50.021 Artificial Intelligence

HDB PUBLIC HOUSING PRICE PREDICTION PROJECT

April 15, 2024

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Contents

1 Introduction

1.1 Background Information

The Housing Development Board (HDB) public housing market in Singapore stands as a vital pillar of the nation's residential landscape, accommodating a significant portion of its population and boasting an impressive 90% homeownership rate. The HDB resale market presents several unique challenges that necessitate precise price prediction. Firstly, the market is influenced by a myriad of factors. These factors interact in complex ways making it difficult to discern their individual impacts on resale prices.

Moreover, the HDB resale market is subject to dynamic shifts influenced by economic trends, government policies and societal changes. Therefore, accurate predictions of HDB resale prices are crucial for various stakeholders like homeowners and prospective buyers.

In response to this challenge, this report presents an AI-driven project focused on predicting HDB public housing prices in Singapore. By leveraging advance machine learning and AI techniques, we aim to develop a predictive model that can provide insights into the factors influencing resale prices and facilitate informed decision-making for homeowners, buyers and policymakers. Throughout this report, we will delve into the various aspects of the project, starting with an exploration of the key factors influencing HDB resale prices, such as location, flat size, type, remaining lease and proximity to amenities. Subsequently, we detail our methodology using the AI project lifecycle framework. Furthermore, we will discuss findings from each of the phases in the project lifecycle and lastly conclude with insights gleaned from our project and recommendations for future research.

1.2 Problem Statement

How can we leverage AI-driven solutions to accurately forecast resale flat prices in Singapore's public housing market, incorporating the impact of nearby amenities to address the need for precision amidst market fluctuations?

1.3 AI Project Cycle

For our project, we adhered closely to the AI project lifecycle to ensure a structured and efficient development process. By adhering to this framework, we were able to navigate through each stage of the project - from problem scoping to evaluation-with clarity and precision. This approach provided us with a systematic methodology for managing tasks, identifying potential risks and ensuring the quality of our AI-driven solutions.

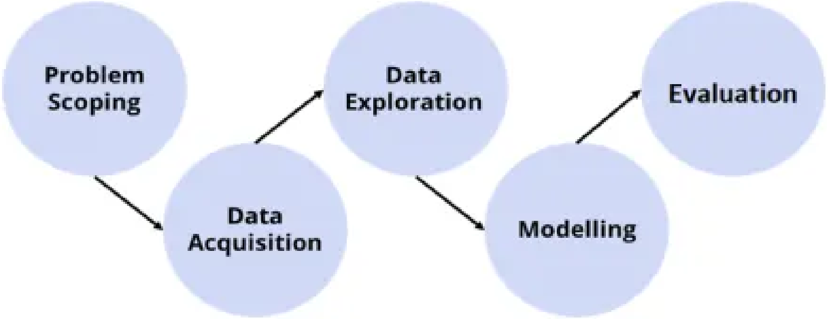


Figure 1: AI Project Lifecycle

1. Problem Scoping

Initially, our project aimed to address the challenge of predicting HDB flat prices, recognizing the broad scope encompassing various buyer demographics. To refine our focus, we decided to concentrate specifically on resale flats, targeting a more defined audience segment. This decision enabled us to tailor our approach to the specific needs and preferences of resale flat buyers, ensuring that our predictive model would be more accurate and relevant.

2. Data Acquisition

In the data acquisition phase, we meticulously gathered multiple datasets to facilitate comprehensive analysis and modeling. Our primary dataset contained crucial information about resale flat prices, including key features such as address, town, flat type, and flat model. Additionally, we acquired supplementary datasets to enrich our analysis further. These included HDB property information, providing insights into building types and amenities like multistory car parks. Furthermore, we collected data on various amenities such as MRT stations, schools, shopping malls, and hawker centers, which added geographic context with address, latitude and longitude coordinates. This extensive dataset acquisition process ensured that we had a robust foundation for our analysis, incorporating both essential pricing data and contextual information relevant to potential buyers.

3. Data Exploration

During the data exploration phase, we delved deep into our datasets using a variety of visualization techniques. Through uni-variate and bi-variate analysis, we sought to understand the intricate relationships between different features and the target variable—resale flat prices. By plotting graphs and charts, we visualized how various factors such as location, flat type, and amenities influenced prices. This exploration process allowed us to identify significant patterns and correlations within the data, providing valuable insights into the drivers of resale flat prices. Moreover, it helped us gain a nuanced understanding of the complex interactions between different variables, laying the groundwork for our subsequent modeling efforts.

4. Modelling

In the modeling phase, we embarked on an extensive experimentation process, exploring a diverse range of models to address the predictive task effectively. Leveraging our comprehensive dataset, we tested five different types of models: Decision Tree Regressor, Gradient Boosting Regressor, Random Forest Regressor, feed-forward neural networks, and ARIMA. By experimenting with multiple models, we aimed to avoid bias and ensure that our final choice was based on empirical performance rather than theoretical assumptions. This approach allowed us to capture the diverse patterns present in the data and select the most suitable model for predicting resale flat prices accurately.

5. Evaluation

For model evaluation, we selected the Root Mean Square Deviation (RMSE) as our primary metric, enabling us to compare the performance of different models objectively. After evaluating the RMSE scores of all models, we determined that the Random Forest Regressor exhibited the best performance. This model leverages ensemble learning techniques, combining multiple decision trees to generate more accurate predictions. By selecting the Random Forest Regressor as our final model, we ensured that our predictive model could effectively capture the complex relationships between various features and resale flat prices, providing valuable insights for potential buyers in the housing market.

2 Datasets

2.1 Base Datasets

For our project, we used 2 primary datasets which served as the foundation of this project before we added other secondary sources.

Singapore Public Housing Resale Flat Prices

This dataset contains a list of Singapore's public housing resale flat prices in Singapore dollars (SGD) from year 2017 to 2023.

Variable	Description
Town	Towns in Singapore with HDB properties
Flat Type	Types of HDB property (2 Room, 3 Room, etc.)
Lease Commence Date	Date of lease of HDB property
Block	Block number of HDB property
Street Name	Street name of HDB property
Flat Model	Different types of models of HDB property
Resale Prices	Resale prices of HDB property
Storey Range	The storey range of HDB property's location
Floor Area	Floor area of HDB property

Table 1: Items in Dataset

HDB Property Information

This dataset contains HDB Property Information such the location of existing HDB blocks, highest floor level, year of completion, type of building and number of HDB flats (breakdown by flat type) per block etc.

Variable	Data Type	Description
blk_no	String	Block number of the building
street	String	Street name where the building is located
max_floor_lvl	Integer	Maximum number of floors in the building
year_completed	Integer	Year when the construction of the building was completed
residential	Boolean	Whether Residential Building
commercial	Boolean	Whether Commercial Building
market_hawker	Boolean	Whether Hawker centre nearby
multistorey_carpark	Boolean	Whether there are multistorey carparks in the building
precinct_pavilion	Boolean	Whether there is a precinct pavilion
bldg_contract_town	String	Town where the building is located
total_dwelling units	Integer	Total number of dwelling units in the building
1room_sold	Integer	Number of one-room units sold
2room_sold	Integer	Number of two-room units sold
3room_sold	Integer	Number of three-room units sold
4room_sold	Integer	Number of four-room units sold
5room_sold	Integer	Number of five-room units sold
exec_sold	Integer	Number of executive units sold
multigen_sold	Integer	Number of multigenerational units sold
studio_apartment_sold	Integer	Number of studio apartments sold
1room_rental	Integer	Number of one-room units rented out
2room_rental	Integer	Number of two-room units rented out
3room_rental	Integer	Number of three-room units rented out
other_room_rental	Integer	Number of other types of rooms rented out

Table 2: Data Dictionary

2.2 Additional Datasets

In our efforts to enhance the accuracy and robustness of our house price prediction model, we have included some supplementary datasets. These datasets cover the different amenities and various parts of the city, giving us a better understanding of what affects housing prices in different areas.

