Predicting HDB Rental Price in Singapore

Lim Ji Chen E0792464
Le Tan Dang Khoa E1124464
Goh Siow Chuen E0146431
Yap Kim Thow E0257961

***Abstract*—In tandem with a general rise in inflation, we have seen an increase in property prices and rental fees in Singapore in recent years. In this paper, we explore the use of several datasets including monthly rentals from year 2021 to 2023, along with information about HDB like its size, location, year of lease commencement, existing MRTs stations and their locations, planned MRTs stations, shopping malls, and primary schools. With these datasets, we fitted three models to predict the rental price of HDB apartments. We first explored a basic linear regression model, Ridge Regression, followed by a Random Forest Regressor, and finally with CatBoost, a variant of gradient-boosting method. All three models are then evaluated with root mean squared error (RMSE) by conducting 10-fold cross-validation. The models' performance improves with the increase in complexity from Ridge Regression, Random Forest and CatBoost. We also identified rental approval date, the region of which the apartment is in, distance to the nearest MRT, distance to the nearest mall, and size or flat type to be the main factors affecting the rental price. The code is available at: https://github.com/cs5228-group-1/cs5228-final-project.**

***Index Terms*—rental price, linear regression, tree-based methods, gradient-boosting**

I. Introduction

A. Background

Renters in Singapore currently face heavy financial strains in the rental market in light of recent price increases. For this reason, prospective renters would want to make well-informed decisions based on their financial situations, and this knowledge could then help them avoid potential rip-offs and/or spot potential bargains in the current rental market. On the other hand, landlords and real estate agents would want to maximise their rental profits.

This project makes use of historical HDB rental prices to make rental price predictions. While the core dataset offers a substantial array of relevant attributes that can contribute to accurate forecasts, the challenge lies in integrating the auxiliary datasets, which may not have causal relations with the core data. Our main focus is on deriving valuable insights from the core dataset while harmoniously amalgamating relevant data sources.

This project has the potential to contribute to a deeper understanding of rental trends, market dynamics, and the influence of various factors on rental rates. By doing this, this initiative will serve to benefit the stakeholders such as prospective renters and landlords by offering them a deeper understanding of the Singapore rental market.

As this project is part of Kaggle competition for National University of Singapore School of Computing CS5228 course, we also compared our experiments against other groups using same datasets given.

B. Goal

The goal of this project is to create a prediction model for Singapore's HDB rental prices. With this, we aim to show how data mining can be performed in a practical setting. Given the core dataset and supplementary datasets, we perform a series of data mining techniques, justify design and implementation issues, before interpreting the results and assessing limitations of our approach.

C. Datasets

The core dataset of rental rates for HDB flats was collected from data.gov.sg by the teaching staff of CS5228 (extended with additional data sources such as the flat type and size, the lease commence date, etc.), containing approved applications by HDB owners to rent out their flat from 2021 to 2023. The core dataset contains essential attributes, including rent approval date, town, block, street name, flat type, flat model, floor area sqm, furnished, lease commence date, latitude, longitude, elevation, subzone, planning area, region, and monthly rent. These attributes offer a comprehensive view of the HDB rental market, encompassing key information about the location, physical attributes of flats, their rental history, and monthly rental rates. The 'latitude' and 'longitude' attributes, in particular, enable geographical mapping, while 'monthly-rent' serves as the target variable for predictive modeling.

Additionally, auxialiary datasets such as locations of existing and planned MRT stations, shopping malls, and primary schools, as well as information related to economic factors, namely Singapore stock prices and Certificate of Entitlement (COE) prices, are given to enrich the core dataset for this project.

These datasets capture potentially important factors determining rental prices like convenience (proximities to MRT stations, shopping malls and primary schools), and the economy on rental rates. By merging these auxiliary datasets with the core data, this project seeks to establish their correlations with the rental prices and gain a comprehensive understanding of the Singapore rental market, ultimately enabling more accurate rental rate predictions and informed decision-making for renters and landlords.

II. Data Preparation

A. Exploratory data analysis

1) Main Dataset: The main dataset given consists of 16 attributes and 60,000 samples. The attributes include

rent_approval_date, town, block, street_name, flat_type, flat_model, floor_area_sqm, furnished, lease_commence_date, latitude, longitude, elevation, subzone, planning_area, region, and monthly_rent. As we are concerned with monthly_rent, we explored various relationships between the attributes and monthly_rent.

From the 16 attributes, rent_approval_date and lease_commence_date are dates represented in string format, while floor_area_sqm, latitude, longitude, elevation and monthly_rent are numerical values, the rest of the attributes are all nominal data. All the values in all samples are available therefore there is no missing value.

