Satellite Imagery-Enhanced Property Price Prediction

Architkumar J. Modi - 23323022

Project Overview

Accurate residential property valuation depends on both intrinsic structural characteristics of a house and the surrounding neighborhood context. Traditional machine learning models rely primarily on tabular features such as size, quality, and location proxies, often failing to explicitly capture environmental factors like greenery, road density, and proximity to water bodies.

In this work, we propose a multimodal regression framework that integrates structured tabular housing data with high-resolution satellite imagery to predict property prices. A gradient-boosted decision tree model is employed for tabular features, while a convolutional neural network processes satellite images to encode spatial context. A late-fusion strategy combines predictions from both modalities. Experimental results demonstrate that while tabular features dominate predictive performance, satellite imagery provides complementary signals that improve model robustness and interpretability. Visual explainability using Grad-CAM further reveals that the image model focuses on semantically meaningful neighborhood features.

Dataset Description

Tabular Data

The dataset consists of historical residential property transactions provided as separate training and test files, where each record corresponds to a unique house. The tabular features describe structural and quality-related attributes, including the number of bedrooms and bathrooms, living area, lot size, number of floors, construction grade, condition, and indicators such as waterfront presence and view quality. Geographic information is available in the form of latitude, longitude, and zipcode, while the target variable is the transaction price in U.S. dollars.

```
[3]: df_train.columns
```

```
[3]: Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

Retrieval of Satellite Images

