Predicting HDB Rental Rates: Finding the Right Flat

CS5228 Final Project

AY 2023/2024 Semester 1

Kaggle Team Name: CS5228 Group(27)

Github Repo Link: https://github.com/sailu1997/CS5228_FinalProject

Edmund Tham Ren Jiat
A0250679U
e0945861@u.nus.edu

Yip Weiheng Darius
A0155793R
e0031930@u.nus.edu

Premi Jeevarathinam
A0268262Y
e1101561@u.nus.edu

Sai Niharika Naidu Gandham
A0268520E
e1101819@u.nus.edu

***Abstract*—HDB rental prices can be predicted by data mining methods based on analysis of market supply and demand factors in addition to traditional data sources. This report will reflect the implementation of EDA, data pre-processing and model training to predict HDB rental prices.**

I. Introduction

HDB rental prices in Singapore have been rising rapidly, leading to an information gap among landlords, tenants, and real estate agents. This project addresses the challenge by implementing a predictive model for HDB rental prices. Utilizing data mining techniques, we analyse a dataset from data.gov.sg, encompassing variables such as rent approval date, town, block, street name, monthly rental, flat type and size, lease commencement date, and geo-data. Additionally, we explored an auxiliary dataset containing geo-data for primary schools to uncover valuable insights. The goal is to establish a level-playing field for all stakeholders in the HDB rental market. This report encapsulates our journey, from navigating complexities to justifying design choices, recognizing that creativity and methodical decision-making are essential in the absence of a singular best approach in data mining.

II. Motivation and Goals

A. Motivation

In this project, we explore the rental market in Singapore, focusing on HDB flats. We understand the financial challenges people face in finding affordable accommodation, especially with rising rental prices. Prospective tenants are looking for clear information, while landlords and agents aim to make informed decisions.

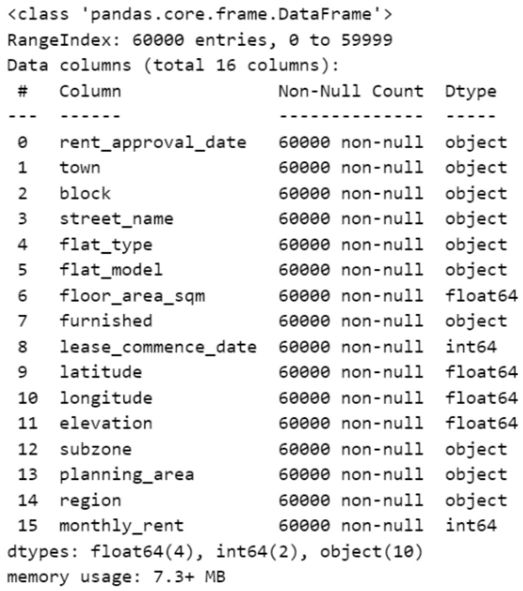
B. Goals

Our main goal is to predict HDB flat rental rates using historical data. We emphasize the importance of carefully justifying, deriving, and evaluating different features, aiming for outcomes beyond just predicting monetary values.

