DBA4813: AI Strategies in Business

AY 2024/2025 Semester 1

Name	Matriculation Number
Low Jow Loon Jovian	A0218112W
Sugimoto Shoujin	A0265946M
Chang Xinzhou Leslie	A0233010H
Cheng Zhibin, Nicholas	A0217486W
Mai Youlian	A0222998M

Content Page

1. Introduction

1.1 Problem Statement

The housing market in Singapore has become a growing concern for many potential homebuyers due to rapidly escalating property prices. The challenge of selecting an affordable yet desirable home has only intensified, particularly against the backdrop of constrained supply and high demand. Limited land, growing population and influx of foreign investments have poised Singapore's property prices to be the most expensive in the Asia Pacific (*Reuters, 2023*). This presents a significant problem for first-time homebuyers, investors, and even homeowners looking to upgrade their properties. To plug this gap, it necessitates a solution which helps prospective homebuyers to navigate the complex market by assisting them in making informed decisions – Both in the present and future.

1.2 Opportunity – Market Sizing

To determine the opportunity of this market, the **Total Addressable Market (TAM)**, **Serviceable Available Market (SAM)**, and **Serviceable Obtainable Market (SOM)** framework is applied.

With a population of over 5.9 million people, Singapore is a densely populated urban city-state. Approximately 80% of the population lives in public housing, and the demand for both public and private housing continues to rise. By targeting homebuyers in Singapore, especially the growing middle-class segment, we can tap into a substantial market.

In summary we segmented the market as follows (*refer to Appendix 1 for full breakdown of calculation*):

- **TAM (Total Market)**: ~4 million potential homebuyers in Singapore.
- **SAM (Active Buyers per Year)**: ~50,000 active homebuyers or transactions per year.
- **SOM (Initial Target Market)**: ~1,500 users (capturing 3% of SAM in early stages).

1.3 Positioned for Success – Competitor Analysis

Several platforms already exist in Singapore to help consumers find homes, such as PropertyGuru, 99.co, and SRX. SimilarWeb presents a comprehensive list of potential competitors in the realtor industry (SimilarWeb, 2024). These platforms provide a range of services, from listing properties to offering basic pricing tools. However, they often lack advanced data-driven insights that predict property trends or offer personalized recommendations.

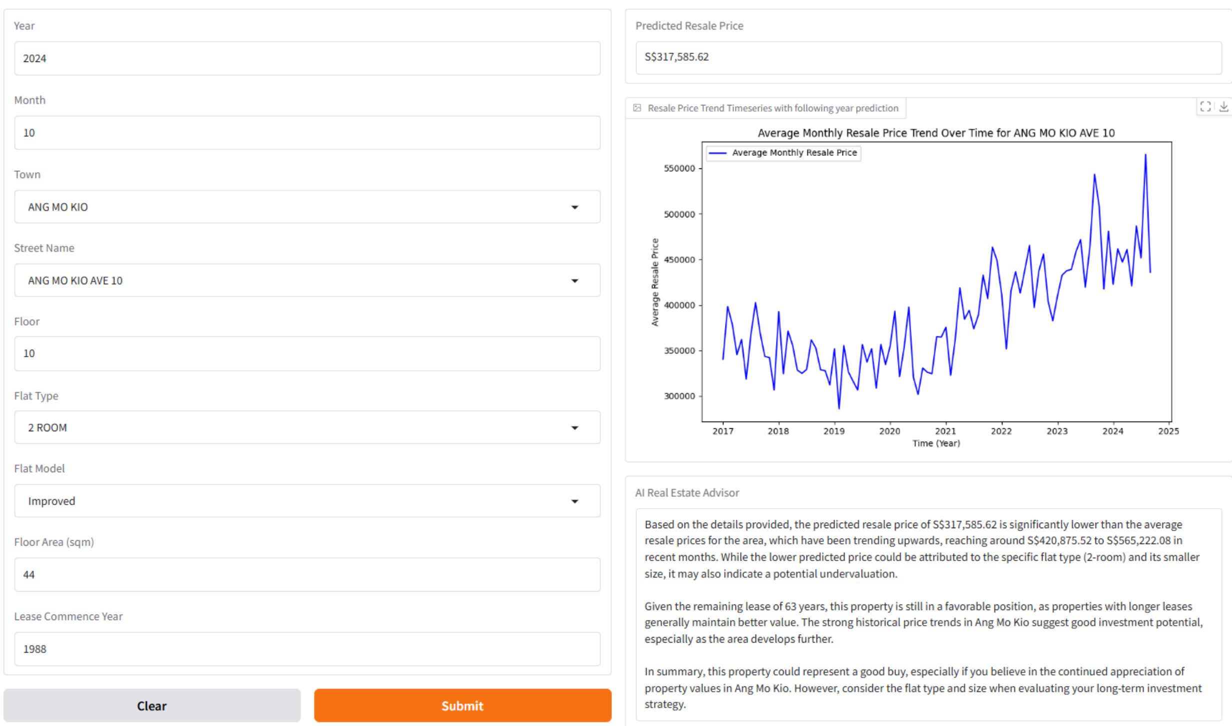
Our solution, **NeuralNest,** differentiates itself by (i) leveraging machine learning to deliver a more tailored, predictive experience, (2) trend analysis for future property prices – instead of just a current snapshot of the prices today. This grants our users a bespoke experience where they can make informed decisions on their big-ticket purchase for future capital appreciation.

2. NeuralNest - The One-stop Solution for Homebuyers

NeuralNest is a cutting-edge platform for homebuyers to find their dream homes in Singapore. Understanding the complexity and challenges of the local real estate market, the team created a data-driven tool to simplify the search process – Empowering buyers to make informed decisions. The platform's predictive engine is built on a dataset retrieved from data.gov.sg, which has comprehensive information on housing Street Name, Floor Area (non-exhaustive features) and its related price over the past 7 years.

NEURALNEST
THE SMARTER WAY TO NEST

By training our model on this extensive data, the team utilised Random Forest and Neural Network algorithms to **(i) Predict current price of ideal home** and **(ii) Predict future resale price**. Taking it a step further, our **(iii) In-house "AI Real Estate Advisor"** succinctly breaks down the analysis into byte sizes to make it comprehensible for all users.



