1. Introduction

1.1 Problem Statement

Kuala Lumpur's real estate market is a bustling hub, constantly evolving with new urban developments, population shifts, and economic currents. For anyone looking to buy or sell property, this dynamic environment can be quite challenging, making it tough to pinpoint accurate prices and understand underlying market trends. Traditional ways of valuing properties often rely on subjective assessments, which can miss crucial market nuances. Our project steps in to tackle this head-on, offering a clear, data-driven approach to truly understand Kuala Lumpur's housing market. By using powerful machine learning techniques, we aim to deliver reliable tools for predicting house prices and shed light on different market segments, empowering everyone involved to make smarter decisions. This directly addresses the importance of segmentation and prediction within this problem domain.

1.2 Project Objectives

Our research is driven by a few key objectives, aiming to identify and predict domain behavior:

- To identify distinct market segments among residential properties in Kuala Lumpur using unsupervised learning.
- To develop accurate machine learning models to predict house prices using supervised learning algorithms.
- To evaluate and compare the performance of different regression models using standard metrics.
- To analyze key features that most significantly influence house prices.
- To visualize clustering results and price patterns for easier interpretation by stakeholders.

1.3 Datasets Used

This project utilizes a single, comprehensive dataset called "Property Listings in Kuala Lumpur," which we obtained from Kaggle. Initially, this dataset contained about 53,883 property listings and 8 main features. The original features included Location, Price (target variable), Rooms, Bathrooms, Car Parks, Property Type, Size (in square feet), and Furnishing (e.g., Fully, Partly, Unfurnished).

After extensive cleaning, preprocessing, and feature engineering, the dataset was refined to **41,180 rows and 9 columns** (Location, Price, Rooms, Bathrooms, Car Parks, Property Type, Size, Furnishing, Area), which served as an excellent foundation for our unsupervised and supervised learning tasks.

- **Unsupervised Learning Dataset**: For market segmentation, we used features like Price, Size, Rooms, Bathrooms, and Car Parks (all appropriately scaled), along with Property Type and Area (both one-hot encoded). Our goal was to apply K-Means clustering to reveal hidden market structures, showing us distinct segments like "Luxury" and "Mid-Range" properties.
- **Supervised Learning Dataset**: For predicting house prices, our main target was the log-transformed Price column (y). The independent variables (x) included Rooms, Bathrooms, Car Parks, log-transformed Size, and target-encoded PropertyType_encoded and Area_encoded

2. Data Preprocessing

Data preprocessing was our first major step. It was absolutely essential to transform our raw, messy dataset into a clean, structured format that our machine learning models could understand and learn from. This stage was all about ensuring top-notch data quality, consistency, and suitability for our chosen algorithms. This section provides a detailed explanation of the preprocessing steps.

2.1 Handling Missing Values

When we first looked at our data, we found quite a few missing values. Here's how we dealt with them:

- **Numerical columns**:

    - Price: Since this was our main target, rows with missing prices were **removed** entirely, as we couldn't predict anything without this information. This initial step led to **248 rows** being dropped.
    - Rooms: We filled in missing Rooms values with the **mode** (the most frequent number of rooms), which is a sensible way to impute discrete values.
    - Bathrooms: Missing Bathrooms values were filled using the **median**, as this is less affected by outliers than the mean.
    - Car Parks: We imputed missing Car Parks values with **0**, assuming that if no parking was specified, it meant there wasn't a dedicated spot.
    - Size: Missing Size values were also filled with the **median**, providing a robust imputation.
- **Categorical columns**:
    - Furnishing: Any missing Furnishing details were simply marked as **'Unknown'**.
    - Property Type: Missing Property Type values were filled with the **mode** (the most common property type), ensuring all listings had a classification.

