**Singapore HDB: Resale Price per Square Meter Prediction**

**Team 4:** Li Rongrong, Zhang Tieyang, Soo Yik Hong

**1. Introduction (Zhang Tieyang, Li Rongrong)**

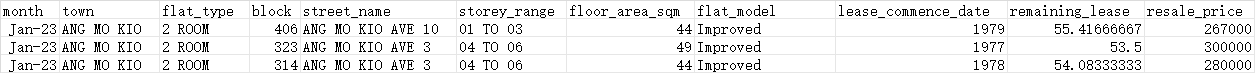
Housing Development Board (HDB) flats are a vital component of the housing market in Singapore, which provided by the government to offer safe and affordable housing options for Singaporen. Predicting the price per sqm of Singapore HDB has real-world significance, for instance, help people decide when to buy or sell a property for the best return, or ensuring social equity in the market and low-income families have chance to buy a suitable housing.

In this report, we will describe the dataset used for this project, how we attempted to get a comprehensive understanding of the dataset, which machine learning models we used, and some comparison and interpretation of our models.

**2. Dataset & Pre-processing (Zhang Tieyang)**

***2.1 Download dataset***

we got our original dataset from the website of Singapore government (<https://beta.data.gov.sg/>), which has following features:



1) ***‘month’***: the date of resold

电脑屏幕的照片上有文字

描述已自动生成2) ***‘town’***: the town where the resold housing is located

3) ***‘flat\_type’***: the flat type of the resold housing

4) ***‘block’***: the block of the resold housing

5) ***‘street\_name’***: the name of the street where the resold housing is located

6) ***‘storey\_range’***: the range of storey of the resold housing

7) ***‘floor\_area\_sqm’***: the area of the resold housing

8) ***‘flat\_model’***: the flat model of the resold housing

9) ***‘lease\_commence\_Date’***: Remaining lease term of the house (year)

10) ***‘remaining\_lease’***: the remaining lease of the resold housing

11) ***‘resale\_price’***: the resale price of the resold housing

We only use data of year 2023 for our project (20429 rows).

***2.2 Expansion of dataset***

To improve the performance of models, we first consider finding additional features that may influence the price per sqm. We first considered to get the latitudes and longitudes of all housing using its ‘street\_name’ and ‘block’. Meanwhile, noticed that the price may be related to the distance of a certain housing to the nearest MRT station and to a popular primary school, we also decided to collect the latitude and longitude of all MRT and LRT stations as well as the famous primary school in Singapore. We then found a dataset from Kaggle contains the geolocation of all stations (<https://www.kaggle.com/datasets/yxlee245/singapore-train-station-coordinates>), and with the help of OneMap API (<https://www.onemap.gov.sg/>), we manage to obtain the following features for each housing:

1/2) ***‘lat/lng’***: Latitude and longitude of the housing

3/4) ***‘min\_mrt\_lat/lng’***: Latitude and longitude of the nearest MRT/LRT station.

5/6) ***‘min\_primary\_lat/lng’***: Latitude and longitude of the nearest popular primary school.

表格

描述已自动生成表格

描述已自动生成表格

描述已自动生成

***2.3 Create dummy variables***

To utilize the columns with qualitative features, we created one hot encoding of categorical variables, made them all to dummy variables. Example:

表格

描述已自动生成

***2.4 Drop outliers / missing values***

To get better performance, we try to drop the outliers (Interquartile Range):

图表, 箱线图

描述已自动生成

Then we try to find rows with missing value. Because the dataset has enough samples, we directly drop all those rows, and we still have 96.5784% data of the original dataset after dropping, which is enough for training models (19370 rows).

***2.5 Get price on per sqm basis***

We divided the resale price by the area to create a new feature ‘resale\_price\_per\_sqm’, which is our new target variable of prediction.