Primary Schools

This dataset contains a list of primary schools in Singapore with their corresponding area. This dataset enables us to assess the proximity of housing units to educational institutions, which is a crucial factor for families with school-going children.

Variable	Type	Description
Name	String	Name of Primary School
Town	String	Town of Primary School

Shopping Malls

This dataset contains a list of shopping malls in Singapore with their latitude and longitude coordinates. This allows us to evaluate the accessibility of retail establishments within the vicinity of residential properties, as proximity to shopping centres often correlates with higher property values due to the convenience and amenities offered.

Variable	Type	Description
Name	String	Name of Shopping Mall
Latitude	Numeric	Latitude of location of Shopping Mall
Longitude	Numeric	Longitude of location of Shopping Mall

MRT Stations

This dataset contains a list of MRT Stations with their latitude, longitude, and the line they belong to. Integration of MRT station data aids in understanding transportation connectivity, which influences housing demand and prices.

Variable	Type	Description
Name	String	Name of MRT/LRT
Latitude	Numeric	Latitude of location of MRT/LRT
Longitude	Numeric	Longitude of location of MRT/LRT

Hawker Centres

This dataset contains a list of Hawker Centres and Markets in Singapore with information such as name, location, and type of centre. Proximity to these food destinations can be indicative of neighbourhood vibrancy and lifestyle preferences, thereby impacting housing demand and values.

Variable Name	Type	Description
name_of_centre	String	Name of Center
location_of_centre	String	Address of Center
type_of_centre	String	Type of Center (e.g., Market only, Hawker Centre only)
owner	String	Owner of Center (e.g., HDB or Government)
no_of_stalls	Numeric	Total number of stalls
no_of_cooked_food_stalls	Numeric	Number of cooked food stalls
no_of_mkt_produce_stalls	Numeric	Number of market produce stalls

3 Utilization of OneMap API

In this project, we utilized the OneMap API to gather information about nearby amenities such as schools, hawker centres, shopping malls and MRT stations. This information was scraped from the API and integrated into our final dataset to enhance the predictive capabilities of our housing price model.

`find_postal(lst, filename)`

To retrieve location details for a list of addresses, we developed the `find_postal` function. This function iterates through the list of addresses, queries the OneMap SG API, and retrieves the location details. The response from the API is then converted into a pandas DataFrame, and the original address is appended to each row. Finally, the data is exported to a CSV file specified by the `filename` parameter.

`clean_address(string, dictionary)`

To clean up the addresses before querying the API, this `clean_address` function was created. This function replaces certain words or phrases found in the address string with their corresponding values from the provided dictionary. This helps to standardize the address format and improve the accuracy of the queries.

`find_nearest(house, amenity)`

This function is responsible for finding the nearest amenity to each address in the `house` DataFrame from a list of amenities in the `amenity` DataFrame. It calculates the distance between each address and every amenity using latitude and longitude coordinates and returns a dictionary containing the nearest amenity for each address along with the distance.

`save_file(data, filename)`

Once the nearest amenities are identified, the `save_file` function takes the data dictionary generated by `find_nearest` and saves it into a CSV file specified by the `filename` parameter. This ensures that the information is stored and accessible for further analysis and integration into our housing price prediction model.

4 Data Pre-processing

Data pre-processing is a critical stage in our data analysis step, involving the transformation, cleaning, and preparation of raw data to ensure its quality and usability for subsequent analysis and modeling. In our project, we meticulously conducted various pre-processing steps to optimize our dataset for predictive modeling while ensuring accuracy and reliability.

4.1 Data Cleaning

Data cleaning involves handling missing values, removing duplicates, and ensuring consistency in the dataset.

(a) **Drop NA rows**: We adopted a systematic approach to handle missing data, opting to drop rows containing NA values to prevent any potential distortions in our predictive models.

(b) **Rename Columns for Datasets**: Additionally, to enhance clarity and ease of use across all stages of our project, we embarked on renaming the original columns, ensuring consistency and facilitating seamless data management throughout the project life cycle.

4.2 Data Integration

Data integration was a crucial aspect of our pre-processing efforts, enabling us to enrich our dataset with additional information relevant to our predictive modeling task. Leveraging supplementary datasets and utilizing the OneMap API, we identified the nearest amenities such as MRT stations, schools, shopping malls, and hawker centers for each resale flat. This involved a meticulous process of data collection and merging, in the creation of a comprehensive dataset that encapsulated key factors influencing resale flat prices.

4.3 Feature Engineering

To enhance the predictive power of our models, we embarked on feature engineering, a crucial step in extracting meaningful insights from our data.

(a) **Storey Range Transformation**: One notable aspect of this process was the transformation of the "StoreyRange" column. Originally containing object values such as "10 TO 12," we decomposed this into two separate integer columns—StartStoreyRange and EndStoreyRange. By breaking down the range into its constituent parts, we minimized the number of categorical columns in our dataset, thereby improving its efficiency and interpretability.

(b) **Dropping Low Occurrence Value**: We began by calculating the number of unique values for each column in our dataset. For flat types, we targeted those with counts below 100, excluding multi-generation and 1 room flats, as they exhibited insufficient representation to contribute meaningfully to our models. Similarly, for flat models, we applied a threshold of below 30 counts, excluding models such as Improved-Maisonette, Premium Maisonette, and 3 Gen. This strategic approach enabled us to declutter our dataset, focusing solely on the most informative and influential features for our predictive models.

(c) **One-Hot Encoding**: To prepare categorical features for inclusion in our predictive models, we applied one-hot encoding. This transformation technique converts categorical variables into a binary format, creating a separate binary feature for each unique category. We applied one-hot encoding to categorical features such as town, flat type, and flat model, ensuring compatibility with the AI models.

4.4 Data Splitting

Data splitting involves partitioning the dataset into separate training and testing sets to evaluate the performance of the model.

Training and testing sets: The dataset is split into training and testing sets, typically with around 80% of the data allocated for training and 20% for testing. This allows for the model to be trained on one subset of the data and evaluated on another to assess its generalization performance.

These pre-processing steps are essential for ensuring the quality and usability of the data for subsequent analysis and modeling tasks. By addressing issues such as missing values, duplicates, and encoding categorical variables, we can prepare the data effectively for building predictive models of housing prices.

By combining the resale prices dataset from Kaggle, the HDB property information dataset from data.gov.sg, and amenity information scraped using the OneMap API, a comprehensive dataset was created for the resale house price prediction AI project. It contains a variety of features related to property characteristics, location, and nearby amenities, which were used to train AI models for predicting resale house prices.

