Predicting HDB Prices in Singapore

Raimi bin Karim  
*A0082893R   
Team ARK  
raimi.karim@u.nus.edu*

Azmi bin Mohamed Ridwan   
*A0225575Y  
Team ARK  
e0576209@u.nus.edu*

Wang Tian Ming Kenneth  
*A0116351R  
Team ARK  
e0573193@u.nus.edu*

In this project, the team was tasked with developing a model to predict the resale prices of a HDB flat. There were auxiliary data and training data given to us to explore and construct the predictive model. We believe that there are many practical use cases for such a predictive model. The model provides property agents, buyers and sellers of resale property flats an opportunity to generate resale price predictions so that all parties can transact at a fair price. We believe that with more transparency, more resale transactions can be carried out with ease and this will benefit all parties in the resale market.

The dataset details the transactions in the last 20 years from January 2000 to November 2020 for 1-5 room, executive and multi-generation flats.

# EXPLORATORY DATA ANALYSIS

In this section, our aim is to better understand the data through Exploratory Data Analysis (EDA) to enable to us to process the data correctly and engineer the relevant features for the modelling pipeline. The following insights and plots are derived through our analysis of the training dataset.

There appears to be a clear distinction in the number of transactions before and after 2011 (refer to Fig X). This could be due to the cooling measures implemented by the government. In January 2011, the LTV limit for individuals with outstanding housing loans was reduced to 60%.

Graphical user interface, chart, line chart

Description automatically generated

1. A clear distinction in the number of transactions before and after 2011

We noticed that there is a visible upward trend in the resale prices across the years (refer to Fig X). This gives us the intuition that the resale prices are non-stationary and the date at which the resale flat transaction occured should be useful. The resale prices are also at an all-time high since the Global Financial crisis in 2008.

Chart, line chart

Description automatically generated

1. Visible upward trend in the resale prices across the years

In addition, there is a clear distinction in the resale prices across the different flat types (refer to Fig X). As the different flat types have differing flat sizes, we believe that their sizes could have contributed to the price disparity. An interesting observation is that the prices for multi-generational flats seem to fluctuate more aggressively.

Chart, histogram

Description automatically generated

1. A clear difference in prices for flats of different flat types

Furthermore, we noticed that there are outliers in the dataset that are situated on higher floors or have outsized floor areas. Based on this observation, we need to take select a model that is able to be resilient to these outliers. Tree based models present themselves as suitable candidates since they are generally more robust to outliers. This is because the tree based models are unlikely to pick extreme feature values (the outliers) when looking for a suitable split in the tree as these values rarely provide a positive information gain.

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

1. Flats that are situated on high floors or have outsized floor areas

We computed the correlation for numerical features and observed that the resale prices have a relative higher correlation with floor area (size of property), lease commence date (age of property) and storey. This gives us an upfront intuition that these features might be useful for our selected model to generate accurate predictions.

Table

Description automatically generated

1. Resale prices have a high correlation with “floor\_area\_sqm”, “lease\_commencement\_data”, “storey” features

Since there are a couple of location-based features that are given to us, we computed the Cramers' V coefficient amongst these features to see if there are any relationship between them. We noticed that these location-based features are higher correlated to one another. This could imply that we need to only select one of the location-based features for our modelling pipeline.

Graphical user interface, application, table

Description automatically generated

1. Resale prices have a higher Cramer’s V coefficient with location-based features

# DATA PREPROCESSING

There were no missing values in the training dataset given to us. While looking through the unique values across all the column features, we noticed that "elevation" and "eco\_category" columns contain only a single value. Since there are no variation in the column features, we chose to drop them as they unlikely provide any information useful for the model.

We have processed the different column features according to their data type. For example, based on our EDA exercise, we noticed that the range of resale prices differ according to the type of flat. Hence we have mapped the flat types to integer values based on their flat sizes with larger integer values mapped to larger flat sizes.

A picture containing diagram

Description automatically generated

1. "flat\_type" feature was mapped to integer values while preserving their ordinality characteristic

"planning\_area" is our selected location-based feature and has been processed by passing the column through a one-hot encoder.

**Strategies to deal with categorical features**

Categorical features by nature have a data type of string. Since most regression methods do not deal with string directly, we need to convert it to a numeric type.

Ordinal features can be easily converted to numbers starting with 0 and ending with the size of cardinality less 1.

