**Overview:**

The objective of this project was to build a predictive model that accurately estimates Airbnb listing prices using structured tabular data. We followed a five-step pipeline: Exploratory Data Analysis (EDA), Data Cleaning, Model Creation, Accuracy Optimization, and Final Evaluation. Find the repository link here

<https://github.com/wasif23ahad/AirBnB-Price-Predictions>

**Step 1: Exploratory Data Analysis (EDA)**

We began by exploring the structure and distribution of the dataset:

* **Data Inspection:** Loaded the dataset and removed irrelevant columns like id, host\_id, and img\_links. Also corrected column naming issues (e.g., toiles to toilets).
* **Missing Values:** Identified missing data in columns such as rating, reviews, host\_name, checkin, and checkout.
* **Type Conversion:** Converted string-based numeric fields (like rating, reviews) to proper numeric types.
* **Data Distribution:** Visualized distributions of numerical features (e.g., price, rating, bedrooms, bathrooms) using histograms and boxplots.
* **Correlation Analysis:** Used a heatmap to identify correlations among features.
* **Categorical Analysis:** Investigated how country and other categorical variables influence price using bar charts and boxplots.

*Reasoning:* This initial step helped identify data issues, discover trends, and understand feature relationships essential for predictive modeling.

**Step 2: Data Cleaning and Feature Preparation**

The dataset underwent systematic preprocessing:

* **Missing Value Handling:** Used SimpleImputer with median for numeric features and mode for categorical features.
* **Outlier Removal:** Removed extreme values from the price column using the IQR method to reduce skew and improve model robustness.
* **Encoding:** Applied one-hot encoding to convert categorical variables (e.g., country) into numeric form.
* **Feature Scaling:** Standardized numeric features using StandardScaler to optimize them for the neural network.

*Reasoning:* ANN models are sensitive to feature scale and noise, so cleaning and normalization are essential for model convergence and generalization.

**Step 3: Model Creation Using ANN**

We developed a custom Artificial Neural Network (ANN) using Keras with the following configuration:

* Input layer: Equal to the number of input features
* Two hidden layers: 128 and 64 neurons with ReLU activation
* Dropout layers for regularization
* Output layer: Single neuron for regression
* Optimizer: Adam with learning rate 0.001

The model was trained for 100 epochs with a batch size of 32 and yielded the following results:

* Test MSE: 35,204,568
* Test R² Score: 0.3770

*Reasoning:* ANN was chosen for its ability to learn complex non-linear relationships in high-dimensional data.

**Step 4: Accuracy Improvement**

To enhance model performance, we applied the following strategies:

* **K-Fold Cross Validation:** Applied 5-fold CV to assess model generalizability.
  + Mean R² Score: 0.3832, Standard Deviation: 0.0164
* **Hyperparameter Tuning:** Used GridSearchCV to test various combinations of epochs, batch sizes, and optimizers.
* **Ensemble Model (Random Forest):** Built a Random Forest Regressor as a benchmark.
  + R² Score: 0.3831

*Reasoning:* These steps provided insight into model consistency and helped fine-tune its learning parameters.

**Step 5: Advanced Accuracy Optimization**

Further improvements were achieved through advanced techniques:

* **Feature Engineering:** Introduced total\_rooms (sum of bedrooms, bathrooms, guests, beds) and room\_density (guests per bedroom).
* **Enhanced ANN Architecture:** Used 3 hidden layers (256, 128, 64), increased dropout regularization, lowered the learning rate to 0.0005, and trained for 120 epochs with batch size 16.
* **Gradient Boosting Regressor (GBR):** Implemented GBR for ensemble comparison. Performance was expected to exceed previous models.

*Reasoning:* Adding domain-specific features and refining model complexity helped extract better patterns from the data.