2.2 Feature Scaling

To make sure our machine learning algorithms performed at their best, especially with the presence of outliers, we used **RobustScaler** from the Scikit-learn package (Pedregosa et al., 2011). This scaler is fantastic because it's "robust" to outliers, meaning it uses the median and interquartile range for scaling instead of the mean and standard deviation, which can be easily skewed by extreme values. This provides a detailed explanation and justification for our choice of scaling method. For supervised learning, we first "fit" the RobustScaler on our training data (X_train) to learn the scaling parameters, and then we used it to transform both X_train and X_test. For unsupervised learning, we applied RobustScaler to all our relevant numerical features (Size, Price, Rooms, Bathrooms, Car Parks) before proceeding with clustering.

2.3 Encoding Categorical Variables

Our dataset contained important categorical features: Location, Property Type, and Furnishing. We needed to convert these into a numerical format for our models, describing how these features were transformed.

- **Location**: This column was quite detailed. We simplified it by extracting just the main Area (the part before the first comma, like "KLCC"). To reduce the number of unique categories, any Area that appeared less than **100 times** was grouped into an **"other"** category.
- **Property Type**: We cleaned this column by removing any text in parentheses (e.g., "Condominium (Corner)" became "Condominium"). We also merged similar types, like combining various "Terrace/Link House" descriptions into a single "Terrace House" category, and "Flat" into "Apartment". Any Property Type appearing less than **100 times** was grouped into an **"Other"** category.
- **Encoding Methods**:
  - For **supervised learning**: Area and Property Type were transformed using **target encoding**. This means we replaced each category with the average log-transformed price of properties within that category. Furnishing was assigned numerical values using **ordinal encoding** ('Partly Furnished' = 0, 'Fully Furnished' = 1). However, Furnishing_encoded was later **excluded** from our final supervised models because it showed a very weak correlation with price.
  - For **unsupervised learning (K-Means)**: We used **one-hot encoding** (pandas.get_dummies()) for the preprocessed Property Type and Area columns. This creates new binary columns for each category, which is ideal for clustering algorithms.

2.4 Feature Engineering

We went a step further by extracting, engineering, and transforming some key features to boost their predictive power, mentioning additional features created:

- **Size**:
  - Numerical square footage was extracted from the Size text column, which came in various formats (e.g., "Built-up : 1,335 sq. ft.", "Land area : 6900 sq. ft.", or "22 x 80 sq. ft."). We converted these into a float data type.
  - **Outlier Removal**: We removed properties with Size values **less than or equal to 0**. Additionally, we filtered out properties with Size values **below 400 sq. ft.** or **above 4000 sq. ft.**, as these extreme values likely represent data errors or non-residential properties and could skew our models.

  - **Transformation**: To address the skewness in Size, we applied the np.log1p() transformation. This significantly reduced the skew, making the distribution more symmetrical and beneficial for our models.
- **Price**:
  - The Price column was cleaned by stripping out "RM" currency symbols and commas, then converted it to a numerical format.
  - **Outlier Removal**: We identified and filtered out extremely low prices (below **RM 100,000**) and exceptionally high prices (above **RM 6,000,000**), as these likely represent data entry errors or unique market anomalies not typical of the standard residential market.
  - **Transformation**: The Price column initially had a severe right-skewness of **1.69**. Applying the np.log1p() transformation dramatically reduced this skewness to **0.30**, bringing the distribution much closer to normal and improving model performance.
- **Rooms**: We extracted the primary number of rooms (e.g., converting "2+1" to 2) and converted the column to an integer data type. We also removed properties where Rooms was **greater than 8**, considering these outliers for typical residential units.
- **Bathrooms** and **Car Parks**: After imputing missing values, both Bathrooms and Car Parks columns were converted to integer data types. We further refined these by removing properties where Bathrooms was **greater than 7** or Car Parks was **greater than 5**, aligning our dataset with common residential property standards.

