50.039 Deep Learning Project

Deep Learning-Based Prediction of HDB Resale Flat Prices with LSTM-GRU Networks

Group 10
Lim Jie Han - 1006246
Ng Xue Min - 1006127
Daniel Yuen – 1006380

GitHub Link:
https://github.com/yjrd21/DL_HDB_ResaleFlat_Price_Forecast

Introduction

In Singapore, prospective buyers of resale HDB flats are not only concerned with affordability and location but are also increasingly interested in the potential return on investment when selling their flats after fulfilling the Minimum Occupation Period (MOP) of five years. Understanding the resale value trajectory of flats can help buyers make more informed decisions. Historical trends in HDB resale prices provide a valuable basis for forecasting future prices and estimating potential profits.

This leads us to our problem statement: *How might we develop a predictive model that forecasts the resale prices of HDB flats 5, 6, or 7 years after a buyer's intended year of purchase, in order to identify which street in a selected town is likely to yield the highest profit?* By leveraging historical resale data, this model aims to support prospective buyers in choosing resale flats with better investment potential.

Dataset

Raw Dataset

The dataset used in this project is sourced from the official HDB website [1] and spans from 2017 to 2025. It comprises approximately 200,600 rows, with each row representing a resale transaction. The features included are:

1. month: Month and year of resale
2. town: Name of the town (e.g. Bedok, Yishun)
3. flat_type: Type of flat (e.g. 3-Room, 5-Room)
4. block: Block number
5. street_name: Name of the street
6. storey_range: Range of the storey (e.g. 04 TO 06)
7. floor_area_sqm: Floor area in square metres
8. flat_model: Model of the flat
9. lease_commence_date: Year the lease commenced
10. remaining_lease: Remaining lease at the time of resale
11. resale_price: Price the flat was sold for

Data Pre-Processing

Identified key features

- Removed ‘block’ feature as we as it was considered redundant, given that the other features provided sufficient information to model the trends in resale prices.

Classified original dataset

To introduce more structured categories:

- *hdb_classification:* Added to differentiate between Mature and Non-Mature estates based on a predefined list.
- *storey_category:* Categorised as Low (1–15) or High (16 and above).
- *floor_area_category:* Grouped floor_area_sqm into buckets of 5 sqm intervals (e.g., 70–74 sqm, 75–79 sqm).

Aggregated classified dataset

To enhance the model's ability to learn general trends rather than overfitting to specific transactions, the dataset was aggregated using the combination of key features:

- month, town, flat_type, street_name, storey_category, and floor_area_category

This aggregation was necessary because performing analysis or forecasting directly on the raw transaction-level data—without grouping—would result in an impractically large and sparse dataset, potentially approaching an infinite number of combinations across all feature dimensions. By aggregating based on a defined combination of features (month, town, flat_type, street_name, storey_category, and floor_area_category), we significantly reduce the dataset’s dimensionality while retaining meaningful structure. This also makes it feasible

to identify and fill in missing values later using interpolation and categorical filling methods, which would have been far more complex and unreliable on ungrouped data.

Aggregation operations included:

- Median for numerical features (resale_price, remaining_lease, lease_commence_date)
- Mode for non-numerical/categorical features (flat_model, hdb_classification)

Filled up missing data in aggregated dataset

To ensure continuity in time series data:

- A full-time grid for each valid combination was created to fill in missing months where the combination exists.
- Numerical values were filled using interpolation within their respective groups (e.g., resale price trends for a specific street and flat type).
- Non-numerical values were filled using a combination of mode, forward fill, and backward fill.

A visual walkthrough of the data pre-processing up to this point can be found in the appendix.

Encoding non-numerical data

One-hot vector encoding

One-hot encoding was applied to:

1. town
2. flat_type
3. storey_category
4. flat_model
5. hdb_classification

This allows non-numerical/categorical features to be used in our deep learning model.

