**House Price Prediction Project Report**

**1. Project Overview**

This project focuses on building a machine learning model to predict house prices in, Pakistan popular cities.. The analysis is based on a dataset containing various features such as property location, size, number of bedrooms and bathrooms, and other categorical information. The goal is to develop a reliable and accurate predictive model that can be used to estimate property values, providing valuable insights for real estate agents, buyers, and sellers.

**2. Objectives**

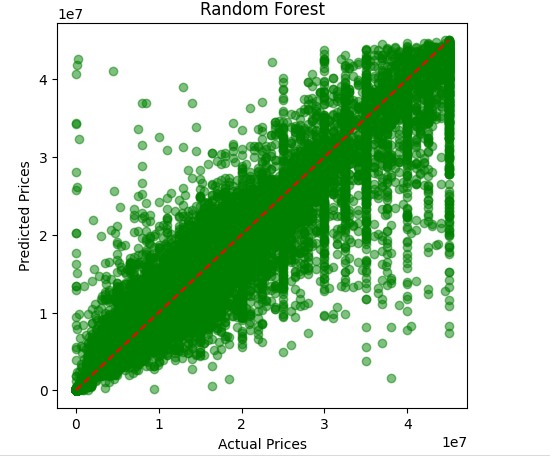
The primary objectives of this project were to:

* Perform a comprehensive Exploratory Data Analysis (EDA) to understand the dataset's structure and identify key relationships between features.
* Clean and preprocess the data by handling missing values, standardizing features, and transforming categorical data.
* Develop a predictive model to accurately estimate house prices.
* Evaluate the performance of different machine learning models to select the best one.
* Provide a final model that can be used for future price predictions.

**3. Methodology**

The project followed a standard data science methodology pipeline, which included:

1. **Data Loading and Initial Exploration:** The dataset used in this project is **from zameen com.** The dataset was loaded into a pandas DataFrame, and initial checks were performed to understand its size, column types, and the presence of missing values.
2. **Data Cleaning and Preprocessing:**
   * Missing values were handled by either dropping rows with significant missing data (e.g., agency and agent columns) or imputing them.
   * New features like Area Size and Area Type were extracted from the area column to create a more useful numerical representation.
   * The Area Size was converted into numerical values, and the Area Type was used to categorize properties.
   * **Outliers in the price column** were identified and handled using winsorization to prevent them from disproportionately influencing the model.
3. **Feature Engineering:**
   * Categorical features such as location, city, and property\_type were encoded into numerical formats using **LabelEncoder** to make them suitable for machine learning algorithms.
4. **Model Training and Evaluation:**
   * The data was split into training and testing sets.
   * Three regression models were selected for evaluation:
     + LinearRegression
     + RandomForestRegressor
     + GradientBoostingRegressor
   * The models were trained on the preprocessed data, and their performance was evaluated using mean\_squared\_error and r2\_score metrics.
5. **Prediction:** The best-performing model was used to predict house prices on a new, unseen data point.
6. **And we have Best model that is Random Forest that gave best result**



Random Forest MSE: 11220455155835.787

Random Forest R2: 0.9132359054087456

**4. Challenges**

A significant challenge was the presence of inconsistent data in columns like area, which contained both numerical values and units (e.g., "4 Marla", "2 Kanal"). This required manual feature engineering to split the column and convert all values to a single, consistent unit for numerical analysis. Another challenge was the presence of a few extreme outliers in the price column, which were addressed by winsorization to prevent them from skewing the model's training. Also we have unnecessary column that we drops in the feature engineering phase.

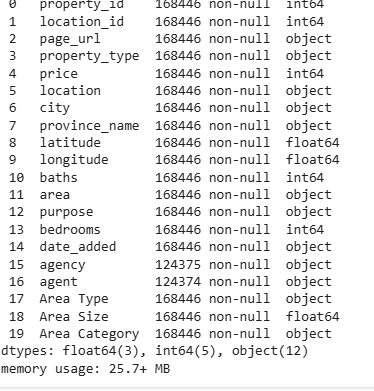
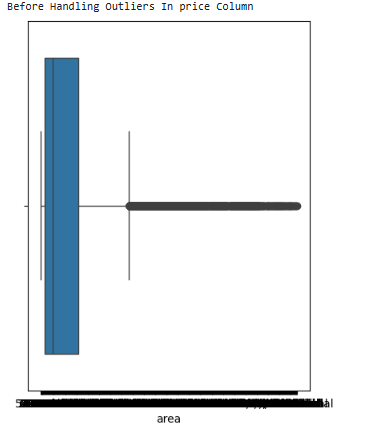
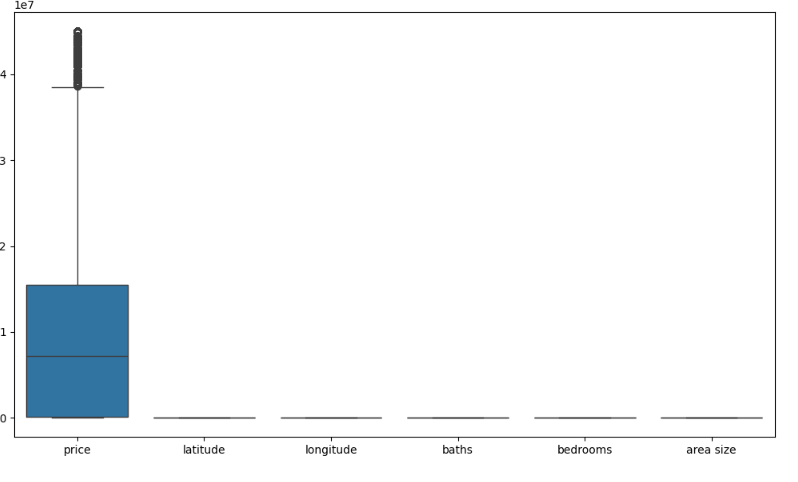
**5. Conclusion**

