# Introduction

The Malaysian real estate market plays a crucial role in the economy. This is fuelled by urbanization, population growth and the growing demand for commercial and residential properties.

With the changes happening in the market, real estate developers, investors and policy makers need to understand the factors impacting house price changes. The factors can optimize valuation of property, highlight causes of high demand and this can guide real estate developers on the expected growth and changes in house prices.

This project uses machine learning techniques to analyse the MYProperty\_sales.csv data to develop predictive models.. Steps taken to prepare the data for the models include data exploration, data cleaning, feature engineering and encoding of categorical variables. This will lay a foundation for predictive models that will give actionable insights for the Malaysian real estate market.

## Business Problem

While the Malaysian real estate market is still lucrative, like any other industry, it has challenges. House prices vary widely across regions due to factors such as property size, location, market demand, amenities, and accessibility to infrastructure.

Key challenges include unpredictable price trends leading to missed market opportunities, valuation disparities due to a lack of robust predictive tools, and proper planning due to the limited understanding of the real estate industry.

By creating a predictive model, the analysis will empower real estate companies and their policy makers to make informed decisions.

## Objectives.

To develop and compare machine learning models that predict house prices in the Malaysian real estate market.

To analyse the price dataset and identify the key factors influencing properties in Malaysia.

To evaluate the feature importance and understand the key variables affecting house prices.

# Methodology

Initial exploration of the property dataset involved inspecting for missing values, outliers, any inconsistencies, and summarizing key statistics. Data cleaning included imputing missing values and handling outliers. Numerical features were standardized, and categorical variables were also encoded.

Derived features like Income Per House Age and Rooms to Bedrooms Ratio were added. For model development, Random Forest (RF) and XGBoost were chosen. Models were evaluated using R² and RMSE metrics, with hyperparameter tuning via GridSearchCV. Cross-validation ensured model generalizability. Feature importance analysis highlighted the most impactful features and compared XGBoost and RF prioritizations.

## 2.1.Data Cleaning

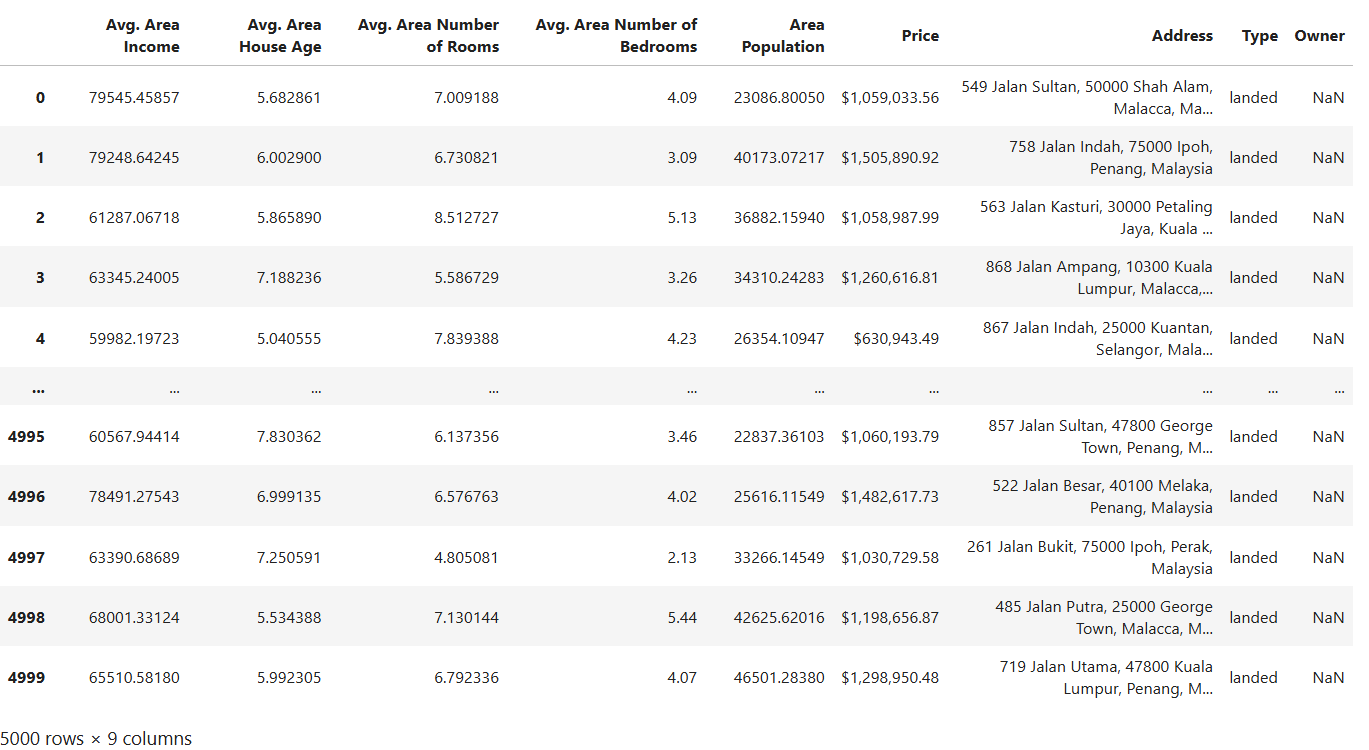
### 2.1.1.Importing Data and Libraries

The first step in this project is importing the dataset that will be used for the analysis and models. The dataset namely MYproperty\_sales.csv. was uploaded using the Python’s pandas.



Machine Learning Figure 1 - Loading the dataset.

Getting an overview of the data helps identify the data types, see the dimensions, and spot missing values. Through loading the data, some of the features are numerical and most of the columns are numerical.



Machine Learning Figure 2- Loaded dataset

The first step after loading the data is checking the summary statistics. This is a summary that identifies the count of every column, mean, standard deviation, the minimum and maximum values in each column.