2.1 Predict Current Price of Ideal Home

One of the main challenges in the home-buying process is sifting through a vast number of listings to find properties that align with a buyer's unique preferences. NeuralNest addresses this challenge through its personalized selection feature, which enables buyers to select features of their ideal home from parameters such as location, floor area, and other specific needs. This would then predict a resale price based on the user's selection, creating a highly tailored experience, making it easier for buyers to find flats that suit their lifestyle and financial requirements.

2.2 Predict Future Resale Price

The team also understands that some users purchase resale flats as an investment, hence NeuralNest offers a predictive pricing feature to estimate the potential for capital appreciation. This feature leverages on the predictive capabilities of our Neural Network model to analyse the correlation of historical price trends with other features. By offering insights into the likely future value of a flat, this feature enables users to consider the long-term investment potential of their purchase.

2.3 AI Real Estate Advisor

A unique addition to NeuralNest's suite of features is the AI Real Estate Advisor – An interface designed to make complex analysis accessible and easy to understand for every homebuyer. While our predictive models provide valuable insights on housing trends, price forecasts, and personalized recommendations, this information can sometimes be difficult for non-experts to interpret. The AI Real Estate Advisor bridges this gap by leveraging on a Large Language Model (LLM) to translate the intricate analytical findings into plain language, making it accessible to all users.

3. Data

3.1 Data Source

The dataset was sourced from data.gov.sg, a Singapore governmental platform that provides open datasets for public use. This dataset describes all of the past transactions related to the resale of HDB flats in Singapore.

3.2 Dataset Description

For our project, only data from 2017 and newer was utilised. In total, the dataset contains 191,443 observations, and 11 columns. The town, flat_type, block, street_name, storey_range, flat_model, remaining_lease are categorical variables that describe various qualities of the flat. floor_area_sqm and lease_commence_date are numerical variables representing additional features of the flat. resale_price is a numerical variable that captures the actual resale price, and will be used as the main target for predictive analysis. Further details are in Appendix 2.

Column Name	Data Type	Description	Example Values	Total Unique Values
town	Categorical	The town where the flat is located, providing geographical context.	"SENGKANG", "PUNGGOL"	26
flat_type	Categorical	The type of flat, indicating the size or room configuration.	"4 ROOM", "5 ROOM"	7
storey_range	Categorical	The range of floors where the flat is located, indicating its approximate height within the building.	"04 TO 06", "49 TO 51"	17
flat_model	Categorical	The model type of the flat, representing different layouts and designs.	"Model A", "Improved"	21

block	Categorical	The block number of the flat, identifying specific buildings within a town.	"406", "315C"	Numerous
street_name	Categorical	The street name where the flat is located, adding geographical detail.	"ANG MO KIO AVE 10"	Numerous
floor_area_sqm	Numerical	The size of the flat in square meters, indicating the total floor area.	44.0, 105.0	-
lease_commence_date	Numerical	The year the flat's lease started, indicating the age of the lease.	1979, 2015	-
remaining_lease	Categorical (Time)	The time left on the lease, usually in years and months.	"61 years 04 months"	Numerous
resale_price	Numerical	The transaction price of the flat in Singapore dollars, representing the resale value.	232000, 738888	-

4. Methods

4.1 Suitability of Algorithms

Random Forests and Neural Networks were selected to be used for the prediction of the dataset as these models are more robust and able to capture more complex patterns of different towns and streets as compared to models such as a simple linear regression.

4.2 Random Forests & XGBoost

4.2.1 Data Preparation

The data is prepared using the following techniques, ensuring that the features are suitable to be fed into the random forest model.

Checking for Missing Values: The code checks for any missing values in the dataset using the isnull() function. Within this dataset, no empty or null values were found, hence no further action was required.

Converting Variables: The *remaining_lease* column, which contains the remaining lease in years and months, is converted to a numerical value representing the total number of years. This is done by extracting the month and year value from the *remaining_lease* column using a looped function, and combines both values into a single decimal value.

Dropping Unnecessary Columns: The *month* column and *block* column are dropped as it is not deemed useful for the model, since month data is already represented in *remaining_lease*, and *block* information as it has no significant correlation with predicting rental pricing.

4.2.2 Encoding Categorical Variables & Feature Scaling

Encoding Categorical Variables: Categorical variables, such as *town*, *flat_type*, *flat_model*, *storey_range*, and *street_name*, are encoded using an OrdinalEncoder. This ensures that the categorical variable can be converted into a numerical one, without the need to increase the number of features, as would be the case if one-hot encoding was used.

This encoder is saved to a file and used across both the neural network and random forest models to ensure consistency in encoding across both the neural network and random forest models, preventing variations that could occur if each model generated its own encoding logic.

Feature Scaling: The floor_area_sqm and remaining_lease features are scaled using a StandardScaler to ensure they are on a similar range, so that the model is able to interpret these features on the same normalized scale.

Similarly, the scaling logic is also saved and used across all models to ensure consistency in scaling.

4.2.3 Train-Test Split

The dataset was split into training and test sets with an 80:20 ratio, with 153,154 rows for training and 38,289 in the test set. This ensures substantial data points for training the model before evaluating its predictive ability on unseen test data.

4.2.4 Model Selection

Two different regression models are utilized for prediction:

Random Forest Regressor: A RandomForestRegressor is initialized with a grid of hyperparameters to be tuned using GridSearchCV.

XGBoost Regressor: An XGBRegressor is also initialized with a grid of hyperparameters to be tuned using GridSearchCV.

A GridSearchCV process is then implemented for both models in order to look for the best hyperparameters to achieve greatest accuracy and predictive ability of the model.

4.2.5 Ensemble Modeling

To further improve the model's performance, a VotingRegressor is created, which combines the predictions of the best XGBoost and Random Forest models, and then averages the individual predictions to form a final prediction. By averaging the predictions, the VotingRegressor reduces the likelihood of large prediction errors, as errors in one model can be compensated by better predictions from the other. This ensemble approach leverages the strengths of both models to improve the performance of the model.

4.2.6 Model Evaluation

The final ensemble model is evaluated using the following metrics:

Mean Absolute Error (MAE),Root Mean Squared Error (RMSE), R-squared (R^2) and Mean Absolute Percentage Error (MAPE).

These metrics provide a good indication of the model's performance on the test set.

4.3 Neural Networks

4.3.1 Data Preparation

Extracting the Number of Years from `remaining_lease`:
The `remaining_lease` column originally contained strings (e.g., "61 years 04 months"), which are not directly usable by machine learning models. To make this feature suitable for the neural network, only the number of years was extracted and converted into an integer format.