However, handling nominal features can be tricky. Here are some experiments conducted using decision tree as the model. We use a tree-based model as it is nonlinear and it helps us get a baseline across the different strategies. The reported RMSE scores are based on the validation set.

*Drop nominal features*

RMSE: $50k - $55k

The reason for high RMSE is probably due to the lack of information to explain the target variable. Nominal features account for about 38% of our data (6 out of 16 columns). We conclude that dropping nominal features is a bad idea, because we are dropping possibly useful information.

*1-of-K encoding*

RMSE: $26k - $28k

Firstly, we drop features that have a cardinality of more than 100 or 200. Affected columns include street name and subzone. We see that this is an improvement from dropping the nominal features entirely. However, ensemble models like random forest tend to be very slow (because the total no. of columns blow up to more than 100).

*Label encoding*

RMSE: $19k - $21k

This converts the string column to numeric by assigning a numeric label ("label encoding"). Like the previous point, we first drop features that exceed a specified cardinality.

Then, we fit this data to models that can handle nominal features. The implementations we used that have such categorical support Gradient Boosting methods: HistGradientBoostingRegressor from scikit-learn and LGBMRegressor from LightGBM.

We see that this is an improvement from dropping the nominal features entirely. We see that this is an improvement from the 1-of-K method in terms of (1) speed and (2) score.

**Strategies to deal with missing values**

After performing a merge with the *HDB Property Information* dataset, we notice about 2000 transactions bear missing values on the new features, possibly because the property information dataset might not have information about these new flats. There are 2 ways to handle the NaN’s: dropping or keeping them.

*Drop missing values*

We did not experiment dropping these rows but we assume that it will reduce the performance of our models.

*Keep missing values*

We chose to keep these missing values and turn to tree-based Gradient Boosting implementations that handle missing values. This turns out to be the same implementations that were able to handle nominal features -- HistGradientBoostingRegressor from scikit-learn and LGBMRegressor from LightGBM. These implementations consider NaN’s as one of the values to consider while splitting. For example, the HistGradientBoostingRegressor learnsat each split point whether samples with missing values should go to the left or right child based on the information gain.

# FEATURE ENGINEERING

*Dataset*

We engineered the following features based on the given dataset:

* before\_covid – we believe that the distribution of resale prices change after Feb 2020, the period for Covid-19 lockadown in Singapore
* average\_storey – the average of the minimum and maximum from the given storey\_range feature
* is\_inauspicious – we block numbers which are 13, 4, 44 or 444 might be considered to be inauspicious to some
* remaining\_lease – this is the difference between the year and lease\_commencement\_date
* can\_use\_cpf – if remaining\_lease is less than 30

*Amenities*

Several auxiliary data representing different amenities were provided as part of the data package in addition to the transaction data. All of the data has positional information (longitude and latitude) which provides opportunities to create additional features based on the distance between a property and an amenity. The amenities provided are:

* Primary Schools
* Secondary Schools
* Significant commercial areas including the campuses for the major IHLs (Institutes of Higher Learnings)
* Markets and Hawker Centers
* Shopping Centers
* MRT and LRT stations

Three features were created for each of the amenity type (total of 18 new columns in the data). The coordinates were first converted from geodetic to cartesian to allow for easier calculations and easy interpretation of the distances (kilometres instead of degrees). For scalability, the amenities were stored in a Quadtree data structure which allows efficient searches for a given query location. These features are:

* Number of each type of the amenity within a 1 km radius of the property
* Number of each type of the amenity within a 2 km radius of the property
* Closest distance from a property to an amenity for each type

These three feature types are obviously very similar in nature, especially the first two. It is postulated that only one feature for each amenity will be kept once feature selection is done and the feature importance is calculated.

*Postal district*

We also postal district information based on information from the URA website here and 3rd party information here and the planning\_area feature.

* district
* prime\_district
* core\_central\_region

*HDB Resale Price Index*

We obtained the HDB Resale Price Index data from [here](https://data.gov.sg/dataset/hdb-resale-price-index). We mapped the data (consisting of quarter and price index) to the dataset’s transaction using a lag of 2 quarters. This gives us a new feature:

* price\_index

*HDB Property Information*

We obtained the HDB Property Information from [here](https://data.gov.sg/dataset/hdb-property-information). We mapped the data’s block and street name to the main dataset (this required us to use the common street abbreviations). This gives us the following features which we believe might help with the prediction:

* max\_floor\_lvl
* total\_dwelling\_units
* rental\_rooms
* building\_age

# DATA MINING METHODS

*Baseline model*

To track our progress, we decided to create a baseline model using the available features in the training dataset.