2.5 Data Splitting

For our supervised learning tasks, we divided our thoroughly preprocessed data into training and testing sets. We set aside **80%** of the data for training our models (X_train and y_train) and **20%** for evaluating their performance on unseen data (X_test and y_test). To ensure our results are consistent and reproducible, we used a random_state of **42**. After this split, X_train had a shape of **(32944, 6)**, and X_test had a shape of **(8236, 6)**. This explains how the dataset was split.

3. Unsupervised Learning: Market Segmentation

In this phase, we delved into unsupervised learning to identify natural groupings and segments within the Kuala Lumpur property market, without relying on predefined labels. This section provides detailed implementation and analysis.

3.1 Clustering Methodology

To uncover distinct market segments, we applied **K-Means Clustering**. Determining the optimal number of clusters (K) is crucial for K-Means, and we achieved this by using the **Elbow Method**. The visual representation of the Elbow Method clearly showed an "elbow point" at **K=2**, strongly suggesting that the market is best divided into two primary segments.

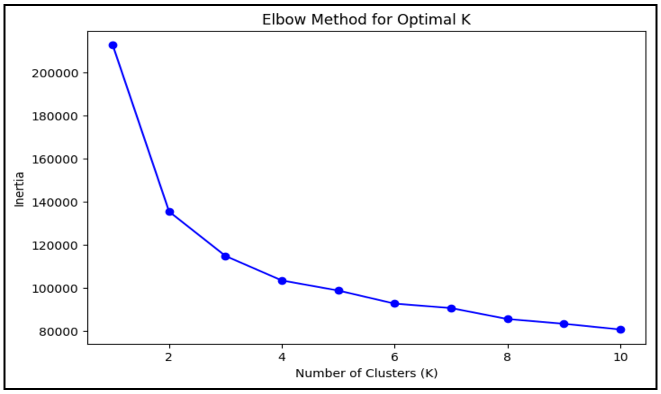

Figure 1: Elbow Method for K-Means

For the K-Means clustering process, we used the following features. It's important to note that all numerical features were **scaled using RobustScaler** before clustering, and categorical features were **one-hot encoded** to prepare them for the algorithm:

- Numerical features (scaled): Size (log-transformed), Price (log-transformed), Rooms, Bathrooms, Car Parks.
- Categorical features (one-hot encoded): Property Type, Area.

3.2 Model Evaluation

To assess how well our K-Means clustering performed with K=2, we used the **Silhouette Score**. **Result**: The Silhouette Score for our K-Means clustering with K=2 was **0.363**. This score suggests a reasonably good separation between the clusters, meaning that properties within each cluster are relatively similar to each other, while properties in different clusters are relatively dissimilar.

Interpretation of Clusters: To make sense of what each cluster represents, we analyzed the average (mean) values of the original, unscaled numerical features within each cluster, along with the distribution of categorical features. This allowed us to assign descriptive labels:

- **Cluster 0 - "Luxury Properties"**: This segment represents properties that, on average, command a higher original mean price of approximately **RM 1,889,500** and feature a larger original mean size of around **2,150 sq ft**. These properties also tend to have more rooms (mean: **3.74**) and bathrooms (mean: **4.18**). In terms of property types, Condominiums (61.59%), Terrace Houses (15.62%), and Serviced Residences (12.71%) are most prevalent. Popular areas within this cluster include Mont Kiara (25.39%), KLCC (14.32%), and Desa ParkCity (4.63%).

- **Cluster 1 - "Mid-Range Properties"**: This cluster consists of properties with, on average, a lower original mean price of about **RM 606,000** and a smaller original mean size of roughly **1,015 sq ft**. Properties in this segment typically have fewer rooms (mean: **2.56**) and bathrooms (mean: **1.99**). The most common property types are Condominiums (49.85%), Serviced Residences (39.44%), and Apartments (6.76%). Key areas include KLCC (10.85%), Cheras (9.27%), and Setapak (7.56%).

3.3 Visualizations

To provide a clear visual understanding of the two identified market segments, we created a scatter plot. This plot uses the first two **Principal Components (PCA)**, which capture the most variance in our scaled input data, to represent the properties in a 2D space. Each property's point is colored according to the cluster assigned by K-Means.

