validation, 20% Test)	0.76	61,010.40	78,551.35	0.76	61,666.22	79,229.98
Normalization	KNN Regression	Manhattan (k = 3)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.91	32,121.39	47,826.02	0.91	32,302.83	49,441.72
		Euclidean (k = 4)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.90	34,479.91	51,406.93	0.89	35,601.21	53,576.74
	Decision Tree	Mean Square Error (D = 16)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.88	35,406.06	55,941.89	0.87	36,684.47	58,688.34
		Absolute error (D = 15)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.88	33,505.09	54,614.84	0.87	35,132.45	56,911.37
	Linear Regression	NA	Percentage Split (64% Train, 16% Validation, 20% Test)	0.76	61,010.40	78,551.35	0.76	61,666.00	79,229.98
Standardization	KNN Regression	Manhattan (k = 4)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.91	32,845.61	48,400.07	0.90	33,357.39	50,309.11
		Euclidean (k = 5)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.89	35,620.83	52,464.08	0.89	35,253.47	52,765.20
	Decision Tree	Mean Square Error (D = 16)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.88	35,210.30	55,541.76	0.87	36,599.19	58,620.43
		Absolute error (D = 15)	Percentage Split (64% Train, 16% Validation, 20% Test)	0.89	33,688.93	53,551.00	0.88	35,056.28	56,715.06
	Linear Regression	NA	Percentage Split (64% Train, 16% Validation, 20% Test)	0.76	61,010.40	78,551.35	0.76	61,666.22	79,229.98

EXPERIMENT 2: FEATURE SELECTION

Here, I will choose three methods for feature selection, and then compare the results of these choices after model training to determine the best feature combination. Pearson coefficient feature selection method (filtering method), decision tree model feature selection method (embedding method), RFE feature selection method (wrapping method).

The first is Pearson coefficient feature selection: The principle is to calculate the Pearson correlation coefficient of each feature to the target value and the P-value of the correlation coefficient and select the features with significant correlation.

This filtering approach uses statistical indicators to score and filter each feature, focusing on the characteristics of the data itself. Its advantage is that the calculation is fast and does not depend on the specific model. The disadvantage is that the selected statistical indicators are not customized for a particular model, so the final accuracy may not be high. Moreover, because the univariate statistical test is carried out, the correlation between features is not considered, and the interaction items cannot be selected.

Table 6 Results by Pearson coefficient feature selection

Threshold	selected columns	R2	RMSE
0.1	['month', 'flat_type', 'block', 'storey_range', 'floor_area_sqm', 'flat_model', 'lease_commence_date', 'remaining_lease', 'region_central', 'region_north', 'region_west'] (11 columns)	0.86	59963.34
0.15	['month', 'flat_type', 'storey_range', 'floor_area_sqm', 'flat_model', 'lease_commence_date', 'remaining_lease', 'region_central', 'region_north'] (9 columns)	0.79	73542.40
0.2	['flat_type', 'storey_range', 'floor_area_sqm', 'flat_model', 'lease_commence_date', 'remaining_lease', 'region_central'] (7 columns)	0.75	80768.66
0.3	['flat_type', 'storey_range', 'floor_area_sqm', 'flat_model', 'lease_commence_date', 'remaining_lease'] (6 columns)	0.54	108776.20
0.35	['flat_type', 'storey_range', 'floor_area_sqm', 'flat_model'] (4 columns)	0.65	94569.19
0.4	['flat_type', 'floor_area_sqm', 'flat_model'] (3 columns)	0.62	99079.00
0.6	['flat_type', 'floor_area_sqm'] (2 columns)	0.49	115376.24

The second is the selection characteristics of the decision tree model: Currently, Gini coefficient or information gain is generally used for classification problems, and RMSE (root mean square error) or MSE (square error) is generally used for regression problems.

This embedded method makes use of the characteristics of the model itself, embedding feature selection into the process of model construction. The accuracy is high, and the computational complexity is between filtering method and wrapping method, but the disadvantage is that only part of the model has this function.