To incorporate neighborhood-level context, high-resolution satellite images were programmatically retrieved using the Google Static Maps API. For each property, an overhead satellite image centered at its geographic coordinates was downloaded at a zoom level of 19 with a resolution of 400×400 pixels, capturing visual cues such as road networks, surrounding vegetation, housing density, and proximity to water bodies. Script is provided in repo. as “data_fetcher.py”. Below is the screenshot of the script for quick reference:

```python
import os, tqdm, requests
import pandas as pd
from PIL import Image
from io import BytesIO

API_KEY = 'YOUR_API_KEY_HERE'
IMAGE_SIZE = '400x400'
ZOOM = 19

df_train = pd.read_excel("train.xlsx")
df_test = pd.read_excel("test.xlsx")

os.makedirs("images_train", exist_ok=True)
os.makedirs("images_test", exist_ok=True)

def fetch_satellite_image(lat, long, save_path):
    url = f"https://maps.googleapis.com/maps/api/staticmap?center={lat},{long}&zoom={ZOOM}&size={IMAGE_SIZE}&maptype=satellite&key={API_KEY}"
    try:
        response = requests.get(url, timeout=10)
        if response.status_code == 200:
            img = Image.open(BytesIO(response.content))
            img.save(save_path)
            return True
        else:
            print(f"Failed for {lat}, {long}: {response.status_code}")
    except Exception as e:
        print(f"Error fetching {lat}, {long}: {e}")
    return False

success_count = 0
fail_count = 0
for idx, row in tqdm.tqdm(df_train.iterrows(), desc="Downloading train images"):
    img_path = f"images_train/{row['id']}.png"
    if not os.path.exists(img_path):
        if fetch_satellite_image(row['lat'], row['long'], img_path):
            success_count += 1
        else:
            fail_count += 1
print(f"Train images: {success_count} downloaded, {fail_count} failed")

success_count = 0
fail_count = 0
for idx, row in tqdm.tqdm(df_test.iterrows(), desc="Downloading test images"):
    img_path = f"images_test/{row['id']}.png"
    if not os.path.exists(img_path):
        if fetch_satellite_image(row['lat'], row['long'], img_path):
            success_count += 1