The dataset consists of rental approval records from January 2021 to July 2023. We also see that floor area size varies from the smallest at 34 sqm to the largest at 215 sqm. In addition, the monthly rental goes as low as SGD 300 to SGD 6,950. One of the oldest HDB in the dataset had its lease commence date of 1966.

From Figure 2a, we see that there are five flat types, with 4-room type being the flat type with the highest share (36.5%). From Figure 2b, we can also see that there is a correlation between flat type and rental price, with the lowest rental prices associated with the smaller flat type (i.e. 2-room with the lowest rental prices) and increases with larger flat types, since we typically pay more for larger floor areas. The executive flat type typically fetched slightly higher rental prices due to it being in a more 'premium' category. There are significant outliers in all flat types at the higher range of rental prices, with the exception of 2-room flat type. With this observation, we can view the 2-room flat type as the 'exclusive budget' category, with no monthly rental price greater than SGD 4000 – and there are other factors in play which helped to drive up **some** units in the other flat types.

From Figure 1, we can also see that the rental prices for all flat types has been steadily increasing over the 3 years period.

2) Location Attributes: As can be shown in Figure 3, most of the higher-priced rental units are located in the central, south-eastern area. The general observation shows that location in general plays a role in rental prices, but we would need to supplement this with other factors (since relatively pricier units and relatively cheaper units can cluster closely as seen in the figure).

3) Auxiliary Data - Existing Train Stations: In the existing train stations dataset, there are five attributes, namely code, name, opening year, latitude and longitude. There are no duplicates in this data and hence there are 162 stations in Singapore. For each row of the main dataset, we calculate the distance from each apartment rented out to the nearest MRT station with latitude and longitude. From the result, we can see that flats that are nearer to MRT stations typically fetched higher rental prices. From Table I, we can see that the HDB flats with the highest rental price in each flat type are all located within several hundred meters from an existing MRT station. We also plotted rent_per_sqm against nearest_mrt_dist in Figure 4 subplot 3, the

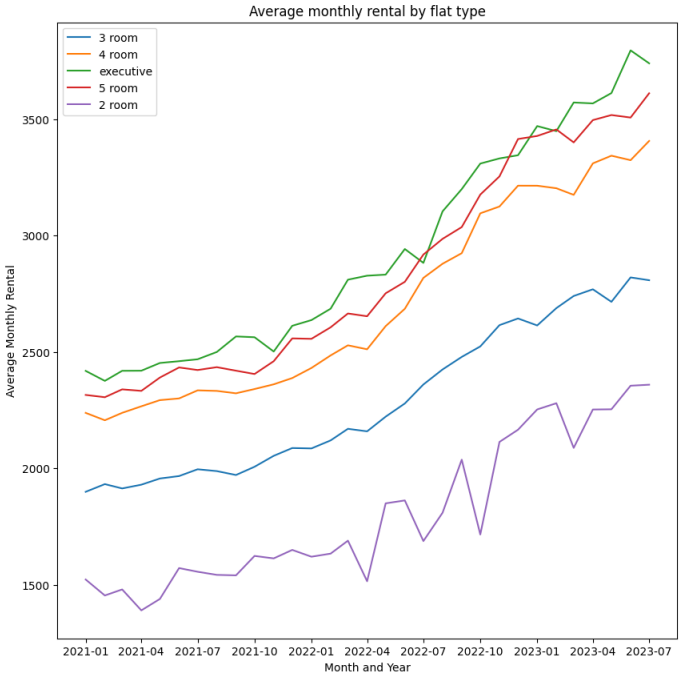
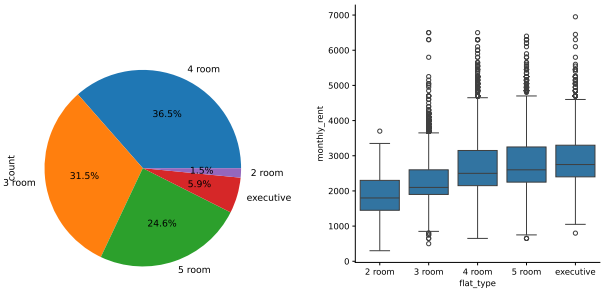


Fig. 1: Average rental by flat type.



(a) Flat type distribution. (b) Flat type vs Monthly Rental.

Fig. 2

mean rental price seems to follow a decreasing trend when the distance to the nearest MRT increases. We make use of this data for a discussion in Section II-A5.