Splitting `month` into `year` and `month`:
The original `month` column contained both year and month data (e.g., "2017-01"). To improve temporal analysis, this column was split into two separate columns: `year` and `month`.

Extracting the Lowest Floor from `storey_range`:
The `storey_range` column provided a range of floor numbers (e.g., "04 TO 06"), which was not directly usable for analysis. By extracting the lowest floor from the range, we simplified this feature into a single numeric value. This transformation is important since the floor level can significantly influence property price, with higher floors often commanding higher prices.

Mapping `flat_type` to Numerical Values:
The `flat_type` feature is categorical, with different types of flats (e.g., "2 ROOM", "3 ROOM") representing a wide price range. To enable the neural network to interpret this feature, the flat types were mapped to numerical values. This ordinal mapping allows the model to differentiate between the price ranges typically associated with each flat type, thus enhancing the model's ability to predict prices.

Mapping `flat_model` to Numerical Values:
Like `flat_type`, the `flat_model` feature was originally categorical. Different flat models (e.g., "Improved", "Model A") typically follow a hierarchy in terms of price. By mapping these categories to a ranked numeric scale, the neural network can learn the relative value of different flat models and incorporate this knowledge into its predictions.

Converting town and street_name to Categorical Data Using Embeddings:
To handle categorical features like town and street_name, embeddings are used to represent these variables as dense vectors. Unlike ordinal encoding, which would assign arbitrary integer values to each category and imply an unintended order, embeddings allow the model to capture complex relationships between categories without assuming any order. This approach not only reduces the dimensionality

compared to one-hot encoding but also enables the neural network to learn meaningful patterns in the data, as similar categories can end up with similar embeddings. Using embeddings prepares the data efficiently for the model, where town and street_name are represented as continuous vectors, while other continuous features are scaled as needed. This setup improves the model's ability to generalize and find nuanced relationships in the data.

4.3.2 Feature Selection

Only the most relevant features were selected for modeling. The chosen features—`year`, `month`, `town`, `floor`, `flat_type`, `flat_model`, `floor_area_sqm`, `remaining_years`—are expected to have the greatest influence on the `resale_price`. By focusing on these core features, we reduce the complexity of the dataset while maintaining the predictive power for the neural network.

4.3.3 Neural Network Fine Tuning

To enhance the predictive accuracy of the neural network model for resale flat prices, we conducted hyperparameter tuning using Keras Tuner's Hyperband algorithm. This approach allows us to optimize the model architecture and key parameters, improving the model's ability to generalize to new data. The tuning process involved adjusting the number of units in each dense layer and the learning rate for the optimizer, aiming to minimize the validation loss.

Optimizer and Learning Rate:

We used **Keras Tuner** with the **Adam optimizer** to optimize our model for predicting resale flat prices, as they efficiently handle the dataset's complexity. Keras Tuner automates hyperparameter tuning, allowing us to find the best neuron count and learning rate, with **Hyperband** improving efficiency by focusing on promising configurations. The **Adam optimizer** adjusts learning rates dynamically, ensuring faster, stable convergence

Early Stopping:

We used EarlyStopping to monitor the validation loss. If the model's performance doesn't improve after a set number of training cycles (10 epochs), training will stop to avoid wasting time and to prevent overfitting. EarlyStopping also saves the best version of the model during training.

4.3.4 Tuning Results

The hyperparameter tuning reduced the validation loss from 3,705,689,344.0 to 1,529,529,088.0, significantly improving model accuracy. The optimal configuration identified—hidden layers with 160, 128, 64, and 16 units, and a learning rate of 0.0075—was subsequently used for model training, achieving a balanced setup for efficient and accurate predictions.

4.4 Quality of Codebase

The quality of our codebase in this project is high, characterized by a well-structured and organized approach to data processing, model training, and evaluation. The use of clear variable names and modular functions enhances readability and maintainability. The use of libraries such as Pandas, NumPy, and Scikit-learn demonstrates adherence to industry standards in data manipulation and machine learning practices. Effective commenting provides context and explanations for complex operations, aiding future developers in understanding the workflow. Additionally, implementing feature engineering, hyperparameter tuning through GridSearchCV, and rigorous model evaluation, including the calculation of various metrics like MAE and R^2, reflects a comprehensive methodology that ensures robust performance and results. Overall, our codebase is efficient, scalable, and demonstrates best practices in data science and machine learning development.

5. Results

5.1 Random Forests & XGBoost

5.1.1 Results and Accuracy of Model

The results of the ensemble model are as follows:

MAE	20112.747881213025
RMSE	28672.77260571221
R-squared	0.973103628837828
MAPE	4.06%

As seen from the results, the ensemble model produced highly promising results across a variety of metrics, demonstrating strong model performance and accuracy in estimating house prices. The Mean Absolute Error (MAE) of 20,112.75 indicates that the model's predictions deviate from actual housing prices by approximately $20,113, which is fairly accurate and consistent.

A high R-squared score of ~0.97 also reinforces the message of a good model fit, where only a small percentage of variance cannot be explained by the model's predictors. Overall, these metrics indicate that the ensemble model is performing well, with low levels of error and a high degree of accuracy in capturing the variance in housing prices.

5.1.2 Feature Importance

A feature importance map was generated for both the Random Forest and XGBoost models (Appendix 4) revealing both models' differing conclusions drawn to feature importance. For instance, Random Forests determined *floor_area_sqm* to be the most important feature in affecting the results of the prediction, while XGBoost concluded that *street_name* had the highest weightage. As such, this reveals the importance of VotingRegressor in helping to aggregate results from both models and prevent extreme results.

5.1.3 Predicted Results vs. Actual Results

The actual and predicted prices were further plotted on a scatter plot (Appendix 5), and reinforces the conclusion that the model was accurate in predicting the price of the house. The distribution of residuals curve also showed a majority of residuals to be limited within $100,000, indicating that a majority of the predicted prices were no more than $100,000 away from the actual price.

5.2 Neural Networks

5.2.1 Training and Validation Loss Over Epochs (Refer to Appendix 6)

The training loss decreases rapidly, stabilizing and plateauing after about 20 epochs, while the validation loss follows a similar trend with minimal fluctuations. The close alignment between training and validation loss suggests that the model is well-fitted without significant overfitting or underfitting.

