**PREDICTIVE MODELLING OF HOUSE PRICES**

***A comprehensive report on machine learning project***

***Prepared by: Oladokun Sunday Obasanjo***

**PROJECT OVERVIEW**

In this capstone project, I applied my data science and machine learning skills to develop a predictive model for house prices. This project involved several key steps, including data cleaning, exploratory data analysis (EDA), feature engineering, model training, and evaluation. By leveraging a real-world dataset, I built a model designed to accurately predict house prices based on a range of features. The goal was to create a robust predictive model that provides valuable insights into the factors influencing property values.

**PROBLEM STATEMENT**

In the real estate market, a lot of factors can affect property values, making it tough to accurately predict house prices. These factors are inclusive of the physical elements relating to the property such as its size, location and state as well as external variables that comprise market trends and economic conditions. The complexity stems from these variables, which interact with one another differently and affect property prices in different proportions (Kiran, 2019; Tiwari, 2018). For example, the number of bedrooms in a house may adjust its price range significantly along with other elements like the overall condition of the house in question or neighborhood facilities (Fayyaz et al., 2020).

Accurate predictions of house prices can offer substantial benefits across multiple stakeholders in the real estate sector. For buyers, precise price predictions can facilitate more informed purchasing decisions, ensuring that they are not overpaying for a property. Sellers can benefit from understanding the fair market value of their property, which helps in setting a competitive price that can attract potential buyers while maximizing their return. Real estate agents gain from accurate price predictions by being able to provide better advice to their clients and streamline the buying and selling process. Financial institutions and investors also rely on accurate property valuations to make sound investment decisions and assess the risk associated with mortgage loans (Chien & Ding, 2013; Li et al., 2021).

This project aims to address the challenge of accurate property valuation by developing a robust machine learning model for predicting house prices. By leveraging various features of the houses, such as their physical characteristics, location, and condition, the project seeks to build a model that can effectively capture the complex relationships between these features and the sale price. The model will be trained and evaluated using a comprehensive dataset, enabling it to learn patterns and make accurate predictions based on historical data. Through this approach, the project intends to provide a reliable tool for stakeholders in the real estate market to better understand and predict property values.

**PROJECT OBJECTIVE**

***General Objective:***

The primary objective of this project is to accurately predict house prices using machine learning models and identify the most effective approach for achieving this goal.

***Specific Objectives:***

Data Cleaning and Exploratory Data Analysis (EDA): Prepare and understand the dataset through thorough cleaning and analysis.

Feature Engineering and Encoding: Enhance the model's predictive capabilities by engineering features and encoding variables appropriately.

Model Training, Evaluation, and Optimization: Train, evaluate, and optimize multiple machine learning models to determine the best-performing one for predicting house prices.

**METHODOLOGY**

***Data Collection:***

The dataset utilized for this project is titled "Housing Sales: Factors Influencing Sale Prices" from Kaggle. It is a comprehensive collection of housing sales data organized in a CSV (Comma-Separated Values) format. Each row represents a unique property sale, while the columns provide extensive information on various property attributes. These attributes include lot size, building type, house style, condition ratings, year of construction, amenities, and sale prices. This dataset is particularly useful for analyzing real estate market trends, understanding how property characteristics and location influence sale prices, and developing predictive models for housing sales. The dataset can be accessed [here](https://www.kaggle.com/datasets/rohit265/housing-sales-factors-influencing-sale-prices).