4) Auxiliary Data - Planned Train Stations: In the planned train stations dataset, there are also five attributes as in existing train stations dataset. There are 74 planned stations with opening year as near as year 2024 and as far as 2040. There

TABLE I: Highest Monthly Rental by Flat Type and Its corresponding distance to nearest MRT

Flat Type	Highest Monthly Rental	Distance to Nearest MRT
2 room	SGD 3,700	584.24m
3 room	SGD 6,500	376.01m
4 room	SGD 6,500	358.87m
5 room	SGD 6,400	341.73m
executive	SGD 6,950	282.20m

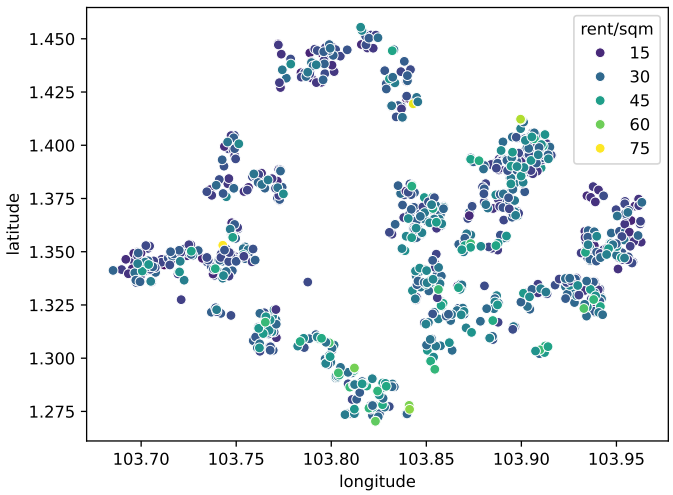


Fig. 3: Relationship between apartment location and monthly rent per squared meters.

are five train stations with no opening year. Similar to existing train stations, we calculated the distance from each flat to the nearest planned station. We were not able to uncover any potential correlation with rental prices from our cursory exploration.

5) Auxiliary Data - Nearest Mall, Schools, and MRTs: As depicted in the Figure 4, the rental price trends for the years 2021 to 2023 exhibit a consistent upward trajectory year-on-year.

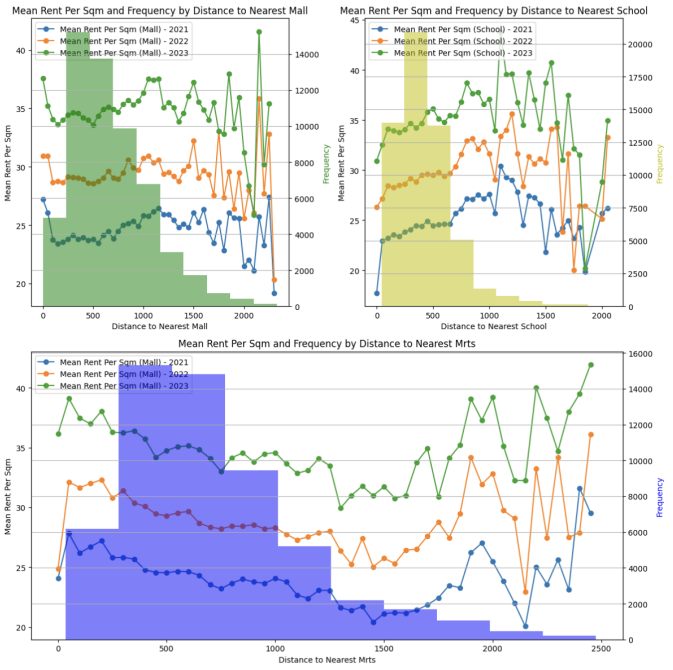


Fig. 4: Mean Rent per Sqm vs Nearest Schools, Malls and MRTs Distance by Year

From Figure 4, we note that the plots of mean rental price per square meter show high variability at both ends of the 'distance' axis (especially at the farthest end) due to the fewer data points – this can be seen from the histograms (right skewed) which show higher number of units near a mall, primary school or MRT station. Disregarding points lying at both ends, we can observe a downward trend in rental prices as units get farther away from an MRT station. The plot also shows that the mean `rent_per_sqm` trends higher when the distance to MRT is more than 2km (even though with few data points). With further exploration, we found out that this is due to the rental price for apartments more than 2km away from MRT are contributed by mainly one block of HDB and that particular block has fetched quite high rental therefore skewing the trend.

As for distance to nearest mall and distance to nearest primary school, there appears to be a weak upward trend in rental prices as units get farther away from a mall or primary school – this is a rather surprising observation. Several factors could account for this discrepancy, including the possibility that the malls or schools in the vicinity are not particularly popular or influential in influencing rental prices. Residents may instead be willing to travel a bit farther to access better shopping centers or educational institutions. While we attempt to draw some clues from Figure 4, it's important to note that this visualization does not account for different types of flats or other factors.