**3. Feature Engineering & Visualization (Zhang Tieyang, Li Rongrong)**

***3.1 Add transit time & distances to station / primary school***

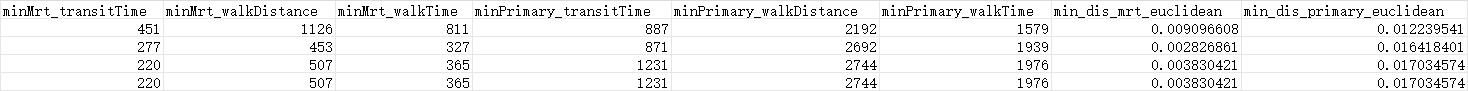
Since we already have the latitude and longitude of the nearest station and good primary school, we then used them to create new features. First, we calculate the Euclidean distance from a certain housing to its nearest station and popular primary school using their latitude and longitude. After that, we used the same API from OneMap to get the transit time, walking time, and walking distance as well, since sometimes, smaller Euclidean distance does not always mean shorter transit/walking time. Finally, we add these features to our dataset. Examples:

1) ***‘min\_dis\_mrt\_euclidean’***: Euclidean distance to the nearest station calculated from latitude and longitude.

2) ***‘minMrt\_transitTime’***: Time to reach the nearest station by public transportation.

3) ***‘minMrt\_walkTime’***: Time to walk to the nearest station.

4) ***‘minMrt\_walkDistance’***: Distance of walking to the nearest station.



***3.2 Drop useless column***

For the future model training process, we need to drop some columns that has been used to generate new features, but is useless for training. So we dropped columns ‘month’, ‘street\_name’, and ‘block’.

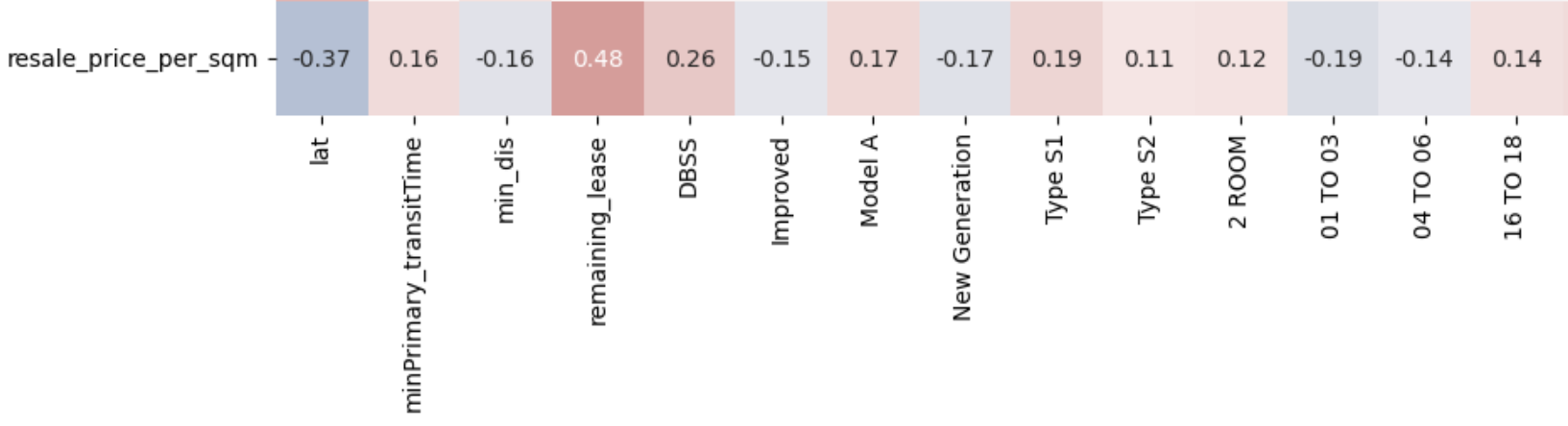
***3.3 Data size and structure***

Our final version of dataset has 19730 rows and 87 columns, most of which are dummy variables. Data types including int64, float64, and Boolean.