The size of the dataset is 18 features and 2,413 observations. Out of the 18 features contained in the dataset, 14 are numerical while 4 are categorical. The features are described below:

1. Lot\_Frontage: Linear feet of street connected to the property.
2. Lot\_Area: Lot size in square feet.
3. Bldg\_Type: Type of building (e.g., single-family, multi-family).
4. House\_Style: Style of the house (e.g., ranch, two-story).
5. Overall\_Cond: Overall condition rating of the house.
6. Year\_Built: Year the house was built.
7. Exter\_Cond: Exterior condition rating of the house.
8. Total\_Bsmt\_SF: Total square feet of basement area.
9. First\_Flr\_SF: First-floor square feet.
10. Second\_Flr\_SF: Second-floor square feet.
11. Full\_Bath: Number of full bathrooms.
12. Half\_Bath: Number of half bathrooms.
13. Bedroom\_AbvGr: Number of bedrooms above ground.
14. Kitchen\_AbvGr: Number of kitchens above ground.
15. Fireplaces: Number of fireplaces.
16. Longitude: Longitude coordinates of the property location.
17. Latitude: Latitude coordinates of the property location.
18. Sale\_Price: Sale price of the property (target feature for prediction).

***Data Cleaning:***

This phase of the project involved:

1. Typographical Error Correction: I corrected typographical errors for categorical features ('Bldg\_Type', 'House\_Style', 'Overall\_Cond', 'Exter\_Cond') by extracting unique values from the dataset using pandas and cross-checking them with the data description on Kaggle. All values were found to be in order.
2. Numerical Feature Validation: I performed a similar check for numerical features ('Lot\_Frontage', 'Lot\_Area', 'Year\_Built', 'Total\_Bsmt\_SF', 'First\_Flr\_SF', 'Second\_Flr\_SF', 'Full\_Bath', 'Half\_Bath', 'Bedroom\_AbvGr', 'Kitchen\_AbvGr', 'Fireplaces', 'Longitude', 'Latitude', 'Sale\_Price') to identify any unusual characters. These features were also found to be clean, indicating that the Kaggle dataset was relatively well-maintained.
3. Handling "Year Built": The "Year Built" feature is an integer, and converting it to a datetime format is unnecessary as it would default to January 1st of each year, which lacks precision. Instead, if needed, I can calculate the age of the houses by subtracting the year from 2024 and adding this information to the dataset later.
4. Data Quality Issues: There were no missing values or duplicate records, but I identified 561 outliers, which constitute a significant portion of the dataset. An analysis of variance (ANOVA) showed a significant difference between data with and without outliers (F-statistic: 17.5570, P-value: 2.8887e-05), suggesting that outliers could impact model performance. Consequently, I decided to work with two datasets: one including the outliers (df) and another with the outliers removed (df2\_without\_outliers). I will evaluate the performance of machine learning models on both datasets.

***Exploratory Data Analysis (EDA)***

This phase explored the different distributions and interactions with the target feature (Sales Price) in the dataset. The key decisions and approaches I used here are:

1. I worked with the df dataframe which is the dataset containing outliers as, I want to explore the dataframe with the outliers as it is a good practice so the outliers don’t affect the insights that would be generated.
2. I splitted the df dataset into training and test set before diving into EDA to prevent data leakage and ensure that the machine learning models have a test set they have not seen before.
3. The EDA was then carried out in 4 different phases- Non-graphical univariate analysis, Graphical univariate analysis, Non-graphical multivariate analysis, and Graphical Multivariate analysis.
   1. Non-graphical univariate analysis explored frequency distributions for categorical features and statistical summaries for numerical features. Skewness and unique values were also checked.
   2. Graphical univariate analysis explored histograms and boxplots for numerical features and bar plots for categorical features.
   3. Non-graphical multivariate analysis explored relationships between features and the target feature majorly to understand how much of an effect they have. Correlation analysis was carried out to get the co-efficients for each of the features. Groupby analysis was also carried and finally analysis of variance was carried out to check if the means of Sale\_Price across different categories of categorical variables are significantly different.
   4. Graphical multivariate analysis explored the use of pairplots to visualize the relationship between multiple features and the target variable (Sales Price). A correlation heatmap was created as well.
4. The major outcome of this section is majorly insights into the distribution of each of the features, how they affect the target feature, and finally which of them would be necessary for machine learning.

***Feature engineering***

After the EDA, new features were created that could enhance the performance of the models that would be used later on. The features that were created are House age and Decade the house was built. This was done for the test set and also the dataframe without outliers that would be utilized during machine learning phase. The new features were correlated with the target feature to see which one would be useful during modelling.