B. Data Preprocessing

We observed that `flat_type` attributes are not consistently formatted. For example, there are `3-room` and `3 room` co-existing in the data. It can easily be handled by replacing – by the white-space character. From Figure 2a, the most common rental apartments are 3-room, 4-room, and 5-room apartments. As mentioned in the previous subsection, the distribution and the average prices of each flat type exhibit a clear trend, indicating this attribute is important.

There are several attributes we opted to remove from the data. Firstly, **furnished** and **elevation** have identical values across all records, namely **yes** and **0.0**, respectively. Since they do not present any variation, they do not offer useful information, hence we omitted these 2 attributes from the data.

1) Null Values and Duplicates: Upon inspection, we did not find any null values for all the attributes in the main dataset as well as auxiliary datasets. There are however 273 duplicates identified in the main training dataset. We opted to remove the duplicates.

2) Date Attributes: To handle the date attribute, namely **rent approval date**, we use one of the following two approaches while experimenting with different models:

- Replacing the date with two numerical attributes representing year and month, e.g. 2021 and 12, respectively.
- Replacing the date with a single numerical attribute representing the date of rent approval with the earliest date of Jan 2021 being 0, and an increment of 1 with each subsequent month, e.g. Feb 2022 would be 13, and Jan 2023 would be 24.

The first approach allows the rent approval date to be captured by year, while allowing the `month` attribute to capture any potential seasonal effects in a year. The second approach allows the rent approval date to be captured in a more natural and more granular time series manner.

The other date attribute which is the **lease commence date** consists of only the year, so this allows us to use this attribute as is when experimenting with tree-based models.

C. Distance Attributes

We want to know whether distance to the nearest amenity such as MRT, shopping malls or primary schools affecting the monthly rental price. To calculate the distance from the provided longitude and latitude of the apartment to other locations, we use the great circle distance:

$$D = R \arccos\left(sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_2 \cos(\Delta\lambda)\right) \quad (1)$$

where $R = 6371000$ meters is the radius of the Earth, $\Delta\lambda = \frac{\lambda_1}{\lambda_2}$. λ_1, θ_1 and λ_2, θ_2 are the longitude and latitude of two points.

D. CatBoost Encoding for Categorical Attributes

In order to utilize all relevant information about apartment types such as flat type, flat model as well as location data, namely subzone name, MRT code, street name, etc, we need an efficient representation of categorical data. During the initial development of the project, we considered one hot encoding in which it creates a sparse vector of size K where K is the number of unique values. However, this approach is not ideal for modeling because of the increase in dimensions, e.g. one-hot encoding street name results in 1083 dimensions, and one-hot encoding subzone results in 152 dimensions. As a consequence, it may introduce 'the curse of dimensionality'. It also hugely increases training times for any linear (e.g. Linear Regression) and non-linear (e.g. tree-based) models.

We then explored other categorical encoding methods and finally selected CatBoost Encoder [1] as the method for transforming categorical data. CatBoost Encoder is derived from target statistic encoding (TS) [2]. TS replaces the category value with the average (or other statistics) of the target attribute belonging in the same category. Unlike one-hot encoding, it does not create additional dimensions to represent each categorical feature. However, it may introduce data leakage (i.e. target leakage) and it may cause overfitting to the training data. To tackle the issue, CatBoost Encoder introduces ordered target statistic encoding which first permutates the order of the input data and consider the new order as "time series". When calculating the statistic, it only use data preceding the current one, similar to calculate the statistic of "history" data points.

If the model encounters unseen values from the test set, it simply replaces it with the mean of the target in the entire training set.

E. Numerical Attributes Scaling

For Ridge Regression, we normalize the data with Min-Max Scaler. As we opted to encode categorical attributes with CatBoost Encoding as mentioned in Secion II-D which uses target encoding, we normalize the numerical attributes with the range of the target, which in our case, the monthly rental, to preserve data range across the training data attributes.

F. Preprocessing

In order to explore the effects of additional data attributes to our models, we have set five different combinations of feature (i.e. feature sets) namely:

- Feature set 1 (V1): Only using existing attributes from provided training data with all preprocessing described in Section II-B. The date attribute of `rent_approval_date` is represented using the first approach as explained in the subsection.
- Feature set 2 (V2): Feature set 1 with existing MRT location: distance to the nearest MRT and its corresponding MRT code.
- Feature set 3 (V3): Feature set 2 with distance to the nearest shopping mall and its name.
- Feature set 4 (V4): Feature set 3 with distance to the nearest primary school and its name.
- Feature set 5 (V5): Only made use of `flat_type`, `floor_area_sqm`, `planning_area`, `nearest_mrt_dist`, `nearest_mall_dist` and `nearest_school_dist`. The feature `age` of the flat is represented by calculating the difference between `rent_approval_date` and `lease_commence_date`. The date attribute `rent_approval_date` is represented using the first approach as explained in the subsection. In V5, most of the categorical data attributes are dropped except for `planning_area`. The features `latitude` and `longitude` are also dropped, with the reasoning that `planning_area` is already encompassing the information provided by other data attributes including `town`, `subzone` , `region`, etc. Thereby keeping information of location represented by a single feature to allow for more straightforward analysis.