Street_name embedding – high dimensionality

The dataset contains 571 unique street names across the various towns in Singapore. Applying one-hot encoding to this feature is not suitable as it results in large sparse vectors which do not contain significant context about the street names.

To combat this issue, we have designed a pre-training procedure that would ideally generate learned embeddings of the street names. The pre-training task in hand would then be classifying the street names into the corresponding towns given the trainable embeddings.

The pre-training setup consists of 2 trainable models:

1. A GRU-based embedding layer that will generate the street name embeddings.
2. A fully connected linear classifier to predict the town given the street name embeddings.

The dataset used in this pre-training task is the set of unique street names along with their corresponding towns as labels. The models are trained using a supervised approach with 80:20 train-test split to help the embedding layer learn the rich semantics of the street names that relate to the corresponding towns. The inputs to this task are the street name mapped to indices, while the outputs are the predicted town indices.

The pre-training models achieved a maximum accuracy of around 65~75%. The trained embeddings are then added to the main dataset based on the corresponding street names.

Train-Validation-Test split

To evaluate the model's performance under realistic temporal conditions, the dataset was divided into three non-overlapping time-based splits:

- Train: 2017 to 2019 (pre-COVID period)
- Validation: 2021 to 2022 (post-restriction recovery phase)
- Test: 2023 to 2025 (post-lockdown normalization phase)

The year 2020 was deliberately excluded from all splits due to the onset of the COVID-19 pandemic, which introduced significant irregularities into Singapore's housing market. This period was marked by circuit breakers, construction delays, policy interventions, and general economic uncertainty—factors that caused atypical fluctuations in resale prices. Including this data could introduce noise and reduce the model's ability to generalize to stable market trends.

Model Architecture

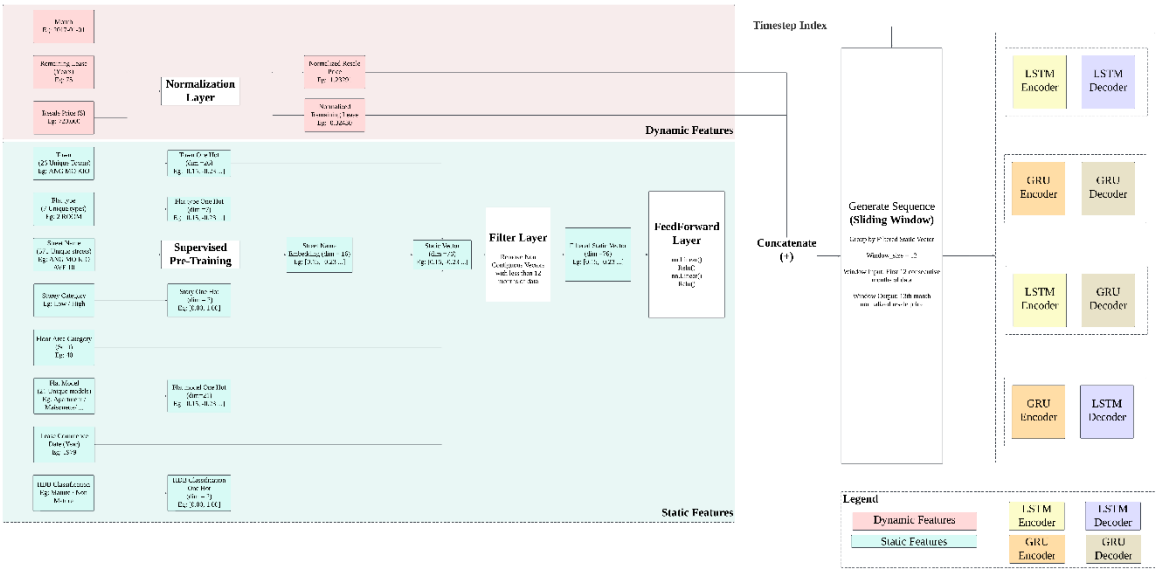


Fig 1: Model Architecture

The architecture diagram (Model_Architecture.pdf) has also been uploaded to the project Google Drive for reference (see Appendix for link). Please refer to it if the image displayed here appears unclear or low in resolution

All input features were grouped into two categories: static features and dynamic features.