III. Exploratory Data Analysis and Pre-processing

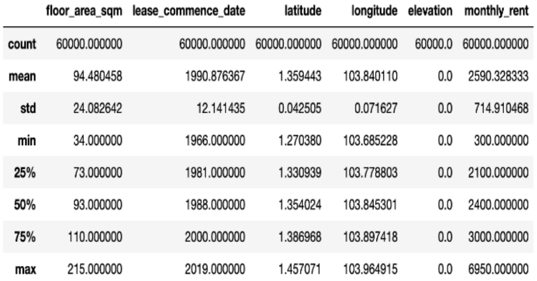
A. Exploratory Data Analysis

In data mining, EDA is an approach to analyse the dataset and visualise to summarize their main characteristics. We have train and test datasets with 60K and 30K samples respectively with 16 attributes in total along with the target attribute 'monthly_rent'. Fig.1 also shows there are no null values in the dataset and it shows all the attributes present and their datatypes.Fig.2 gives the description of the numerical data.Fig.3 shows the hist plots of numerical data, in which 'elevation' values are same across all the samples and the average monthly rent of most of the samples are equal to 2590. Table.1 shows the unique counts of values of each categorical attribute implying that 'block' and 'street_name' have relatively higher unique values. Also 'furnished' is same across all the samples.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60000 entries, 0 to 59999
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   rent_approval_date   60000 non-null  object 
 1   town                 60000 non-null  object 
 2   block                60000 non-null  object 
 3   street_name          60000 non-null  object 
 4   flat_type            60000 non-null  object 
 5   flat_model           60000 non-null  object 
 6   floor_area_sqm       60000 non-null  float64
 7   furnished            60000 non-null  object 
 8   lease_commence_date  60000 non-null  int64  
 9   latitude             60000 non-null  float64
 10  longitude            60000 non-null  float64
 11  elevation            60000 non-null  float64
 12  subzone              60000 non-null  object 
 13  planning_area        60000 non-null  object 
 14  region               60000 non-null  object 
 15  monthly_rent         60000 non-null  int64  
dtypes: float64(4), int64(2), object(10)
memory usage: 7.3+ MB
```

Fig 1. info() of training dataset



	floor_area_sqm	lease_commence_date	latitude	longitude	elevation	monthly_rent
count	60000.000000	60000.000000	60000.000000	60000.000000	60000.0	60000.000000
mean	94.480458	1990.876367	1.359443	103.840110	0.0	2590.328333
std	24.082642	12.141435	0.042505	0.071627	0.0	714.910468
min	34.000000	1966.000000	1.270380	103.685228	0.0	300.000000
25%	73.000000	1981.000000	1.330939	103.778803	0.0	2100.000000
50%	93.000000	1988.000000	1.354024	103.845301	0.0	2400.000000
75%	110.000000	2000.000000	1.386968	103.897418	0.0	3000.000000
max	215.000000	2019.000000	1.457071	103.964915	0.0	6950.000000

Fig 2. describe() of training dataset

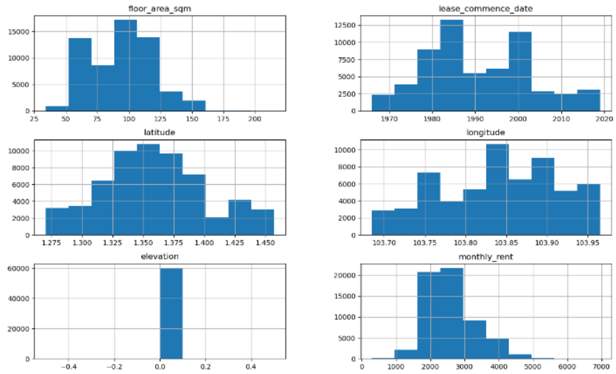


Fig 3. Histograms of numerical attributes

Categorical Attribute	Unique Values
town	26
block	2553
street_name	1083
flat_type	9
flat_model	19
furnished	1
subzone	152
planning_area	29
region	5

TABLE 1. Unique Count values for categorical attributes