III. Experiments

A. Models

We made two baseline models in the form of linear regression and random forest in order to establish some good baselines before adopting a more advanced model in the form of CatBoost to improve the predictive performance.

1) Linear Regression: We start with basic linear regression model with 5 settings. As ridge regression in scikit-learn can only take in numerical data, additional data processing steps have to be taken here to encode categorical data. We made use of CatBoostEncoder as mentioned in Section II-D to encode all categorical data as one-hot encoding is adding too many features to the training data. Numerical data attributes are also normalised as mentioned in Section II-E.

2) Random forest regressor: Random forest regressor is an ensemble algorithm that operates by constructing multiple decision trees during training and then combining their predictions to make more accurate and robust final prediction. Each decision tree in the random forest is trained using a random subset of the training data and a random subset of the features. So, this helps to reduce overfitting and improve generalization of the model. It can generally offer good accuracy because of the ensemble of multiple decision trees (each of them differs from one another due to the random subset of training samples and features).

3) CatBoost: For the project, we decided to experiment with CatBoost [1]. CatBoost's motivation is to deal with target leakage issue by proposing a modification of gradient boosting named ordered boosting. To improve the performance further, it also deploys several techniques such as symmetric tree construction, outliers handling. In addition, it also proposes a new target-based (supervised) [2] encoding method for categorical data as mentioned in Section II-D.

B. Experimental Settings

K-fold cross-validation with 10 folds are used across all settings. We report the mean root mean square error (RMSE) of each run with its standard deviation to observe how each configuration performs. We have selected this as our primary evaluation metric as it provides a good measure of predictive accuracy for our regression models. The evaluation begins in computing the average magnitude of errors from the predict values in reference to the actual values which is the labels of the test dataset. In addition to that, regression models are designed to predict and estimate a continuous numerical values which makes RMSE the suitable metric. On the other hand, RMSE may not be suitable for other approaches like classification models where metrics like accuracy or F1-score are frequently used. By utilising RMSE, we can make sure that our assessment is in line with the particular objectives and characteristics of our regression problem, allowing us to assess how well the model reduces prediction errors. Furthermore, we opted for this measurement metric to align with the benchmark metric employed in the Kaggle challenge, ensuring a more harmonized and consistent evaluation process.

For **Linear Regression**, we chose to use Ridge Regression with several parameters combinations from alphas of 0.01, 0.1, 1, 10, and 100 and solvers of `svd`, `cholesky`, `sparse_cg`, and `sag`. This helps us identify the level of regularization and the efficient solver for the linear model. By using GridSearch and 10-fold cross validation, we identify the best parameters combination for each settings.

For **Random Forest**, we explored some combinations of hyperparameters before performing a grid search with `n_estimators=300`, `max_depth={25, 30}`, and `min_samples_leaf={5, 7, 9}`. The hyperparameter `max_depth` is the minimum number of samples required to split an internal node, while `min_samples_leaf` (default value of 1) is the minimum number of samples required to be at a leaf node. As noted in the Scikit Learn documentation, the `min_samples_leaf` has the effect of smoothing the random forest model, especially in regression – therefore it is important in this project for us to allow a value greater than 1 to improve its generalization.

TABLE II: Cross-validated RMSE means and standard deviations of training set (10-fold).

	Ridge Regression Feature set combination				
	V1	V2	V3	V4	V5
Mean	510.25	508.49	508.37	508.32	522.80
Standard Deviation	7.88	7.27	7.25	7.25	6.89

Categorical (nominal) features are one-hot encoded when training and making predictions. We opted to use a relatively simpler set of features for better clarity in explainability: Numerical features of `date` (the second approach for encoding `rent_approval_date`; integer), `floor_area_sqm` (float), `lease_commence_date` (integer), `nearest_mrt_dist` (float), `nearest_mall_dist` (float) and `nearest_school_dist` (float), and ordinal features of `planning_area`, `flat_type`, `flat_model`, `nearest_mrt_code`, `nearest_mall_name` and `nearest_school_name`.

For **CatBoost**, we used learning rate of 0.05 on 5000 iterations with early stopping. L_2 regularizer is set at 50.0 to alleviate the effect of overfitting while the maximum depth of each stump is limited to 5. For categorical encoding, we use the default method from the package, which is the ordering statistic target encoding.