Static features refer to attributes that do not change over time for a given flat profile. These include town, flat type, flat model, storey category, floor area category, lease commence date, HDB classification, and a learned embedding of the street name. These features were encoded using a combination of one-hot encoding and learned embeddings, then concatenated into a unified static vector.

Before using these vectors in downstream processing, a filtering step was applied to ensure sufficient temporal coverage. Specifically, only flat profiles with at least 12 consecutive months of transaction data were retained. This filtering step ensured that the model received temporally consistent input sequences and helped reduce noise from fragmented profiles. The resulting static vectors were then passed through a feedforward neural network to generate compressed static embeddings, which captured the long-term profile characteristics in a lower-dimensional space.

Dynamic features—namely normalized resale price and normalized remaining lease—were computed for each monthly transaction. These features vary across time and capture short-term market behaviour. To prepare the data for sequence modelling, a sliding window approach was used. For each flat profile, the first 12 months of dynamic data were extracted as input, and the 13th month's resale price served as the prediction target.

The static embedding generated earlier was repeated across all 12-time steps and concatenated with the dynamic features to form a composite input sequence. This allowed the model to jointly learn from both the temporal fluctuations of a flat's resale history and the invariant characteristics encoded in its static profile.

The combined sequences were passed into a family of sequence-to-sequence (seq2seq) forecasting models. Each model variant comprised an encoder and decoder, allowing the architecture to learn complex temporal dependencies. Four model configurations were evaluated:

- LSTM Encoder → LSTM Decoder
- GRU Encoder → GRU Decoder
- LSTM Encoder → GRU Decoder
- GRU Encoder → LSTM Decoder

These models were trained to predict future resale prices over configurable forecast horizons. The encoder processed the input sequence and produced hidden states, which were then used by the decoder to generate future predictions in an autoregressive manner.

Models

LSTM encoder + LSTM decoder

This architecture uses separate LSTM recurrent neural networks (RNNs) as encoder and decoder of the model.

The encoder processes the input sequences and encodes the context from these sequences into two hidden vectors, representing the final short-term and long-term memories generated from the input sequences.

The decoder receives the hidden vectors from the encoder and the last step of the encoder input as the initial inputs. Following that, the decoder generates the forecasted sequence autoregressively until the target output length is reached.

GRU encoder + GRU decoder

This architecture is adapted from the previous LSTM+LSTM model by swapping the LSTM RNNs with GRU models. GRU models only generate one hidden vector as opposed to two in LSTM.

The fundamental workings remain the same despite this difference.

LSTM encoder + GRU decoder

With the basic models constructed, we explored mixing and matching different RNNs. In the LSTM+GRU architecture, an LSTM model is used as an encoder, while a GRU model acts as a decoder.

The key difference of this architecture from the previous two architectures is the mismatch in the number of hidden vectors generated from the encoder and the number of hidden vectors expected by the decoder. In this LSTM+GRU approach, the encoder outputs two hidden vectors, while the GRU decoder only expects one hidden vector.

To connect the two models, the two memory vectors are concatenated into a single vector that is twice the dimension of the original hidden vectors, which should contain the intrinsic context from the encoding process. This means that the hidden vector input for the GRU decoder must be doubled to accommodate the additional information. The decoding process could then resume as usual.

GRU encoder + LSTM decoder

This model faces the opposite issue from the LSTM+GRU model, where the encoder outputs one hidden vector while the decoder expects two. This problem is easily solved by using the same hidden vector encoding for both hidden inputs of the decoder. As before, the decoding process can then proceed as usual.