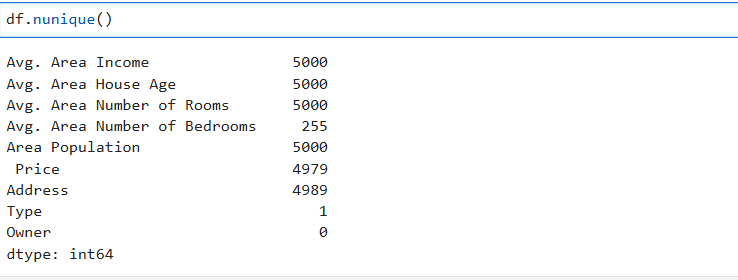
A screenshot of a computer

Description automatically generated

Machine Learning Figure 3- Summary Statistics.

Our property dataset has key observations that may need further investigation, it is easy to identify that Owner has 0.0 count. The average income is 68,583 yet the standard deviation . this indicates variability in the income levels. Most homes are between 5.32 and 6.65 years. The smallest population is 173 and largest 69,622 which means there is wide range.

With these insights, the next step would involve checking unique values in the dataset as summary statistics only shows numeric values only. Checking unique values is crucial to understand the diversity and variability within the whole dataset.



Machine Learning Figure 4- Unique

Type and Owner columns have zero and one unique value consecutively meaning that they have no useful information for distinguishing our datapoints and they lack variability. The columns need to be dropped leaving the dataset with 7 columns.



Machine Learning Figure 5- Number of columns

After dropping the irrelevant columns, it is important to understand the data types in our daatset. This is important for subsequent analysis and modelling tasks.

Columns states as (float64) contain numerical data and only the ‘Price’ and ‘Address’ are non-numerical data (object).

A screenshot of a computer code

Description automatically generated

Machine Learning Figure 6- Understanding data Types

Ideally, the price column should be numeric but there seemed to be inconsistencies as it is a non-numeric column. Steps to ensure the price column was shifted to a numerical column were taken.

Figure 7 shows the names of the columns before stripping.

A white background with black text

Description automatically generated

Machine Learning Figure 7– data types for stripping

Stripping is the process of removing spaces between character names. Extra spaces may cause mismatches which could lead to issues in the preprocessing and modelling .

A screenshot of a computer program

Description automatically generated

Machine Learning Figure 8 -After stripping

The ‘Price’ column is currently non-numeric due to the presence of dollar signs and commas. To address this, displaying the first few rows will help understand the format of the ‘Price’ column and know which characters need to be removed to allow conversion to numeric.

A screenshot of a computer code

Description automatically generated

Machine Learning Figure 9

To ensure that the ‘Price’ column is suitable for analysis, the unwanted characters are removed, the data is converted and verification of the changes is done. Finally, checking of ‘Nan’ values was important since the ‘Nan’ values cannot be converted to numeric and the results show that there were 20 missing values in the Price column as shown(Figure 10)

A screenshot of a computer

Description automatically generated

Machine Learning Figure 10- Post price stripping

After identifying that there are missing values in the ‘Price’ column, it is essential to investigate further and address any more missing values. Missing data can impact data analysis and distort conclusions drawn from models when left in the data.

After investigating the missing values, it is revealed that there are 20 missing values in the ‘Price’ column and 4 missing values in the ‘Address’ column.

A screenshot of a computer

Description automatically generated

Machine Learning Figure 11- missing values

To handle missing values in the ‘Price’ column, I plotted a histogram that will show how the prices are distributed and understand if the data follows a normal distribution, if it is skewed or if there are any abnormal peaks.

A graph of a red and black line

Description automatically generated with medium confidence

Machine Learning Figure 12- Histogram of Price

The histogram follows a normal distribution hence choosing to choose MEAN as the value that will be used to fill in the missing values. This will reserve the integrity of our data and ensure the missing values has been filled.

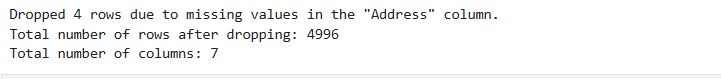
To fill in the missing values in the ‘Price’ I executed some codes and the output in the figure below confirmed that there were no missing values in the ‘Price’ column.



Machine Learning Figure 13- Missing values with mean

Initially, our dataset had five thousand rows and Address only has four missing values which means that only 0.08% of our data was missing. 0.08% of the data is negligible and may not have a significant impact in our analysis and it cannot change our output.

Dropping them was a prudent decision to maintain the integrity of the dataset. This enabled me to continue the analysis with confidence ensuring our results are complete and dependable.



Machine Learning Figure 14

After the data exploration process, I did a comprehensive overview of my data, df\_dropped to understand its size, structure, and completeness.

The data frame now contains 4996 rows and 7 columns, 6 of which are numerical and one containing string values, the ‘Address’ column. All my entries are non-null entries meaning there are no missing values.

A screenshot of a computer

Description automatically generated

Machine Learning Figure 15

## Data Exploration

This is a step aimed at understanding patterns and characteristics of a dataset. One of the primary tools of data exploration is visualization that which allows one to grasp the distribution and behaviour of different variables.

In this dataset, I used Histograms (figure 16) to show the distribution of some variables such as Income, House Age, Number of Rooms, and the Area Population.

A group of graphs showing different sizes of numbers

Description automatically generated with medium confidence

Machine Learning Figure 16

The image above shows four histograms with different variables. Some of the patterns include:

Distribution of Avg. Area Income is symmetrically distributed and has a peak around 60,000. There is no significant skewness observed. This may suggest that the potential buyers are middle income earners and this may influence the pricing strategies of houses. Understanding the income distribution is important for investors and developers to determine affordability thresholds and set prices.

Distribution of Avg. Area House Age is also normally distributed and has a peak at around 6 years meaning that most property are around 6 years. This means that most houses are recently built. Newer houses means that they have modern amenities which makes them desirable. Lack of skewness could indicate that there is a range of house ages from new to older houses.

Distribution of Avg. Area Number of rooms has a normal distribution and a peak at around 7 rooms. The histogram is symmetric and there is no significant skewness. Big and spacious homes usually have many rooms which is a good selling point because it supports customer preferences. Understanding that sizes of houses influence investment decisions is a guide for investors on what customers want.

Lastly, Distribution of Area Population that is normally distributed and has a peak within 40,000 and no notable skewness is observed. This is an indicator that the areas are moderately populated. Developers can use this information to prioritize regions with moderate population.

## Feature Engineering and Transformation

Feature engineering and transformation are important steps in preparing data for machine learning. It involves creating new features from existing data and structuring data in a format that can be used in models.