5.2.2 Predicted Results vs. Actual Results (Refer to Appendix 6)

The points closely align along the red diagonal line, indicating high prediction accuracy. The strong linear relationship, reflected in an R-squared value of 0.973, demonstrates that the model effectively captures the underlying patterns in the data, explaining 97.3% of the variance in resale prices.

5.2.3 Residual Distribution (Refer to Appendix 6)

The residual distribution, showing the difference between predicted and actual prices, is centered around zero with a slight symmetrical spread, indicating that most residuals fall within a narrow range. Most predictions lie within $100,000 of the actual resale price, suggesting that errors are relatively small and evenly distributed. This confirms the model's accuracy and reliability in predicting resale flat prices.

In summary, the model demonstrates high predictive accuracy with low error, as evidenced by the low and stable validation loss, close alignment between predictions and actual values, and the centered residual distribution. These results confirm that the model is well-suited for estimating resale flat prices.

6. Implementation

6.1 Exporting Model, Scaler, and Encoder via Joblib

The first step involves exporting the trained model, the scaler, and the encoder mappings into .h5 and .pkl format. The joblib library is useful for saving large datasets, models, and transformations because of its efficient file handling. Here's how this can be done:

6.2 Importing Model, Scaler, Encoder, and Dataset for Time Series Plot

In this step, we load the necessary components (model, scaler, encoders) and the dataset for plotting the time series trend. These elements are crucial for re-processing the input data and predicting the resale price.

6.3 Process, Scale, and Encode Data in the Same manner as the Model Training

Data preprocessing involves encoding categorical features (e.g. town and street_name) and scaling numerical features to match the format used during model training. This ensures that the inputs to the neural network are consistent, enabling accurate predictions.

6.4 Predict Prices Using Neural Network and Generate Time Series Plot

The function NN_predict uses the preprocessed inputs to predict the resale price. It passes these inputs to the neural network, which outputs a price prediction. A time series plot is generated using historical resale prices to visualize trends and make future predictions via function plot_time_series.

6.5 Call OpenAI API to Generate Advice Using Predicted Price and Historical Data

The function get_advise takes the predicted price and other property details and sends them to OpenAI's API for advice. The response from the model serves as a real estate advisor, assessing the property's investment value based on current market trends.

6.6 Output the Results on a User Interface Using Gradio

Using Gradio, a simple interface is created to take user input and display predictions and advice. It links each part of the process into a cohesive app where users can enter property details, view predicted resale prices, see time series trends, and receive investment advice.

7. Conclusion – Limitations and Future Improvements

Limitation 1: Limited Historical Data
The model relies on only seven years of data, limiting its capacity to capture long-term market cycles essential for accurate forecasts. Attempts to use time-series models like SARIMA were hindered by the short-term data's lack of cyclical patterns.
***Improvement*:** Expanding historical data with indicators such as GDP and interest rates would allow more effective use of models like SARIMA or LSTM, enabling continual retraining for improved accuracy and robustness.

Limitation 2: Limited Data for Real Estate Advice
Current advice relies on limited property and historical data, lacking inputs like government policies, infrastructure plans, and buyer preferences, which impact investment decisions.
***Improvement*:** Adding data on policies, developments, and user preferences would enable more tailored advice. External sources like market news would make AI recommendations more dynamic and responsive.

Conclusion
NeuralNest provides a data-driven solution for Singapore's housing market by predicting property prices, forecasting trends, and offering personalized advice. Using a neural network model pre-trained on 20,000 data points and a large language model, the platform delivers tailored insights and simplifies complex analyses for homebuyers. While limited historical data and market factors present some challenges, future improvements such as expanded data sources and real-time policy integration will enhance NeuralNest's accuracy and relevance, empowering users to make informed property investment decision

Appendix

Appendix 1 - Market Sizing TAM SAM SOM Framework:

- **Total Addressable Market (TAM)**:
  TAM refers to the entire potential market if there were no competition or constraints. In the context of the housing market in Singapore, this could be represented by all potential homebuyers and investors, including both public and private housing sectors.
    - In Singapore, about **80% of the population** lives in public housing (HDB) and **20% in private housing**. Assuming an adult population of around 4 million (since most potential homebuyers are adults), this gives us approximately **3.2 million public housing buyers** and **800,000 private property buyers**.
    - This makes the **TAM** the total number of potential homebuyers, both new and repeat buyers.
    - For simplicity, let's assume that **all adult residents** are within the TAM, which could approximate to **4 million** potential users of a home selection app.
- **Serviceable Available Market (SAM)**:
  SAM is the portion of TAM that your product or service can serve, based on geographic, demographic, or other restrictions.
    - In this case, the SAM represents potential buyers who are actively engaged in house hunting or property investment in a given year.
    - For example, based on transaction data from previous years, there are approximately **25,000-30,000 public housing resale transactions** per year, and around **20,000 private property transactions**.
    - Thus, we can estimate the **SAM** to be around **50,000 potential transactions per year** in Singapore.
    - These numbers represent active buyers or investors who are more likely to use the app.
- **Serviceable Obtainable Market (SOM)**:
  SOM represents the realistic portion of SAM that you can capture within a certain time frame, usually the early stages of product launch.
    - For a new app, gaining even **1-5%** of the market in the first few years can be considered a reasonable goal.
    - Assuming we capture **3% of SAM** (i.e., 3% of 50,000 transactions annually), this translates to **1,500 active users** or transactions.
    - These could be users who heavily engage with the app for home recommendations and insights, providing a clear initial user base to scale from.

<u>Summary:</u>

- **TAM (Total Market)**: ~4 million potential homebuyers in Singapore.
- **SAM (Active Buyers per Year)**: ~50,000 active homebuyers or transactions per year.
- **SOM (Initial Target Market)**: ~1,500 users (capturing 3% of SAM in early stages).

Appendix 2 - Dataset Description

The dataset `new_resale_flat_prices.csv` includes categorical and numerical data columns that provide details about resale flats in Singapore.