Results

	Best model parameters	Test Results
LSTM encoder to LSTM decoder	Epoch 5 Train Loss: 0.0322 Val Loss: 0.0298 MAE: 20007.24 R^2: 0.97	MAE: 18547.87, R^2: 0.98, Loss: 0.0215
GRU encoder to GRU decoder	Epoch 4 Train Loss: 0.0313 Val Loss: 0.0286 MAE: 19976.26 R^2: 0.97	MAE: 19040.36, R^2: 0.98, Loss: 0.0213
LSTM encoder to GRU decoder	Epoch 4 Train Loss: 0.0296 Val Loss: 0.0252 MAE: 18775.43 R^2: 0.98	MAE: 17774.02, R^2: 0.98, Loss: 0.0190
GRU encoder to LSTM decoder	Epoch 3 Train Loss: 0.0295 Val Loss: 0.0261 MAE: 19055.35 R^2: 0.98	MAE: 17449.26, R^2: 0.98, Loss: 0.0191

Fig 2: Best model parameters and Test results

The results across all four encoder-decoder configurations—LSTM-to-LSTM, GRU-to-GRU, GRU-to-LSTM, and LSTM-to-GRU—demonstrate consistently strong predictive performance in forecasting HDB resale prices. Each model achieved a high R^2 score of 0.98 on the test set, indicating excellent goodness-of-fit. Among the models, GRU-to-LSTM yielded the lowest test MAE of $17,449.26, closely followed by LSTM-to-GRU at $17,774.02, suggesting that hybrid architectures may slightly outperform homogeneous pairings. All models converged rapidly, with major improvements observed within the first few epochs. Notably, even the highest test MAE observed—$19,040.36 in the GRU-to-GRU model—remained within a tight range, affirming the robustness of the training process and the effectiveness of combining static profile embeddings with temporal input sequences.

Conclusion

Through this project, we explored various time-series forecasting models that could perform forecasting of HDB resale flat prices reliably. Our work demonstrated the possibility of using deep learning models that are trained on extensive data to predict future trends of resale flat prices with high accuracy.

We have also discovered the potential of hybrid architectures, which performed better than single-architecture sequence-to-sequence models in our experiments. Nonetheless, all of the models that we have looked at demonstrated strong predictive performance with minimal errors. They showed rapid convergence and minimal overfitting, and the consistently high R^2 scores across test sets reinforce the reliability of the learned temporal patterns and static profile embeddings in modelling resale value trajectories.

Overall, our work affirms the feasibility of using sequence-to-sequence models for fine-grained, flat-level price forecasting in the Singapore HDB market.

User Instructions

Pre-Requisites

Before running any code in this repository, ensure you have the following installed:

- Python 3.8 or higher
- Jupyter Notebook or JupyterLab or VS Code with Jupyter extension
- Required Python libraries (see below)
    - Pandas
    - NumPy
    - PyTorch
    - Matplotlib
    - Scikit-learn

Installation

You may find the link to our GitHub repository in Page 2 above.

To clone the repository, you may open a terminal window and navigate to your desired directory, then run the following command:

```
git clone https://github.com/yjrd21/DL_HDB_ResaleFlat_Price_Forecast.git
cd DL_HDB_ResaleFlat_Price_Forecast
```

In the same terminal, install the required dependencies using the following command:

```
pip install -r requirements.txt
```

Dataset

Data source: Resale flat prices based on registration date from Jan-2017 onwards, HDB (https://data.gov.sg/datasets/d_8b84c4ee58e3cfc0ece0d773c8ca6abc/view)

The dataset is exported from the source and saved in 'resale_flat_prices_original.xlsx'.

During data preparation, the following files are generated as intermediary files:

- 'resale_flat_prices_classified.xlsx'
- 'resale_flat_prices_aggregated.xlsx'
- 'resale_flat_prices_final.xlsx'

The final processed and cleaned data with the trained embeddings is stored in 'resale_flat_prices_one_hot_vector_and_embed.xlsx' and will be used to train the forecasting models.

All dataset files can be found at https://drive.google.com/drive/folders/16yzyB7VBLktP9UFbtng2bhYS_2SJL5_g. It is advised to only download 'resale_flat_prices_one_hot_vector_and_embed.xlsx', but feel free to explore the other data files.