B. *Visualisation of patterns in rental prices based on the Auxiliary dataset*

For data visualisation, we plotted out the rental prices using KeplerGL to investigate if there were any obvious patterns in rental prices (such as closer proximity to MRT stations which resulted in higher rental prices).

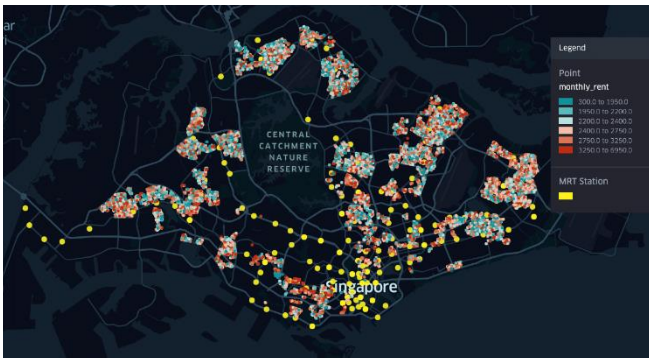


Fig 4. Rental prices visualisation using KeplerGL

From the visualisation, it indicates that while some rental prices are a result of being near a MRT station, it is not always the case. Furthermore, we see more rentals with higher prices towards the south of Singapore, suggesting that there might be some prime locations with generally higher rental prices compared to the rest of Singapore.

Taking a closer look at the distribution of rental prices over time, we saw that there was a general increasing trend in rental prices (see figures 8, 9 and 10 in appendix), but this increasing trend in rental prices over time was not captured in any of our independent variables. Hence, this suggested that there was a need to include a feature that captured this increase in price index over time.

C. *Data Pre-processing*

From the exploratory data analysis, we believe that the dataset was in rather good shape and did not require much pre-processing steps before it could be used. In our case, the data pre-processing steps consisted of the following:

- Parsing the ‘rent_approval_date’ into a datetime object
- Converting spaces to underscores
- Converting all strings to lowercase

IV. Dataset Feature Preparation

A. *Feature Engineering on Existing Attributes*

Feature engineering is the process of transforming the given raw data so it can be used in training and prediction. Based on our problem we may need to transform or encode the samples as required. For numeric features, we decided to leave them as they are, since normalization did not really contribute to the performance of our result. Attributes ‘elevation’, ’furnished’ are 0 and ‘yes’ respectively for all the sample hence dropped due to the lack of variability.

While algorithms like decision trees can directly learn from numerical data, most machine learning algorithms cannot be operated on categorical data, hence the non-numerical or categorical attributes need to be handled properly. Initially, we merged 'flat_type' and 'flat_model' due to their 39 unique values, and subsequent analysis involved both one-hot encoding and target encoding. However, the outcomes remained consistent even after dropping these features. Similarly, 'subzone' and 'planning_area' were excluded for the same reason as combining and target encoding failed to enhance the model's performance.

‘flat_type’ and ‘region’ are one-hot encoded for 5 unique values. The other columns ‘town’, ‘block’, ‘street_name’, ‘flat_model’, ‘subzone’ and ‘planning_area’ all have high cardinality and are dropped as they all exhibit high cardinality. Attempts to one-hot encode them would result in many resulting feature columns.