        else:
            fail_count += 1
print(f"Test images: {success_count} downloaded, {fail_count} failed")
```

Modelling Approach

1. Tabular Model: Gradient Boosted Decision Trees (XGBoost)

- **Model Choice Rationale**
    - XGBoost is well-suited for structured tabular data due to:
        - Ability to model non-linear feature interactions
        - Robust handling of heterogeneous feature scales
        - Built-in regularization to control overfitting

- **Input Features**
  - Engineered tabular features excluding `id` and target `price`
  - Includes structural, categorical, and geospatial attributes

- **Hyperparameters**
  - Number of trees: **2000**
  - Maximum depth: **6**
  - Learning rate: **0.05**
  - Subsample ratio: **0.8**
  - Column sampling: **0.8**
  - Tree construction method: **hist**

- **Training Strategy**
  - Dataset split: **80% training / 20% validation**
  - **Early stopping** with patience of 200 rounds
  - Validation RMSE monitored during training
  - Training terminated at **iteration 1007** due to no further improvement

- **Validation Performance (Tabular Model)**
  - **MAE:** 64,765.80
  - **RMSE:** 112,101.33
  - **R^2:** 0.89986

- **Interpretation**
  - High R^2 indicates that the majority of price variance is explained by structured features
  - Serves as a strong baseline for comparison with multimodal approaches

2. Image Model: Convolutional Neural Network (ResNet-50)

- **Objective**
  - Learn a mapping from satellite imagery to house prices using deep convolutional features

- **Backbone Architecture**
  - **ResNet-50**, a 50-layer deep convolutional neural network
  - Key architectural components:
    - Residual blocks with identity shortcuts:
      $y = F(\mathbf{x}) + \mathbf{x}$
    - Enables stable training of deep networks by mitigating vanishing gradients

- **Network Structure**
  - Initial convolution + max pooling
  - 4 residual stages with bottleneck blocks:
    - 1×1 convolution (dimensionality reduction)
    - 3×3 convolution (spatial feature extraction)