***Feature selection for machine learning***

This phase was used to select features that are necessary for my machine learning as a result of their effect on the target variable as established from the exploratory data analysis carried out. Three datasets were generated with only the features selected:

1. training\_ml\_df: The training set from the df with outliers
2. test\_ml\_df: The test set from the df with outliers
3. df2\_without\_outliers\_ml: Original dataframe with outliers dropped

The three dataframes above would be used for the machine learning phase.

***Machine Learning***

This phase explored three models across the two datasets created initially- the splitted dataset that has both train and test with outliers in them, the second dataset is the dataset without outliers.

The first step here was to encode all of the categorical variables in the dataset using OneHotEncoder for the nominal categorical variables and then using the map function for the ordinal categorical data. This step is very crucial because the models only work with numerical values

The three models I explored and the reasons are below:

1. Linear Regression:
   1. Simplicity and Interpretability: Linear Regression is straightforward and provides clear insights into the relationships between features and the target variable. This makes it easy to interpret how each feature influences house prices.
   2. Baseline Model: It serves as a good baseline model. If the Linear Regression model performs well, it can indicate that the relationships between features and prices are relatively simple and linear.
   3. Efficiency: It is computationally efficient and requires less processing power compared to more complex models.
2. Random Forest Regressor
   1. Handling Non-Linearity: Random Forest Regressor can model complex, non-linear relationships between features and the target variable, which is useful for capturing intricate patterns in housing data.
   2. Feature Importance: It provides insights into feature importance, helping to understand which attributes are most influential in predicting house prices.
   3. Robustness: It is less prone to overfitting compared to individual decision trees and can handle noisy data and missing values better.
3. XGBoost Regressor
   1. Performance: XGBoost is known for its high predictive accuracy and efficiency. It often performs better than other models due to its advanced boosting technique.
   2. Handling Large Datasets: It can handle large datasets and high-dimensional features effectively, making it suitable for complex housing datasets.
   3. Feature Engineering: XGBoost can automatically handle various feature interactions and is robust to overfitting, which enhances model performance and generalization.

The three models were used to train and test the data with and without outliers, they were then evaluated using root mean square error, mean squared error, and R2 scores.

Finally, scatter plots and bar charts were created to evaluate the best model

**RESULTS AND DISCUSSION**

***Categorical Variables Analysis***

1. Bldg\_Type:

The dataset reveals that "OneFam" is the most common building type, comprising 82.12% of the properties with 1,585 occurrences. This is followed by "TwnhsE" at 8.03% and 155 instances. The types "Twnhs" and "Duplex" make up 4.40% and 3.32% respectively, while "TwoFmCon" is the least common, representing only 2.12% of the dataset.

1. House\_Style:

"One\_Story" is the predominant house style, appearing in 49.64% of the cases with 958 instances. "Two\_Story" follows at 29.84% with 576 occurrences. Other styles like "One\_and\_Half\_Fin" account for 10.83%, while "Two\_and\_Half\_Fin" is the rarest at just 0.21%.

1. Overall\_Cond:

The majority of properties are rated "Average" in overall condition, representing 53.89% of the dataset. "Above\_Average" and "Good" follow at 19.07% and 15.18% respectively. "Very\_Good" comprises 5.18%, with "Poor" and "Very\_Poor" being quite rare, each below 0.2%.

1. Exter\_Cond:

The "Typical" exterior condition is predominant, covering 86.79% of the dataset with 1,675 instances. "Good" is the next most frequent at 10.47%, while "Fair" and "Excellent" are less common. "Poor" is exceedingly rare, appearing in only 0.05% of the cases.

***Descriptive Statistics***

1. Lot\_Frontage:

The average lot frontage is approximately 55 feet, with values ranging from 0 to 313 feet. The standard deviation of 33.82 feet indicates moderate variation in lot size.