Categorical attribute	Handling method	Reason
town	Dropped	Arguably like planning_area
block	Dropped	Higher granularity
street_name	Dropped	Higher granularity
flat_type	One-hot encoded	5 unique values only
flat_model	Dropped	High cardinality
furnished	Dropped	Single value across the entire dataset
subzone	Dropped	High cardinality
planning_area	Dropped	High cardinality
region	One-hot encoded	5 unique values only

TABLE 2. Handling categorical attributes

B. *Source of Additional Dataset CPI Housing Data*

To understand how housing costs change over time, we explored additional datasets. However, none of them precisely represented the overall housing price trend in Singapore. Although COE prices and stock prices might be relevant, we wanted a feature that specifically reflects the general housing price level based on factors like housing demand and supply. To address this, we included data from SingStat's website, which provides the Housing Consumer Price Index (CPI) for Singapore.

A simple visualisation of the additional dataset reflecting the CPI for housing from 2020 to 2023 confirmed our belief that there was a trend of increasing housing prices over time.

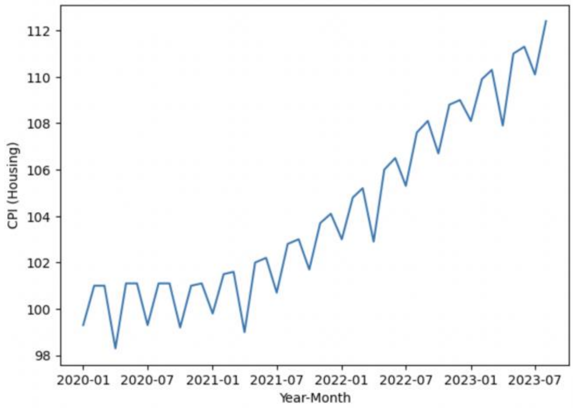


Fig 5. CPI (Housing) from 2020 to 2023

C. *Generation of New Features*

We computed the distances to existing and planned MRT stations, primary schools, and shopping malls within a 1 km radius for each flat. Despite this effort, subsequent analysis revealed no notable performance improvement. Consequently, we opted to exclude these features from our final models.

We created a new feature 'remaining_lease_years_at_rent' that captures the remaining years of lease the house has left. The intuition is - greater the number of years the lease has left, the more valuable the house would be, translating to higher rental prices.

A second feature created would be 'distance_from_dg' (Distance from Dhoby Ghaut mrt). From our finding in the exploratory data analysis section above, we discussed the possibility of higher rentals in prime locations. From our domain knowledge, it is generally known that housing/rental prices would increase the closer the unit is to the Central Business District. Therefore, we created a new feature "distance_from_dg" which reflects the distance of the unit from the Central Business District, using Dhoby Ghaut MRT station as a proxy to represent the location of the Central Business District. Another factor to consider would be popular housing districts in Singapore, as we noticed that there were some districts more high rental prices. Arbitrary, we selected "Pioneer", "Redhill", "Sengkang", "Tampines" and "Ang Mo Kio" MRT stations as the popular housing districts and created new features that reflected the distance of the unit from these stations.