3.2.1 Categorical Columns

1. "*Month":* This is a categorical variable that indicates the date of the resale transaction, recorded in YYYY-MM format.
2. "town": Represents the town where each flat is located. There are 26 unique towns, with Sengkang and Punggol being the most common.
3. "*flat_type*": Indicates the type of flat, such as "4 ROOM" or "5 ROOM." There are 7 unique types, with "4 ROOM" being the most prevalent.
4. "*storey_range*": Describes the floor range of the flat, such as "04 TO 06" or "10 TO 12." There are 17 unique ranges, with "04 TO 06" and "07 TO 09" having the highest counts. The highest range recorded in the data is "49 TO 51", which represents the tallest flats
5. "*flat_model*": Specifies the flat's model type, such as "Model A" or "Improved." There are 21 unique models, with "Model A" being the most frequent.
6. *"remaining_lease"*: Thisindicates the remaining lease period for each flat, typically expressed in years and months (e.g., "61 years 04 months"

3.2.2 Numerical Columns

1. "*floor_area_sqm*": The floor area of the flat in square meters, with values ranging from 31 to 366.7 sqm. The average floor area is around 97 sqm.
2. "*lease_commence_date*": The year when the flat's lease began, ranging from 1966 to 2020. The average commencement year is approximately 1996, indicating most flats are around 25 years old.
3. *"resale_price"*: The sale price of the flat in Singapore dollars, ranging from $140,000 to $1,588,000. The average resale price is about $504,250, with a standard deviation indicating a significant range in prices based on location, size, and other factors.

3.2.3 Location Columns

1. "*block*" contains unique identifiers for each building within a town. Although it may consist of numbers, it does not represent a quantitative value but rather serves as a label.

"*street_name*" provides the name of the street where the flat is located. Each unique street name represents a category within the dataset.

Appendix 3 - Results of XGBoost and Random Forests model

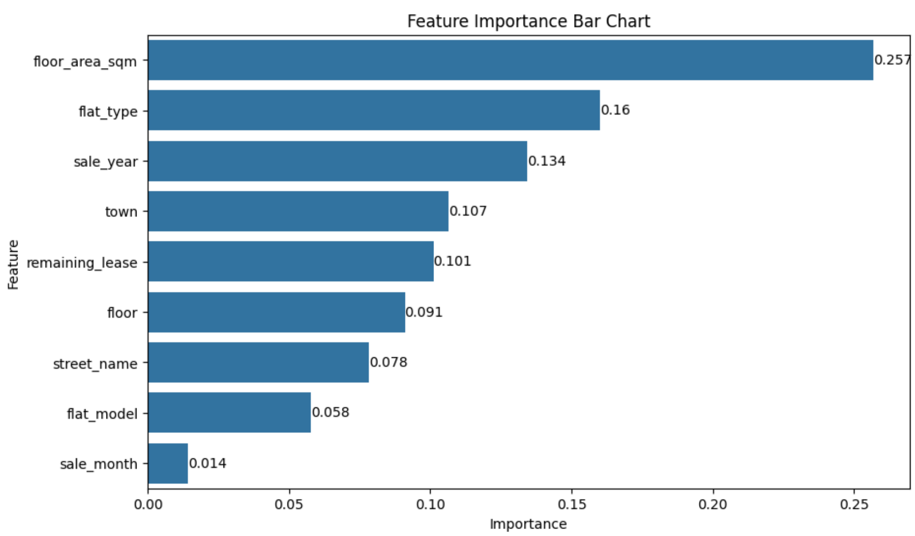
The XGBoost and Random Forests models were individually executed as well to compare the results, which are shown below:

	Random Forests	XGBoost
MAE	22997.77251540341	20775.2113787962
RMSE	34095.287042947806	28736.81380909158
R-squared	0.9619685389094035	0.9729833475109039
MAPE	4.58%	4.23%

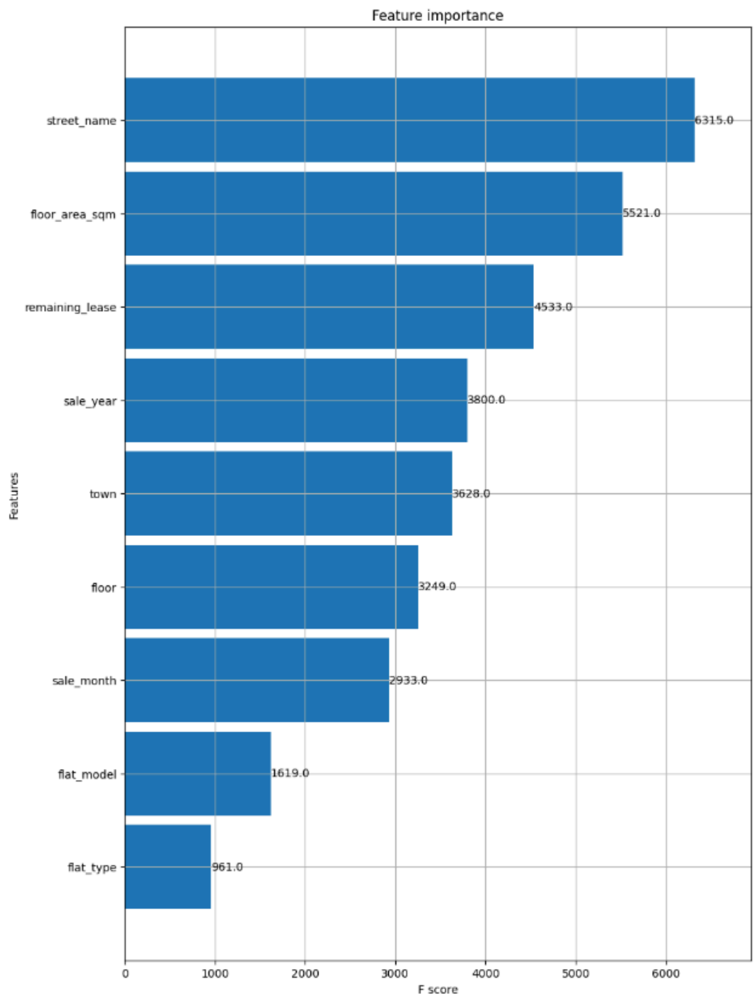
Appendix 4 - Feature Importance Output

The feature importance chart developed by both the Random Forest and XGBoost models are shown below:

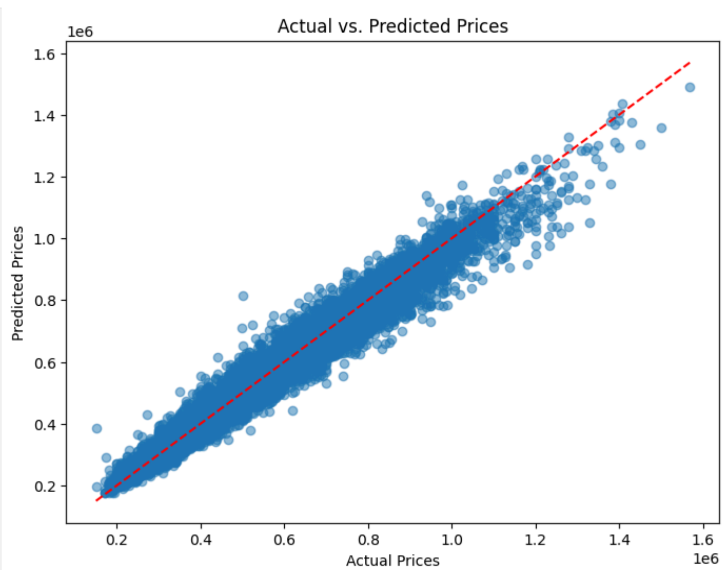
Random Forests:

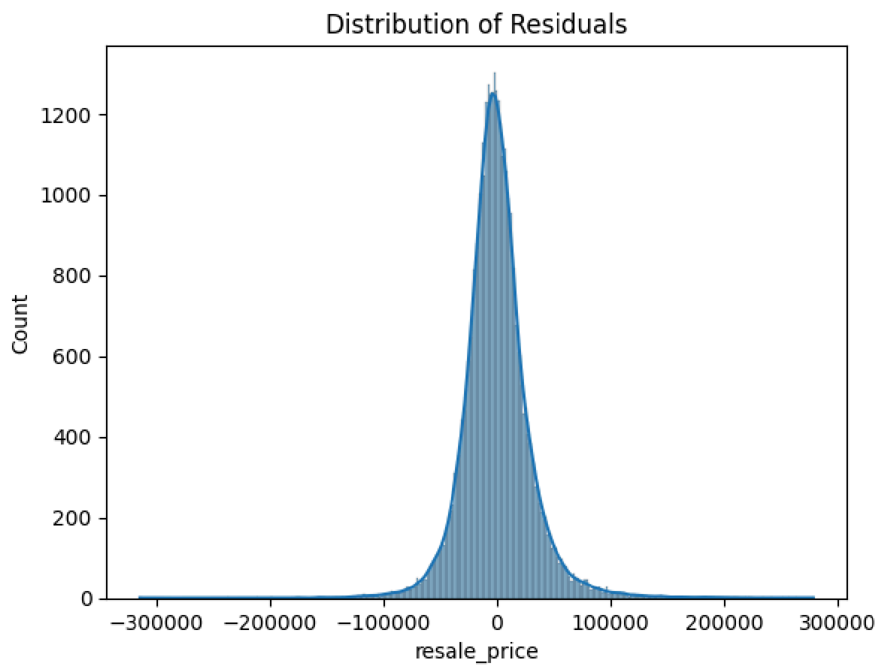


XGBoost:



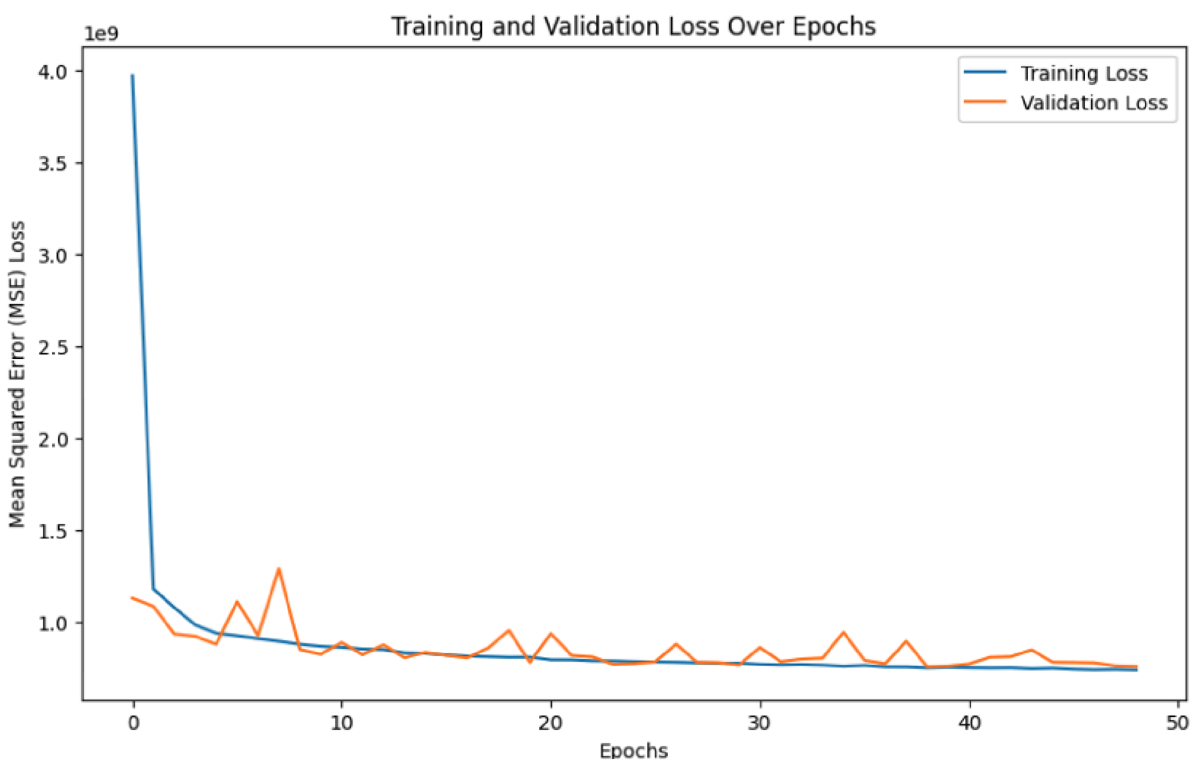
Appendix 5 - Results of the VotingRegressor model



