The dataset used in this study was sourced from Kaggle and includes thousands of residential properties from across Saudi Arabia. SQL queries were used to filter the data down to only Riyadh, with irrelevant columns removed and the focus narrowed to apartments and villas — the most common residential types in the city. After filtering and cleaning, 128,121 records remained, representing a robust and diverse sample for predictive modeling.

The predictive model used various carefully selected features, including property size in square meters, the number of rooms, living spaces, bathrooms, property type (apartment or villa), neighborhood, street width, and precise geographic coordinates. Additional coordinates for the nearest services (schools, hospitals, parks, grocery stores) were included. These coordinates were sourced from OpenStreetMap (OSM), an open-source geospatial platform widely used for urban and geographic analysis. The Python library osmnx was employed to retrieve service locations within Riyadh’s administrative boundaries using specific OSM tags such as amenity=school for schools, amenity=hospital for hospitals, leisure=park for parks, shop=supermarket for grocery stores, and shop=mall for malls. Each type of service was downloaded as a set of geographic points (latitude and longitude), and for every residential property in the dataset, the distance to each point was computed using the Haversine formula. This method calculates the spherical distance between two coordinates, capturing real-world proximity without the need for detailed road networks or routing data. After computing all distances, only the nearest point of each service type was retained for every property, and both the actual distance in meters and the coordinates of the closest service point were added to the dataset. As a result, new numerical features such as distance\_to\_school, distance\_to\_hospital, distance\_to\_park, distance\_to\_grocery, and distance\_to\_mall were created, along with corresponding latitude and longitude fields. These spatial variables added meaningful location-based context to the model, enhancing its ability to account for the real-world impact of service proximity on property values.

Distances were calculated using the Haversine formula, offering high accuracy based on latitude and longitude. These distances were added to the model as numerical features, considering their potential impact on property value.

The preprocessing phase began with handling missing values using the mode — a suitable statistical method for categorical columns with frequent values — preserving structural consistency without introducing bias. Next, outliers were addressed, particularly in price, area, and street width columns. Descriptive analysis and visual exploration revealed skewed distributions, justifying the use of IQR-based filtering to detect out-of-range values.

However, the traditional Tukey method (1.5 \* IQR) was not used due to its assumption of symmetrical distributions, which was not observed in this dataset. Each column was treated individually, with IQR thresholds adjusted based on distribution shape to avoid removing valid but extreme data. This approach is supported by literature from The AAPS Journal: “The approach suggested by Tukey (1977)... assumes an approximately symmetrical distribution, which is not the case for skewed CP datasets.” Similarly, Hubert and Vandervieren proposed modified boxplots with skew-adjusted whiskers, providing more accurate bounds for non-symmetric data distributions [6][7].

Following this logic, upper outliers were removed, while lower outliers were imputed using a spatially aware method with the cKDTree algorithm. Missing values for price and area were replaced with those from the nearest property of the same type in the same neighborhood. For street width, nearest values were used without distinguishing between property types. This minimized data loss and maintained logical consistency without introducing bias.

Categorical encoding presented a challenge with the "neighborhood" column, which had over 150 unique values. One-hot encoding would have drastically increased dimensionality, so Frequency Encoding was used instead, converting neighborhoods into numerical values based on frequency. The "city" column, which only contained “Riyadh,” was removed due to lack of predictive value.

Although tree-based models like XGBoost and Gradient Boosting do not require feature scaling, normalization was tested experimentally. A study in IEEE Access found that “RF and XGBoost generate similar performance with both scaling techniques, with slight advantages when standardized.” Hence, normalization was included for consistency and to support future models relying on distance-based metrics like SVM and k-NN [8].

The modeling phase followed a staged approach. The first experiment tested whether proximity alone could predict prices. Models were trained using only distance features (to nearest school, hospital, mall, park, grocery). Internal features were excluded to isolate spatial effects. However, results were weak, with R² not exceeding 0.50 and high RMSE, indicating proximity alone is insufficient.

The second experiment combined spatial and internal features: property type, room counts, area, street width, encoded neighborhood, and distances. This significantly improved performance, with R² rising to 0.91 and RMSE dropping to 241,665 — appropriate for Riyadh’s dynamic market.

Three algorithms were compared: Random Forest, Gradient Boosting, and XGBoost. XGBoost outperformed the others in accuracy and speed, handling imbalanced distributions more effectively. Final model parameters included:  
n\_estimators=300, max\_depth=10, learning\_rate=0.1, subsample=0.8, colsample\_bytree=0.8, min\_child\_weight=2, and random\_state=42.

To ensure model stability, 5-fold cross-validation was used. The model achieved 97% training accuracy and 92% testing accuracy, with minimal overfitting — acceptable in applied research contexts involving noisy and diverse data.

Numerical features were normalized using StandardScaler, and log-transformation was applied to the price variable to reduce skewness, then reversed with exp() after prediction. To interpret the model’s decisions, SHAP (SHapley Additive exPlanations) was used. The most influential factors were property type, park distance, area, neighborhood encoding, and mall distance, with lesser influence from school and hospital distance, street width, and room count. SHAP provided interpretable outputs, even for non-technical users.