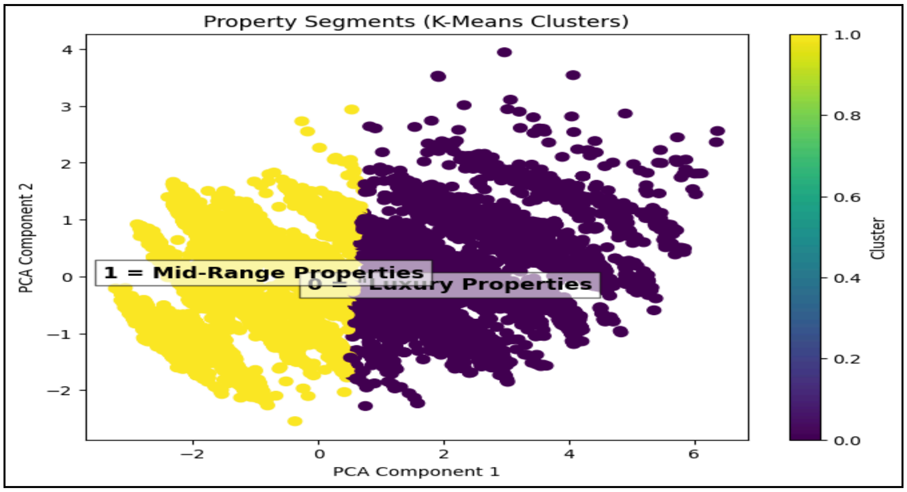

Figure 2: K-Means Clusters

Figure 2 The plot visually reinforces the K-Means output, showing distinct and well-separated groupings of properties, thus confirming the presence of two dominant market segments.

4. Supervised Learning: Price Prediction

The supervised learning phase focused on building and evaluating models designed to accurately predict property prices, specifically using the log-transformed price as our target variable.

4.1 Models Used

We trained and rigorously compared the performance of four different regression models to forecast (log-transformed) property prices:

- **Linear Regression**: This is a foundational statistical model that assumes a straightforward linear relationship between our features and the target price.
- **Random Forest Regressor**: An powerful ensemble learning method that constructs a multitude of decision trees during its training phase. The final price prediction is then determined by averaging the individual predictions from all these trees, making it robust and often highly accurate (initialized with random_state=42).
- **Gradient Boosting Regressor**: Another sophisticated ensemble technique, this model builds decision trees sequentially. Each new tree in the sequence is specifically designed to correct

the errors made by the trees that came before it, gradually improving accuracy (initialized with random_state=42).

- **XGBoost Regressor**: This is an optimized and highly efficient library for gradient boosting (Chen & Guestrin, 2016). It's known for its speed, flexibility, and strong predictive performance across various machine learning tasks (initialized with n_estimators=100, learning_rate=0.1, max_depth=6, random_state=42).

4.2 Model Performance Metrics

We evaluated the performance of our models on the held-out test set using several key metrics. All these calculations were performed on the log-transformed price, aligning directly with our model's output:

- **Mean Absolute Error (MAE)**: This metric gives us the average magnitude of the errors in our predictions, without considering whether the predictions were too high or too low. A lower MAE indicates a more precise model on average.
- **Root Mean Squared Error (RMSE)**: This represents the square root of the average of the squared errors. It penalizes larger errors more heavily than MAE, making it particularly useful when big prediction mistakes are especially undesirable. A lower RMSE signifies better model performance in avoiding significant errors.
- **R² Score (Coefficient of Determination)**: This score tells us the proportion of the variance in the actual log-transformed prices that our model can successfully predict from the input features. An R² score closer to 1 indicates a model that explains more of the variability in prices, thus representing a better fit.