1. Lot\_Area:

The mean lot area is about 10,051 square feet, with a broad range from 1,470 to 215,245 square feet. The high standard deviation of 8,520 square feet suggests considerable variability in lot sizes.

1. Year\_Built:

Most homes are constructed around 1970, with the data spanning from 1872 to 2010. The 25th percentile is 1954, and the 75th percentile is 1998, indicating a mix of older and more recent constructions.

1. Total\_Bsmt\_SF:

The average basement area is 1,028 square feet, with a standard deviation of 415 square feet. Basements range from 0 to 3,206 square feet, showing a wide variety in basement sizes.

1. First\_Flr\_SF:

The average first-floor area is 1,137 square feet, with a broad range from 334 to 3,820 square feet. This reflects significant differences in the size of first floors among homes.

1. Second\_Flr\_SF:

The mean second-floor area is 335 square feet. The high standard deviation of 424 square feet indicates many homes have no second floor or a small one.

1. Full\_Bath:

Homes typically have about 1.55 full bathrooms, with a range from 0 to 4. The standard deviation of 0.54 shows moderate variability in the number of full bathrooms.

1. Half\_Bath:

The average number of half bathrooms is 0.38, with most homes having none or one. The low standard deviation of 0.50 suggests consistency in the number of half bathrooms.

1. Bedroom\_AbvGr:

On average, homes have 2.86 bedrooms above ground, with a range from 0 to 6. The standard deviation of 0.81 reflects moderate variation in bedroom counts.

1. Kitchen\_AbvGr:

Most homes have just over one kitchen above ground, with a low standard deviation of 0.20, indicating a common kitchen count.

1. Fireplaces:

Homes average about 0.61 fireplaces, with a range from 0 to 4. The standard deviation of 0.64 reveals a wide variation in fireplace counts.

1. Longitude and Latitude:

Longitude values slightly vary around -93.64, and latitude values around 42.03, with narrow ranges and low standard deviations, indicating the dataset is geographically concentrated.

1. Sale\_Price:

The average sale price is $176,115, with significant variation (standard deviation of $72,495) and a range from $39,300 to $755,000. This indicates a broad disparity in property values.

***Distribution of Features***

1. Lot\_Frontage:

The distribution of lot frontage is nearly symmetric, with a skewness of -0.081 and kurtosis of 1.166, showing a balanced spread of values with slightly heavier tails.

1. Lot\_Area:

The distribution is highly positively skewed (skewness: 13.394) with heavy tails (kurtosis: 270.466), reflecting a concentration of smaller lot sizes and a few very large ones.

1. Year\_Built:

The distribution is slightly negatively skewed (skewness: -0.587), indicating a tendency for more recent construction. The kurtosis of -0.439 shows fewer extreme values.

1. Total\_Bsmt\_SF:

The distribution is moderately positively skewed (skewness: 0.456) with heavier tails (kurtosis: 1.716), indicating a wide range of basement sizes.

1. First\_Flr\_SF:

The first-floor area distribution is positively skewed (skewness: 1.043) with heavy tails (kurtosis: 2.257), reflecting a concentration of smaller areas and some larger outliers.

1. Second\_Flr\_SF:

The distribution is moderately positively skewed (skewness: 0.804) and slightly platykurtic (kurtosis: -0.559), indicating a prevalence of small second-floor areas.

1. Full\_Bath:

The distribution is slightly positively skewed (skewness: 0.245) and platykurtic (kurtosis: -0.570), with a small right tail and fewer extreme values.

1. Half\_Bath:

The distribution shows moderate positive skewness (skewness: 0.664) and platykurtic kurtosis (-1.163), indicating a concentration of fewer half bathrooms.

1. Bedroom\_AbvGr:

The distribution is nearly symmetric (skewness: 0.184) with slightly heavier tails (kurtosis: 1.472), showing a balanced range of bedroom counts.

1. Kitchen\_AbvGr:

The distribution is highly positively skewed (skewness: 4.682) with extremely heavy tails (kurtosis: 21.975), reflecting a majority of homes with one kitchen and a few with more.