By extracting new patterns and addressing the data complexity, I improve the interpretability of the data by the models.

The first step in this process is extracting ‘Address’ column into individual components. I breakdown the address into 'street', 'postal\_code', 'city', 'state', and 'country'. This adds new columns into our dataset providing more detailed information about the addresses.

After processing the address data, figure 17 shows a snippet of the data part with the new columns that will be used for various analytical purposes and help make more informed decisions.

A screenshot of a computer

Description automatically generated

Machine Learning Figure 17

After creating new columns, it is important to recheck unique values for data quality assessment, to understand the data distribution and check for any redundancy.

A screenshot of a computer

Description automatically generated

Machine Learning Figure 18

Identifying unique columns helps identify columns that are irrelevant to our analysis. In this case, Country has one unique feature hence it needs to be dropped. Conversely, splitting the 'Address' column into its components has introduced unnecessary complexity to our dataset.



Machine Learning Figure 19

### Creating new features

New features capture complex relationships that can provide additional context to the machine learning models. I created two features ‘Rooms to Bedrooms Ratio’ and ‘Income per House Age’ .

**Rooms to Bedrooms Ratio**:

The general formula for this is Avg. Area Number of Rooms / Avg. Area Number of Bedrooms. This captures how the number of rooms in the house compares to the number of bedrooms. A higher ratio may suggest houses with more rooms apart from the bedrooms, indicating the presence of additional living spaces such as living rooms, dining rooms, and home offices.

On the other hand, a low ratio would indicate that most rooms are bedrooms, suggesting a more straightforward home. The Rooms to Bedrooms Ratio combines two key features into a single metric that provides a more nuanced understanding of the housing structure. It will also capture housing preferences, reflecting how buyers prefer homes with more communal rooms relative to bedrooms, and improve the predictive power of the models.

Understanding the rooms to bedrooms ratio can help developers determine the market appeal for different house layouts. Houses with higher ratios may attract buyers who need more living space, whereas a lower ratio indicates that buyers prioritize more bedrooms. In pricing, houses with spacious layouts tend to be pricier than the compact ones.

**Income per House Age:** The general formula for this is Avg. Area Income / Avg. Area House Age. This identifies how ‘affluent’ an area is. It can provide insight into the relationship between wealth and housing development trends. A high Income per House Age index indicates a region with higher income and newer houses, showing that affluent families are moving to areas with better house developments. A low Income per House Age shows a region with lower average income and/or older houses, indicating that this type of area does not keep up with modern housing trends.

Having an area with a low ratio will pinpoint areas where old homes dominate, leading to lower property prices. The Income per House Age will capture economic patterns in an area and emphasize the joint influence of house prices. Lastly, this feature can help the model capture the economic strength and the impact housing age has on property prices.

After adding the new features, Figure 20 displays the first few rows.

A screenshot of a computer

Description automatically generated

Machine Learning Figure 20

At this point, I had added a few columns especially after splitting the ‘Address’ column hence the need to understand the structure of the columns and know how to ensure the categorical variables are understandable by the models. After adding a few new columns, I needed to identify the unique structures in the ‘city’ and ‘state’ columns. The main reason for doing this is to understand which features can be generated from the unique values in the specific columns.

The output in figure 21 shows the unique cities and states in the data.

A group of words on a white background

Description automatically generated

Machine Learning Figure 21- unique in city and states

Machine learning models use numerical features to interpret data hence the need to encode the state columns in a way understandable to the models. Before that, I decided to split the state column to regions instead to understand if house prices differ in certain regions.

Malacca and Johor would be southern, Penang and Perak would be Northern, Kuala Lumpur and Selangor would be Central and lastly Pahang would be on the eastern side. Figure 22 is an output of the first 5 column to confirm if the process was successful.

A white background with black text

Description automatically generated

Machine Learning Figure 22 -Encoded Regions

The remaining parts of ‘Address’ column that are yet to be managed are postal\_code, city and street which will be dropped since they provide hyper-specific information. The information in these three columns is already captured by the Region hence keeping these three columns will only introduce redundancy.

## Correlation Analysis

This displays the correlation coefficient between multiple variables in a dataset. The correlation coefficient ranges between -1 and 1 where 1 indicates perfect positive correlation, -1 shows perfect negative correlation and 0 means no correlation at all.

Before checking the correlation, it is essential to focus on numerical variables only as the metric is meaningful for numerical data only.

A screenshot of a computer program

Description automatically generated

Machine Learning Figure 23- data types

Figure 24 is the correlation matrix showing the correlations in our data.

A screenshot of a graph

Description automatically generated

Machine Learning Figure 24 – correlation matrix

Strong positive correlations observed between Price and Avg. Area Income(0.64) and Price per House Age(0.65). Moderate Positive correlations between Price and Average House Age(0.45) and Price and average no. Of rooms (0.33). some have weak positive corrlations like Average Area No. Of Bedrooms.

## Outliers

Outliers are the extreme values that could potentially distort our analysis, despite having a normal distribution, there are high chances of having outliers in our numerical features.

### Identifying Outliers

I utilized the Interquartile Range method to identify the outliers in the numerical columns. Based on the Distribution histograms done earlier, the numerical columns show a normal distribution hence the need to use IQR.

The IQR method uses this method below to identify the outliers.

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

All values below the lower bound and above the upper bound are considered outliers.