The following features were used as it is: "flat\_type", "lease\_commence\_data", "storey\_range", "floor\_area\_sqm", "planning\_area", "storey\_range". The "storey\_range" feature was used to generate two additional features "min\_storey" and "max\_storey" which represent the lower and upper bound of "storey\_range". "month" was used to derive a new feature "year" to represent the year at which the transaction happened. The other columns were dropped for simplicity. As mentioned in the data pre-processing section, "flat\_type" was mapped to integer values to preserve its ordinality characteristic and "planning\_area" was passed through a one-hot encoder.

Due to our observation that there are outliers in the "storey\_range" and "floor\_area\_sqm" columns, we are inclined to use tree-based models to generate our predictions as we believe that they are more resilient to handling outliers. For our baseline predictions, we ensembled the predictions of the "RandomForestRegressor" and "GradientBoostingRegressor" models to generate a final prediction for prediction. RandomSearchCV was first used to identify promising ranges of parameter values, before using GridSearchCV to narrow the search to specific parameters values. With the baseline models and features, we managed to obtain a RMSE score of $25,076 for the test dataset.

*Models with categorical support*

We used `HistGradientBoostingRegressor` as its implementation supports splitting of nominal features. We obtained a val RMSE of $17,000. This implementation is the fastest, with a runtime of about 30s.

Another implementation that supports nominal features is the LGBMRegressor from LightGBM. The best val RMSE obtained is $18,329. This implementation is much slower than HistGradientBoostingRegressor.

We also had an ensemble model of the following:

* HistGradientBoostingRegressor with 100,000 trees and L2 regularization 0.05
* HistGradientBoostingRegressor with 10,000 trees, 160 max\_bins and 30 min\_samples\_leaf
* HistGradientBoostingRegressor with 1000 trees, 160 max\_bins and 50 min\_samples\_leaf
* HistGradientBoostingRegressor with 500 trees and 40 min\_samples\_leaf (found via scikit-learn’s GridSearchCV)
* LGBMRegressor with 10,000 trees and default parameters

and obtained a val RMSE of $???.

# MODEL EVALUATION & INTERPRETATION

Since this is a regression task, we feel that a suitable metric will be the Root Mean Squared Error (RMSE).

We were inclined to use the native tree-based regressor's feature importance metrics to determine which features were more important and to selectively drop features that were not important. However, based on our research, we found out that there is a possibility that the feature importance algorithm can inflate the importance of features with high cardinality. Furthermore, impurity-based importance is computed on the train dataset and may not generate the same insights on the test set. As such, we decided to utilise sklearn's permutation importance as well to see if there it provides a different result.

While training the baseline model, we did some empirical analysis and noticed that both the tree-based regressor's feature importance and sklearn's permutation importance algorithms provide the same insights (top 5 most importance features are the same). Hence, for the sake of illustration, we will only provide the output of the native tree-based regressor's feature importance for the features used in our final predictions.

Diagram

Description automatically generated with low confidence

1. Permutation importance plots for RandomForestRegressor and GradientBoostingRegressor

Diagram

Description automatically generated with medium confidence

1. In-build feature importance plots for RandomForestRegressor and GradientBoostingRegressor

*Feature importance based on LGBMRegressor*

We use the LGBMRegressor model to evaluate feature importance (the HistGradientBoostingRegressor does not have such a feature). Here are the top 5 features:

1. Feature
2. Feature
3. Feature
4. Feature
5. Feature

# BREAKDOWN OF WORKLOAD

Raimi bin Karim

* Data preprocessing: experimenting different strategies of handling categorical features
* Feature engineering: dataset-based features, URA district, price index
* Data mining methods: Histogram-based Gradient Boost, LightGBM, ensembling, hyperparameter tuning
* Generate predictions for final model

Azmi bin Mohamed Ridwan

* XX

Wang Tian Ming Kenneth

* Created the pipeline to:
  + Run random search and grid search
  + Generate predictions
  + Generate feature importance and permutation importance results
* Generated predictions for baseline model

# APPENDIX

As a future work, we hope to come up with a Random Forest implementation that can handle splitting of nominal features.

##### Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.