5 Visualisations

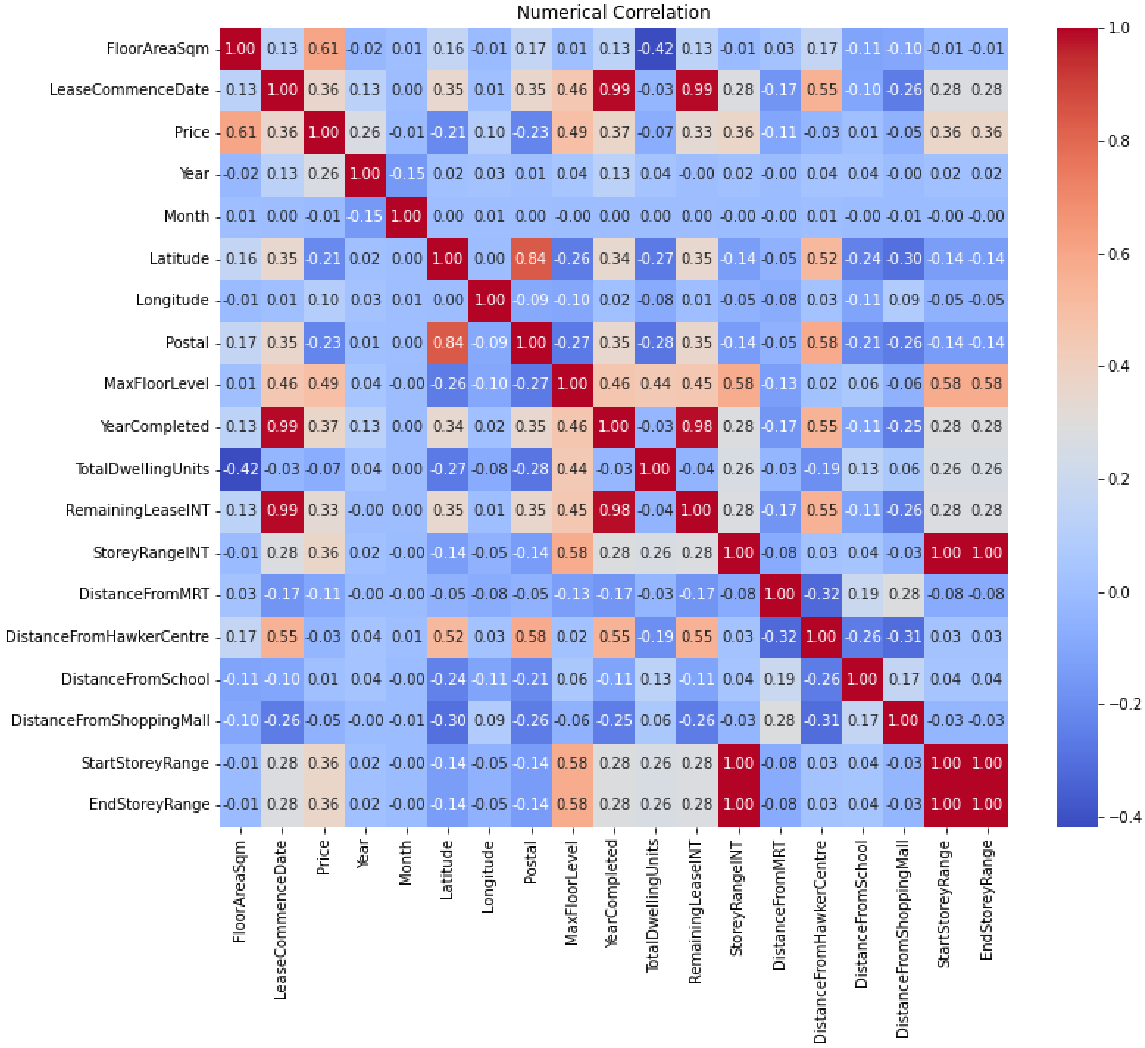


Figure 2: Correlation Matrix of Numerical Variables

The correlation matrix offers valuable insights into the relationships between variables, particularly their influence on HDB property prices. By examining correlation coefficients, which range from -1 to 1, we can discern the strength and direction of these relationships. Coefficients closer to 1 indicate a strong positive correlation, signifying that as one variable increases, the other tends to increase as well. Conversely, coefficients close to -1 suggest a strong negative correlation, implying that as one variable increases, the other tends to decrease. These correlations provide crucial guidance for feature selection in predictive modeling tasks, enabling the inclusion of only the most relevant variables while disregarding those with little to no impact, as indicated by coefficients close to 0. Ultimately, this process ensures that predictive models are built on meaningful relationships, enhancing their accuracy and interpretability in predicting HDB property prices.

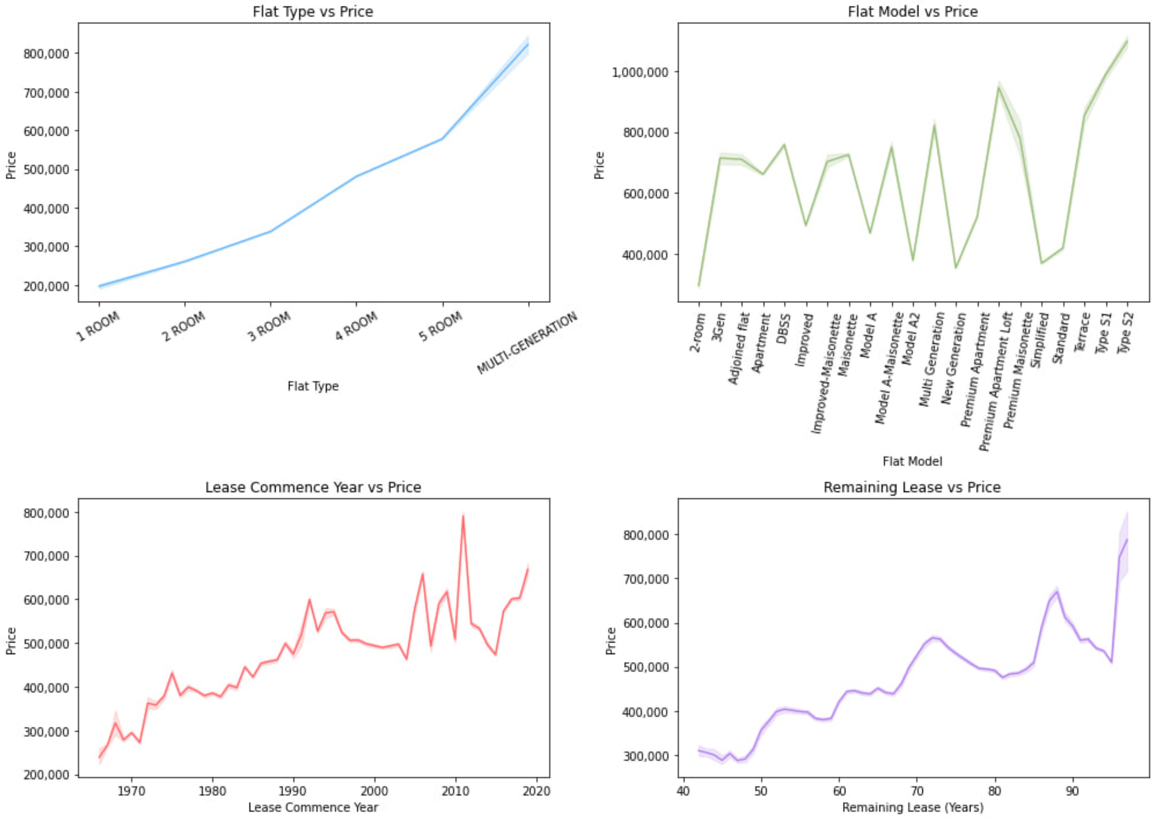


Figure 3: Subplots of Flat Information vs Price

The subplot analysis of Flat Information versus Price reveals insightful trends.In the subplot analysis of Flat Type, Flat Model, Lease Commencement Year, and Remaining Lease versus Price, distinct trends emerge regarding their impact on HDB property prices. Firstly, in the Flat Type versus Price graph, there's a noticeable steady increase in prices, with the lowest observed for 1-room flats at $200,000 and peaking at multi-generation flats at above $800,000. Secondly, examining Flat Model versus Price reveals fluctuations in prices across different models. Notably, prices are lowest for 2-room flat models at around $200,000 and highest for Type S2 flat models at above $1,000,000. Lastly, Lease Commencement Year and Remaining Lease versus Price demonstrate a staggered increase in prices as lease commencement year and remaining lease years increase. Price peaks around 2010 for lease commencement year at around $800,000 and when the remaining lease is around 100 years at $790,000. These observations offer valuable insights into how various flat attributes correlate with fluctuations in price, providing crucial information.