The new feature "cpi" (Consumer Price Index) was created by mapping the month and year that the rental transaction happened to the housing CPI for that same month and year in our auxiliary Housing CPI dataset.

With the newly created features, we can proceed to examine the correlation between the numerical features in the dataset and the monthly rental prices.

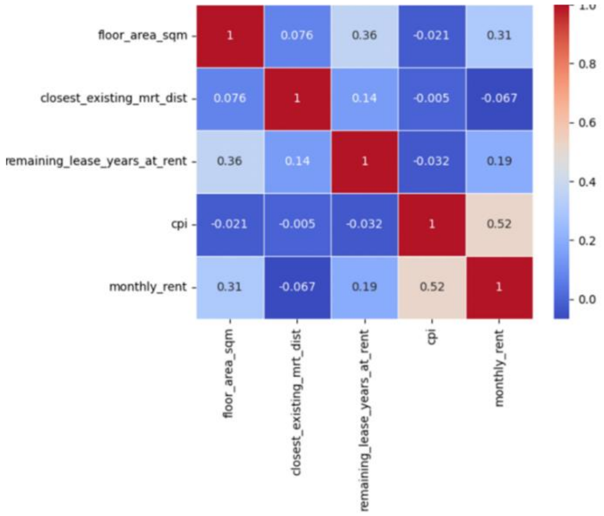


Fig 6. Correlation Heatmap between numerical attributes and target

D. *Feature Selection*

In our final dataset, we included all the engineered features but dropped some columns that were either uninformative (such as "furnished" and "elevation"), as well as categorical variables that had high cardinality and would result in many features if we were to one-hot-encode them. From this final dataset, we proceeded to fit our considered models, focusing mainly on tree-based models that would automatically select the best features to split on.

V. Data Mining Methods

A. *Model Selection*

With the final dataset prepared, including the engineered features and the inclusion of auxiliary data, we proceeded with the fitting of the dataset to train machine learning models. We mainly considered state-of-the-art tree-based models as their underlying algorithms included a feature selection element and non-linearity. However, in the initial report we used a linear regressor so we would still be keeping the linear regressor model for comparison purposes.

The various models we tried are

- Linear Regression
- AdaBoost
- Random Forest
- XGBoost

For the purposes of selecting a final model, we focused on RMSE as the main metric to select the best-performing model. For each machine learning model considered, we trained and validated the different models using the same dataset after splitting and repeated the process 10 times using different splits (Table 1 in Appendix) With the results, we selected the model with the lowest RMSE across the 10 times.

Singstat (https://tablebuilder.singstat.gov.sg/table/TS/M212882)

Among the models under consideration, the XGBoost Regressor had a lower average RMSE as compared to the Random Forest Regressor, Adaboost and Linear Regressor. Hence, the XGBoost Regressor was selected as the final model which we would optimize further.

Model	Mean RMSE Value
Linear Regression	7228742.574346982
AdaBoost	543062.5104637535
Random Forest	281523.30019723676
XGBoost	244500.58510952623

TABLE 3. Mean RMSE vakyes from various models

B. *Cross-validation and Hyperparameter Tuning*

Hyperparameter tuning and cross-validation are crucial for improving the model's performance. We did hyperparameter tuning on our best performing model XGBoost. We explored GridSearchCV, Optune and Hyperopt for tuning the parameters.