Table 7 Results by model-based feature selection

Threshold	selected columns	R2	RMSE
0.01	['month','block','storey_range','floor_area_sqm','flat_model','lease_commence_date','remaining_lease','region_central', 'region_north'] (9 columns)	0.85	62034.85

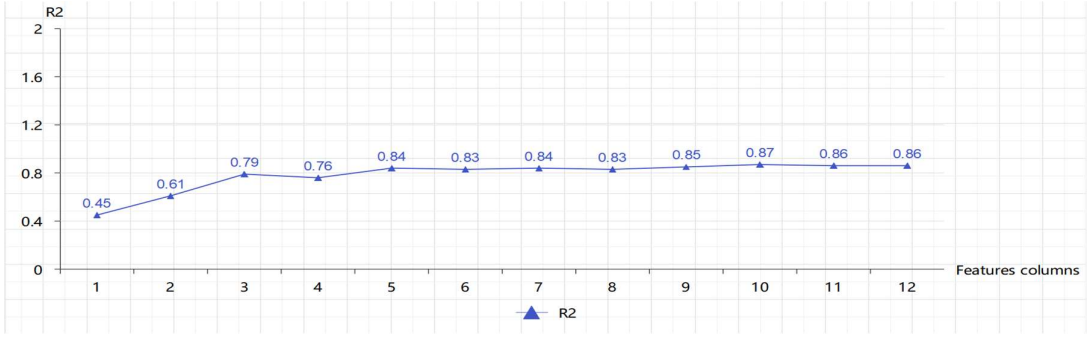
0.02	['month','block','storey_range','floor_area_sqm','flat_model','lease_commence_date','remaining_lease','region_central'] (8 columns)	0.83	66710.92
0.05	['floor_area_sqm','flat_model', 'region_central'] (3 columns)	0.79	73476.45
0.1	['floor_area_sqm','flat_model','region_central'] (3 columns)	0.79	73474.31
0.2	['floor_area_sqm'] (1 column)	0.45	119646.4
mean	['floor_area_sqm', 'flat_model', 'region_central'] (3 columns)	0.79	73476.45
median	['month','block','floor_area_sqm','flat_model','lease_commence_date', 'remaining_lease', 'region_central'] (7 columns)	0.84	64338.36

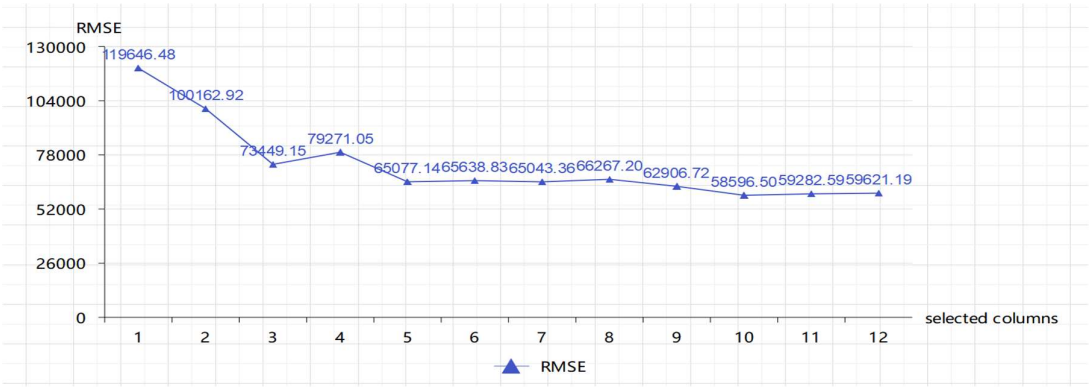
The third is the RFE feature selection method: This wrapping approach uses the model to filter features by constantly adding or deleting features and testing model accuracy on the validation set to find the optimal feature subset. Because of the direct participation of the model, the wrapping method usually has high accuracy. However, because the model needs to be retrained every time a feature is changed, the calculation cost is high. Another disadvantage of the wrapping method is that it is easy to overfit.

RFE： RFE ranking: [7 12 5 8 1 2 6 4 3 11 9 13 10]

1	floor_area_sqm	6	remaining_lease	11	region_east
2	flat_model	7	month	12	flat_type
3	region_central	8	storey_range	13	region_north_east
4	lease_commence_date	9	region_north		
5	block	10	region_west		

Table 8 Results by RFE feature selection





By observing the three models, RFE with the highest R2 score is 0.87, so we choose the corresponding top 10 features.

EXPERIMENT 3: EMSEMBLE LEARNING

In this part, we will examine if ensemble learning can help improve the model's performance.

To maintain the fairness of the result, this part uses the same learning algorithm as in Experiment 1, namely K-Neighbors Regressor, Decision Tree Regressor and Linear Regression.

The 3 ensemble learning methods used in this project are:

1. Stacking

For stacking, K-Neighbours Regressor and Decision Tree Regressor are used as Level 0 and Linear Regression is used as Level 1.

2. Blending

For blending, K-Neighbours Regressor, Decision Tree Regressor and Linear Regression are used to train the dataset.

3. Boosting - XGBoost

XGBoost (extreme Gradient Boosting) was chosen for ensemble learning in this project because XGBoost is an advanced implementation of the gradient boosting algorithm that has high predictive power and is almost 10 times faster than the other gradient boosting techniques.