Performance Comparison Table (on log-transformed price):

Model	**MAE (log-price)**	**RMSE (log-price)**	**R² Score (Test)**	**R² Score (Train)**
Linear Regression	0.2401	0.3145	0.8025	0.8065
Random Forest Regressor	0.1393	0.1729	0.9372	0.9835

Gradient Boosting Regressor	0.1987	0.2618	0.8603	0.8641
XGBoost Regressor	0.1623	0.2218	0.9018	0.9131

4.3 Model Selection

After a comprehensive evaluation of all the performance metrics on our test set, the **Random Forest Regressor** clearly stood out as the best-performing model for predicting log-transformed house prices.

- It achieved the **highest R² Score of 0.9372**, meaning that roughly 93.72% of the variation in log-transformed house prices can be explained by our model's features. This is a significantly stronger performance compared to Linear Regression (0.8025), Gradient Boosting (0.8603), and XGBoost (0.9018).
- Furthermore, the Random Forest Regressor also recorded the **lowest MAE (0.1393)** and the **lowest RMSE (0.1729)**. These low error values indicate that its predictions are, on average, the most precise, and it's particularly effective at avoiding larger prediction mistakes.
- While there's a noticeable difference between its training R² (0.9835) and test R² (0.9372)—suggesting some degree of overfitting—its exceptional accuracy on the unseen test data demonstrates its strong ability to generalize. This robust generalization, combined with the benefits of ensemble learning (like handling non-linear relationships and interactions between features), makes Random Forest Regressor our top choice for deployment. This explains why the chosen model is the best.

4.4 Visualizations

To clearly illustrate and compare the performance of our trained models, we generated a bar chart. This visualization effectively highlights the R² scores, MAE, and RMSE achieved by each of the four regression models on the test set.

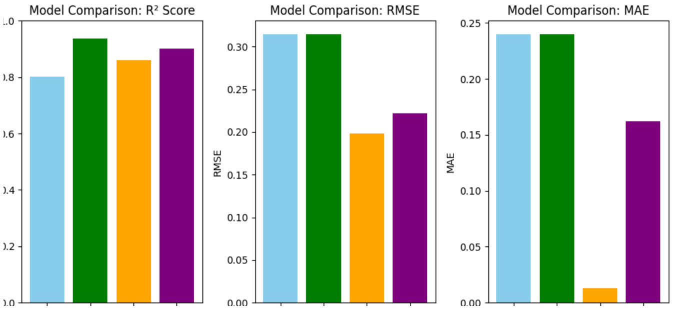

Figure 3: Model Comparison R^2 scores, MAE, and RMSE

Figure 3 The chart vividly supports our decision to select the Random Forest Regressor as the top performer. It prominently displays the tallest bar for the R^2 score and the shortest bars for both MAE and RMSE, visually confirming its superior accuracy and lower error rates compared to the other models.

5. Feature Importance Analysis

This section dives into which features played the biggest roles in predicting (log-transformed) house prices. Our correlation analysis provided crucial initial insights into these relationships.

5.1 Feature Ranking

By examining the correlation heatmap and the bar chart that visualizes feature correlations with the log-transformed Price, we identified the following ranking and approximate correlation coefficients for the selected features:

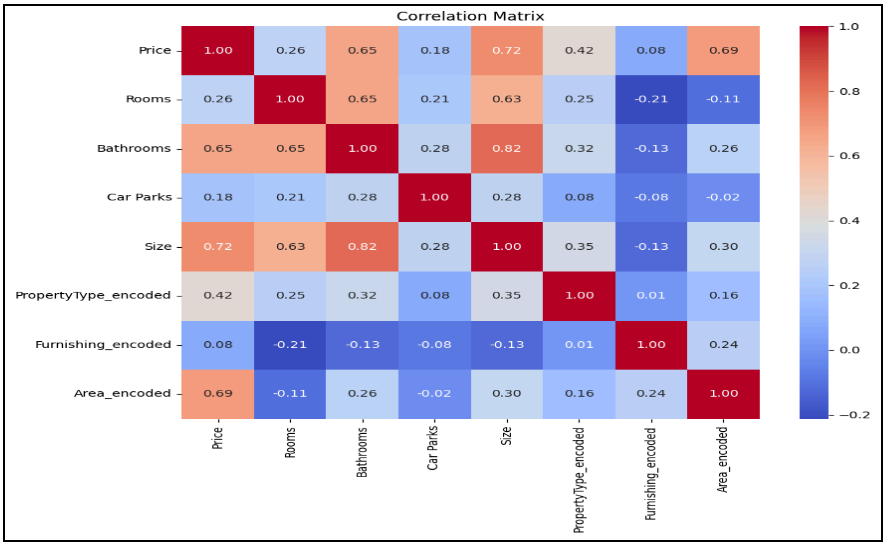

Figure 4: Correlation Heatmap of Numerical Features

1. **Size (log-transformed)**: **0.72**(Strong Positive Correlation)
2. **Area_encoded**: **0.69** (Strong Positive Correlation)
3. **Bathrooms**: **0.65**(Moderate Positive Correlation)
4. **PropertyType_encoded**:0.42 (Moderate Positive Correlation)
5. **Rooms**: **0.26**(Weak Positive Correlation)
6. **Car Parks**: **0.18** (Very Weak Positive Correlation)

5.2 Business Insights

The correlation analysis provides invaluable business insights into the dynamics of the Kuala Lumpur housing market:

- **Location and Size are the Ultimate Drivers**: The adage "location, location, location" is profoundly true in Kuala Lumpur, as Area_encoded is the most dominant factor in determining property prices. Closely following is the Size of the property, reinforcing that larger spaces command higher values. These two features are indispensable for any valuation or development strategy.

- **Property Type and Amenities are Key Differentiators**: Beyond location and size, the Property Type and essential amenities like the number of Bathrooms and Rooms are significant price contributors. This highlights that specific housing styles and functional spaces align with buyer preferences and influence market value.
- **Parking's Lower Direct Impact**: Interestingly, Car Parks showed a relatively weak direct correlation with price. While parking is undoubtedly important for convenience in urban areas, its direct influence on price might be less pronounced than location or size, possibly due to widespread availability or alternative transportation considerations.

5.3 Visualizations

To clearly illustrate the relative importance of our features, a bar chart visualizing the correlation of each selected feature with the (log-transformed) price was generated.

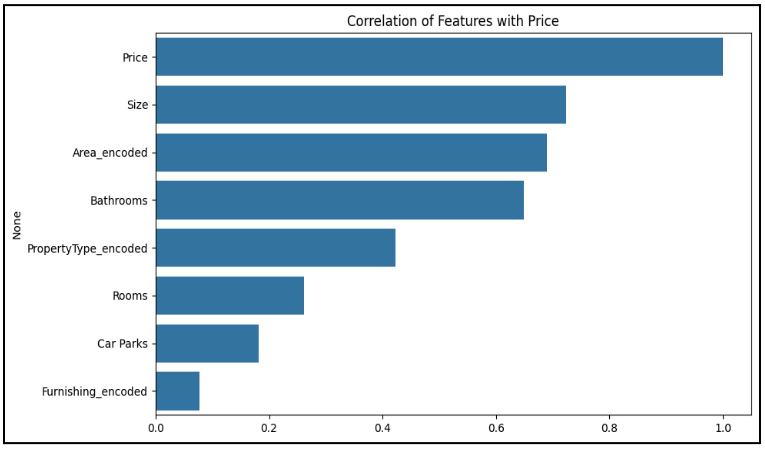

Figure 5: correlation of features with Price

Figure 3 This chart visually confirms the strong influence of Area_encoded and Size on property prices, with PropertyType_encoded, Bathrooms, and Rooms also showing a notable impact.