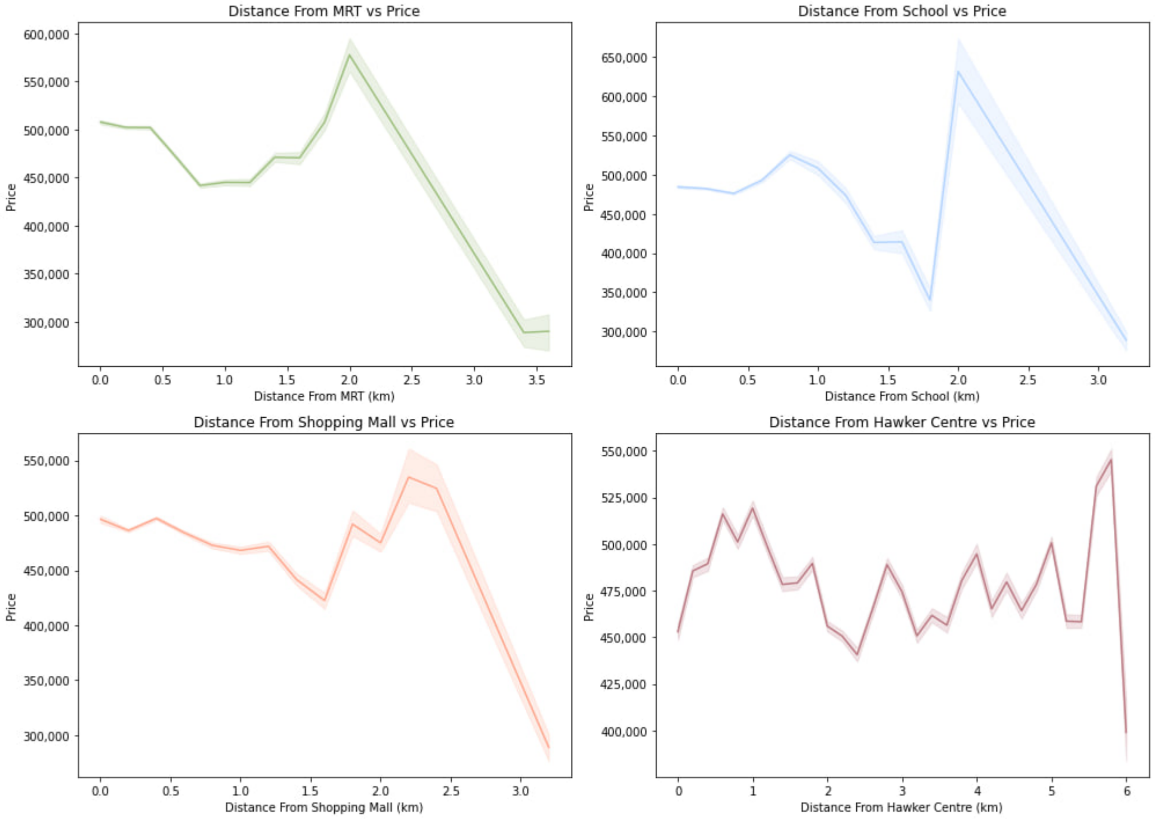


Figure 4: Distance from Amenities vs Price

The subplot analysis of Distance from Amenities vs Price reveals insightful trends. In the graph for Distance from MRT versus Price, it's evident that prices peak around $575,000 when the distance is 2.0 km, gradually decreasing thereafter and reaching a low below $300,000 at 3.5 km. Similarly, for Distance from School, prices decrease gradually from 0 km to around 1.8 km, peak at $625,000 at 2.0 km, and decrease again beyond 3.0 km, dropping below $300,000. In contrast, Distance from Shopping Mall shows a gradual decrease in prices from 0 km to 1.7 km, followed by an increase to $540,000 at 2.3 km, then another decrease beyond 3 km dropping below $300,000. For Hawker Centre distance, prices fluctuate across all distances, peaking around 5.8 km at $545,000 before sharply dropping to $400,000 at 6 km. These observations highlight the significant impact of proximity to amenities on HDB property prices, underscoring the importance of location considerations.

6 Modelling and Evaluation

The modeling stage is a pivotal phase where the actual machine learning models are constructed based on the data collected and preprocesed in earlier stages. This stage involves selecting appropriate algorithms, designing the model architecture and tuning hyperparameters to optimize model performance. In our project, we followed a systematic approach during the modeling stage to ensure the development of robust and effective AI models. We employed various regression and time-series forecasting techniques to address the project objectives.

We utilised a diverse set of models, including Decision Tree Regression, Random Forest Regression, Gradient Boosting Algorithm, FeedForward Neural Network and ARIMA (AutoRegressive Integrated Moving Average). Each model offers distinct capabilities.

6.1 Regression Models

Models	Library
Decision Tree Regression Random Forest Regressor Gradient Boosting Algorithm	scikit-learn

Table 3: Models used and corresponding libraries

6.1.1 Decision Tree Regression

Decision Tree Regression works by recursively partitioning the data into subsets based on the values of input features, creating a tree-like structure to make predictions on continuous target variables. Default hyperparameters were used, but none were initialized. After running the model, we found the following metrics:

- **Mean Squared Error (MSE)**: 1484379363.8
- **Root Mean Squared Error (RMSE)**: 38527.6

6.1.2 Gradient Boosting Algorithm

Gradient boosting is a type of machine learning boosting. It relies on the intuition that the best possible next model, when combined with previous models, minimizes the overall prediction error. The key idea is to set the target outcomes for this next model in order to minimize the error. For gradient boosting algorithm, 'max_iter' is a hyperparameter that specifies the maximum number of iterations or boosting stages. Each iteration adds a weak learner (decision tree) to the ensemble in a way that corrects the errors made by the existing ensemble. By limiting the number of iterations, we can control the complexity of the model and prevent overfitting. In our model, this parameter was set to 100 (max_iter = 100), indicating that the boosting process will stop after 100 iterations. After running the model, we found the following metrics:

- **Mean Squared Error (MSE)**: 1435226818.2
- **Root Mean Squared Error (RMSE)**: 37884.4

6.1.3 Random Forest Regressor

A non-linear ensemble learning method, comprised of a collection of decision trees, where each tree is built independently and makes its prediction.The final prediction is often an average/weighted average of the predictions of all the trees in the forest. For random forest regressor, a hyperparameter called ‘random_state’ is used to set the seed for the random number generator. By setting a specific value, we can ensure that the random splitting of data during the construction of trees is reproducible. This is essential for obtaining consistent results across different runs of the model therefore it can be any arbitrary integer value. For our model, we have set ‘random_state’ = 42, and left all other hyperparameters to their default values.After running the model, we found the following metrics:

- **Mean Squared Error (MSE)**: 841155398.3
- **Root Mean Squared Error (RMSE)**: 29002.7

6.2 Neural Networks Model

Model	Library
FeedForward Neural Networks	PyTorch

Table 4: Models used and corresponding libraries

6.2.1 Feedforward Neural Networks

A Feedforward neural network (FNN) is a fundamental type of artificial neural network where connections between the units do not form a cycle. It consists of an input layer, one or more hidden layers, and an output layer. In the process of defining the FNN class, we specify the architecture of the network including the number of hidden layers, the size of each hidden layer, and the dropout rate to prevent overfitting. Hyperparameters like the learning rate, weight decay, and number of epochs are set for training the model. During the training loop, the model iterates through the dataset in batches, computing predictions, calculating the loss (such as Mean Squared Error), and adjusting the model's weights using backpropagation. After training, the model is evaluated on a separate test dataset to assess its performance using metrics like MSE (Mean Squared Error), which measures the average squared difference between predicted and actual values. The lower the MSE, the better the model's performance.

Hyperparameters

- **Input Size (input_size):** Dimensions of input feature vector. Set to 68, including multiple encoded categorical features.
- **Number of Hidden Layers (num_hidden):** Set to 1, which can approximate any continuous function mapping from one finite-dimensional space to another with arbitrary accuracy.
- **Size of Each Hidden Layer (hidden_dim):** Set to 34, approximately the average of the input and output sizes.
- **Dropout Rate (dropout):** Dropout rate is set to 0.2, which means during training, 20% of the units in the hidden layer will be randomly set to zero to prevent overfitting.
- **Optimizer (ffnn_optimizer):** Adam optimizer is chosen for its adaptive learning rate and ability to update weights during training.
- **Weight Decay (weight_decay):** Set to 0.01, indicating the extent of L2 regularization, which penalizes large weights to prevent overfitting and improve generalization.
- **Loss Function (ffnn_loss_func):** Mean Squared Error (MSE), commonly used for regression tasks.
- **Number of Epochs (num_epochs):** Set to 100, allowing the model to learn from the data over multiple iterations to improve performance.
- **Batch Size (batch_size):** Set to 10, improving model efficiency, stability, and providing a form of regularization.

After running the model, we found the following metrics:

- **Mean Squared Error (MSE)**: 22342254592.00
- **Root Mean Squared Error (RMSE)**: 149473.3

6.3 Time-Series Model

Model	Library
Arima	Statsmodels

Table 5: Models used and corresponding libraries

6.3.1 Autoregressive integrated moving average

ARIMA, or AutoRegressive Integrated Moving Average, is a set of models that explains a time series using its own previous values given by the lags (AutoRegressive) and lagged errors (Moving Average) while considering stationary corrected by differencing (oppossite of Integration.)

Hyperparameters

- **p (Lag Order)**: Represents the number of lag observations included in the autoregressive (AR) model. It accounts for autocorrelation, with a value of 5 indicating significant influence from the previous 5 observations.
- **d (Degree of Differencing)**: Represents the number of times raw observations are differenced to achieve stationarity. In this case, it's set to 0 as the data is already stationary.
- **q (Order of the Moving Average)**: Specifies the size of the moving average window in the moving average (MA) model. It's set to 2, indicating influence from the errors of the previous 2 observations.

After running the model, we found the following metrics:

- **Mean Squared Error (MSE)**: 3364000000
- **Root Mean Squared Error (RMSE)**: 58005.816

6.4 Results discussion

We have decided to compare across the models based on the robustness of the model to our dataset as well as then Root Mean Squared Error metric.

Table 6: Comparison of Models

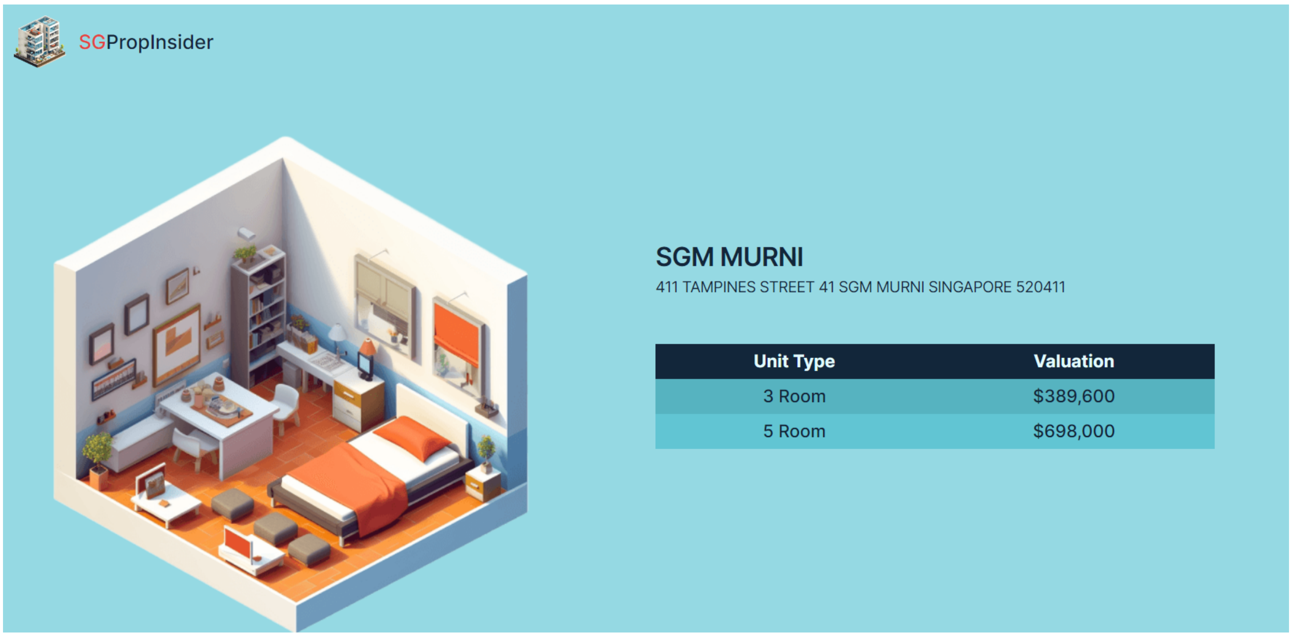
Model Name	RMSE
Decision Tree Regression	38527.6
Gradient Boosting ALgorithm	37884.4
Random Forest Regressor	29002.7
Feedforward Neural Network	149473.3
Autoregressive Integrated Moving Average	58005.8

The results indicate that the Random Forest Regressor (RFR) achieved the lowest RMSE, followed by the Feedforward Neural Network (FNN) , Decision Tree Regression, Gradient Boosting Algorithm, and Autoregressive Integrated Moving Average (ARIMA).

Our group decided that Random Forest Regressor is the best as expected, The Random Forest Regressor is favored among regression models due to its ability to combine predictions from multiple decision trees, resulting in more accurate predictions while minimizing overfitting. This ensemble approach is good for tasks with intricate relationships, such as predicting housing prices where we have multiple different input features and a very large dataset. Its robustness against noisy data made it suitable for our large dataset.