The results of this experiment can be found in the following table:

Table 9 Results of Experiment Set 3

Regression	Validation R2	Validation MAE	Validation RMSE	Test R2	Test MAE	Test RMSE
KNN	0.914	32,123.860	47,826.324	0.910	32,302.830	49,441.720
DTREE	0.884	35,226.976	56,046.878	0.870	36,684.470	58,688.340
LINEAR REGRESSION	0.760	61,010.400	78,551.350	0.760	61,666.000	79,229.980
STACKING	0.920	30,024.102	44,926.325	0.925	28,984.851	44,014.570
BLENDING	0.866	58,121.492	34,806.092	0.910	48,261.393	31,813.520
BOOSTING - XGBOOST	0.939	39,312.303	27,219.183	0.940	39,478.857	27,434.476

For a better comparison, the results of Experiment 1 are listed in the first three rows of the table above. The best results are highlighted in yellow.

In general, all three ensemble learnings improve the model when training the dataset in our project. Among the three ensemble learnings used, XGBoost gives the best result, either among all ensemble learning methods or compared to the algorithm without ensemble learning.

This could be due to the fact that boosting is an ensemble learning method that combines several weak learners into one strong learner to minimise training errors.

EXPERIMENT 4: VARYING TRAINING SAMPLE SIZE

The experiment is to examine the correlation between quantity of training data and the model's performance.

The method of incrementing the training sample is as follows:

1. Randomly subsample 10% of the full training data and perform training and testing of models.
2. Randomly subsample another 10% of the training data from the remaining 90% of training data. The subsample is added to the previous subsample, resulting in a subsample of 20% of total training data. Use the subsample to perform training and testing of models.
3. Repeat steps 1 and 2 until all training data is used.

The three best models from Experiment 1 to 3 are selected for evaluation:

1. XGBoost
2. Stacking (Base models: kNN and decision tree, meta model: linear regression)
3. Blending (Base models: linear regression, kNN, and decision tree; meta model: linear regression)

Table 10 Results of Experiment Set 4

				Validation			Testing		
Feature Scaling	Regressor	Feature Subset	Train Data %	R^2	RMSE	MAE	R^2	RMSE	MAE
MinMax Scaler	XGBoost	Full	10	0.870	57208.960	39484.992	0.873	57340.535	39262.014
			20	0.900	50253.718	34222.654	0.899	51024.619	34458.888
			30	0.907	48314.712	32814.766	0.907	49078.033	33196.455
			40	0.917	45782.248	31541.898	0.916	46494.319	31882.367
			50	0.924	43902.398	30081.708	0.924	44409.410	30576.456
			60	0.925	43484.593	29575.441	0.927	43344.075	29628.851
			70	0.930	41918.682	28818.014	0.930	42405.836	29290.212
			80	0.934	40706.053	28380.434	0.933	41489.745	28527.866
			90	0.936	40271.748	27882.142	0.935	41040.266	28293.609
			100	0.939	39312.303	27219.183	0.938	40058.954	27674.691
MinMax Scaler	Stacking	Full	10	0.840	63574.889	43732.282	0.845	63384.408	43261.875
			20	0.868	57798.097	39476.160	0.872	57532.668	38916.757
			30	0.882	54481.448	36970.580	0.881	55469.040	36992.004
			40	0.891	52398.862	35383.499	0.892	52937.930	35123.826
			50	0.896	51111.231	34127.132	0.895	52042.067	34269.226
			60	0.907	48334.478	32517.191	0.904	49761.834	32767.455
			70	0.910	47596.653	31801.219	0.906	49236.006	32255.862
			80	0.915	46377.912	31021.339	0.912	47838.479	31408.672
			90	0.916	45943.740	30908.274	0.915	46969.622	31017.219
			100	0.921	44674.626	29994.775	0.916	46630.689	30594.637
MinMax Scaler	Blending	Full	10	0.834	64728.006	45932.024	0.840	64338.853	46091.493
			20	0.857	60064.034	41981.381	0.860	60214.972	42145.417
			30	0.874	56448.640	39099.897	0.874	57015.834	39550.760
			40	0.877	55625.356	38095.958	0.878	56150.575	38491.176
			50	0.884	54077.915	36335.464	0.888	53925.241	36354.616
			60	0.893	51878.536	34941.516	0.891	52987.918	35391.802
			70	0.897	50958.192	34400.833	0.896	51970.338	34872.866
			80	0.901	50048.879	33733.645	0.899	51062.751	33847.824
			90	0.905	49016.988	32775.655	0.906	49253.845	32931.466
			100	0.908	48168.193	32165.023	0.909	48475.096	32192.129