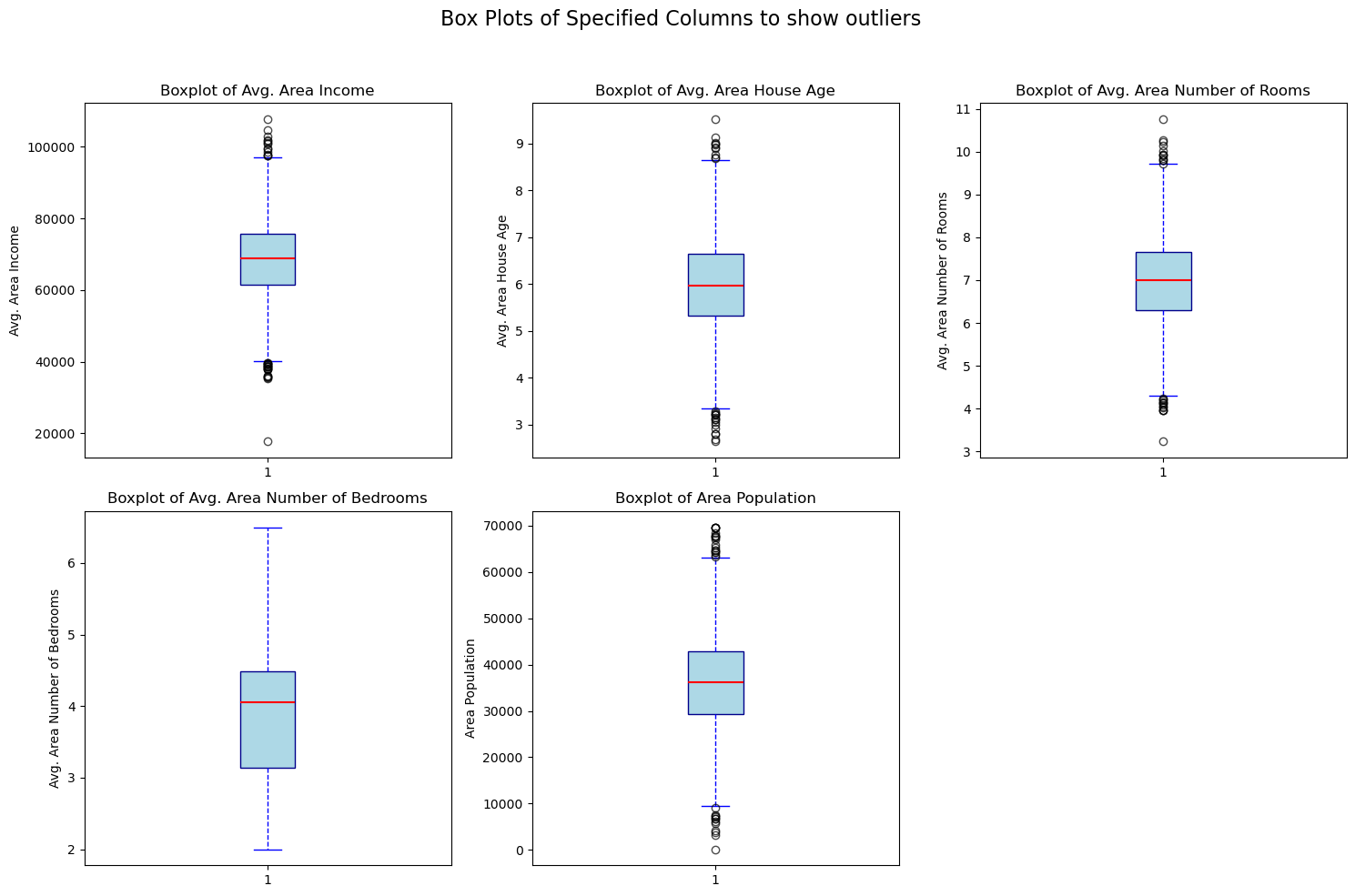
Using the IQR method, there are few outliers stated in figure 25.

A screenshot of a computer code

Description automatically generated

Machine Learning Figure 25 - Outliers

Visualizing the outliers to make spot them easily and to understand the distribution is important to decide on how to manage them . I used box plots to visualize the outliers in the numerical features. Figure 26 shows the outliers in the following columns. 'Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms' and 'Area Population'.