6.5 Comparison with State-Of-The-Art

The comparison between our model and existing works, including the HDB Price Predictor by Housing Development Board and an Instant Valuation by SG Prop Insider. To compare between the three, we chose 15 datapoints from our test set, and input these values into the GUI of the three predictors. In the Images below, As an example, we retrieve the valuation for a 3-Room flat located in Block 411, Tampines Street 41.

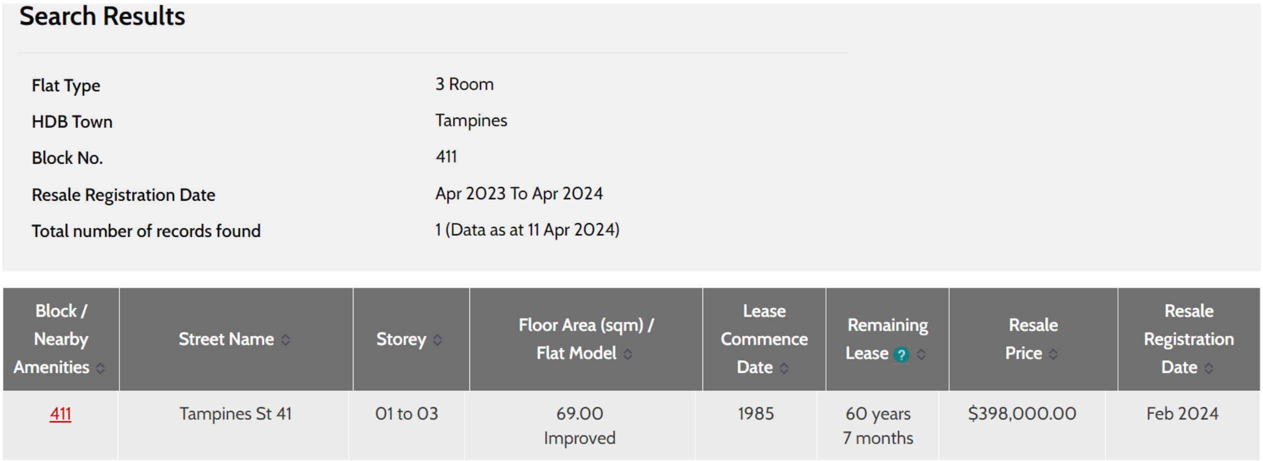


SGM MURNI

411 TAMPINES STREET 41 SGM MURNI SINGAPORE 520411

Unit Type	Valuation
3 Room	$389,600
5 Room	$698,000

(a) SGPropInsider



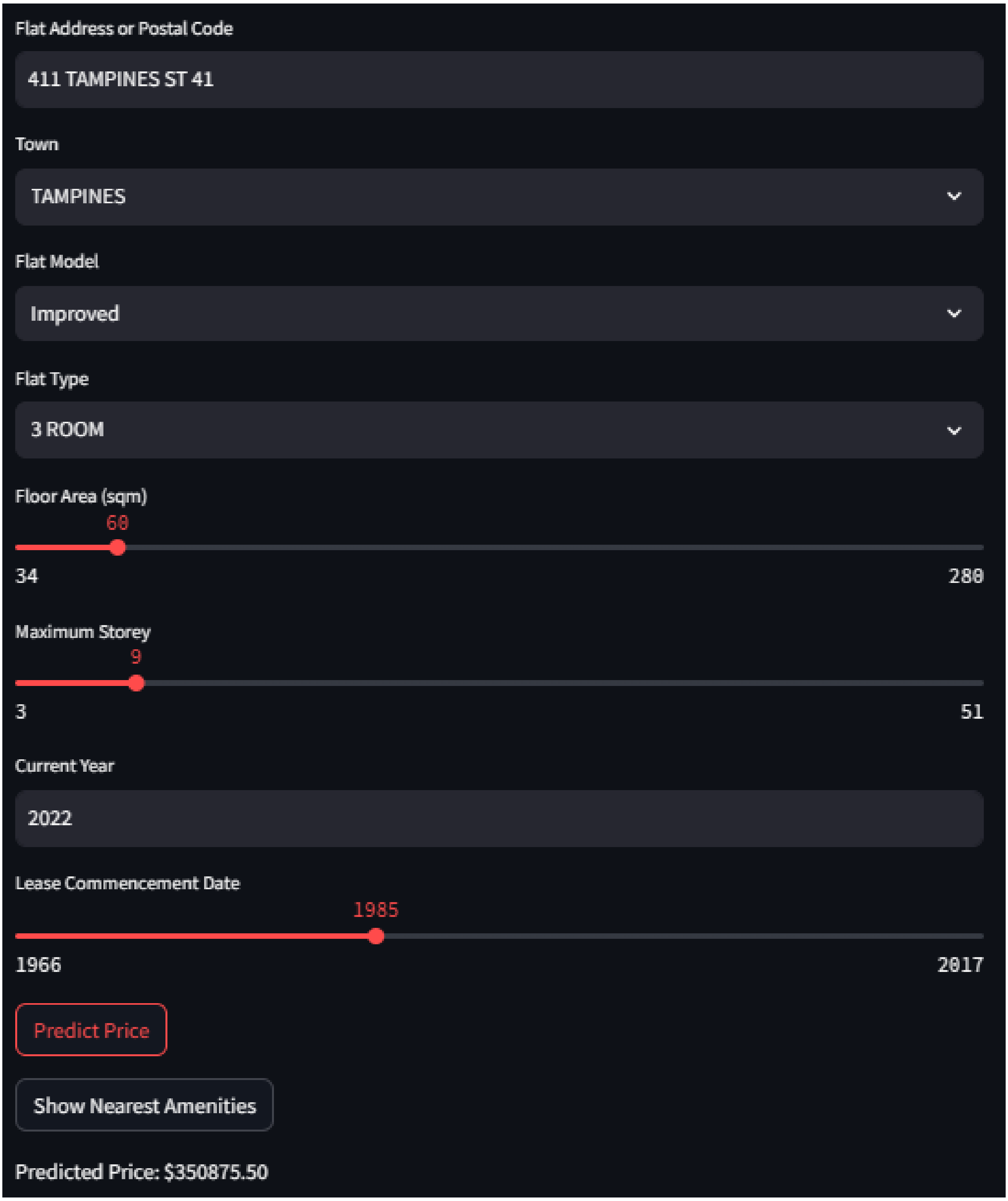
Search Results

Flat Type	3 Room
HDB Town	Tampines
Block No.	411
Resale Registration Date	Apr 2023 To Apr 2024
Total number of records found	1 (Data as at 11 Apr 2024)

Block / Nearby Amenities	Street Name	Storey	Floor Area (sqm) / Flat Model	Lease Commence Date	Remaining Lease	Resale Price	Resale Registration Date
411	Tampines St 41	01 to 03	69.00 Improved	1985	60 years 7 months	$398,000.00	Feb 2024

(b) HDB Price Predictor

Figure 5: Comparison of existing models: SGPropInsider and HDB Price Predictor



Flat Address or Postal Code

411 TAMPINES ST 41

Town

TAMPINES

Flat Model

Improved

Flat Type

3 ROOM

Floor Area (sqm)

60

34 280

Maximum Storey

9

3 51

Current Year

2022

Lease Commencement Date

1985

1966 2017

Predicted Price: $350875.50

Figure 6: Comparison with our model: Random Forest

Shown below is a table comprising of actual prices [Target] along with the predictions by our model and the existing works based on the input parameters. The metric to compare across the 3 works is Root Mean Squared Error.