Running the Notebooks

There are 2 notebooks to be run:

1. 'data_preparation.ipynb': contains data preparation and cleaning steps

2. 'main.ipynb': contains forecasting models and training process

How to Run the Notebooks:

1. Open this repository in Jupyter Notebook or JupyterLab or VS Code with Jupyter extension.
2. In the preferred interface, open the 'data_preparation.ipynb' file.
3. Follow the instructions in the notebook to execute each cell. Ensure that you run the cells sequentially for proper execution.
4. Repeat Steps 2-3 with the 'main.ipynb' file.

Note: if you encounter any issues, verify that all dependencies above are installed and compatible with your Python version.

Team Members Contribution

Member	Contribution
Ng Xue Min	Data pre-processing, report
Daniel Yuen	Train-Val-Test split, Sliding window, initial Seq2Seq Model, report
Lim Jie Han	Street_name encoding, Seq2Seq Models, report

Appendix

References

[1]: https://data.gov.sg/datasets/d_8b84c4ee58e3cfc0ece0d773c8ca6abc/view

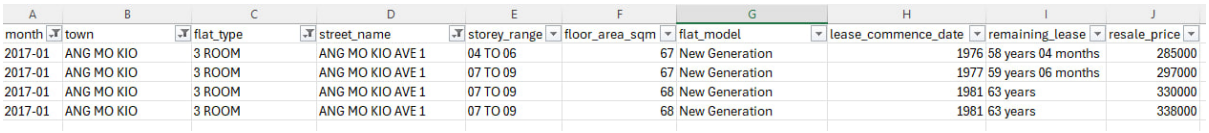
Data Pre-Processing

Google Drive link

- The excel files for each step of the data pre-processing can be found here: DL_Project_Group10 - Google Drive

Original dataset

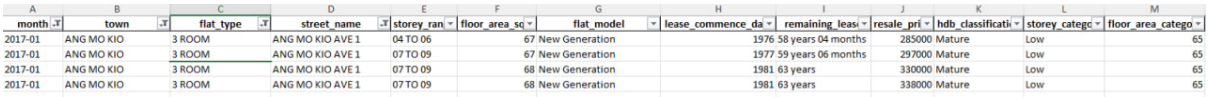
- This screenshot is filtered based on a given combination: month (2017-01), town (ANG MO KIO), flat_type (3 ROOM), street_name (ANG MO KIO AVE 1) storey_range (<= 15) and floor_area_sqm (65-69).



month	town	flat_type	street_name	storey_range	floor_area_sqm	flat_model	lease_commence_date	remaining_lease	resale_price
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	04 TO 06	67	New Generation	1976	58 years 04 months	285000
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	07 TO 09	67	New Generation	1977	59 years 06 months	297000
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	07 TO 09	68	New Generation	1981	63 years	330000
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	07 TO 09	68	New Generation	1981	63 years	338000

Classified dataset

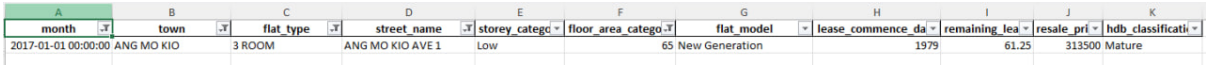
- This screenshot shows the original dataset with added features (hdb_classification, storey_category and floor_area_category).



month	town	flat_type	street_name	storey_ran	floor_area_sq	flat_model	lease_commence_da	remaining_leas	resale_pri	hdb_classificati	storey_catego	floor_area_catego
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	04 TO 06	67	New Generation	1976	58 years 04 months	285000	Mature	Low	65
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	07 TO 09	67	New Generation	1977	59 years 06 months	297000	Mature	Low	65
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	07 TO 09	68	New Generation	1981	63 years	330000	Mature	Low	65
2017-01	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	07 TO 09	68	New Generation	1981	63 years	338000	Mature	Low	65