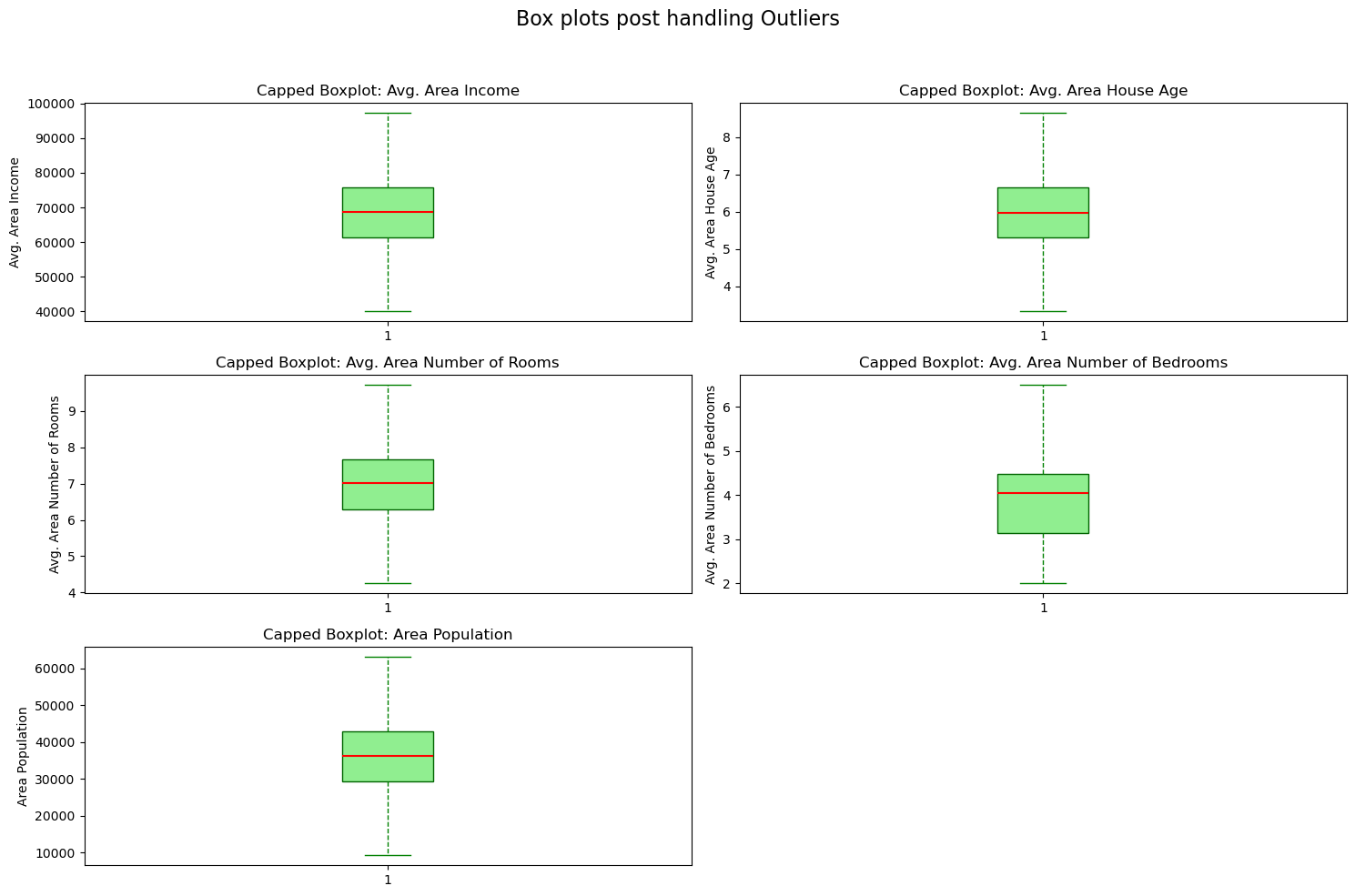


Machine Learning Figure 26 Box plots for outliers

### Handling Outliers

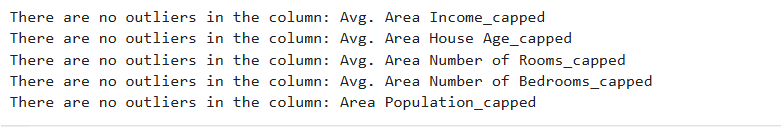
The handling of outliers was done using capping and flooring method based on the Interquartile method. Any data above the upper bound were capped at the upper bound and the data below the lower bound was floored to the lower bound as well. This method of handling outliers ensures that the data remains within a sensible range and preserves the overall structure and distribution.

Figure 27 shows the capped columns to highlight the changes and check if the process was successful.



Machine Learning Figure 27- After Handling outliers

Since box plots are only for visual purposes, it is important to do a quantitative confirmation that there are no more outliers. This is an extra step to ensure the handling of the outliers was thorough.



Machine Learning Figure 28

## Encoding

Categorical variables should be converted to a numerical format so that they can be used by machine learning algorithms. Most machine learning models like linear regression and decision trees use mathematical computations. This is why variables like ‘region’ need to be converted into a format the algorithms can understand.

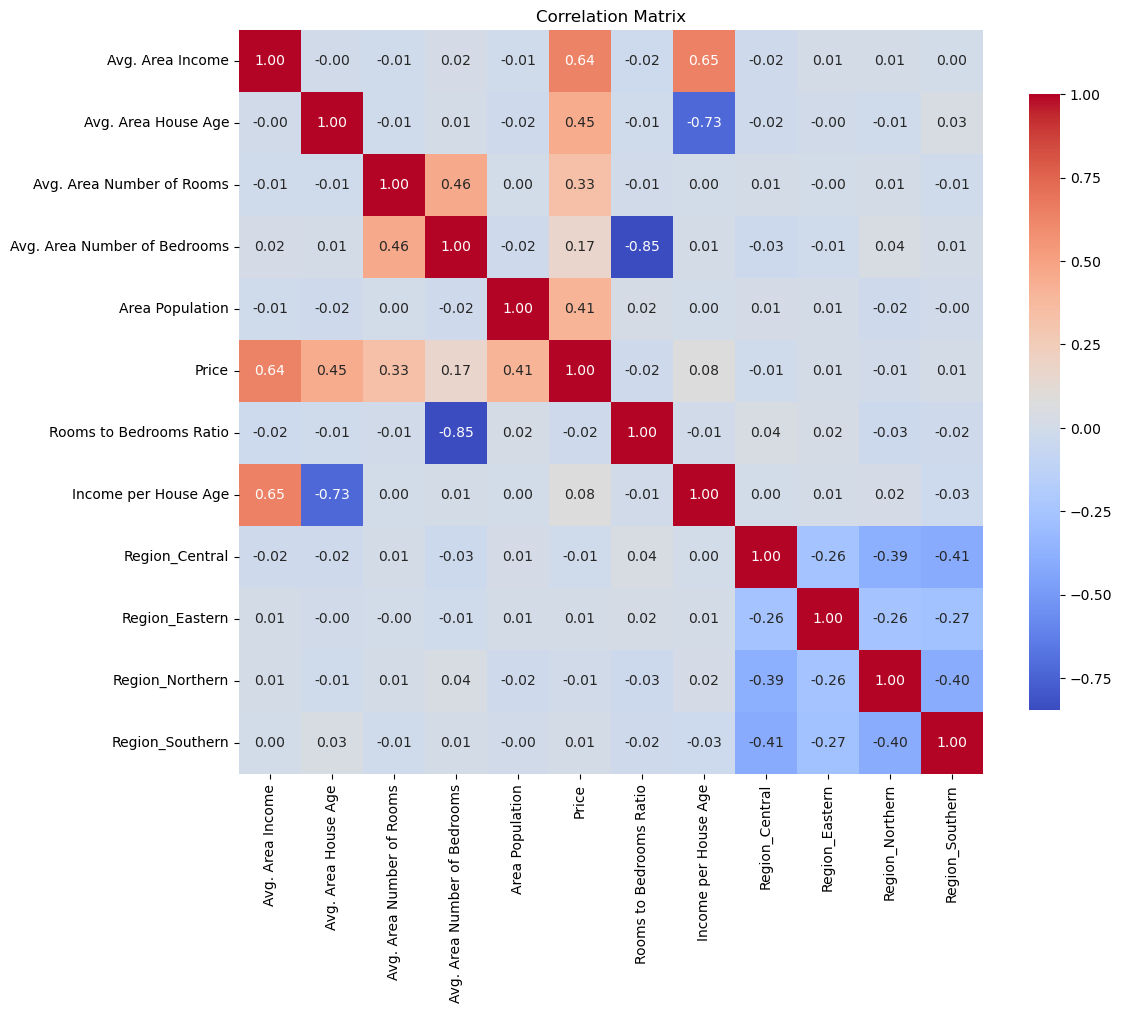
The method used to encode ‘Region’ was one-hot encoding. This is because there is not ordinal relationship between the regions.

