Predicting HDB Rental Rates: Finding the Right Flat

CS5228 Final Project (AY 2023/2024 Semester 1)

Group 27
Edmund Tham Ren Jiat (A0250679U)
Premi Jeevarathinam (A0268262Y)
Sai Niharika Naidu Gandham (A0268520E)
Yip Weiheng Darius (A0155793R)

Background

HDB rental prices have been trending upwards rapidly in recent years, potentially resulting in a mismatch of information between landlords, tenants and real-estate agents.

This project aims to implement a predictive model for HDB rental prices in Singapore to provide a level-playing field for all stakeholders through the application of data mining techniques to analyze a core dataset of rental rates for HDB flats provided by data.gov.sg in order to identify correlating factors that affect HDB rental prices.

The dataset contains rent approval date, town, block, street name, monthly rental, flat type and size, lease commencement date, and geo-data. Auxiliary dataset containing geo-data for primary schools will also be analyzed to uncover further data insights.

Exploratory Data Analysis

Method

The data was analyzed using statistical and visualization techniques

Observation

Analysis revealed potential correlation between HDB rental prices and the following factors:

- Location
- Size
- Amenities
- Transportation
- General increase in rental prices over time

Data cleaning and preprocessing

Method

The data was **cleaned** and **preprocessed** to remove errors and inconsistencies

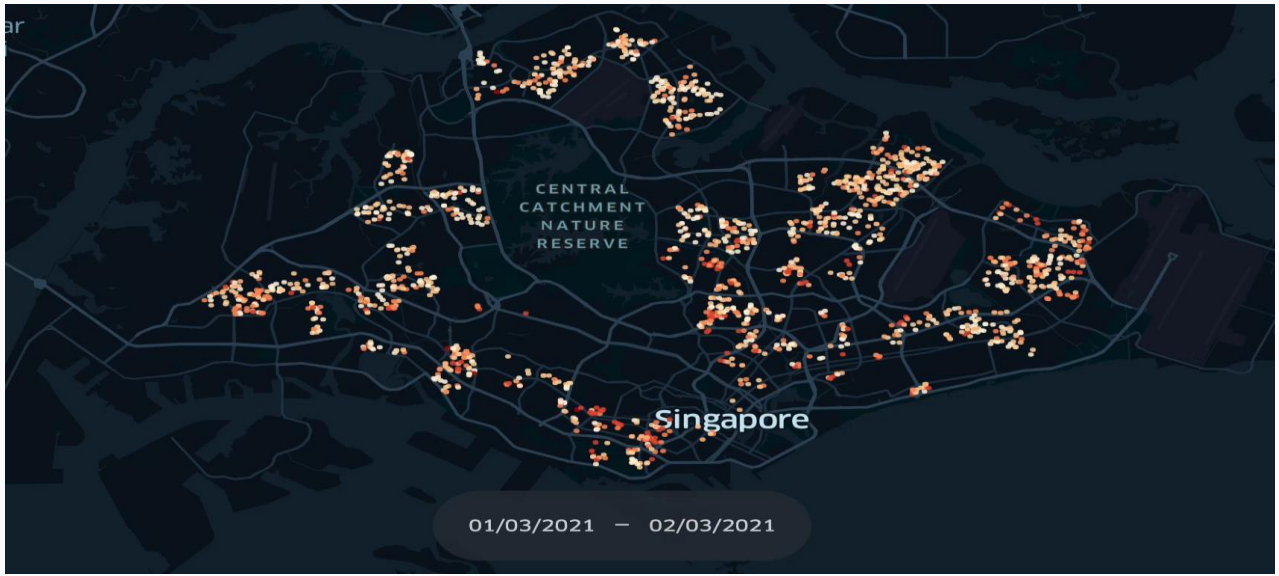
Observation

- White spaces were replaced with underscores
- Uppercase strings were corrected to lowercase
- Data was **standardized** to make it easier to compare different variables.