Hyperopt outperformed by showing superior model performance than the other techniques. Hyperopt employs Bayesian optimization with the Tree-structured Parzen Estimator (TPE) algorithm by exploring the search space that more likely lead to better results. Also, it employs automated search strategy than GridSearchCV which employs manual search grid. Using Hyperopt we tuned n_estimators, learning_rate, max_depth, min_child_weight, sub_sample, reg_lambda, reg_alpha, objective based on their impact on the performance of the XGBoost model. With Objective parameter reg:gamma model performed best when compared to reg:squarederror and reg:tweedie which shows that the target variable is gamma distributed.

Cross-validation is a technique for assessing the model performance and generalization across multiple subsets of the dataset. It provides more robust evaluation of model performance compared to single train-test split. We employed 5-Fold and 10-Fold cross validation strategies.10-Fold gave the best results compared to 5-Fold. Each fold involved randomly splitting the dataset into training and validation sets and then tuning for the best hyperparameters with the objective of minimizing the RMSE.

Following are the set of tuned hyperparameters at which our model performed best with the lowest RMSE score of 481.01677 on the test data.

Hyperparameters	Values
n_estimators	850.0
learning_rate	0.05865350580518565
Max_depth	4.0
Min_child_weight	1.895116573861646
Sub_sample	0.9715714637190287
Reg_lambda	0.4555658474213259
Reg_alpha	4.894859202308267

TABLE 4. Set of tuned hyperparameters of our model

VI. Evaluation and Interpretation

After the process of data analysis, feature engineering, and model selection, here is the evaluation and interpretation on the performance of our final model.

- The chosen XGBoost Regressor, with a mean RMSE of 244,500.59, outperformed other models, showcasing its robust predictive capabilities for HDB flat rental rates.
- Hyperparameter tuning using Hyperopt, a Bayesian optimization technique, proved superior in fine-tuning model parameters, contributing to enhanced predictive accuracy.
- The 10-Fold cross-validation strategy validated the model's generalization capabilities, ensuring reliability across diverse subsets of the dataset.
- Key features influencing rental predictions include "floor_area_sqm," "remaining_lease_years_at_rent," and "cpi," as highlighted by the correlation heatmap. External data, specifically the Housing Consumer Price Index (CPI) dataset, dynamically captured variations in accommodation prices over time, addressing the need to reflect the general increase in rental prices observed.
- Visualization tools, like KeplerGL, aided in identifying patterns in rental prices, reinforcing correlations with factors such as location and amenities. The model's ability to capture the overall increasing trend in rental prices over time underscored the importance of incorporating temporal features.