C. Results

We start with two baseline methods which serve as good baselines for the performance by utilizing simpler, less advanced models. They serve to help us understand the effects of various features before attempting a more advanced method in the form of CatBoost, a variant of gradient boosting method.

1) Ridge Regression: The training is done on the feature set combinations as mentioned in II-F. Additional categorical data encoding is performed for each settings before training with CatBoost encoding. Table II and Figure 5 show the mean and standard deviation RMSE across 5 different settings used. We can see that the result improve with addition of new attributes to the training data. In setting 5, we have removed quite a few nominal data attributes and kept only one of them, the removal of the data attributes, `latitude` and `longitude` has negatively impacted the performance of the model.

2) Random forest regressor: We trained on the same feature set combinations as mentioned in Section II-F. Table III shows the mean and standard deviation RMSE across all feature sets with a 10-fold cross-validation. Figure 6 illustrates the RMSE values of all folds in each setting.

3) CatBoost: We trained on the same feature set combinations as mentioned in Section II-F. Table IV shows the mean and standard deviation RMSE across all feature sets while Figure 7 illustrates the RMSE values of all folds in each setting. We can see that the result improves with addition

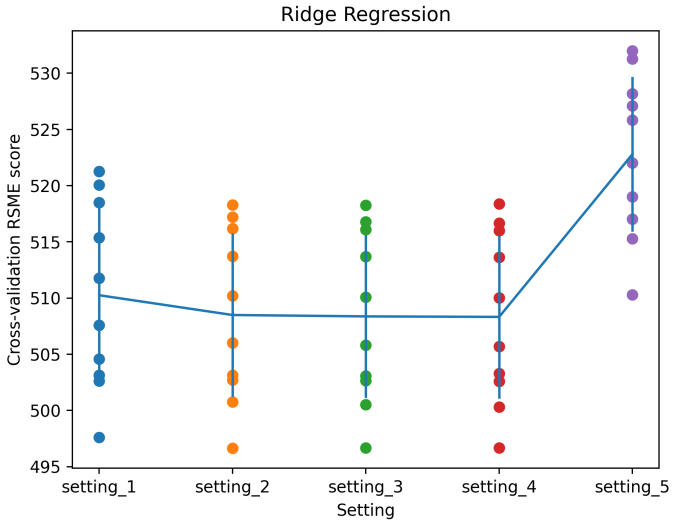


Fig. 5: Cross-validation RMSE score of Ridge Regression on all settings.

TABLE III: Cross-validated RMSE means and standard deviations of training set (10-fold).

	Random Forest Feature set combination				
	V1	V2	V3	V4	V5
Mean	492.64	491.58	491.84	492.08	496.13
Standard Deviation	8.01	8.05	8.00	7.97	9.50

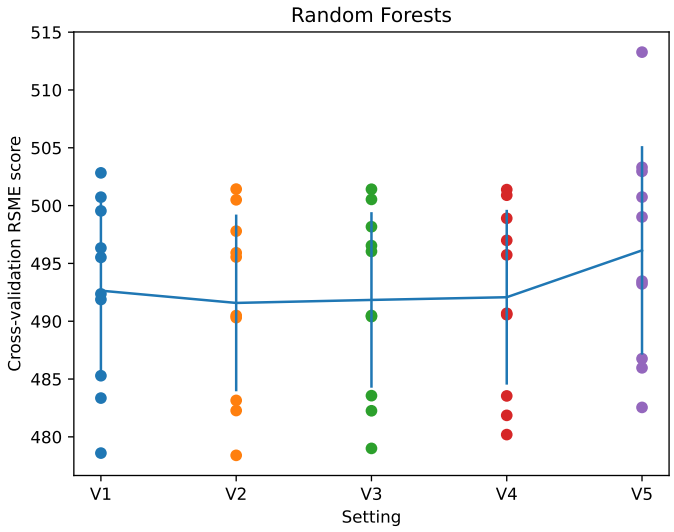


Fig. 6: Cross-validation RMSE score of Random Forest on all settings.

TABLE IV: Cross-validated RMSE means and standard deviations of training set (10-fold).

	CatBoost Feature set combination				
	V1	V2	V3	V4	V5
Mean	481.12	480.50	480.39	480.49	482.60
Standard Deviation	5.61	5.74	5.76	5.78	5.97

of new attributes, namely distances to amenities and facility name, to the training data. However, the addition of primary school attributes reduces the performance slightly, from 480.39 to 480.49. Catboost is showing the best result compared to Ridge Regression and Random Forest .

