Homestay Price Prediction

Link to the Dataset: Homestays_Dataset

Download the **“Homestays_Data.csv”** which contains the dataset.
Note: The entire dataset has multiple columns and has more than 70,000 rows. Processing the following tasks on a Jupyter notebook on your local environment (your personal computer) might be time consuming due to limited processing power. We encourage you to use cloud-based tools such as **Google Colab**

Complete the following tasks:

Objective: Build a robust predictive model to estimate the `log_price` of homestay listings based on comprehensive analysis of their characteristics, amenities, and host information.

First make sure that the entire dataset is clean and ready to be used.

1. **Feature Engineering:** Enhance the dataset by creating actionable and insightful features. Calculate ‘Host_Tenure’ by determining the number of years from ‘host_since’ to the current date, providing a measure of host experience. Generate ‘Amenities_Count’ by counting the items listed in the `amenities` array to quantify property offerings. Determine ‘Days_Since_Last_Review’ by calculating the days between `last_review` and today to assess listing activity and relevance.

Tip: There are some similar amenities in the list, you can use the concept of string matching used in Task-1 for removing duplicate/similar strings.

2. **Exploratory Data Analysis:** You can refer to this **EDA Blog** to grasp the basics about EDA. Find out the underlying patterns in the dataset. Analyse how pricing (`log_price`) correlates with both categorical (such as `room_type` and `property_type`) and numerical features (like `accommodates` and `number_of_reviews`). Utilise statistical tools and visualisations such as correlation matrices, histograms for distribution analysis, and scatter plots to explore relationships between variables.

3. **Geospatial Analysis:** Investigate the geographical data to understand regional pricing trends. Plot listings on a map using `latitude` and `longitude` data to visually assess price distribution. Examine if certain neighbourhoods or proximity to city centres influence pricing, providing a spatial perspective to the pricing strategy.

Tip: You can use clustering algorithms like the ones used in Task-2 to find out log_prices in different locations (cities/localities).

4. **Sentiment Analysis:** Use sentiment analysis, to judge how the description is affecting the log_price. For sentiment analysis, it would be preferable to use the NLTK library. You can also build a sentiment analyzer on your own, but using a well known one is preferable. Use the sentiment score obtained to use it as a feature.

5. **Amenities Analysis:** Thoroughly parse and analyse the `amenities` provided in the listings. Identify which amenities are most associated with higher or lower prices by applying statistical tests to determine correlations, thereby informing both pricing strategy and model inputs.

6. **Categorical Data Encoding:** Use a Label-Encoder/One-Hot Encoder to obtain features suitable for model training. Apply it to categorical columns like ‘room_type’, ‘city’, ‘property_type’, etc.

7. **Model Training and Evaluation:**

    - Since many of you have not ventured into Deep Learning, we are limiting the models to be used to: **Random Forest, LGBM, XGBoost, Adaboost and CATBoost. (This is to ensure fairness in competition)**
    - Select the best correlating features out of the features available to ensure that the model can capture all the relevant and necessary information (you can use the SelectKBest function in scikit-learn).
    - Use for example: a split of 70-10-20 (train-val-test) for training the model (you can vary it according to yourself).
    - Tune the different hyperparameters to optimise the model and improve performance.
    - Critically evaluate the performance of the final model on a reserved test set (20% of the original data for example).
    - Metrics such as Root Mean Squared Error (RMSE) , F1 Score, R-squared to assess accuracy and goodness of fit should be **submitted**.

8. **Submission**

    Submit the jupyter notebook containing the code for the given problem statement. It should contain all the graphs, tables, etc. and whatever is required.

    Evaluation will be based on the quality of EDA, the scores associated with the model evaluation and presentation clarity.