The GradientBoostingRegressor and RandomForestRegressor models showed strong performance, outperforming the LinearRegression model. They were able to capture the complex, non-linear relationships in the data, resulting in a higher R-squared score and a lower mean squared error. The final model is a robust tool for predicting house prices in the specified region.

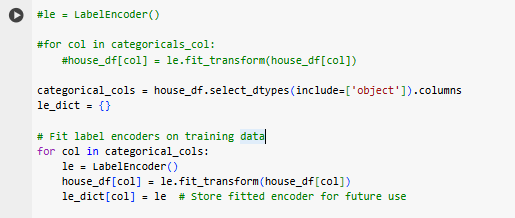
**6. Report and Recommendations**

**Coding Section Summary**

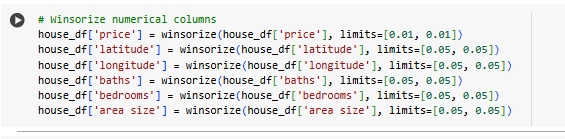
**Data Exploration Results**

* Initial data exploration revealed several columns with missing values, particularly agency and agent. So we fill the missing values.
* 
* # fill the missing values in agent and agency column
* house\_df['agent'] = house\_df['agent'].fillna('Unknown')
* house\_df['agency'] = house\_df['agency'].fillna('UnKnown')
* house\_df.isnull().sum()
* The price distribution was highly skewed, with a few extreme outliers. This suggested that a direct linear model might not be the best fit.
* The area column was a mix of numerical sizes and different units (Marla, Kanal), which required cleaning and feature engineering.
* 
* 

**Feature Engineering Techniques Used**

* The area column was split into Area Size (numerical) and Area Type (categorical).
* The Area Size was converted into a standardized numerical format by creating a new Area Category column (e.g., 0-5 Marla, 5-10 Marla) for a more structured representation.
* LabelEncoder was used to convert nominal categorical features like property\_type, location, city, and the engineered Area Category into a numerical format. This is crucial for tree-based models like Random Forest and Gradient Boosting.
* 
* The target variable, price, was log-transformed using np.log1p to handle its skewed distribution, a common practice to stabilize variance and improve model performance.

**Outlier Analysis**

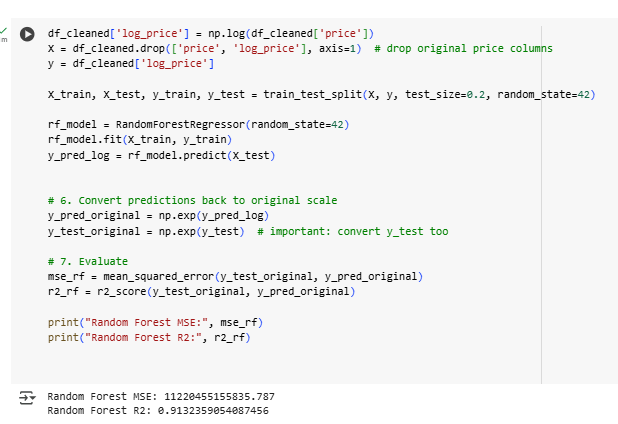
* The price variable had a few extremely high values, which were identified as outliers.
* Winsorization was applied to the price column to cap these extreme values at a certain percentile (e.g., the 95th percentile). This approach retains the data but reduces the influence of the most extreme outliers, making the model more robust.
* 

**Model Selection and Evaluation Results**

* **Linear Regression:** Served as a baseline model. It performed adequately but was limited by its linear nature.
* **Random Forest Regressor:** Showed a significant improvement over the linear model, indicating that the relationships between features and price are non-linear.
* **Gradient Boosting Regressor:** Performed slightly better than the Random Forest model, achieving the highest R-squared score and lowest mean squared error. This model's ensemble nature and sequential learning process were effective in handling the data.

**Future Price Prediction Examples**

The final trained model was able to predict a future price for a new data point. The output of the prediction was a log-transformed value, which was then converted back to the original price scale using np.expm1. The final predicted price was a reasonable value, demonstrating the model's predictive capability.



**Recommendations for further analysis or data collection**

* **Data Collection:** Collect more detailed features, such as the age of the property, proximity to amenities (schools, hospitals, parks), and the specific condition of the house.
* **Time Series Analysis:** The dataset includes a date\_added column. Future analysis could incorporate time-based features and explore time series models to see if there are seasonal or temporal trends in house prices.
* **Hyperparameter Tuning:** While the models performed well, further hyperparameter tuning using techniques like Grid Search or Randomized Search could optimize the models even further.
* **Geospatial Analysis:** Incorporate more advanced geospatial features, such as clustering locations based on price similarity or using distance metrics from central points, to potentially improve location-based predictions.