***3.4 Covariance matrix and feature selection***

Covariance is a statistical measure that quantifies how two variables change together, by generating covariance matrix for our dataset, we can visualize which variables are correlated with ‘resale\_price\_per\_sqm’. We first noticed there are many features that has very little effect on price, and some of them are highly correlated with each other. Finally, Taking correlations of 0.1 and 0.168 (absolute value) as thresholds, we selected two sets of feature sets:

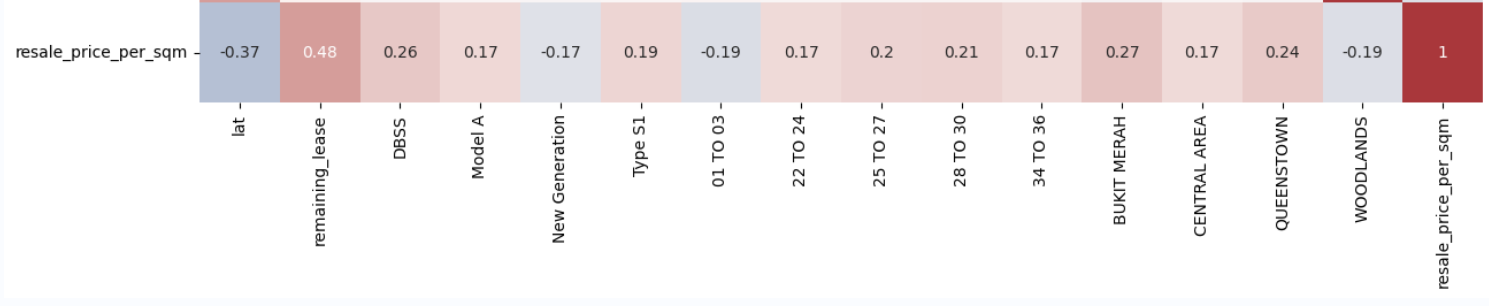
**Feature set 1** with higher complexity, has **33** **features** in total:



图片包含 图表

描述已自动生成

**Feature set 2** with lower complexity, has **15 features** in total:



***3.5 Map / Plot to verify the insights***

Our new feature ‘lat’ does play a important role in predicting price per sqm, the lower the latitude, the higher the price per sqm; Vice versa.

地图

描述已自动生成

|  |  |
| --- | --- |
| Ordinary Linear Regression | MAE |
| 33 Features include ‘lat’ & ‘minPrimary\_transitTime’ | 506.50 |
| 32 Features exclude ‘lat’ | 550.99 |
| 31 Features exclude ‘lat’,‘minPrimary\_transitTime’ | 551.04 |

Also, ‘remaining\_lease’ is highly correlated with price per sqm. The longer the remaining lease, generally the higher the price per sqm.

图表, 散点图

描述已自动生成

**4. Modelling & Optimization**

We use feature set 1 with 33 features and feature set 2 with 16 features.

***4.1.1 Linear Regression (Soo Yik Hong)***

A close-up of a document

Description automatically generated

***Result***

Table 1. Features with best linearity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Type | Column name | Estimate | Std. Error | t\_value | Pr (> | t |) |
| numerical | Remaining\_lease | 805.1 | 7.795 | 103.288 | < 2e-16 |
| binary | 40 TO 42 | 2706.8 | 171.031 | 15.827 | < 2e-16 |

The linear regression model analysis revealed that the binary feature denoted as ‘40 TO 42’ possesses the largest coefficient of 2706.8 when its value is TRUE. Both features in the table above have the most significant positive effect on resale price, and they are statistically significant. The calculated RMSE of 670.4898 is indicative of the model’s predictive performance and signifies that there is room for improvement.

***4.1.2 PCA + Linear Regression (Soo Yik Hong)***

A black text on a white background

Description automatically generated

***Result***

Table 2. Variance ratio of each PC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PC 1 | PC 2 | PC 3 | PC 4 |
| Variance ratio | 0.35259718 | 0.25444489 | 0.23506246 | 0.15789547 |

Based on table 2 above, the minimum principal components used for regression modelling must be 4 to cover more than 95% of the variance in the training data. The PCA implies that the numerical columns: ‘lat’, ‘minPrimary\_transitTime’, ‘min\_dis’ and ‘remaining\_lease’ are equally important and cannot be reduced dimensionally. In addition, the RMSE will be larger if fewer principal components (<4) are used for modelling.