After encoding, the first 5 rows of the dataset to show the new encoded columns were displayed A screenshot of a computer

Description automatically generated

Machine Learning Figure 29

## Correlation Heatmap after encoding.



Machine Learning Figure 30

After encoding, I created a heatmap to explore how the key variables in the dataset are related. The heatmap will reveal the features that will impact the house prices.

Avg. Area Income and Price has a correlation 0.64 . this means that an increase in the average area income results to an increase in House Pirces around the same area/ Avg. Area Number of Rooms and Price have a correlation of 0.33 meaning that more rooms mean a bigger house meaning higher price of the houses.

Avg. Area House Age and Income per House age has a strong negative correlation of 0.73. Older houses might be less expensive because they also need more repairs and maintenance.

After doing the correlations, I ensured all the columns are arranged nicely in the dataset. This was to ensure the columns are in the preferred order.

After comprehensive data preprocessing where I did the data cleaning , outlier handling, feature engineering, encoded categorical variables and rearranged the columns, i saved the DataFrame to a CSV file for future analysis.

The execution of the steps ensures that the dataset is now well- structured and ready for any analytical tasks or machine learning applications.



Machine Learning Figure 32

# MODEL TRAINING AND TESTING

In the second part of my project, I developed a predictive model using a cleaned dataset with a goal to predict house prices in Malaysia. The process begins with the essential steps of data preparation which include loading the cleaned dataset, importing the necessary libraries, and conducting preliminary checks to ensure the data is clean.

The first step was to load the dataset titles PropertyEncoded as shown in figure 45. This data contains encoded and feature-engineered data that will later be used to train the models to predict house prices.



Machine Learning Figure 33

Following the successful loading of our dataset, Figure 46 is a preview of our dataset and it shows that all values are numerical and prepared for training and testing.

A screenshot of a computer screen

Description automatically generated

Machine Learning Figure 34

Before proceeding to training and testing, it is essential to conduct a thorough check of our data. This includes checking for any missing values or any duplicates. This is to ensure the data is complete, to avoid any bias and ensure data integrity.

After an examination of the dataset, it is revealed that there are no missing values and nu duplicate rows. This means that the dataset is well-prepared for the steps of model training and testing.



Machine Learning Figure 35



Machine Learning Figure 36

## 3.1.Training and Test

In this section, the dataset is prepared for training and testing machine learning models.

The data was split into features and target because machine learning models learn to predict the target variables based on the features . the data is split into training and testing sets. In this case, 80% of the data was allocated for training and 20% was allocated for testing. The training part is used to train the model and the test is used as unseen data that will later be used to assess how well the model works.

An instance of the ‘StandardScaler’ was created to standardize features . It ensures that all features contribute equally by preventing features with larger ranges to distabilize and influence the model negatively. ‘Fit\_transform’ is also applied to the training data . it computes the mean and standard deviation for each feature in the training set and scales it. Lastly ‘Transform’ is applied to the test data to ensure consistency.

**Random Forest**

The first model to train is the Random Forest so that it can predict the ‘price’ which is the target variables. Random Forest combines multiple decision trees to improve predictions and avoid overfitting.

The ‘RandomForestRegressor’ is created . The trees used to make predictions are set to one hundred and a parameter of 42 is made to ensure results are reproducible. The model is trained and later the ‘predict’ method is used to make predictions on the test data.

The metrics used to evaluate Random Forest are R^2 score and Root Mean Squared Error.

The R^2 measures how well the model’s prediction matches the actual values. When the R^2 is 1, it means perfect prediction and 0 means no variability is explained by the model.

On the other hand, RMSE calculates the mean squared error between actual values and the predicted values. A lower RMSE indicates a better performing model. Figure 54 shows the performance of Random Forest.

R2 is 87.7 % and the RMSE is 122096.156.

A number on a white background

Description automatically generated

Machine Learning Figure - 37 RF

**XGBoost**

The second model to train is XGBoost. It is a powerful and efficient implementation of gradient boosting .

The parameter is set to 42 to ensure reproducibility of results. The model is then trained by allowing it to learn the relationships between features and the target variable. After making predictions with our model, the predictions are then compared against the actual values to see how well the model performed.

A number and a number

Description automatically generated with medium confidence

Machine Learning Figure – 38 XGB

Evaluation of the model is done the same way as the Random Forest. Using R^2 and RMSE. In XGBoost, the output in figure 56 shows that R^2 IS 86.9% and the RMSE is 125899.41

**K-Nearest Neighbors Model**

The third model choice was KNN Regression to predict the target variable ‘Price’. It makes predictions based on the average of the target values of the nearest neighbours in the features space.

The first step is to train the KNN model by allowing it to learn the relationships between features and target variable. The ‘predict’ method is used to make predictions.

The ‘score’ method is used to calculate the R^2 to measure how well the model predictions match the actual values. The other method OF evaluation used is RMSE. The results in figure 58 show that R^2 78% and RMSE is 163289 .

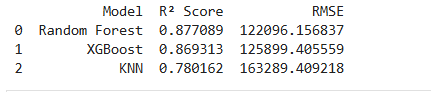


Machine Learning Figure 38 – KNN

In the above steps, I successfully trained three distinct machine learning models: Random Forest, XGBoost and K-Nearest Neighbours. Each model was initialized and trained using the cleaned dataset. Predictions were also made on the test data and assessed the performance of the models using R^2 and RMSE.

The next steps involve more comprehensive performance evaluation. The steps will include hyperparameter tuning, regularization and cross validation to further optimize the models. The feature importance of the best performing models will be compared and this will help in selecting the most effective model for predicting house prices.

Based on the R2 and RMSE, here is a summary of the performance of the three models.



Machine Learning Figure 39

So far, the Random Forest exhibits the highest R^2 meaning at it explains approximately 87.71% of the variance in the target variable. This shows a strong predictive capability. The RMSE IS 122,096.16 means that the model deviates from actual values by this much.

XGBoost follows with a R^2 OF 86.93% which is lower than that of Random Forest. However, its RMSE is higher as it is 125,899.41. this means its a strong performer but its accuracy does not match what random Forest offers.