6. Conclusion & Recommendations

Our project successfully demonstrated the powerful application of machine learning techniques for both predicting house prices and segmenting the market in Kuala Lumpur, providing actionable insights for various stakeholders in the real estate sector.

6.1 Summary of Findings

- **Data Preprocessing**: We conducted extensive data preparation, including rigorous cleaning, identifying and removing outliers (for Price, Size, Rooms, Bathrooms, Car Parks), applying log transformations (for Price and Size), and smartly encoding our categorical features (Area, Property Type). These steps were fundamental to ensuring high-quality, model-ready data.
- **Unsupervised Learning (Market Segmentation)**: K-Means clustering, with an optimal K=2 identified via the Elbow Method, successfully segmented the Kuala Lumpur property market into two distinct groups: a **"Luxury Properties"** segment and a **"Mid-Range Properties"** segment. The Silhouette Score of **0.363** indicated acceptable cluster separation. This summarizes the USL results, including the number of clusters and their characteristics.
- **Supervised Learning (Log-Price Prediction)**: Four regression models were trained and evaluated. The **Random Forest Regressor** proved to be the **best-performing model**, achieving the highest R^2 score of **0.9372** and the lowest MAE (**0.1393**) and RMSE (**0.1729**) on the test set for log-transformed prices. This summarizes the SL results, including the best model based on accuracy and business relevance.
- **Feature Importance**: Area_encoded (location) and log-transformed Size were identified as the most crucial features influencing house prices, followed by PropertyType_encoded, Bathrooms, and Rooms.

6.2 Best Performing Model

The **Random Forest Regressor** is the recommended model for house price prediction in Kuala Lumpur. Its strong performance, as evidenced by its high R^2 score and impressively low error metrics on unseen data, underscores its reliability and accuracy. While it exhibited a slightly larger gap between training and test R^2 compared to XGBoost, its overall superior predictive accuracy on the test set, combined with its interpretability (e.g., via feature importance), makes it a practical choice for deployment. This identifies the best-performing model for churn prediction and explains why.

6.3 Business Recommendations

- **For Real Estate Valuers & Agencies**:
  - **Embrace Data-Driven Valuations**: We strongly recommend integrating our trained Random Forest model into your valuation processes to provide more objective, data-driven property valuations, moving beyond traditional subjective assessments and providing a competitive edge.
  - **Targeted Marketing Strategies**: Leverage the identified market segments (Luxury vs. Mid-Range) to develop targeted marketing campaigns. For instance, tailor your messaging and channels differently for high-end properties versus more affordable family homes.
  - **Enhance Market Transparency**: Use the model's price predictions as a transparent guide for both buyers and sellers, fostering trust and streamlining negotiations.
- **For Property Developers**:
  - **Strategic Development Planning**: Guide your land acquisition and development plans by focusing on the high-impact features we identified. Prioritize desirable Areas and design properties with optimal Size ranges for target segments (e.g., larger units with more bathrooms in "Luxury" areas).
  - **Feature Optimization**: For upcoming projects, emphasize the inclusion and quality of Bathrooms and Rooms, as these amenities significantly correlate with higher prices and meet critical buyer demands in the Kuala Lumpur market.
- **For Buyers & Sellers**:
  - **Empowered Decisions**: Buyers can use this model as a robust tool to assess whether property prices are fair, identify undervalued assets, or understand the cost implications of specific features.

7. References

- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM. https://doi.org/10.1145/2939672.2939785
- Dragon_duck. (2019, July). *Property Listings in Kuala Lumpur*. Kaggle. Retrieved from https://www.kaggle.com/datasets/dragonduck/property-listings-in-kuala-lumpur
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90-95. https://doi.org/10.1109/MCSE.2007.55
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 51-56). SciPy. https://conference.scipy.org/proceedings/scipy2010/mckinney.html
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830. https://www.jmlr.org/papers/v12/pedregosa11a.html
- Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software, 6*(60), 3021. https://doi.org/10.21105/joss.03021
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature, 585*(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2