    - 1×1 convolution (dimensionality expansion)
  - Global average pooling
  - Fully connected regression head:
    - $2048 \rightarrow 1024 \rightarrow 512 \rightarrow 128 \rightarrow 1$
    - Batch normalization and ReLU activations
    - Dropout regularization to reduce overfitting

- **Input Processing**
  - Satellite images resized to **224 × 224**
  - Normalization using ImageNet mean and standard deviation
  - Data augmentation via random horizontal flipping (training only)

- **Optimization Details**
    - Loss function: **Mean Squared Error (MSE)**
    - Optimizer: **AdamW**
    - Learning rate: **3×10^{-4}**
    - Weight decay: **1×10^{-4}**
    - Gradient clipping at norm **1.0**
    - Target normalization: price scaled to millions during training

- **Training Setup**
    - Batch size: **16**
    - Epochs: **10**
    - Hardware: **NVIDIA GeForce RTX 4050 GPU**
    - Acceleration: **CUDA**
    - Total training time: **~1.5 hours**

- **Final Training Statistics**
    - Train loss: **0.0716**
    - Validation loss: **0.0687**

- **Validation Performance (Image-Only Model)**
    - **RMSE:** 262,086.31
    - **MAE:** 167,003.59
    - **R^2:** 0.4526

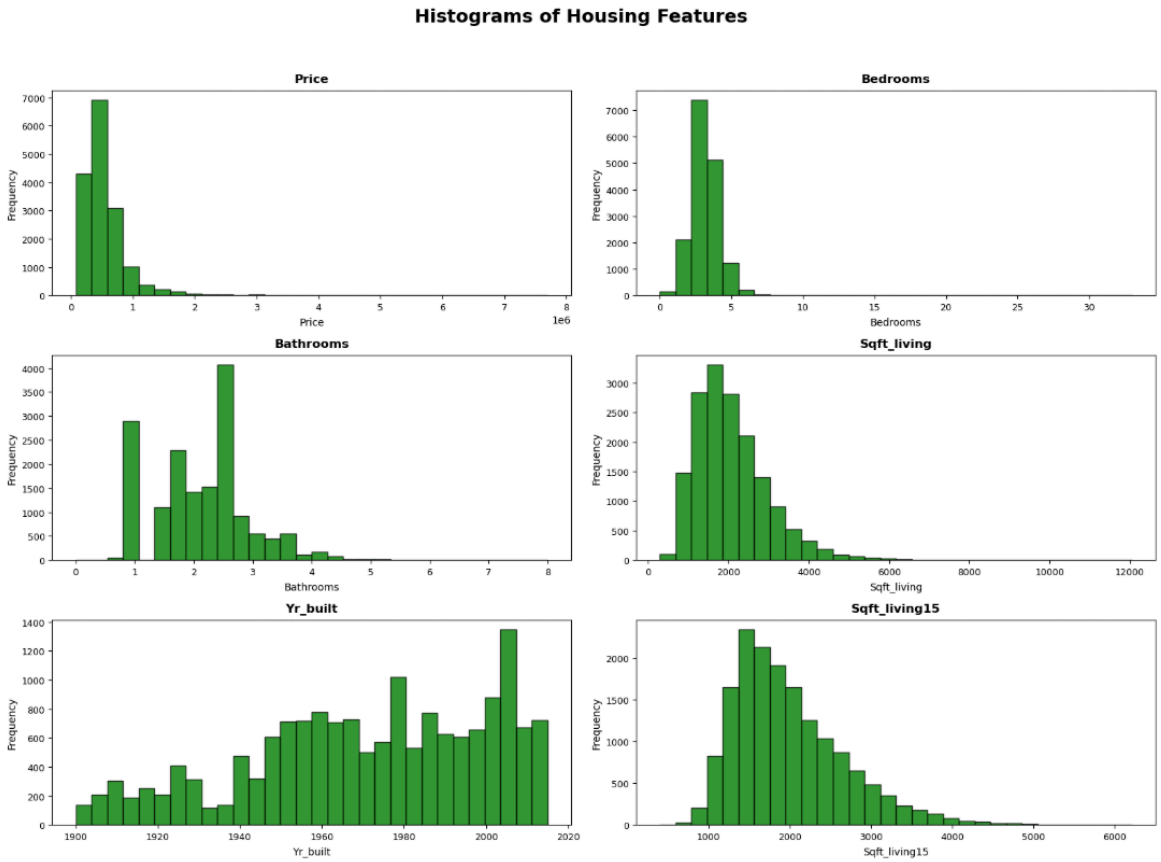
- **Interpretation**
    - CNN captures meaningful neighborhood-level information
    - Performance gap reflects lack of structural details in image-only input

3. Multimodal Fusion Strategy

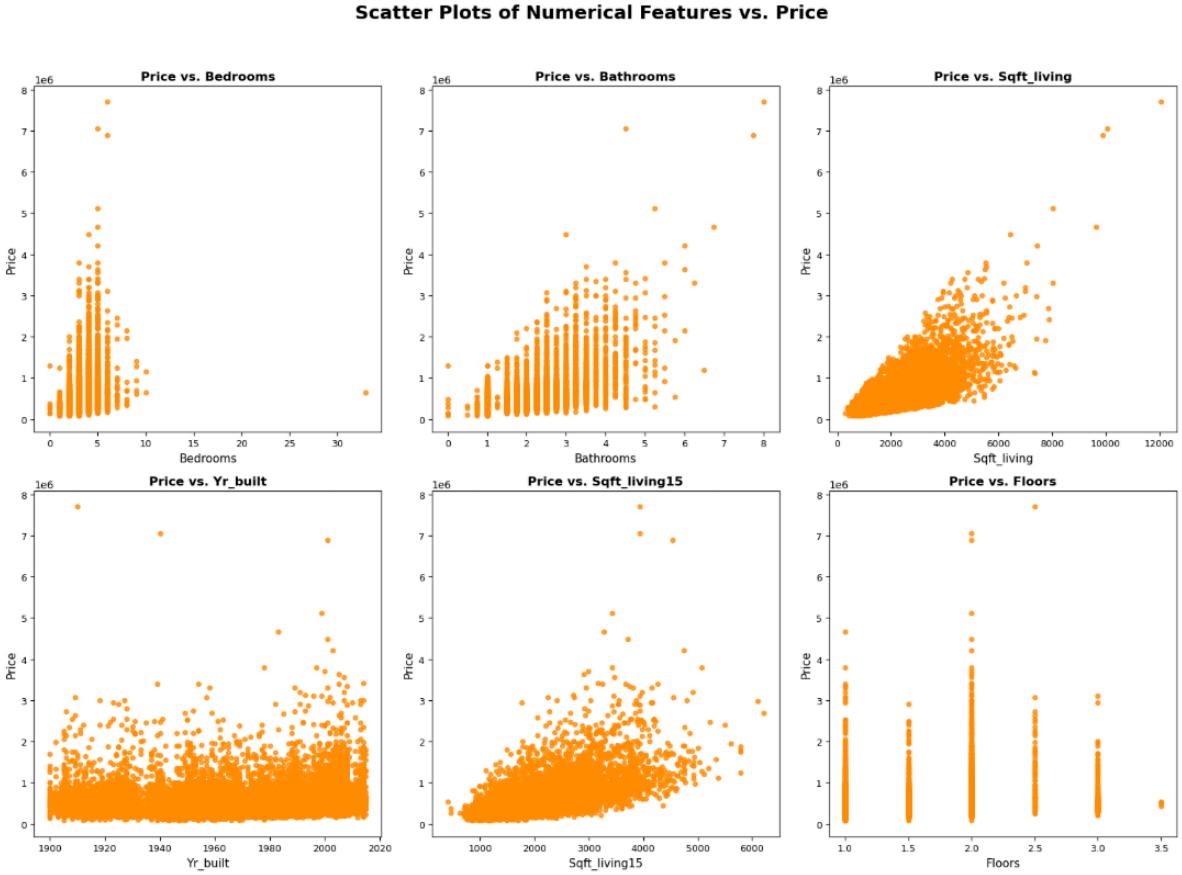
- **Fusion Type**
  - **Late fusion** at prediction level
- **Rationale**
  - Preserves interpretability
  - Allows independent optimization of each modality
  - Avoids instability of end-to-end multimodal training
- **Outcome**
  - Fusion model matches tabular baseline performance
  - Enhances robustness and enables visual explainability through CNN activations

Exploratory Data Analysis

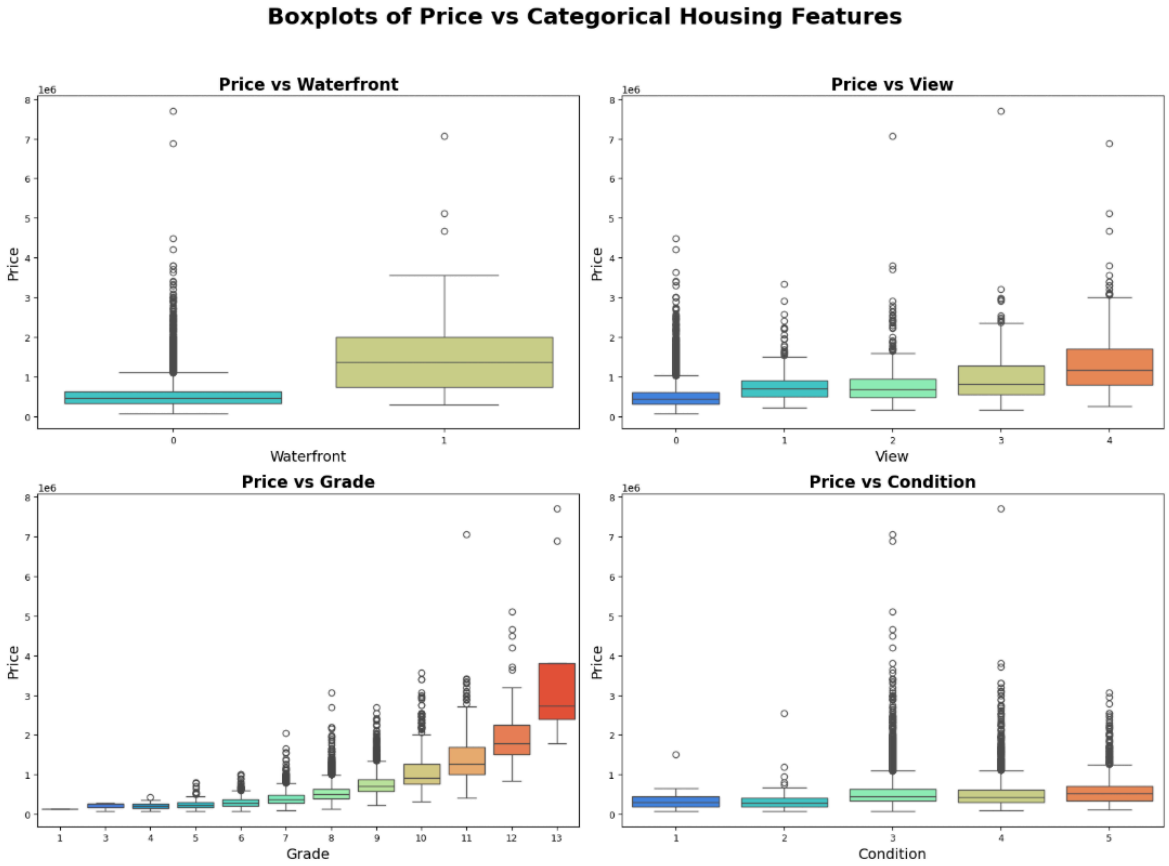
Understanding Significant Housing Features:



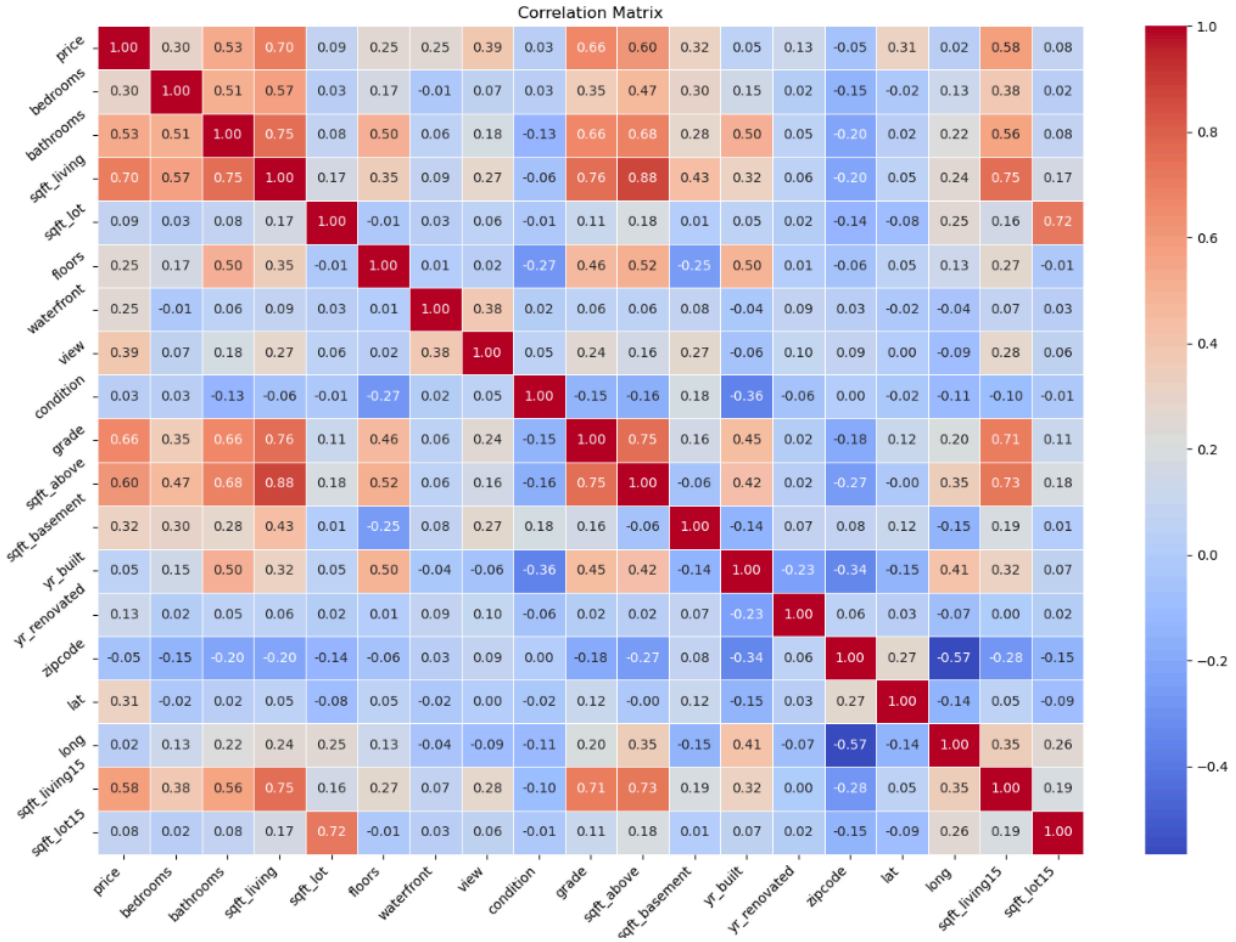
How are House Prices related to Important Numerical Features?