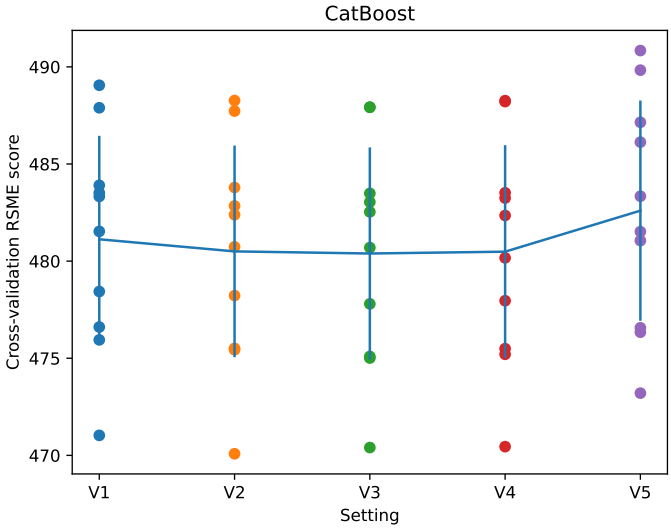


Fig. 7: Cross-validation RMSE score of CatBoost on all settings.

4) Result summary: Table V is a summary of the best mean RMSE achieved with the three models on a 10-fold cross-validation. We can see that the result improves with increasing complexity of the models.

TABLE V: Best RMSE mean and standard deviations for all models (10-fold cross-validation).

	RMSE	
	Mean	Standard Deviation
Ridge Regression	508.32	7.25
Random Forest	491.58	8.05
CatBoost	480.39	5.76

D. Discussions

1) Ridge regression - coefficients: We compared the coefficients for each of the attributes used in the 4 settings. We had expected that `floor_area_sqm` and `year` to be positive as in EDA we saw that the rental increases with increase in value in both the size of the apartment and increase in rental in recent year. We also see that `nearest_mrt_dist`

TABLE VI: Coefficients from Ridge Regression model for attributes in different settings (rounded up to 2 decimals).

	Ridge Regression Coefficients			
	V1	V2	V3	V4
block	0.21	0.20	0.19	0.19
street_name	0.20	0.17	0.17	0.16
flat_type	0.71	0.71	0.71	0.71
flat_model	0.05	0.07	0.06	0.06
subzone	0.16	0.10	0.10	0.11
floor_area_sqm	0.04	0.05	0.05	0.05
lease_commence_dt	0.03	0.05	0.04	0.04
latitude	-0.08	-0.08	-0.08	-0.08
longitude	0.004	0.01	0.01	0.01
year	0.15	0.15	0.15	0.15
month	0.06	0.06	0.06	0.06
nearest_mrt_dist	-	-0.04	-0.04	-0.04
nearest_mrt_code	-	0.06	0.04	0.04
nearest_mall_dist	-	-	-0.01	-0.01
nearest_mall_name	-	-	0.00	0.02
nearest_sch_dist	-	-	-	0.01
nearest_sch_name	-	-	-	-0.03

TABLE VII: Coefficients from Ridge Regression model for attributes in setting 5 (rounded up to 2 decimals).

	Ridge Regression Coefficients
	V5
flat_type	0.80
flat_model	0.11
planning_are	0.26
floor_area_sqm	0.03
date	0.19
age	-0.02
nearest_mrt_dist	-0.05
nearest_mrt_code	0.24
nearest_mall_dist	0.01
nearest_mall_name	0.20
nearest_sch_dist	0.03
nearest_sch_name	0.17

and nearest_mall_dist have negative coefficients as expected, which corroborate our hypothesis that flats with shorter distance to both MRTs or malls will fetch higher rental. Although some of the coefficients captured our suspicions about the data, we do not have full explainability insights for all of the attributes. The linear relationship between the attributes and rental price that we had expected is not confirmed with the observations of coefficients. Further experiments with other methods could help investigate more relationship between the attributes and the rental price.

2) Random forest regressor - feature importance: For random forest regressor, feature importance are provided by the fitted attribute feature_importances_ and they are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree. We plot the simpler feature set V5 with the fewer number of features (excluding features like block, street_name, latitude and longitude) in Figure 8 in order to understand the importance of the various features. From Figure 8, we see that date (i.e. rent_approval_date) and flat_type are by far the most important compared to the other features. The most important feature date captures an important increasing trend as seen earlier in 1, whereas the feature rent_type encodes information of unit size and and if the unit is 'premium' (i.e. executive) as seen earlier in 2b.

The next important feature in V5 is nearest_mrt_code which is somewhat surprising, since it suggests that it matters not (as much) whether a unit is near a MRT station, but to which MRT station it is nearest. This implies that units nearer to certain MRT stations, say, Queenstown, allows the model to capture information with regard to the locations that could command higher prices – so this feature nearest_mrt_code is essentially a proxy for location in this model. The feature planning_area is the next important feature, and it also captures location information but at a higher level (29 planning areas) than nearest_mrt_code which does it at a more fine-grained level (162 existing MRT stations).