1. Fireplaces:

The distribution is moderately positively skewed (skewness: 0.739) with near-normal kurtosis (0.138), showing variation in the number of fireplaces.

1. Longitude:

The longitude distribution is slightly negatively skewed (skewness: -0.338) and platykurtic (kurtosis: -0.973), indicating a left tail and fewer extreme values.

1. Latitude:

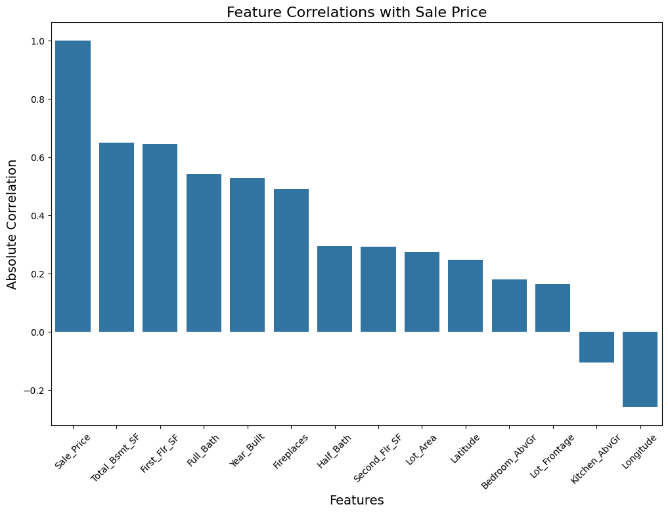
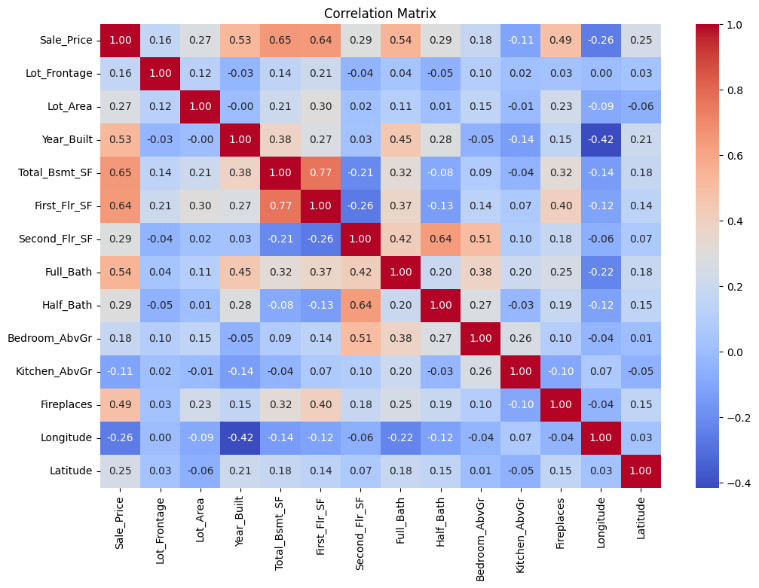
The latitude distribution is moderately negatively skewed (skewness: -0.508) with nearly normal kurtosis (-0.083), reflecting detailed location data with minor deviations.

1. Sale\_Price:

The sale price distribution is highly positively skewed (skewness: 1.745) with very heavy tails (kurtosis: 5.826), indicating a range of lower sale prices with some very high values.

***Correlation Analysis***

This section revealed the various relationship each of the features has with the target feature.



