Building a Housing Price Prediction Model

Introduction:

In this project, the aim was to predict housing prices using machine learning techniques. The data for this project was scraped from ***Zameen.com***, a popular real estate website. Our focus is on cleaning the data, handling outliers, and building a predictive model. Here's a step-by-step guide to how I approached this project.

1. Understanding the Data

- **Source**: The dataset was scraped from Zameen.com.
- **Features**:
  - area_sq_ft: The size of the house in square feet.
  - num_of_bedrooms: The number of bedrooms in the house.
  - num_of_bathrooms: The number of bathrooms in the house.
  - price_in_rupees: The price of the house.

2. Data Preprocessing

Before building our model, we need to clean the data.

- **Handle Missing Values**:
  - In our dataset, 0 was used to represent missing values in the area_sq_ft, num_of_bedrooms, and num_of_bathrooms columns.
  - We replaced these 0s with NaN and dropped the entire row containing NaN values.

- **Merge Columns**:
  - We combined the num_of_bedrooms and num_of_bathrooms columns. This is because these two features are highly correlated.
  - Combining them helps to reduce multicollinearity, which improves model performance.

3. Handling Outliers

Outliers can significantly impact the performance of a machine learning model. To manage outliers, we used **Winsorization**:

- **Winsorization**:
  - Winsorization is a technique that limits extreme values in data to reduce the effect of possible outliers.
  - We applied Winsorization only to the area_sq_ft column to handle outliers in house sizes.

4. Building the Model

Now that our data is clean and outliers are handled, we can build our model:

- **Model Selection**:
    - First we applied Ridge Regression and XGBRegressor on our training data. XGBRegressor performed better in terms of Ridge Regression.

- **Pipeline Setup**:
    - We created a pipeline to streamline the preprocessing and modeling steps. The pipeline consists of:
        1. **Winsorization** of the area_sq_ft column.
        2. **Standardization** of the features using StandardScaler.
        3. **Modeling** with XGBRegressor, Ridge Regression.

5. Hyperparameter Tuning

To improve the model's performance, we used **Grid Search** to find the best value for the hyperparameters in XGBRegressor and Ridge Regression.

- **Best Value**:
    - The grid search helped us identify the best values for *max_depth*, *min_child_weight*, *learning_rate*, *n_estimators* that minimizes the error in predictions for XGBRegressor.
    - Similarly, grid search helped in finding the best values for *alpha* in Ridge Regression

6. Model Evaluation

Finally, we evaluated our model using the following metrics:

- **Mean Absolute Error**:

    - Ridge Regression: -0.3265
    - XGB Regressor: -0.27989

    XGB Regressor performed better than Ridge Regression in terms of MAE.

7. Conclusion

This project demonstrates how to build a robust housing price prediction model by:

- Cleaning and preprocessing the data.
- Handling outliers using Winsorization.
- Reducing multicollinearity by merging correlated features.
- Using XGB Regressor for prediction.