***4.1.3 Linear Regression + Ridge/Lasso Regression (Soo Yik Hong)***

A close-up of a test

Description automatically generated

***Result***

Table 3. Ridge parameters

|  |  |
| --- | --- |
|  | Value |
| Max. λ | 1396 |
| Best. λ | 1.6755 |

The Ridge regression model demonstrates a 9.6% improvement over ordinary linear regression, addressing multicollinearity through pre-processing. Removal of highly correlated features reduces L2 penalty effect, yielding a modest enhancement with an RMSE of 650.4424. Lasso regression, employing same set of lambda values, achieves a comparable RMSE (650.9404) but exhibits overfitting with the smallest lambda (1e-6). The L1 penalty forces some coefficients to zero, beneficial for high-dimensional data but unsuitable here.

**Conclusion**

|  |  |
| --- | --- |
| Model | Test Error (RMSE) |
| Linear Regression | 670.4898 |
| PCA + Linear Regression | 670.4898 |
| Linear Regression + Lasso regularization | 650.9404 |
| Linear Regression + Ridge regularization | 650.4424 |

The best model for the project is the combination of linear regression with Ridge regularization, as it exhibits a balance by effectively reducing the multicollinearity between numerical and binary categorical columns, resulting in improved performance without overfitting.

***4.2 Decision tree (Li Rongrong)***

According to code results in process trying different models to fit, it was found that tree-based models performance were better than linear models generally. To select which model is the most appropriate, we started from decision tree. In the *sklearn* library, many parameters of the tree can be adjusted:

*ccp\_alpha*: parameters for cost complexity pruning

*max\_depth*: maximum depth of the tree

*max\_features*: maximum number of features to consider when splitting each node

*max\_leaf\_nodes*: maximum number of leaf nodes allowed

*min\_samples\_leaf*: minimum number of samples on leaf nodes

Using bagging to increase effect of decision tree is also a good choice. Thanks to convenience of the *sklearn* library, we could building and try it easily. Because parameters between decision tree and bagging are highly similar, we do not introduce them again. To avoiding overfitting, we used 5-fold cross-validation during building all tree-based models.

After many adjustments, the best parameter and error are shown in the table below.

|  |  |  |
| --- | --- | --- |
| Model | The best parameters set | MAE |
| Decision tree | {'max\_depth': [15], 'min\_samples\_split': [17],  'min\_samples\_leaf': [2], 'ccp\_alpha':[0.1], 'random\_state':[817]} | 323.2293 |
| Bagging | {"n\_estimators":[285], "max\_features":[28], 'random\_state': [817]} | 271.6265 |

***4.3 Random forest (Li Rongrong)***

Random forest is convergence of different decision tree, a special situation of bagging. If bagging perform well, random forest can be predicted having great effect. Like decision tree, *sklearn* library also provide many parameters for users to adjust. Because these parameters are highly same as Decision tree, here we only introduce some features that are different from decision trees:

*bootstrap*: specify whether to use bootstrap sampling for training

*max\_depth*: the maximum depth of each tree contained in a random forest is also the same as a single decision tree.

*n\_estimators*: number of base decision trees included in the random forest

In the table showing improvement process, all parameters adjusted are displayed.

|  |  |  |
| --- | --- | --- |
| Model | The best parameters set | MAE |
| Random forest | {"n\_estimators":[290], "max\_features":[21],  'ccp\_alpha': [0.2],'max\_depth': [21], 'random\_state': [817] } | 271.1573 |

***4.5 Neural Network: Back Propagation – Mini-Batch Gradient Descent (Zhang Tieyang)***

In theory, neural networks have the capability to approximate almost any function. However, such a powerful ability comes at the cost of sacrificing interpretability and longer training times.

For this project, also as the first time working on NN, we visited the website for TensorFlow v2.14.0 at <https://www.tensorflow.org/api_docs> and looked up for all the available parameters. Finally, we decided to attempt training the NN using stochastic descent algorithm with mini-batches in the backpropagation process.