The correlation analysis highlights the relationships between the features and the target variable, Sale\_Price. Here’s a detailed examination of these correlations:

* Total\_Bsmt\_SF (0.6482): There is a strong positive correlation between basement area and sale price. Larger basement spaces tend to be associated with higher property values.
* First\_Flr\_SF (0.6448): The first-floor square footage also shows a strong positive correlation with sale price. This suggests that homes with larger first floors generally command higher prices.
* Full\_Bath (0.5410): The number of full bathrooms has a moderate positive correlation with sale price, indicating that properties with more full bathrooms are likely to have higher sale prices.
* Year\_Built (0.5284): The year a home was built is positively correlated with its sale price. Newer homes tend to have higher prices, reflecting modern features and updates.
* Fireplaces (0.4897): The number of fireplaces shows a moderate positive correlation with sale price. Homes with more fireplaces typically achieve higher sale prices.
* Half\_Bath (0.2926): The presence of half bathrooms has a weaker positive correlation with sale price. While there is a relationship, it is less significant compared to other features.
* Second\_Flr\_SF (0.2904): The size of the second floor has a weak positive correlation with sale price. Homes with larger second floors are somewhat more likely to have higher sale prices.
* Lot\_Area (0.2724): Lot size has a weak positive correlation with sale price. Larger lots are associated with higher property values, but the effect is less pronounced.
* Latitude (0.2455): The latitude of a property shows a weak positive correlation with sale price. This may reflect geographical variations in property values.
* Bedroom\_AbvGr (0.1784): The number of bedrooms above ground has a weak positive correlation with sale price. While more bedrooms generally contribute to higher prices, the effect is relatively minor.
* Lot\_Frontage (0.1631): Lot frontage has a weak positive correlation with sale price. Larger frontages are associated with higher values, but the relationship is not strong.
* Kitchen\_AbvGr (-0.1076): The number of kitchens above ground shows a weak negative correlation with sale price. This suggests that having more kitchens might slightly detract from the property's value, though the effect is minimal.
* Longitude (-0.2588): Longitude displays a weak negative correlation with sale price. This indicates that properties located further east might have lower prices, though the relationship is weak.

***Analysis of variance for categorical variables***

* Bldg\_Type: The very low p-value indicates statistically significant differences in Sale\_Price among building types. The type of building substantially influences property values.
* House\_Style: The p-value is extremely low, showing significant differences in Sale\_Price based on house style. House style significantly affects the sale price.
* Overall\_Cond: The low p-value confirms significant differences in Sale\_Price across overall condition ratings. The condition of a house strongly impacts its sale price.
* Exter\_Cond: The very low p-value suggests significant differences in Sale\_Price depending on exterior condition ratings. Better exterior conditions are associated with higher sale prices.

The above results and discussions led to the next stage which was to select features for the machine learning phase, all of the categorical variables were utilized as they have significant effect on the target feature (Sales Price).

The numerical features on the other hand are totally different as some of them didn’t correlate significantly with the target feature, those were dropped. The features dropped as a result of that were:

1. Latitude: Correlation of 0.245

2. Bedroom\_AbvGr: Correlation of 0.178

3. Lot\_Frontage: Correlation of 0.163

4. Kitchen\_AbvGr: Correlation of -0.108 (negative correlation)

5. Longitude: Correlation of -0.259 (negative correlation)

6. HouseAge: Correlation of -0.528 with `Sale\_Price` (moderate negative relationship)

The numerical features kept for machine learning:

1. Total\_Bsmt\_SF: Correlation of 0.648 (high positive correlation)

2. First\_Flr\_SF: Correlation of 0.645 (high positive correlation)

3. Full\_Bath: Correlation of 0.541 (moderate to high positive correlation)

4. Year\_Built: Correlation of 0.528 (moderate to high positive correlation)

5. Fireplaces: Correlation of 0.490 (moderate positive correlation)

6. Half\_Bath: Correlation of 0.293 (low positive correlation)

7. Second\_Flr\_SF: Correlation of 0.290 (low positive correlation)

8. Lot\_Area: Correlation of 0.272 (low positive correlation)

9. DecadeBuilt: Correlation of 0.526 with (moderate positive relationship)

***Machine learning models***

Linear Regression Model

1. With Outliers: This model performs reasonably well, with a high R² indicating that it explains 84.2% of the variance in the data. However, the presence of outliers slightly increases the MSE and RMSE, suggesting that these outliers negatively affect the model's performance by introducing more prediction errors.

