Project Documentation: House Price Predictor

1. Summary

The **Bangalore House Price Predictor** is a machine learning-based web application designed to estimate property values in Bangalore, India. By analyzing historical housing data, the system provides accurate price valuations for "Ready to Move" properties. The application features a user-friendly web interface powered by **Flask** and utilizes a **Random Forest Regressor** for high-precision predictions.

2. Key Features

- **Real-Time Valuations:** Users can input specific property details (Location, BHK, Square Footage, etc.) to receive instant price estimates in both Lakhs and Crores.
- **Market-Relevant Filtering:** The system exclusively filters for "Ready To Move" properties to ensure predictions reflect the current active market rather than future developments.
- **Automated Model Training:** The application includes a self-healing mechanism that automatically retrains the machine learning model upon startup if no pre-existing model is detected.
- **Interactive Frontend:** A responsive, clean user interface built with HTML, CSS, and JavaScript allows for easy data entry and visualization of results.

3. System Architecture

3.1 Technical Stack

- **Language:** Python 3.8+
- **Web Framework:** Flask
- **Machine Learning:** Scikit-Learn (Random Forest Regressor)
- **Data Processing:** Pandas, NumPy
- **Frontend:** HTML5, CSS3, JavaScript

3.2 Directory Structure

The project is organized as follows:

- **app.py**: The core Flask application entry point. Handles routing, model loading, and prediction logic.
- **data/**: Contains raw datasets (Bengaluru_house_price_zoned.csv) and the data cleaning script (clean_data.py).
- **frontend/**: Houses the index.html interface and static assets (style.css, script.js).
- **requirements.txt**: Lists all necessary Python dependencies.

4. Installation & Setup Guide

Follow these steps to set up the development environment on a Windows machine.

Step 1: Configure PowerShell Permissions

To allow the activation of virtual environments, the execution policy must be adjusted for the current process.

Set-ExecutionPolicy Unrestricted -Scope Process

Step 2: Create Virtual Environment

Initialize a new Python virtual environment to isolate dependencies.

python -m venv venv

Step 3: Activate Environment

Activate the virtual environment to begin installing packages.

.\venv\Scripts\activate

Step 4: Install Dependencies

Install all required libraries listed in the requirements file.

pip install -r requirements.txt

Step 5: Launch Application

Start the Flask server.

(Note: If running for the first time, the model will automatically begin training).

python app.py

Step 6: Access Interface

Open your preferred web browser and navigate to the local server address:

http://localhost:5000

5. User Manual

How to Generate a Prediction

1. **Select Location:** Use the dropdown menu to search for and select a specific Bangalore neighborhood.
2. **Verify Zone/Area:** The "Area Type" and "Zone" fields will automatically populate based on the selected location.
3. **Input Property Details:** Enter the following specifications:
    - **BHK:** Number of bedrooms (e.g., 2, 3).
    - **Total Sq. Ft.:** The total area of the property.

- **Baths:** Number of bathrooms.
- **Balcony:** Number of balconies.

4. **Calculate:** Click the **"Calculate Valuation"** button.
5. **View Results:** The estimated price will appear in the result panel, displaying the value in Lakhs, Crores, and Price per Sq. Ft.

6. Technical Specifications

6.1 Data Pipeline

The system uses a dedicated cleaning script (clean_data.py) to process raw data. Key preprocessing steps include:

- **Unit Conversion:** Normalizing area ranges (e.g., "2100-2850") into single average values.
- **Outlier Removal:** Filtering out anomalies based on price per square foot and square footage per bedroom thresholds (minimum 300 sq. ft. per BHK).

6.2 Machine Learning Model

- **Algorithm:** Random Forest Regressor
- **Configuration:** 100 Estimators, Max Depth of 15.
- **Input Features:** Location, Area Type, Availability, Zone Name, Total Sq. Ft, Bath, Balcony, BHK.
- **Encoders:** Label Encoding is used for categorical variables (location, area_type, zone_name). Standard Scaling is applied to numerical inputs.