So, the evaluation affirms the effectiveness of our predictive model in navigating the challenges posed by rising HDB rental prices. The combination of advanced machine learning techniques, thoughtful feature engineering, and meticulous hyperparameter tuning has yielded a model that not only accurately predicts rental rates but also provides valuable insights for stakeholders in the HDB rental marke

VII. Conclusion

In conclusion, this project has tackled the complexities of the HDB rental market in Singapore by creating a predictive model. The process involved careful analysis of data, crafting features, and choosing the right model to offer insights for landlords, tenants, and real estate agents. The journey was a learning adventure, exploring different machine learning models, understanding metrics like RMSE, and trying out hyperparameter tuning techniques. Applying data mining to address the issue of rising HDB rental prices showcased the practical impact of our skills. Overall, from selecting models to using efficient search strategies, each step contributed to our growth as data scientists. This project not only produced a predictive model but also equipped us with valuable insights and problem-solving skills for future challenges in the field of data mining.

VIII. APPENDIX

A. Figures and Tables

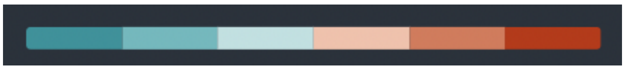


Fig.7. colour scale for visualisation

Key:

- *blue for lower rental prices*
- *red for higher rental prices*

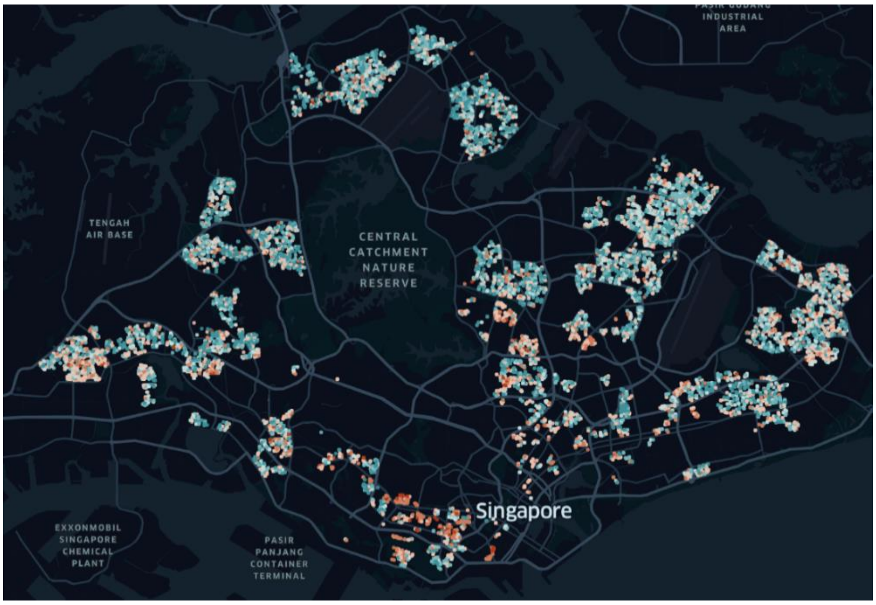


Fig.8. rental prices in 2021

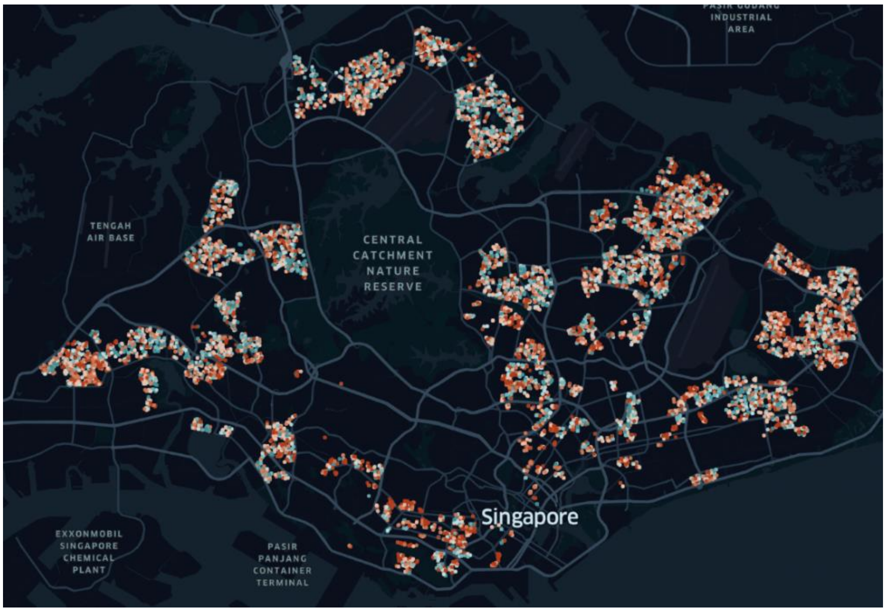


Fig.9. rental prices in 2022

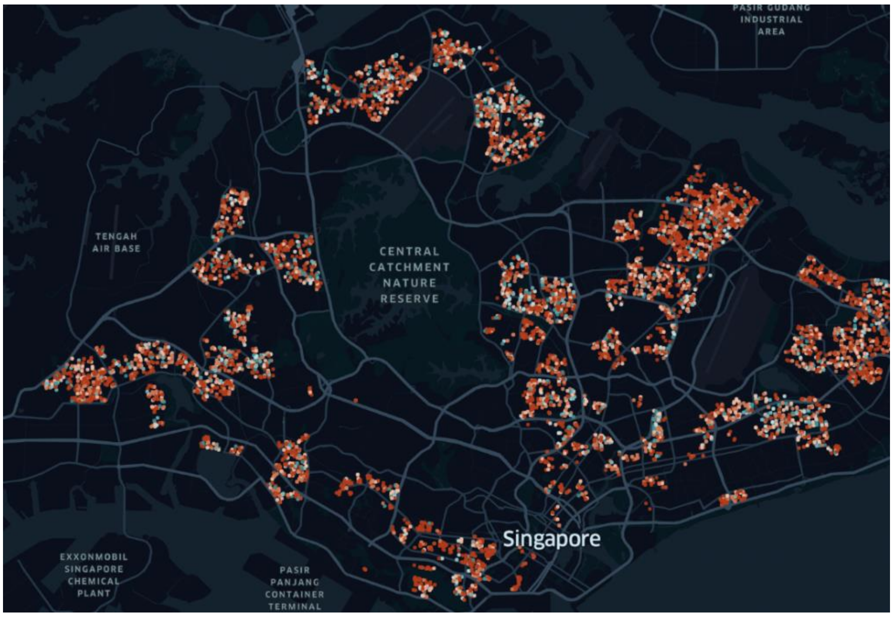


Fig.10. rental prices in 2023

RMSE values of Different models across 10 runs				
Runs	Linear Regression	**AdaBoost**	**Random Forest**	**XGBoost**
1	7243685.618082804	833714.0046877615	282201.80067520525	244808.77634824015
2	7180018.7981989775	539966.0974384731	284755.45676403417	246342.72812565867
3	7229596.024170011	465269.54629164573	279765.44557552354	241341.7719114932
4	7188829.198136076	576316.5494726268	281166.3527842966	242716.94123488117
5	7254826.820952475	710000.8371002846	281166.8413925027	244934.50785692647
6	7177782.678544397	425829.9849321949	281965.2694567983	243793.16771906576
7	7298734.93535373	502553.155868073	279720.1904464856	244182.4558603669
8	7205988.383469392	436434.5806127574	285598.9306780734	247645.2362555199
9	7212445.958990271	479664.9506110832	278070.48425483203	241761.96823874107
10	7295517.327571693	460875.3976226352	280822.2299446162	247478.29754436895

Table.5. RMSE table for respective model

B. Breakdown of Workload

Team Members details				
Name	Edmund Tham Ren Jiat	Premi Jeevarathinam	Sai Niharika Naidu Gandham	Yip Weiheng Darius
Student Id	A0250679U	A0268262Y	A0268520E	A0155793R
NUSNET Id	E0945861	E1101561	E1101819	E0031930
Workload Split				
EDA	✓	✓	✓	✓
Data Preprocessing	✓	✓	✓	✓
Model Training	✓	✓	✓	✓
Hyperparameter Tuning	✓	✓	✓	✓
Report Writing	✓	✓	✓	✓