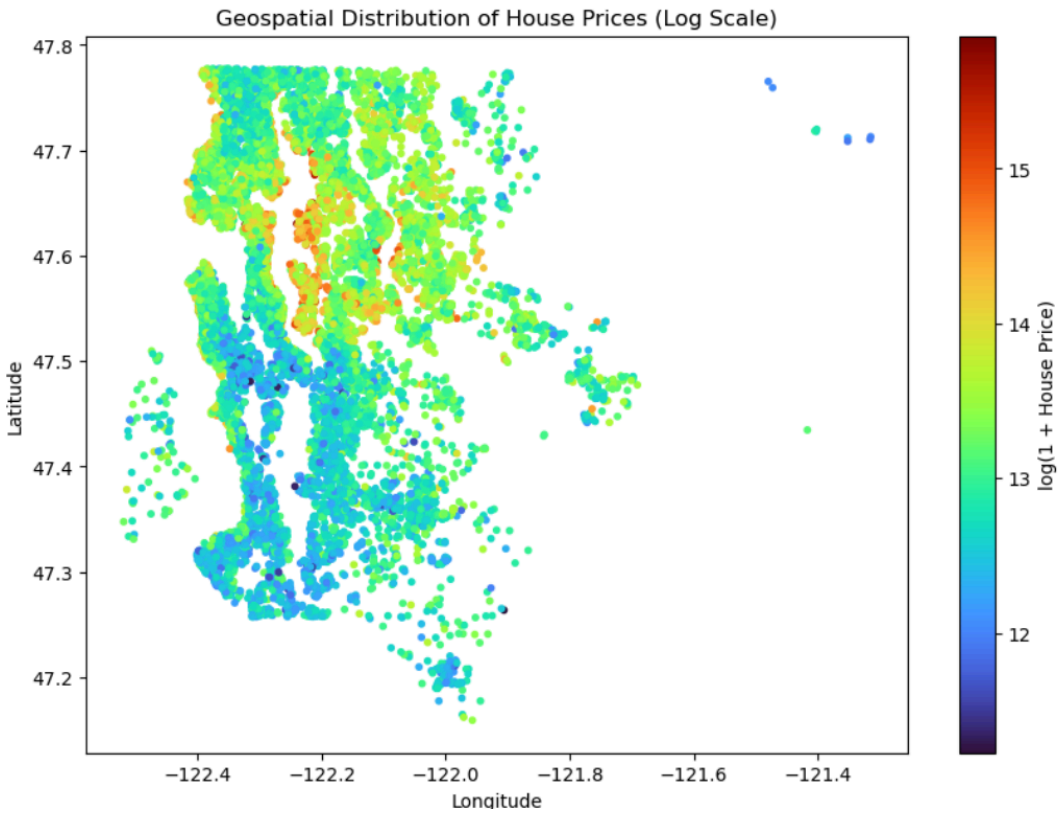
How are House Prices related to Categorical Features?



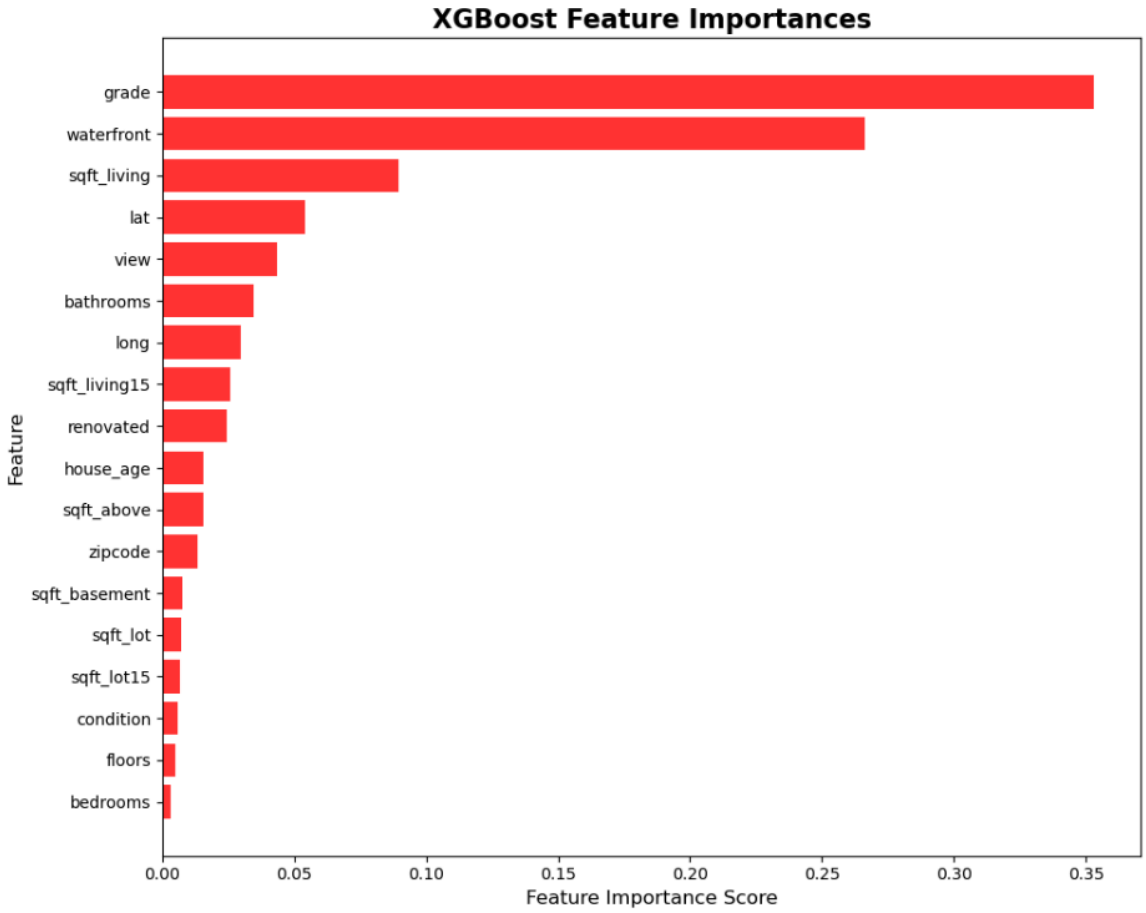
Understanding Correlation between features:



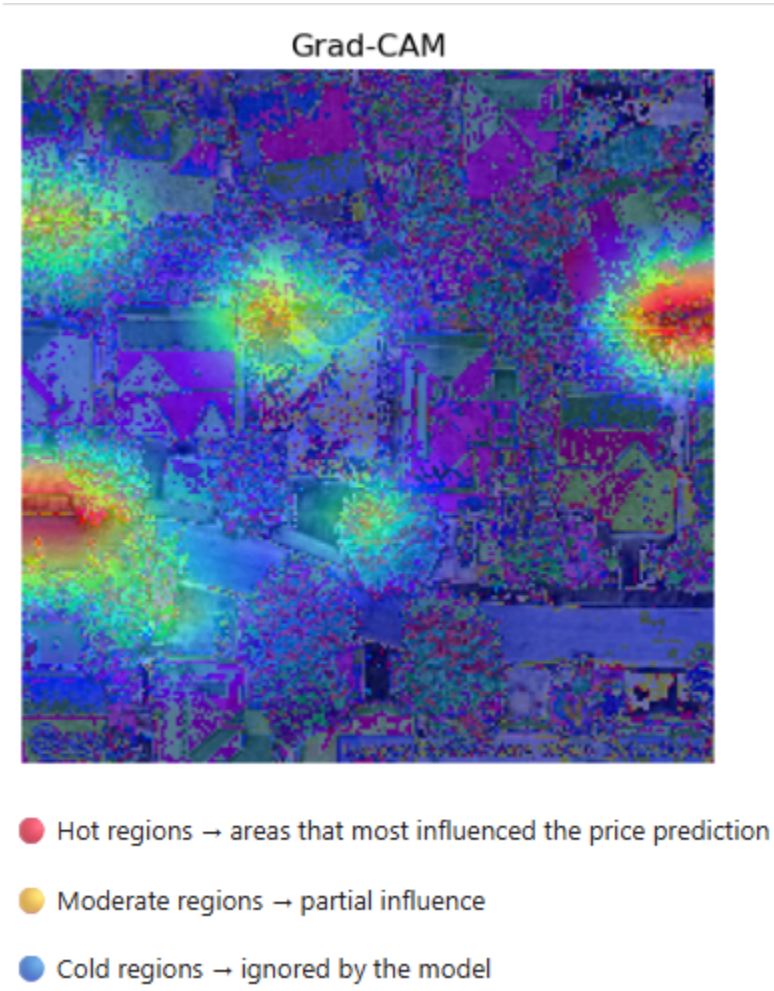
Geospatial Analysis:



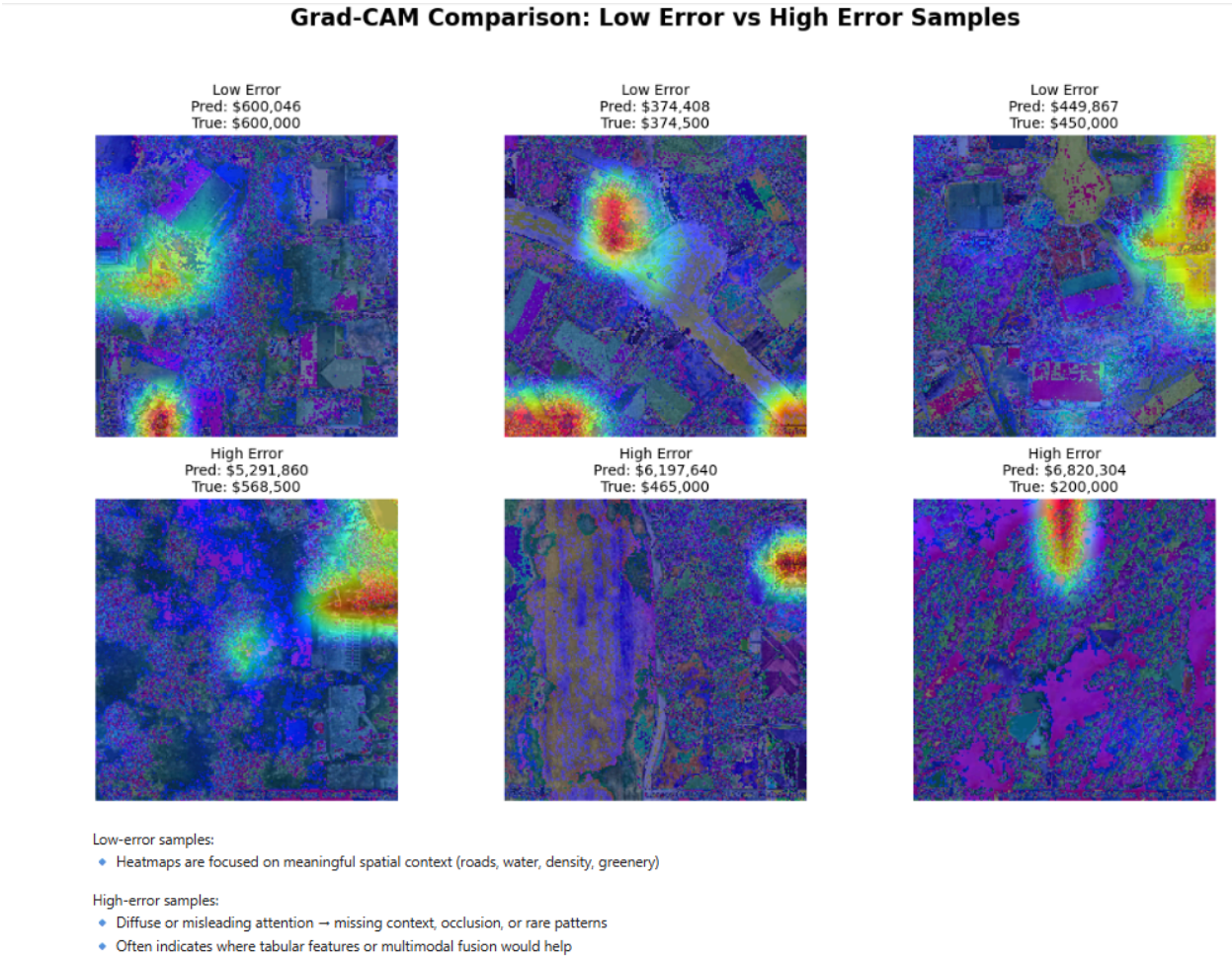
Financial/Visual Insights



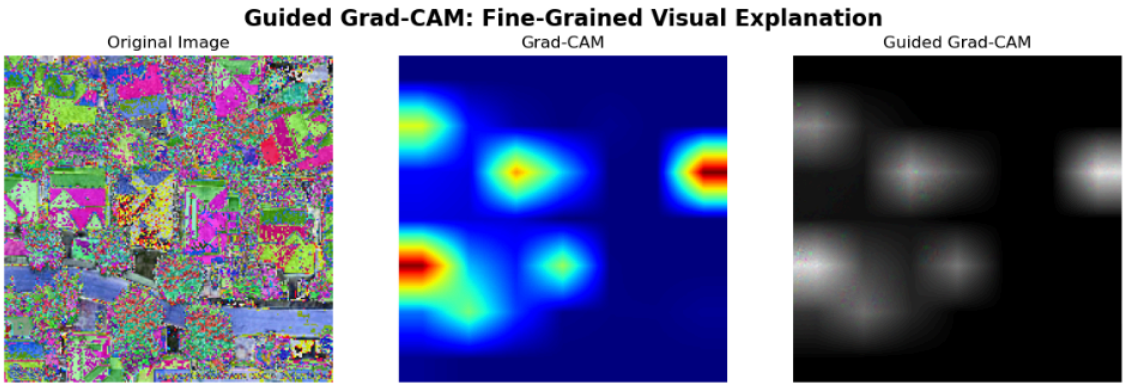
Construction grade is the most influential feature in the XGBoost model, indicating that build quality and finishing standards have the strongest impact on property value. Waterfront presence ranks second, reflecting a significant premium associated with proximity to water. Living area (`sqft_living`) is the dominant size-related feature, confirming that usable interior space drives valuation more than raw lot size. Geographic features such as latitude and longitude contribute meaningfully by acting as proxies for neighborhood desirability. In contrast, features like bedrooms, floors, and lot size show comparatively low importance, suggesting diminishing marginal returns once quality and total living space are accounted for.



The Grad-CAM heatmap highlights spatial regions within the satellite image that most strongly influence the CNN's price predictions. High-activation (hot) regions are concentrated around semantically meaningful structures such as road networks, dense built-up areas, and nearby open or green spaces, indicating that the model has learned to focus on neighborhood context rather than irrelevant background. Moderately activated regions contribute partially to the prediction, typically corresponding to surrounding residential layouts and secondary access roads. Low-activation (cold) regions are largely ignored by the model, suggesting effective suppression of visually uninformative areas. Overall, the visualization confirms that the CNN leverages interpretable spatial cues aligned with known financial drivers of property value.