Handling Categorical Data

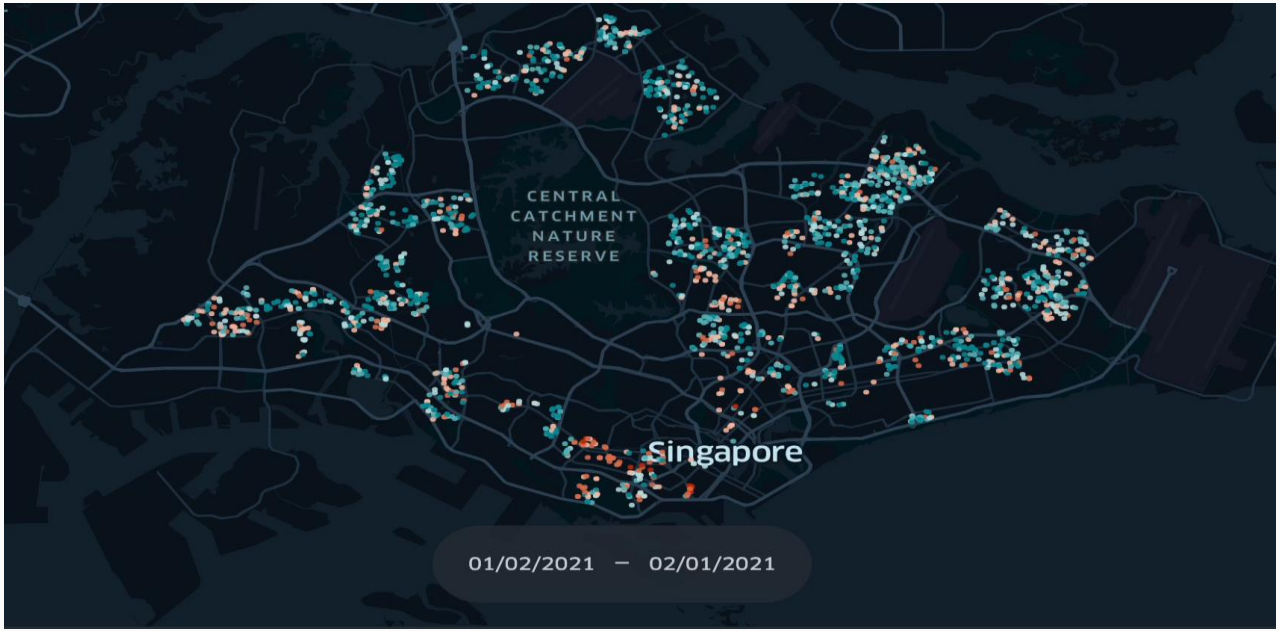
Attribute	Action	Rationale
block	Drop	High number of unique values (~2K), granularity is less meaningful for finding similarities.
elevation	Drop	Lack of variability unlikely to provide meaningful data for predictive modeling
flat_age	New feature	Calculated from `lease_commence_date` and `rent_approval_date`
flat_model	Merge with `flat_type`	Resultant feature *one-hot encoded* for 39 unique values; 12 itemsets dropped after comparison with test dataset
flat_type	Merge with `flat_model`	
furnished	Drop	Lack of variability unlikely to provide meaningful data for predictive modeling
planning_area	Merge with `subzone`	Resultant feature *target encoded*. *One-hot encoding* was rejected due to high cardinality that would result in a sparse matrix impacting model learning ability.
region	One-hot encoding	Attribute has only 5 unique values
street_name	Drop	High number of unique values (~1.8K), granularity is less meaningful for finding similarities.
subzone	Merge with `planning area`	Refer to rationale for `planning_area`.
town	Drop	Duplicate with `planning_area`

Visualization (Approval date vs rent)



Lower rent to higher rent

Visualization (Approval date vs flat age)



Younger house age to older house age

Fit-to-base Linear Regression

Method

We have observed these performances with the above shown encoding techniques and on a basic *Linear Regressor*.

Observation

Most of the columns are one-hot encoded and only 2 columns are actual features ("flat_age" and "floor_area_sqm"), this resulted in sub-optimal performance of a Linear Regressor model against the test dataset.

Next Steps

Further Approach	Rationale
Target encoding for categorical attributes	For attributes with high cardinality, one hot encoding would result in a sparse matrix which might impact the model's ability to learn.
Feature engineering including auxiliary data	Auxiliary data, such as data on the local economy and demographics, will be used to engineer new features. The new features were used to train an improved model.
Exploration of other regression models	Decision Trees for Regression: Decision trees split the data into subsets based on input features. They can handle non-linear relationships and provide interpretable rules. K-Nearest Neighbors (for Regression): KNN can be used for regression tasks by taking the average (or weighted average) of the 'k' nearest training examples.
Hyperparameter tuning	After we have identified the model that performs the best on our dataset. We would consider performing parameter hypertuning, to determine what are the best arguments to use for parameters such as number of estimators, learning rate, max_depth etc.

Thank you :)