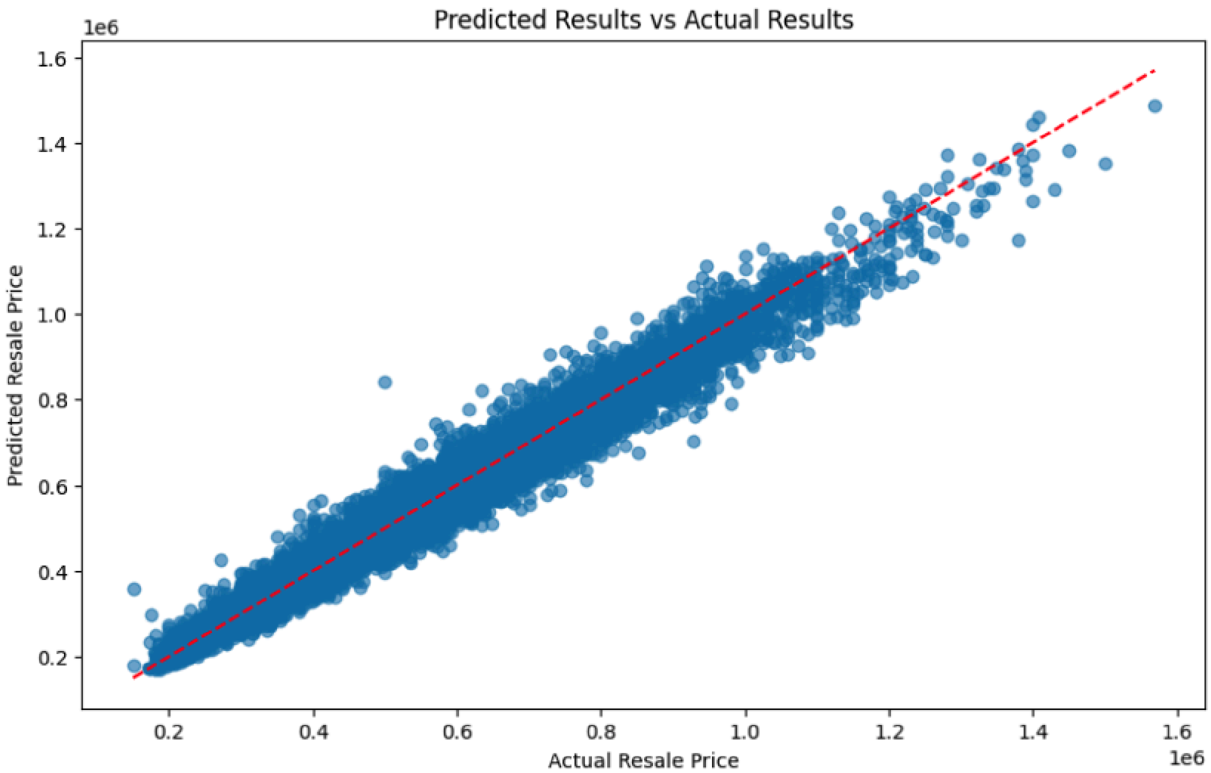


Appendix 6 - Results of the Neural Network model

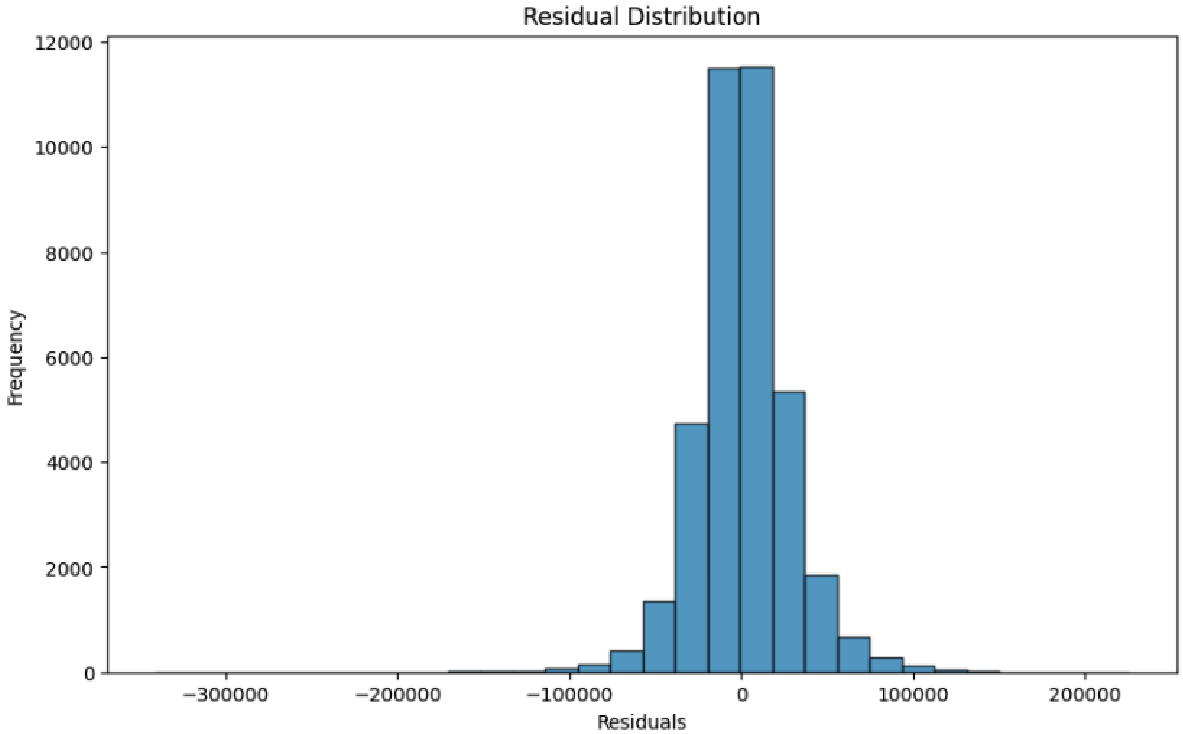
5.2.1 Training and Validation Loss Over Epochs