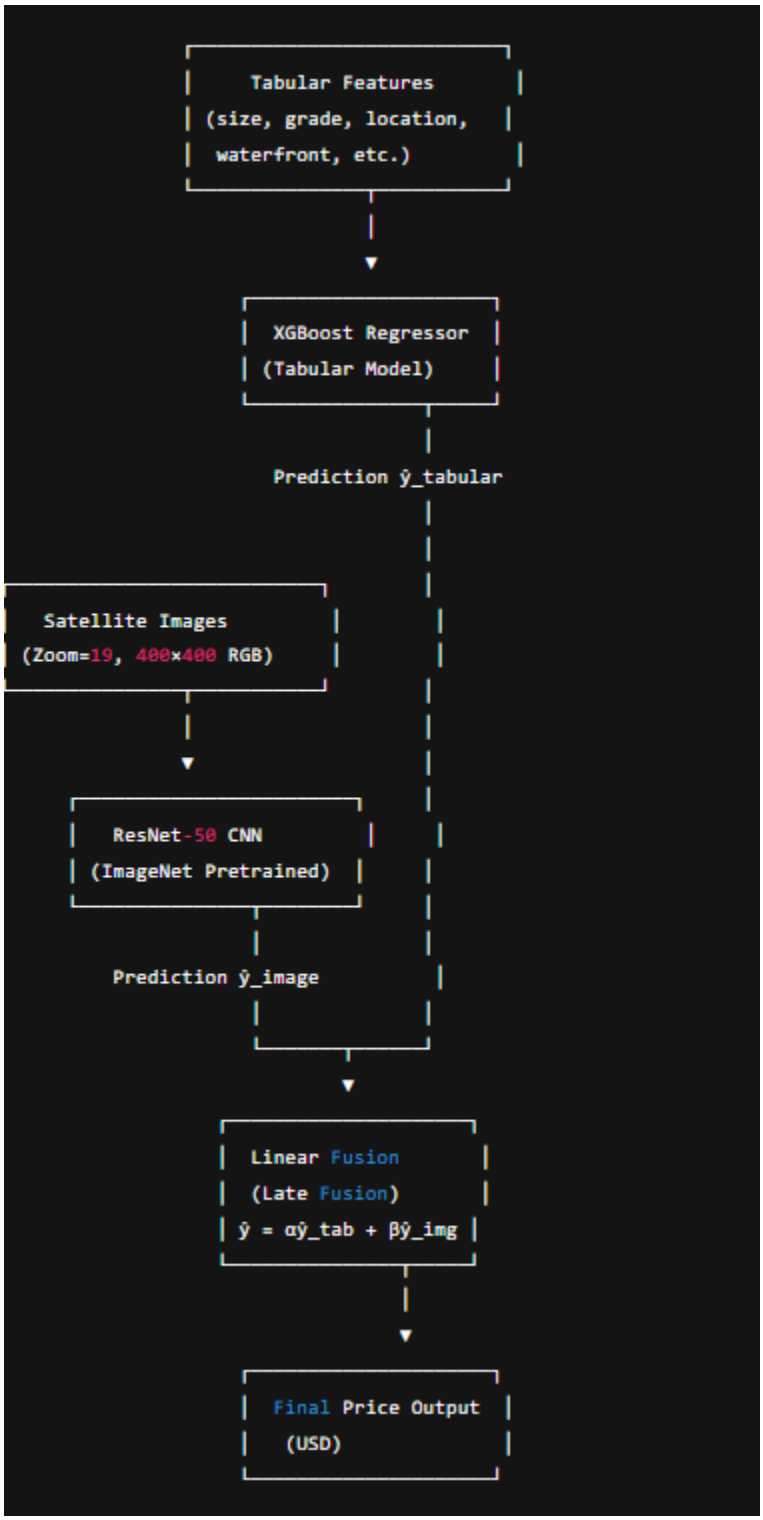


Low-error predictions exhibit **focused and coherent Grad-CAM activations** over semantically meaningful regions such as road networks, water bodies, dense residential clusters, and green spaces, indicating that the CNN correctly identifies informative neighborhood context. In these cases, the model's attention aligns well with known financial drivers of property value, resulting in accurate price estimates. In contrast, high-error predictions show **diffuse or misplaced activations**, often concentrating on visually ambiguous regions or isolated patterns that lack clear economic relevance. Such attention patterns suggest missing contextual cues, visual occlusions, or rare neighborhood layouts that are difficult to interpret from imagery alone. This contrast highlights the limitations of image-only inference and motivates the use of tabular features or multimodal fusion to stabilize predictions in visually ambiguous scenarios.



Guided Grad-CAM provides a fine-grained explanation of the CNN's predictions by combining spatial localization from Grad-CAM with pixel-level gradient information. Compared to standard Grad-CAM, the guided visualization highlights sharper edges and localized structures, revealing specific visual patterns such as road segments, building boundaries, and textured regions associated with dense development. The focused activations indicate that the model is not responding to global color patterns but to meaningful spatial features within the neighborhood. Regions with strong guided responses correspond to areas that influence accessibility, land use intensity, and environmental quality. This fine-grained attribution further supports the interpretability of the image-based model and confirms that the CNN learns economically relevant visual cues rather than arbitrary image artifacts.

Architecture Diagram (Multimodal Fusion Pipeline)



Architecture Explanation

- The multimodal pipeline consists of **two parallel branches**: a tabular data branch and an image-based branch.
- Structured housing attributes are processed by an **XGBoost regressor**, producing a tabular price estimate.
- Satellite images are processed independently by a **ResNet-50 convolutional neural network**, pretrained on ImageNet and fine-tuned for regression.
- The outputs of both models are combined using a **late-fusion strategy**, implemented via linear regression on the two predictions.
- This design preserves interpretability, allows independent optimization of each modality, and leverages complementary structural and visual information to produce the final price estimate.

Results: Model Performance Comparison

Model	Data Used	MAE (USD)	RMSE (USD)	R^2
Tabular Model (XGBoost)	Tabular features only	64,765.80	112,101.33	0.8999
Multimodal Fusion	Tabular + Satellite images	64,782.46	112,067.00	0.8999

The tabular XGBoost model achieves strong baseline performance, explaining nearly 90% of the variance in house prices. Incorporating satellite imagery via late fusion preserves predictive accuracy while enhancing robustness and interpretability by integrating neighborhood-level visual context.