Aggregated dataset (given month)

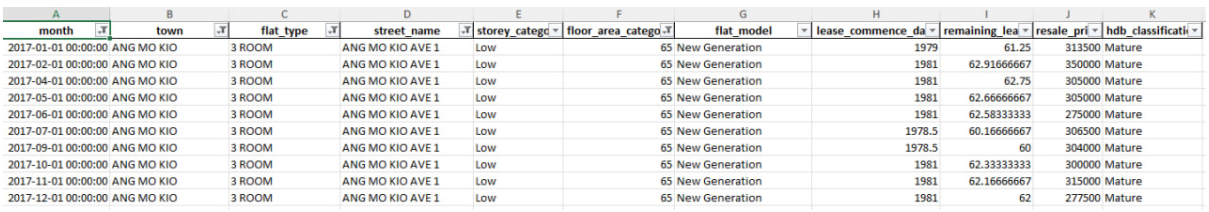
- This screenshot shows the aggregated data for the given combination.



month	town	flat_type	street_name	storey_catego	floor_area_catego	flat_model	lease_commence_da	remaining_lea	resale_pri	hdb_classificati
2017-01-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1979	61.25	313500	Mature

Aggregated dataset (given year)

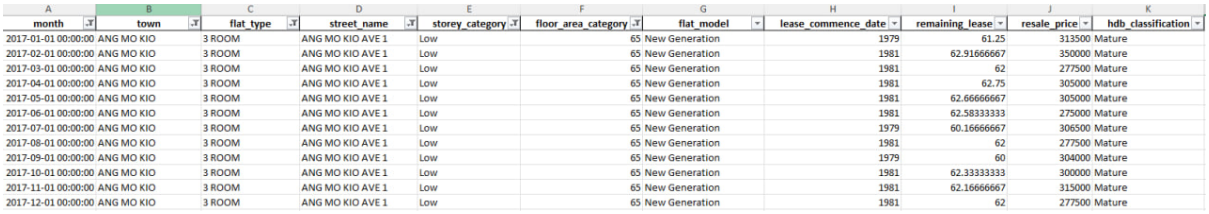
- This screenshot shows the aggregated data for the given combination but for all months in the year. This data is not contiguous. There are missing data points for the months 2017-03-01 and 2017-08-01.



month	town	flat_type	street_name	storey_catego	floor_area_catego	flat_model	lease_commence_da	remaining_lea	resale_pri	hdb_classificati
2017-01-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1979	61.25	313500	Mature
2017-02-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.91666667	350000	Mature
2017-04-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.75	305000	Mature
2017-05-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.66666667	305000	Mature
2017-06-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.58333333	275000	Mature
2017-07-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1978.5	60.16666667	306500	Mature
2017-09-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1978.5	60	304000	Mature
2017-10-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.33333333	300000	Mature
2017-11-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.16666667	315000	Mature
2017-12-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62	277500	Mature

Final dataset with data in missing months filled up

- This screenshot shows the final dataset with data present in all months by filling up the missing data using interpolation (numerical data) and forward-fill/backward-fill/mode (non-numerical/categorical data).



A	B	C	D	E	F	G	H	I	J	K
month	town	flat_type	street_name	storey_category	floor_area_category	flat_model	lease_commence_date	remaining_lease	resale_price	hdb_classification
2017-01-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1979	61.25	313500	Mature
2017-02-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.91666667	350000	Mature
2017-03-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62	277500	Mature
2017-04-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.75	305000	Mature
2017-05-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.66666667	305000	Mature
2017-06-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.58333333	275000	Mature
2017-07-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1979	60.16666667	306500	Mature
2017-08-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62	277500	Mature
2017-09-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1979	60	304000	Mature
2017-10-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.33333333	300000	Mature
2017-11-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62.16666667	315000	Mature
2017-12-01 00:00:00	ANG MO KIO	3 ROOM	ANG MO KIO AVE 1	Low	65	New Generation	1981	62	277500	Mature