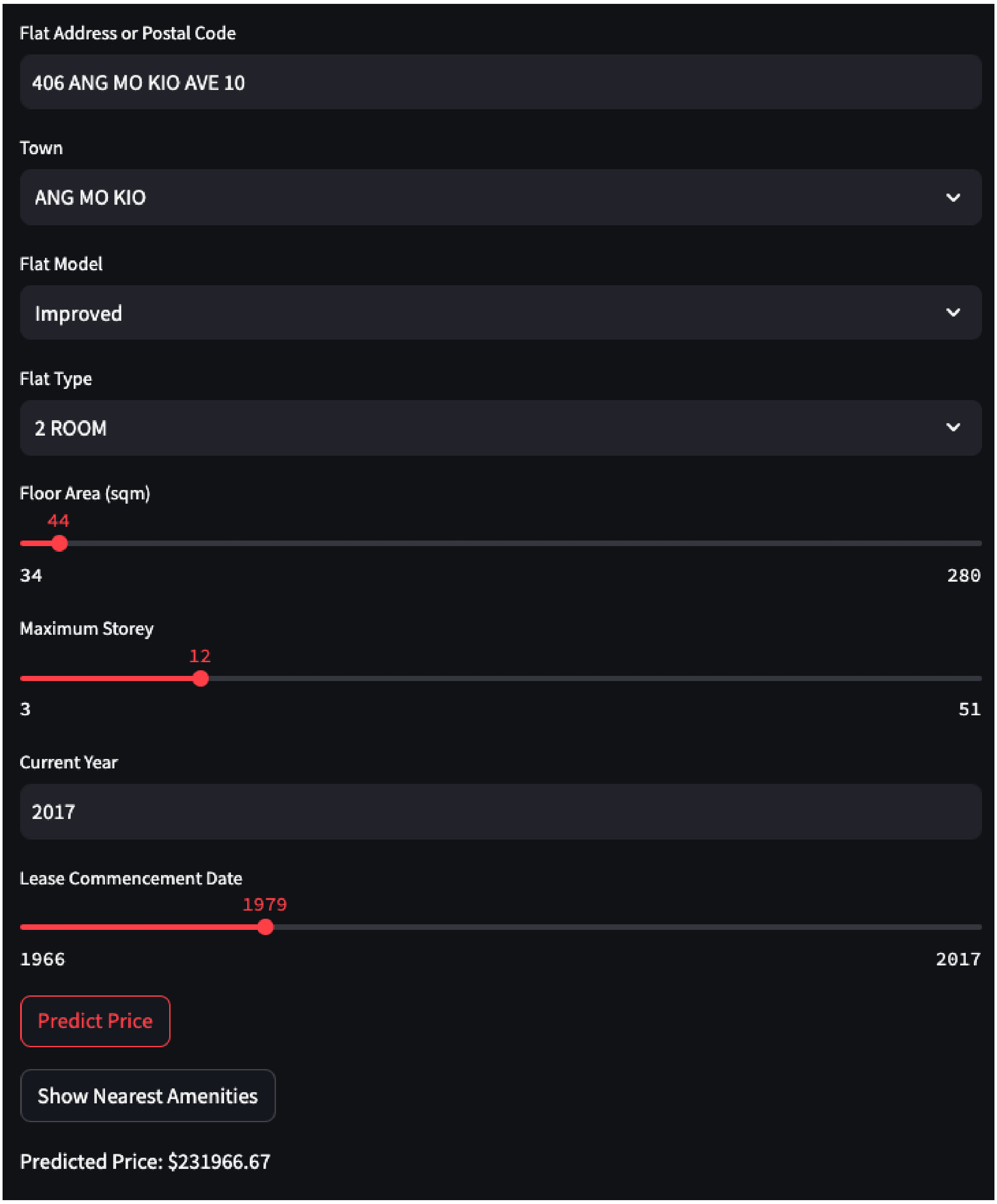
Target	Random Forest	Error	SG Prop Insider	Error	HDB Price Predictor	Error	Error
$370,000.00	$350,875.50	$19,124.50	$389,600.00	-$19,600.00	$398,000.00	-$28,000.00	-$47,124.50
$550,000.00	$563,689.00	-$13,689.00	$731,000.00	-$181,000.00	$708,000.00	-$158,000.00	-$45,689.00
$488,000.00	$473,070.00	$14,930.00	$490,200.00	-$2,200.00	$482,000.00	$6,000.00	$15,070.00
$550,000.00	$556,157.56	-$6,157.56	$515,000.00	$35,000.00	$575,000.00	-$25,000.00	$18,157.56
$453,000.00	$421,678.88	$31,321.12	$575,000.00	-$122,000.00	$575,000.00	-$122,000.00	$31,321.12
$408,000.00	$422,692.16	-$14,692.16	$553,750.00	-$145,750.00	$555,000.00	-$147,000.00	$14,307.84
$438,000.00	$439,562.76	-$1,562.76	$481,148.00	-$43,148.00	$450,000.00	-$12,000.00	$11,562.76
$336,500.00	$389,328.33	-$52,828.33	$390,107.00	-$53,607.00	$465,000.00	-$128,500.00	-$52,828.33
$467,000.00	$489,670.52	-$22,670.52	$593,667.00	-$126,667.00	$590,000.00	-$123,000.00	-$22,670.52
$480,000.00	$479,454.88	$545.12	$495,000.00	-$15,000.00	Na	$0.00	-$545.12
$640,000.00	$615,257.76	$24,742.24	$739,000.00	-$99,000.00	$610,000.00	$30,000.00	$24,742.24
$548,000.00	$509,905.00	$38,095.00	$672,500.00	-$124,500.00	Na	$0.00	$38,095.00
$500,000.00	$508,692.34	-$8,692.34	$758,888.00	-$258,888.00	$758,888.00	-$258,888.00	-$8,692.34
$620,000.00	$605,089.97	$14,910.03	$650,967.00	-$30,967.00	$710,000.00	-$90,000.00	$14,910.03
$525,000.00	$531,371.59	-$6,371.59	$640,000.00	-$115,000.00	$640,000.00	-$115,000.00	-$6,371.59
RMSE		$22,686.59		114895.68		110,735.49	

Table 7

The HDB Price Predictor showed an RMSE value of 110,735.49, while the SG Prop Insider had an RMSE of 114,895.68. In contrast, our model achieved a significantly lower RMSE of 22,686.59, indicating its superior ability to provide more accurate predictions.

7 Graphical User Interface (GUI)

We utilized the Streamlit library for GUI design, using its interface to create an interactive web application effortlessly. Streamlit's features, such as sliders, dropdown menus, and seamless map integration, enhanced the user experience and facilitated smooth navigation, making it an ideal choice for our project.