2. Without Outliers: Removing outliers improved the model slightly, as indicated by the lower MSE and RMSE and a marginally higher R². The model is better at predicting values closer to the actual data without the influence of extreme outliers.

Forest Regression Model

1. With Outliers: This model performs slightly better than linear regression in handling outliers, as seen by a slightly lower MSE and RMSE and a marginally higher R². The model is more robust to outliers, thanks to the averaging of multiple decision trees.

2. Without Outliers: Removing outliers significantly improved the model's performance, as shown by a substantial reduction in MSE and RMSE and an increase in R². This indicates that the random forest model can achieve better accuracy when outliers are excluded from the dataset.

XGBoost Regressor Model

1. With Outliers: Similar to the random forest model, XGBoost handles outliers well, as reflected in comparable MSE, R², and RMSE scores. The performance is stable but does not significantly improve over random forest with outliers present.

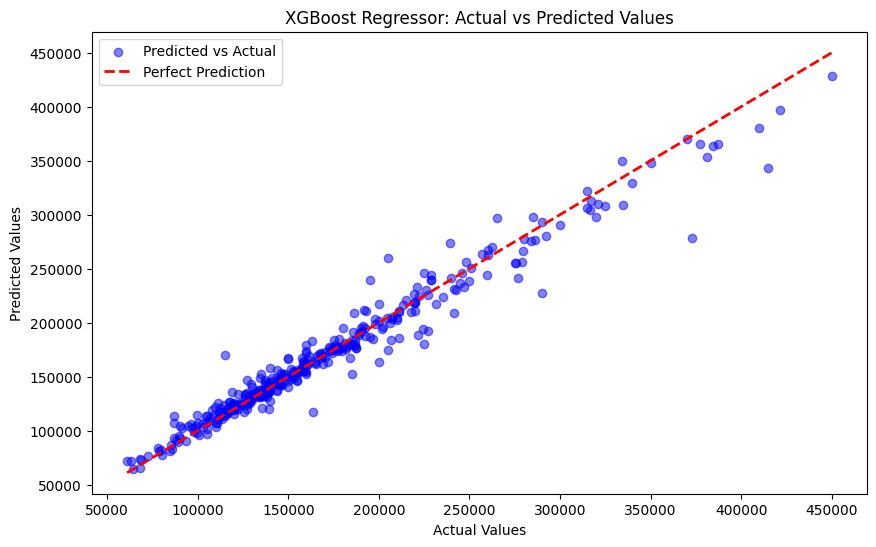
2. Without Outliers: XGBoost performs exceptionally well without outliers, with the lowest MSE and RMSE and the highest R² among all models. This suggests that XGBoost is the most effective model in this context, particularly when outliers are removed, as it captures the underlying patterns in the data more accurately.