5.2.2 Predicted Results vs. Actual Results

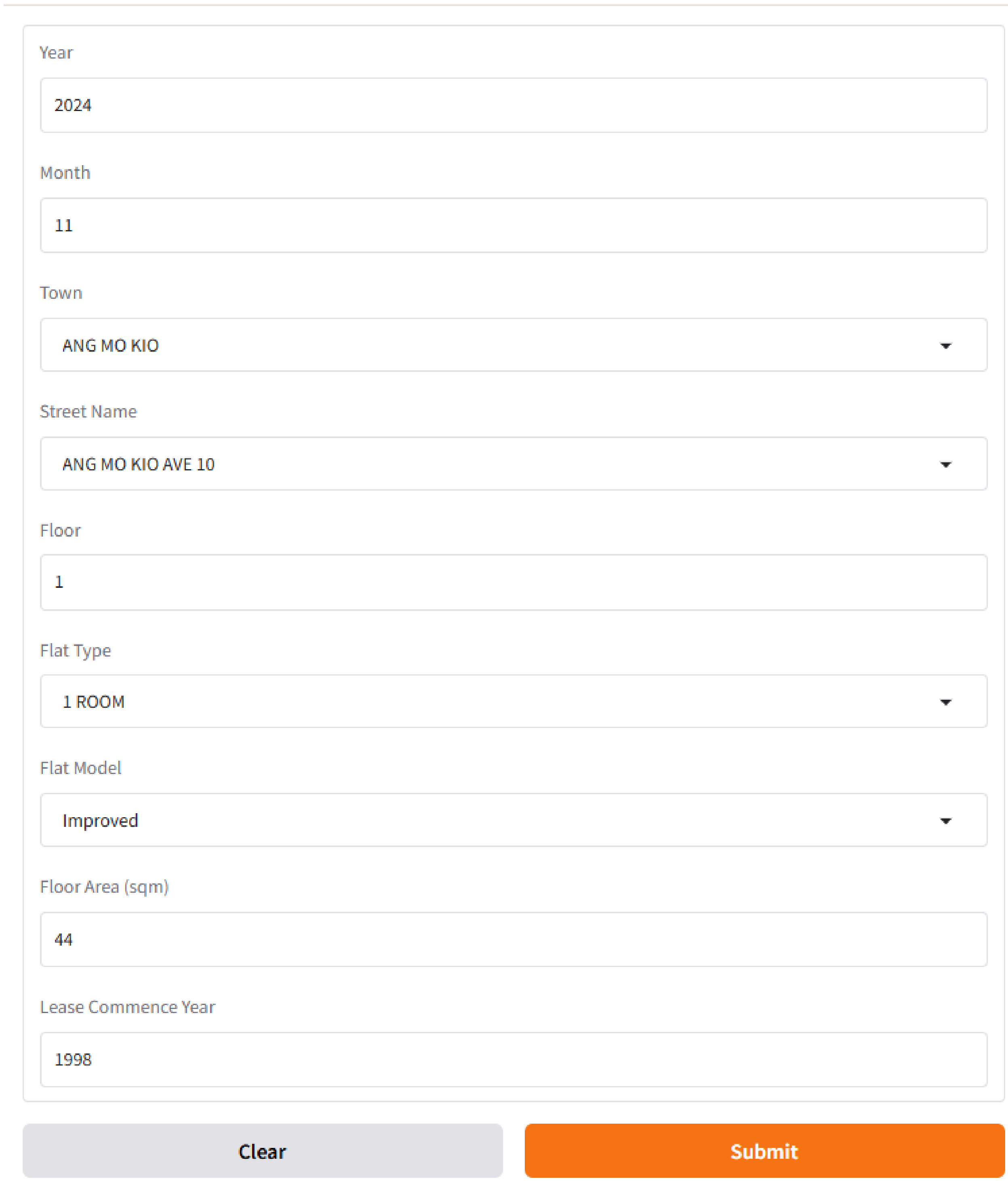


5.2.3 Residual Distribution



Appendix 7 - NeuralNest WebApp User Interface:

Input Field:

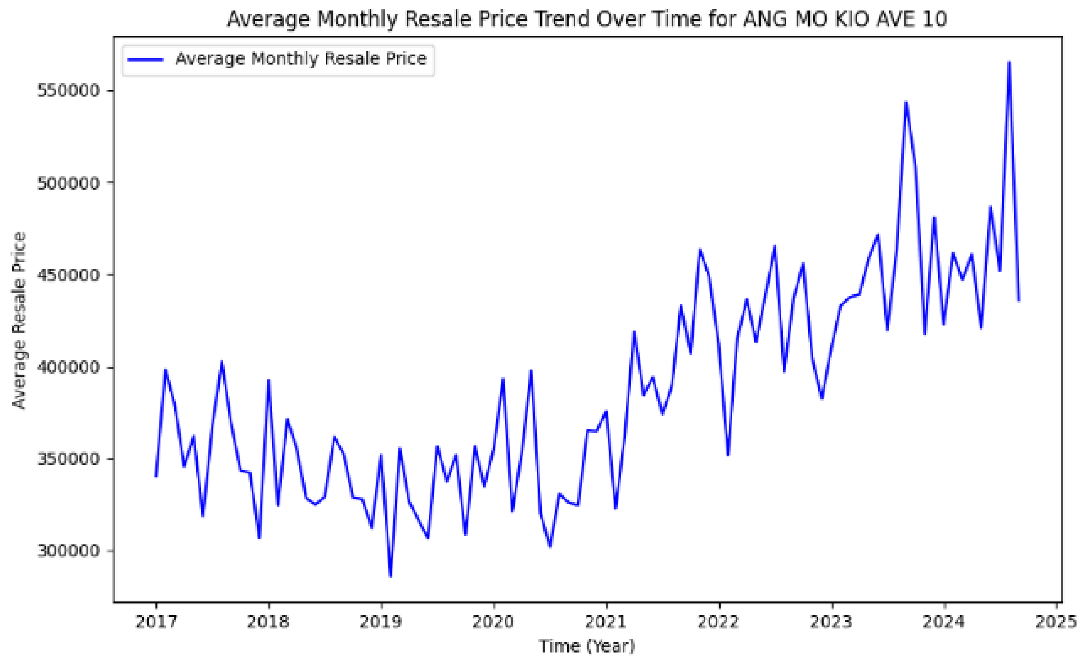


Output Field:

Predicted Resale Price

S$305,524.78

Resale Price Trend Timeseries with following year prediction



AI Real Estate Advisor

Based on the provided details, your property in Ang Mo Kio has a predicted resale price of S$305,524.78, which is significantly lower than the average resale prices in the area, which range from approximately S$340,000 to S$565,000. This suggests that your property may be undervalued compared to similar properties.

Considering the flat type (1 room) and the remaining lease (73 years), it is important to note that smaller flats can appeal to a specific market segment, particularly for singles or young couples. However, the lower floor may deter some buyers who prefer higher floors for better views and ventilation.

Investment potential appears positive given the average resale trends, which show an overall increase in prices. If the property is indeed undervalued and the area maintains its appeal, it could provide good capital appreciation in the long term.

In conclusion, if you are comfortable with the unique characteristics of the property and the potential for value appreciation based on market trends, it could be considered a good buy.