KNN has the least R^2 OF 78.02%. The RMSE is significantly higher , 163,289.41, indicating that its predictions are further from the actual value meaning it might not be the best choice.

## Hyperparameter Tuning and Regularization of the Models

Hyperparameters are settings that are used to control the learning process of machine learning algorithms. Doing hyperparameter tuning improves model performance, it helps mitigate the issues of overfitting and underfitting and helps ensure the model performs consistently across different datasets.

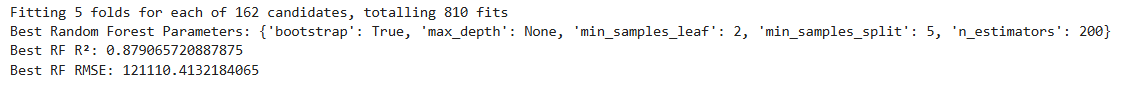
Regularization are techniques used to prevent overfitting by adding a penalty to the loss function during model training. It can be implemented in many machine learning models.

**Random Forest**

Hyperparameter tuning has a substantial impact on Random Forest and this can be explored by exploring different combination of hyperparameters. Through this, we can identify settings that optimize the model’s ability to generalize to new data.

Regularization helps control the complexity of the trees and prevent overfitting. the results after these processes lead to identification of optimal parameters that will enhance the model’s predictive performance.

The results after hyperparameter tuning and regularization indicate a successful optimization of the model’s parameters. R^2 of approximately 87.91% and RMSE of 121,110 indicates that the model’s predictions are closer to the actual house prices. This process highlights the importance of fine-tuning machine learning models to achieve optimal performance.



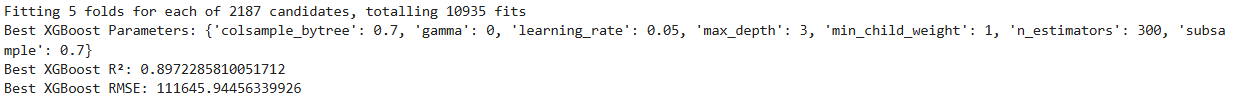
Machine Learning Figure 40

**XGBoost**

Like Random Forest, XG Boost undergoes the same process of hyperparameter tuning and regularization. It will enhance the predictive capabilities.

The results from the process indicate a successful optimization of the model’s parameters. The R^2 is approximately 89.72% which is higher than the previous one. The RMSE is 111,645.94 indicating that the model price predictions are close with that figure.

Figures below 41 shows outputs of XGBoost after hyperparameter tuning and regularization.

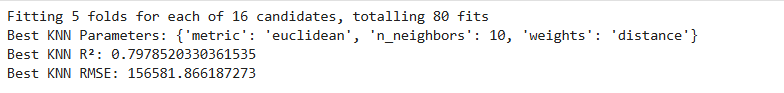


Machine Learning Figure 41

**KNN**

It was also necessary to perform the hyperparameter tuning and regularization on KNN to observe if the model performance would improve. The process ensures the model is robust against overfitting leading to better performance.

The results show that R^2 is 70.70% which is still low compares to Random Forest and XG Boost. The level of error is high as well considering the RMSE IS 156,581. Figure 65 shows the output after hyperparameter tuning and regularization.



Machine Learning Figure 42

## Train and Test Performance

To evaluate the performance of the three models further, I compared the performance of the Random Forest, XGBoost and KNN models on both their training and test datasets. Despite having done this, using the training datasets as well helps identify areas for potential improvement and understand how well the model has learned from the data.

Based on the steps taken so far, XGBoost and Random Forest are our preferred models.

1. Random Forest. According to figure 67, it now has R^2 of 87.91% on the test performance whereas the training has an R^2 of 97.31%. The training performance has RMSE of 57,896 but the test shows a larger error of 121,110 which may suggest some overfitting.

A screenshot of a computer program

Description automatically generated

Machine Learning Figure – Training and Test performance 43

1. XGBoost training has an R^2 OF 92.88% and test has 89.72%. This is slightly better than Random Forest. The RMSE on training is 94,200 and on test is 111,645 making it a well performing model on the unseen data.
2. Lastly, KNN has R^2 is100%.and 0.0 on RMSE suggesting that it is making perfect predictions on training data. R^2 has 79.79%. and RMSE of 156,582 and this indicates degradation in KNN performance.

This eliminates KNN as it is overfitted. It is performing perfectly on training but unable to generalize. This shows that the model is not fit for this task.

## Cross Validation

This is a technique in machine learning used to estimate the capabilities of models. This part provides additional insights into the performance and stability of the top performing models , Random Forest and XGBoost. Cross Validation aims to provide a more accurate estimate of the model’s ability to perform well on unseen data.

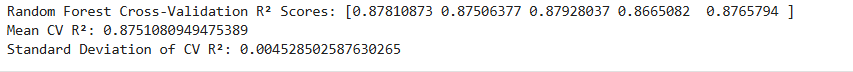
In this step, low standard deviation indicates consistent performance of the models.

**Random Forest**

Despite the previous superior performance of the Random Forest, it is necessary to continue and verify how better it can perform hence the need to check the scores of cross-validations executed with codes.

The cross-validation R^2 scores, [0.87810873, 0.87506377, 0.87928037, 0.8665082, 0.8765794], represent the results obtained from each of the 5 folds of the cross-validation. They are all high and close to **one** meaning that Random Forest explains a high proportion of the variance in the Target variable which is the House prices.

A standard deviation of 0.0045 shows that the model is consistent across all the subsets of the data. A small standard deviation means that the model is stable and its performance does not vary significantly.

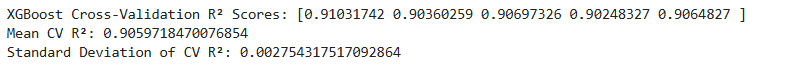


Machine Learning Figure 44

Overall, Random Forest shows is proving to have strong predictive performance which means it could potentially be a good model for predicting house prices.

**XGBoost**

Applying the cross-validation technique on the XGBoost will provide a more robust estimate of the model’s predictive power which will mitigate issues like overfitting. these results can easily be compared with the Random Forest to evaluate which model is best for predicting house prices.