Finally, a user-friendly web interface was developed using Streamlit. Users can input property details — type, size, neighborhood, facilities — and immediately view a predicted price. This integration of AI with an interactive UI offers a practical decision-support tool for buyers, investors, and regulators, backed by interpretable, data-driven insights.

**Dataset**

The dataset used in this study was sourced from Kaggle and initially included thousands of residential properties from various cities across Saudi Arabia. SQL queries were applied to extract only the records relevant to Riyadh, and all irrelevant columns were removed. The focus was narrowed down to apartments and villas, as they represent the most common residential property types in the city. After the filtering and data cleaning processes were completed, 128,121 records remained, providing a robust and diverse sample for predictive modeling. These records covered a wide range of property features, locations, and price levels, offering a solid foundation for developing an accurate and generalizable machine learning model.

**Geospatial Feature Engineering**

The model used a carefully selected set of internal features including property size in square meters, number of rooms, number of living spaces, number of bathrooms, property type (apartment or villa), neighborhood, street width, and precise geographic coordinates. In addition to these structural features, I manually collected external geographic data to represent the property's proximity to key services. I used OpenStreetMap (OSM), an open-source geospatial platform widely utilized for urban and spatial analysis. Through the Python library osmnx, I extracted service locations within Riyadh’s administrative boundaries using specific OSM tags such as amenity=school for schools, amenity=hospital for hospitals, leisure=park for parks, shop=supermarket for grocery stores, and shop=mall for malls. Each type of service was downloaded as a set of geographic points with latitude and longitude.

For every property, I computed the distance to each point of every service category using the Haversine formula, which calculates spherical distance between two coordinates on Earth. This method offered a reliable approximation of proximity without requiring road network data. After calculating all distances, only the nearest point for each service type was retained. I stored both the shortest distance in meters and the exact coordinates of each nearest service point. As a result, the dataset was enriched with new numerical features such as distance\_to\_school, distance\_to\_hospital, distance\_to\_park, distance\_to\_grocery, and distance\_to\_mall, along with their corresponding latitude and longitude fields. These location-based features provided important spatial context that significantly enhanced the model’s ability to capture real-world influences on property value.

**Preprocessing**

Once the feature set was complete, I began the preprocessing stage by handling missing values. For categorical variables with frequent values, such as property type or neighborhood, I used the mode to fill in missing entries. This method helped preserve structural consistency without introducing statistical bias. I then addressed outliers, focusing on columns like price, area, and street width—variables that have a strong impact on property value and are prone to manual entry errors. Upon conducting descriptive analysis and visual inspections, I observed skewed distributions, particularly with property prices showing heavy tails.

To detect and treat outliers, I applied the Interquartile Range (IQR) method. However, I did not use the traditional Tukey approach (1.5 \* IQR), since it assumes symmetric distributions, which was not the case for my data. Instead, I treated each column individually and adjusted the thresholds based on the actual shape of the distribution. This decision was informed by studies such as those published in *The AAPS Journal*, which noted that the Tukey method is not suitable for skewed data, and by the methodology proposed by Hubert and Vandervieren who introduced skew-adjusted boxplots for more accurate outlier detection.

Following this adjusted IQR-based approach, I removed extreme upper outliers. For lower outliers or missing values in columns like price and area, I used a spatial imputation method involving the cKDTree algorithm. This allowed me to find the nearest property of the same type and in the same neighborhood to impute missing values contextually. For missing values in street width, I used the nearest value regardless of property type, since this feature is relatively independent. These spatially informed techniques allowed me to minimize data loss while maintaining logical consistency across the dataset.

**Encoding and Normalization**

I faced a significant challenge when encoding the “neighborhood” column, which included over 150 unique values. Using one-hot encoding would have dramatically increased the dimensionality of the dataset, which could have resulted in model inefficiency and overfitting. To overcome this, I applied frequency encoding, converting each neighborhood to a numeric value based on its occurrence in the dataset. I also removed the “city” column, which only contained “Riyadh,” as it added no predictive value.

Although models like XGBoost and Gradient Boosting are generally robust to unscaled data, I decided to experiment with normalization. According to a study published in *IEEE Access*, both Random Forest and XGBoost yield similar performance with or without normalization, but with slight improvements when standardized. Therefore, I applied normalization using the StandardScaler from scikit-learn to enhance consistency and allow future compatibility with distance-based algorithms like SVM and k-NN.

**Modeling Phase**

The modeling process consisted of two experimental phases. In the first phase, I tested whether spatial proximity alone could predict property prices. I trained models using only the distance-based features and excluded all internal features such as area and number of rooms. This allowed me to isolate the effect of location and proximity. The results showed that proximity alone was not sufficient. The R² values remained below 0.50, and RMSE values were high, indicating weak predictive power.

In the second phase, I combined both internal and spatial features. I included variables like property type, number of rooms, area, street width, encoded neighborhood, and all distance-based features. This significantly improved the model’s performance. The R² value rose to 0.91, and the RMSE dropped to 239,548.17

, which is considered a strong result for a complex and varied market like Riyadh. These results confirmed that combining structural and spatial data yields much better prediction accuracy than relying on either type alone.