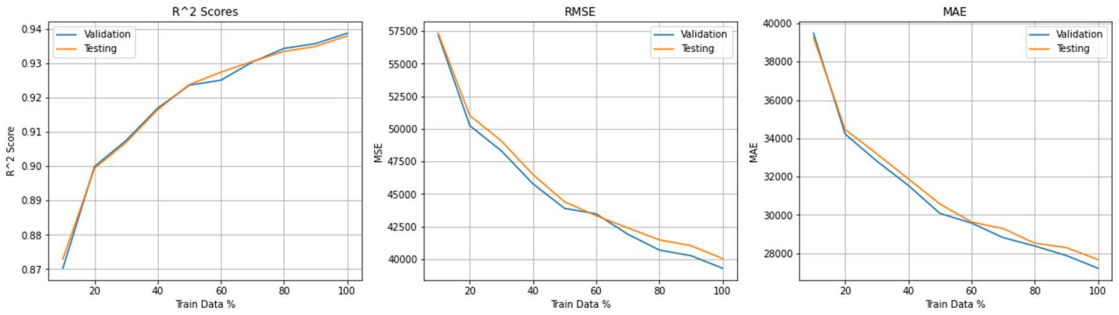


Figure 4 XGBoost - Scores vs. training data %

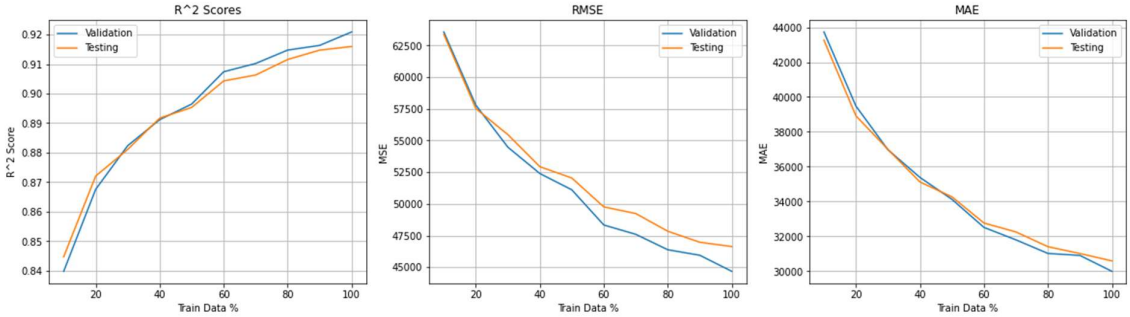


Figure 5 Stacking - Scores vs. training data %

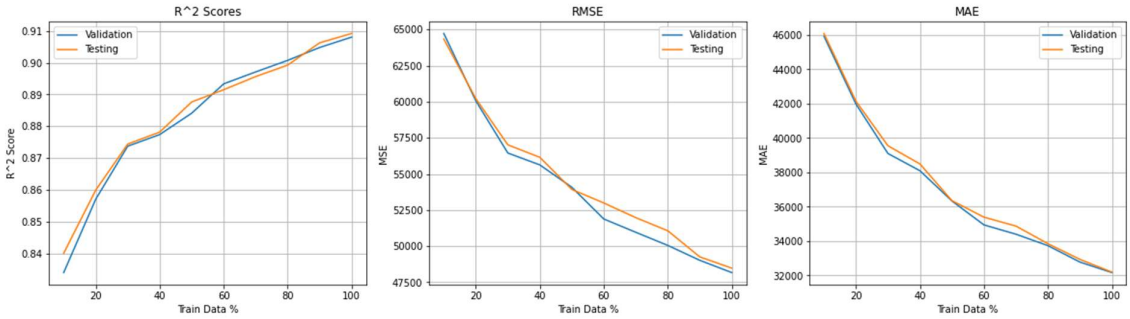


Figure 6 Blending - Scores vs. training data %

From the experiment above, we can see that the performance improved with the amount of training data size. This may be due to more training data can better represent the distribution of the whole dataset. We may expect that the performance can be further improved by adding more data samples, the models may converge at an optimum before becoming overfit. Further experiments can be carried out by training the models with even more training data to validate the hypothesis.

CONCLUSION

Few conclusions can be made from the 4 experiments we conducted in our project. First, feature scaling has improved the model and the KNN regressor gives a better result than the decision tree regressor and linear regression in predicting Singapore housing prices in our project. Second, feature selection does not really help to improve model performance. Third, ensemble learning helps improve the performance of the model, with boosting (XGBoost) giving the best result. Last but not least, increasing the size of the dataset in this project also helps to improve the performance of the model.