Machine Learning Figure 45

Figure shows the performance of the model. The cross-validation scores, [0.91031742 0.90360259 0.90697326 0.90248327 0.9064827 ], reflects the proportion of the target variable that the model can explain. The mean of the values if approximately 90.6% which can be considered excellent performance in most regression tasks.

A standard deviation of 0.00275 is a positive sign of how dependable the model is. The model is stable across different data splits thus reinforcing confidence in generalization capability.

**Feature importance**

This step involves understanding the contribution of different variables in the predictive modelling. Through this, insights on which factors are most influential in predicting house prices in the Malaysian real estate market. Three models were trained and two deserve a fair comparison of the feature importance due to their performance.

The models were used to predict house prices based on several factors such as Area Income, House Age, Population and Region.

The scores of both models, XGBoost and Random Forest, were compiled into a single dataframe for easier comparison. The bar graph in figure() represents the important scores.

A graph of a bar chart

Description automatically generated with medium confidence

Machine Learning figure 45- Feature Importance

### Key Observations

The most prominent features in XGBoost are Average Area Income and Average Area House Age. This means that higher income areas strongly correlate with higher house prices. Affluential residentials usually have higher property values. House Age suggests that there is a demand for newer properties.

Average Area Income and Area House Age are also the most prominent features in Random Forest. Area Income is stronger on Random Forest than XGBoost meaning that In Random Forest, this may make Random Forest less sensitive to other features leading to a slightly imbalanced model.

Area Population and Number of Rooms are moderate on both models. More rooms usually mean a bigger house which would automatically impact the prices. Generally, houses with more rooms are more desirable hence the positive correlation with the house prices.

An interesting finding is that Income Per House Age was identified by XGBoost but not by Random Forest. This is because XGBoost captures complex and non-linear interactions whereas Random Forest builds trees independently. Income per House Age had a less direct relationship with house prices which may have been overlooked by Random Forest.

Random Forest may have overlooked the Income Per House Age due to reliance on variance reduction during its use of trees. The ability to capture Income per House Age gives it an advantage of being a better model for understanding factors influencing House Prices.

### Business Insights

Since Income directly affect prices of houses , the first focus would be focusing marketing and property development in affluent neighbourhood. Similarly, purchasers of homes understand that homes in developed areas are likely to be pricier.

Newer properties have a higher demand hence there should be a focus on ensuring modern homes are well- constructed since they are pricier than older homes. Real estate developers should ensure they develop premium houses .

Policy makers need to establish urban development policies to improve infrastructure in all areas, this would increase demand of houses and automatically the value of houses would rise.

Investors should have a look out on areas with upward trend. Developing houses in such areas will for sure result to pricier homes, meaning more profits from houses.

### Limitations

Since the data is only limited to Malaysia, it restricts generalizability of the findings to other countries /regions.

Limitations of features. Proximity to amenities, security in the areas and political stability in the areas were not captured in the models.

Model Assumptions . Random Forest in my model relies on reduction of variance and XGBoost sensitivity relies on hyperparameter, this may introduce bias in the predictive models.

# *APPENDIX*

*Dropping the Type and Owner Column*

*A white background with black text

Description automatically generated*

*Checking data types*

**

*Stripping the columns*

*A screenshot of a computer program

Description automatically generated*

*Stripping the price column*

*A screenshot of a computer code

Description automatically generated*

***Missing values***

*A screenshot of a computer

Description automatically generated*

***Price Histogram before handling missing values.***

*A computer screen shot of a code

Description automatically generated*

***Replacing mean***

*A screenshot of a computer code

Description automatically generated*

***Dropping missing values in Address***

*A computer screen shot of a computer code

Description automatically generated*

***Number of Rows***

*A screen shot of a computer code

Description automatically generated*

***Histogram of numerical features***

*A screenshot of a computer code

Description automatically generated*

***Extracting new columns from ‘Address’***

*A computer screen shot of text

Description automatically generated*

*A screen shot of a computer code

Description automatically generated*

***Creating new feature***

*A screenshot of a computer

Description automatically generated*

***Correlation Matrix map***

*A screenshot of a computer program

Description automatically generated*

*Identifying Outliers*

*A screenshot of a computer program

Description automatically generated*

*Visualizing outliers using Box Plots*

*A screenshot of a computer code

Description automatically generated*

*Box plots post outliers*

*A screenshot of a computer program

Description automatically generated*

*Confirming there are no outliers*

*A screenshot of a computer

Description automatically generated*

*Encoding Region*

*A group of colorful text

Description automatically generated*

*Correlation matrix post encoding*

*A computer screen shot of a computer code

Description automatically generated*

*Reorganized my columns in my data and saved it as Property Encoded*

*A screenshot of a computer

Description automatically generated*

*SPLITTING TRAIN AND TEST*

*A screenshot of a computer program

Description automatically generated*

*RF TRAIN*

*A screenshot of a computer program

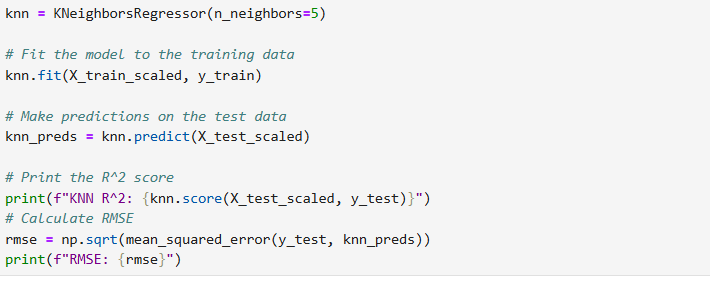
Description automatically generated*

*XGB TRAIN*

*A screenshot of a computer program

Description automatically generated*

*KNN train*

**

*Evaluating the 3 models performance*

*A computer screen shot of a program

Description automatically generatedHyperparameter tuning and regularization*

*OF Random Forest*

*A screenshot of a computer program

Description automatically generated*

*Of XGBoost*

*A computer screen shot of a program code

Description automatically generated*

*Of KNN*

*A screenshot of a computer code

Description automatically generated*

*Training and Test comparison*

*A screenshot of a computer program

Description automatically generated*

*Cross Validation*

*A screenshot of a computer code

Description automatically generated*

*Feature Importance*

*A computer screen shot of text

Description automatically generated*