Flat Address or Postal Code

406 ANG MO KIO AVE 10

Town

ANG MO KIO

Flat Model

Improved

Flat Type

2 ROOM

Floor Area (sqm)

44

34 280

Maximum Storey

12

3 51

Current Year

2017

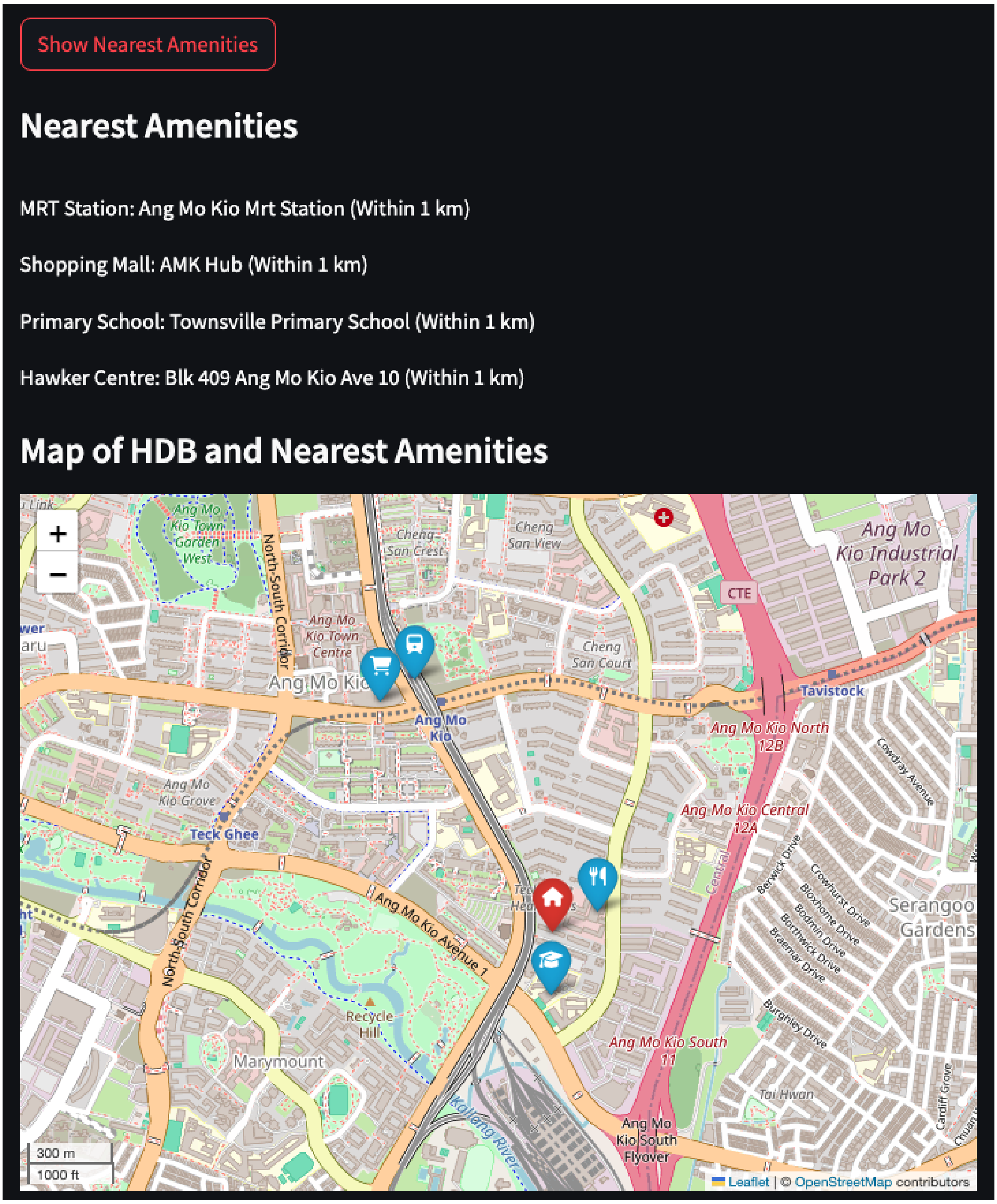
Lease Commencement Date

1979

1966 2017

Predicted Price: $231966.67

Figure 7: GUI User Input Fields and Displaying Price Prediction



Nearest Amenities

MRT Station: Ang Mo Kio Mrt Station (Within 1 km)

Shopping Mall: AMK Hub (Within 1 km)

Primary School: Townsville Primary School (Within 1 km)

Hawker Centre: Blk 409 Ang Mo Kio Ave 10 (Within 1 km)

Map of HDB and Nearest Amenities

Figure 8: GUI Displaying Nearest Amenities and Map

GUI Features

1. User Inputs:
Within the Streamlit framework, we designed the user input interface to be intuitive and accessible. Users are provided with options to input the flat address or postal code and the current year. Dropdown menus are available for selecting the town, flat model, and flat type, catering to users' varying familiarity with these options. Additionally, sliders enable users to specify the floor area, maximum storey, and lease commencement date, ensuring flexibility and precision in inputting parameters.

2. Price Prediction :
To facilitate price prediction, we integrated a "Predict Price" button into the GUI. This button triggers the prediction functionality, utilizing the Random Forest Regressor model to estimate the price based on the selected options and user inputs. The Random Forest Regressor model was selected for its robust performance and accuracy in predicting HDB flat prices, ensuring reliable and insightful results for users.

3. Nearest Amenities:
Another key feature of the GUI is the "Show Nearest Amenities" button, which enables users to view the nearest amenities associated with the selected HDB flat. Upon clicking this button, a list is generated, providing details of the closest MRT station, shopping mall, primary school, and hawker center, along with their respective distances from the specified flat location such as Within 1 km or More than 1 km.

4. Map Integration:
The map interface is integrated to provide users with a visual representation of the neighborhood surrounding the selected HDB flat. The map displays the location of the resale flat with a red pin, featuring a house icon. Additionally, pins with icons representing various amenities, such as MRT stations, shopping malls, hawker centers, and schools, are displayed on the map. Users can interact with the map by zooming in, zooming out, and exploring different areas to gain a comprehensive understanding of the neighborhood layout and proximity to amenities. This map functionality enhances the overall user experience, allowing for easy visualization and exploration of the surrounding area.

References

1. Guide to the Gradient Boosting Algorithm. DataCamp.
   https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm

2. How can you use Feedforward Neural Networks (FNNs) for Predictive Analysis? LinkedIn.
   https://www.linkedin.com/advice/0/how-can-you-use-feedforward-neural-networks-predictive-jwf0e#:~:text=Feedforward%20Neural%20Networks%20(FNNs)%20are,data%20to%20capture%20complex%20relationships

3. Random Forest Regression. Towards Data Science.
   https://towardsdatascience.com/random-forest-regression-5f605132d19d

4. Python Decision Tree Regression using sklearn. GeeksforGeeks.
   https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/

5. How to Build ARIMA Model in Python. ProjectPro.
   https://www.projectpro.io/article/how-to-build-arima-model-in-python/544

6. Shopping Mall Coordinates. Kaggle.
   https://www.kaggle.com/datasets/karthikgangula/shopping-mall-coordinates

7. Primary School Finder. Ministry of Education Singapore.
   https://www.moe.gov.sg/schoolfinder?journey=Primary%20school

8. List of Government Markets & Hawker Centres. GitHub.
   https://github.com/teyang-lau/HDB_Resale_Prices/blob/main/Data/list-of-government-markets-hawker-c
   csv

9. MRT Stations and their corresponding coordinates. GitHub.
   https://github.com/hxchua/datadoubleconfirm/blob/master/datasets/mrtsg.csv

10. Singapore Public Housing Resale Flat Prices. Kaggle.
    https://www.kaggle.com/datasets/wildboarking/singapore-public-housing-resale-flat-prices/data

11. Data.gov.sg - Singapore Government Data.
    https://beta.data.gov.sg/datasets/d_17f5382f26140b1fdae0ba2ef6239d2f/view