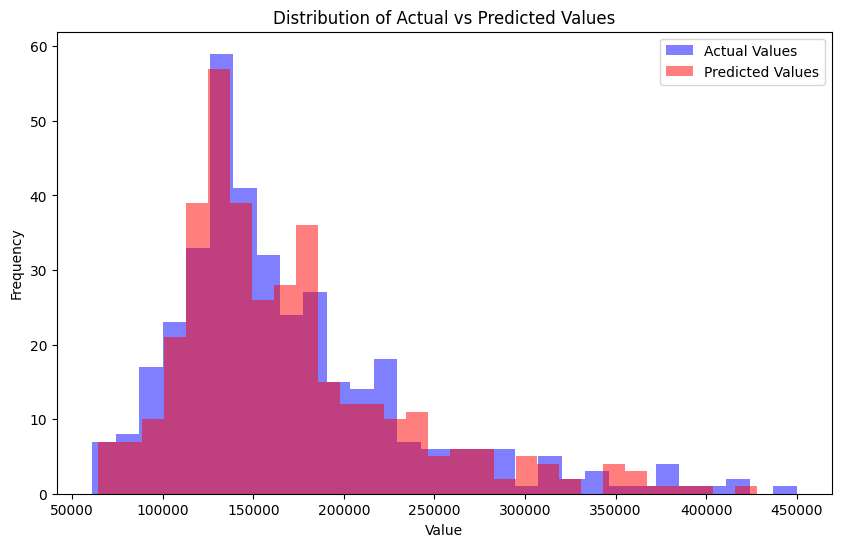
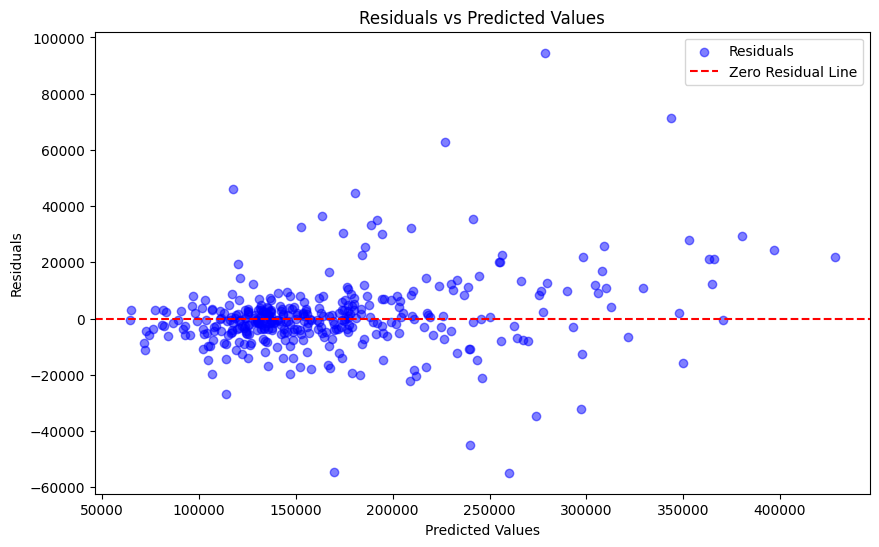
**CONCLUSION**

Summarily, the XGBoost Regressor emerged as the best performer in our analysis, particularly when trained on the dataset without outliers. This model demonstrated significantly lower errors, as indicated by the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), and exhibited superior explanatory power with a higher R-squared (R²) value.

The impact of outliers on model performance was substantial. Generally, removing outliers led to improved performance across all models. However, the most notable enhancement was observed in the XGBoost model, highlighting its sensitivity to outliers and its ability to benefit greatly from their removal.

In terms of model robustness, both the Random Forest and XGBoost models showed greater resilience to outliers compared to Linear Regression. Despite this, all models experienced performance gains when outliers were removed, underscoring the importance of outlier management in achieving accurate predictions.





**FURTHER RESEARCH**

* Feature Engineering: Explore advanced techniques to create or transform features for better predictive accuracy.
* Model Tuning: Optimize hyperparameters across models to improve performance further.
* Handling Non-linearity: Investigate methods like polynomial features or neural networks to better capture complex relationships in the data.
* External Data Integration: Incorporate additional data sources like economic indicators or market trends to enhance model accuracy.
* Spatial Analysis: Apply spatial analysis techniques to account for location-specific factors influencing house prices.
* Time-Series Analysis: Consider a time-series approach to analyze trends in house prices over time.

**REFERENCES**

Chien, C. F., & Ding, H. Y. (2013). Predicting the selling price of residential properties using a hybrid machine learning approach. International Journal of Information Technology & Decision Making, 12(4), 871-889.

Kiran, R. (2019). Real estate price prediction using machine learning algorithms. Journal of Engineering Research and Application, 9(1), 31-36.

Li, L., Wu, L., & Zhao, X. (2021). Predicting house prices using machine learning algorithms: A comparative study. Proceedings of the 2021 IEEE International Conference on Data Mining (ICDM), 234-243.

Tiwari, P. (2018). House price prediction using machine learning algorithms. International Journal of Computer Applications, 179(27), 1-7.