Satellite Imagery-Based Property Valuation

OVERVIEW

A Real Estate Analytics firm aims to improve its valuation framework by developing a **Multimodal Regression Pipeline** that predicts property market value using both tabular data and satellite imagery.

You are provided with historical housing data (including coordinates) and must programmatically acquire visual data to capture environmental context. The goal is to build a model that accurately values assets by integrating "curb appeal" and neighborhood characteristics (like green cover or road density) into traditional pricing models.

This project moves beyond standard data analysis by challenging you to combine two different types of data—numbers and images—into a single, powerful predictive system.

OBJECTIVE

- ***Build a multimodal regression model*** to predict property value (Target: Price).
- ***Programmatically acquire satellite imagery*** using latitude/longitude coordinates to capture visual environmental context.
- ***Perform exploratory and geospatial analysis*** to understand how visual factors (e.g., proximity to water, density) influence price.
- ***Engineer features*** using Convolutional Neural Networks (CNNs) to extract high-dimensional visual embeddings from the images.
- ***Test and compare fusion architectures*** (e.g., combining image data and tabular data at different stages) to find the most accurate method.
- ***Ensure Model Explainability*** by using tools like Grad-CAM to visually highlight the specific areas in the satellite imagery that influenced the model's price prediction.

DATASET

You will be working with a hybrid dataset created by you during the project.

- **Base Data (Tabular):**
  Source: train(1).xlsx , test2.xlsx
  Key Features: price (Target), bedrooms, bathrooms, sqft_living, lat (Latitude), long (Longitude).

- **Visual Data (Image):**
  You are required to use the lat and long columns from the base dataset to fetch satellite images for each property using an API.
  Suggested APIs: Google Maps Static API, Mapbox Static Images API, or Sentinel Hub.
- Dataset Description
  `sqft_living` **:** The total interior living space.
  `sqft_above` **:** The interior space **above ground level** (excluding the basement).
  `sqft_basement` **:** The interior space **below ground level**.
  Note: `sqft_living` = `sqft_above` + `sqft_basement` .
  `sqft_lot` **:** The total land area (lot size).
  `sqft_living15` **&** `sqft_lot15` **:** The average living and lot sizes of the **nearest 15 neighbors**.
  Significance: This captures the "neighborhood density." A big house in a small-house neighborhood is valued differently than a big house in a wealthy estate area.
  `condition` **(1–5):** How well-maintained the house is (trash vs. tidy).
  `grade` **(1–13):** The **construction quality** and architectural design.
  1–3: Poor construction / Cabin.
  7: Average quality.
  11–13: High-quality custom design.
  `view` **(0–4):** Rating of the view from the property (0 = No view, 4 = Excellent view).
  `waterfront` **:** Binary (0/1) indicating if the house overlooks the water.

DELIVERABLES

1. Prediction File (CSV)

- A CSV file containing your final price predictions on the test dataset.
- Format: id, predicted_price

2. Code Repository

- A clean, reproducible GitHub repository containing:
  data_fetcher.py : The script used to download images from the API.
  preprocessing.ipynb : Data cleaning and feature engineering.
  model_training.ipynb : The training loop for the multimodal model.
  README.md : Clear instructions on how to set up the project and run the code.

3. Project Report (PDF)

- **Overview :** Your approach and modeling strategy.
- **EDA :** Visualizations of price distribution and sample satellite images.
- **Financial/Visual Insights :** Analysis of which visual features (e.g., trees vs. concrete) drive value.
- **Architecture Diagram :** A simple diagram showing how you connected the image model (CNN) with the data model.

- **Results :** Compare the performance of "Tabular Data Only" vs. "Tabular + Satellite Images."

EVALUATION CRITERIA

- **Model Performance (40%) :** Accuracy of the price predictions (measured by RMSE and R^2 Score).
- **Engineering Quality (30%) :** Quality of the data fetching pipeline and the neural network architecture.
- **Analysis & Explainability (30%) :** Depth of the EDA and the clarity of the visual explainability (Grad-CAM).

TECH STACK

- **Data Handling :** Pandas, NumPy, GeoPandas
- **Deep Learning :** PyTorch or TensorFlow/Keras (essential for image processing)
- **Image Processing :** OpenCV, PIL
- **Machine Learning :** Scikit-learn, XGBoost
- **Visualization :** Matplotlib, Seaborn

NOTE : These are just some recommendations. We are open to your unique approach as well!

SUBMISSION GUIDELINES

- **Repository :** Push complete code to a public GitHub repository
- **Report :** Upload the PDF report to the submission portal
- **Deadline :** 5 January 2026

We value your unique approach, so please ensure your submission is your own original work. While open-source tools are welcome, direct copying will unfortunately lead to disqualification.