Our NN has 1 input layer, 1 hidden layer, and 1 output layer, the parameters we decide to tune for it are shown below:

*Optimizer:* the algorithm we used for training, we only tried 2 of them: ‘\*SGD’ & ‘\*Adam’

*Metrics:* the loss function (calculate the gradients) we used for training, we only tried 2 of them: ‘mae’ & ‘mse’

*Units:* the # of unit of hidden layer, we tried 7, 8, 9

*Batch\_size:* the # of batches we used for each epoch, we tried 16, 32, 64

*Learning\_rate:* rate for both optimizers to change the weights in NN, we tried 0.008, 0.01 for ‘sgd’ + ‘mae’; 0.0008, 0.001 for ‘adam’ + ‘mse’

*Model:* we tried feature group with size 33 & 15

*Activation function:* always use ‘relu’

The best result appears when using sgd + mae with model of 33 features. We then record the best model with 8 units for hidden layer, 32 batches for each epoch, and learning rate of 0.01 using sgd algorithm and mean absolute error as metric.

Store this model into ‘best\_model.h5’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| units | batch\_size | learning\_rate | minimum\_mae\_error | R^2 |
| 8 | 32 | 0.01 | 381.7575 | 0.842157 |

1. **Prediction & Interpretation (Zhang Tieyang, Li Rongrong)**

***5.1 Comparison between best models***

***Validation set approach:***

|  |  |  |
| --- | --- | --- |
|  | MAE | R^2 |
| Ordinary Linear regression | 506.496 | 0.757 |
| Linear Regression - Ridge | 509.611 | 0.754 |
| Linear Regression - Lasso | 509.632 | 0.754 |
| Decision Tree | 349.386 | 0.871 |
| Bagging | 280.463 | 0.921 |
| Random Forest | 280.729 | 0.921 |
| FNN(Mini Batch-SGD) | 381.757 | 0.842 |

***Cross-Validation (5-fold):***

|  |  |
| --- | --- |
|  | *MAE* |
| Linear Regression - Ridge | 508.593 |
| Linear Regression - Lasso | 508.623 |
| Decision Tree | 351.592 |
| Bagging | 288.216 |
| Random Forest | 284.332 |

**As a result, we concluded that Random Forest might be the best model for this project.**

***5.2 Discovery on features importance***

For the question “Adding which features could improve the most when using only 5 features (whose covariance > 0.3) fitting model”, if it only allowed to add one variable, it would be '01 TO 03’, a dummy variable converted from storey range. If it only allowed to add two variables, it would be '01 TO 03’ and '04 TO 06’. This shows that apart from the features that have the greatest influence to resale price, the condition of the HDB flat itself has a greater impact on its selling price than external factors.

This discovery can be easy to explain: when people make decision on purchasing what kinds of flats, they firstly care about flats themselves, such as their condition, their floor and their available area. After that, people will pay attention to other factors like flats location to judge whether living in these flats is convenient.

***5.3 Something unreasonable but appears***

Of course. As is known to us all, the longer remaining lease is, the higher price buyer should pay. But based on this data set, thing doesn't seem to be the case. (picture processed by Matlab)

图形用户界面

描述已自动生成

It is obvious that when the remaining lease exceeds a certain point between 80 and 90, the position of the dense area of the scatter plot does not rise but falls. Figures from code also show the same result: R-square of 2 degree polynomial (0.1925) is larger than linear model (0.1529). After investigation, the reason for this phenomenon is that Singapore’s policy stipulates that the sum of the remaining lease and the age of the purchaser cannot exceed a certain integer. Therefore, when remaining lease increasing to a high level, HDB must reduce its price to make sure someone is willing to buy it.

**APPENDIX**

***Which primary schools are “great”?***

Singapore's policy is to determine the primary school that children should attend according to the town. Naturally, parents want their children to go to the best school and they would rather purchase HDB flats near great primary schools. But which primary schools can be defined as “great”? We need a specific concept, or a numeric judgement to select primary schools. According to government policy, there are three phase of registration for primary schools in Singapore and every phase has unique requirements.

***Phase A***: Children whose siblings are studying will be given priority.

***Phase B***: Children’s parents who have a relationship with the school or have special social status.

***Phase C***: All citizens and non-PR international students.

About phases B and C, there is a ratio of the number of applicants admitted to the number of planned admissions in the related website. This percentage can indirectly reflect the popularity of the school, which could be regarded as a concept of “great”. We obtained 56 primary schools that not only appears in the top part of Phase B but also in Phase C and regard them as “great” primary schools.