We also note that random forest models can suffer from high cardinality bias [3], that the model would tend to overestimate the importance of features with a high number of unique values – as we would face if we use one-hot encoding for the categorical features. Therefore we have chosen to use CatBoost Encoder as explained earlier.

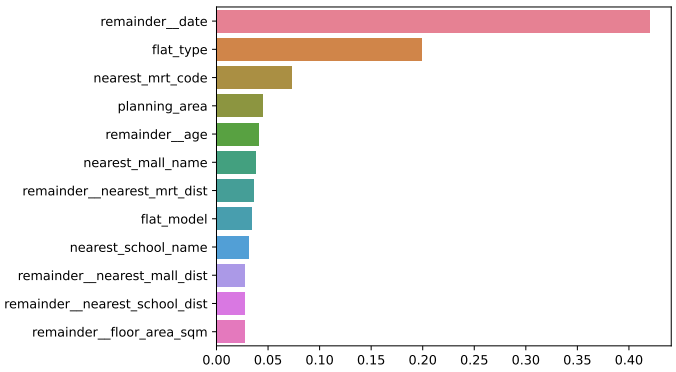


Fig. 8: Random forest regressor feature importance.

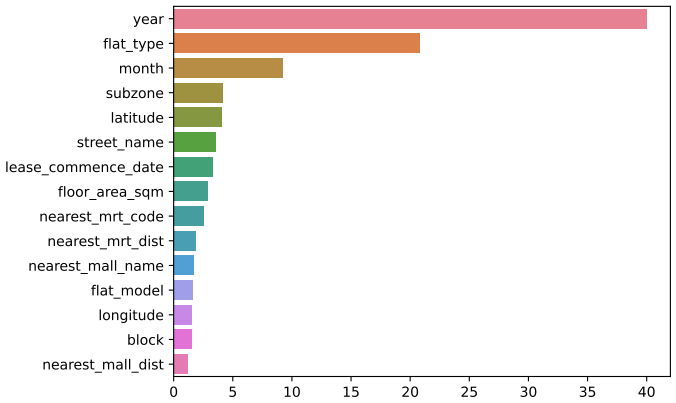


Fig. 9: CatBoost feature importance.

3) Catboost - performance and feature importance: Figure 9 shows how CatBoost evaluates the importance of attributes in the training data. Similar to what we observed while doing EDA, the **time approval date** attributes plays an important signal for the model to perform well. Next is **flat type** and information related to apartment locations such as subzone, street name. Surprisingly, CatBoost does not consider **apartment size** as significant as we initially hypothesized.

As shown in Table V, CatBoost's models achieved the best RMSE at 480.39 on V3 setting. However, the algorithm itself works as a black-box for prediction. Many of the parameters are automatically selected by the algorithm based on problem setting such that loss function, attribute sizes. Therefore, it is hard to really interpret the results of produced by CatBoost due to large amount of factors controlling the method.

E. Final submissions

The final submission on Kaggle is created by a CatBoost model using V3 setting and the aforementioned parameters on all training data without cross-validation. On the public validation, we got RMSE score at 477.66. On one happy accident during experimentation phase of the project, we luckily got score at 477.52 which is top-1 at the moment of writing this report (09/11/2023).

IV. Conclusions

With datasets related to flat properties, MRT locations, malls locations, and primary location, we are able to build models to predict HDB flat rental price up to a level of accuracy. We have seen in our experiments that with additional data attributes, the prediction results improve. The error in predictions also reduces with an increase in the complexity of a model. From the experiments, the CatBoost model achieved the best accuracy according to the blind test score on Kaggle as well as in our 10-fold cross-validation with training data. We are hopeful that this model could be used to predict rental prices for both landlords and prospective renters. However, as we saw from the EDA and the model evaluation, year of the rental plays a significant role, the model could become less accurate with the progress of time, and would require retraining with up-to-date data in the future. Otherwise, the models could be affected by temporal data shift. For example, when government applies cooling measure to help lower the property prices, the rental price might drop instead of increase. Furthermore, while we saw little evidence in the effect of planned MRT stations to rental prices, it could come into effect when (or closely before) they start operating. This, along with many other factors, should be accounted for when considering future prediction or model training.

References

[1] Hancock, John T., and Taghi M. Khoshgoftaar. "CatBoost for big data: an interdisciplinary review." Journal of big data 7.1 (2020): 1-45.

[2] Micci-Barreca, Daniele. "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems." ACM SIGKDD Explorations Newsletter 3.1 (2001): 27-32.

[3] Understanding Random Forests: From theory to practice. PhD thesis. Gilles Louppe (2015).