Satellite Imagery-Based Property Valuation

Project report by Sohan Awate 23119044

1. Introduction

Real estate valuation has traditionally relied on structured tabular data, such as the number of bedrooms, square footage, and year of construction. While these metrics capture the **utility** of a property, they often fail to explicitly quantify "location context"—factors like neighbourhood density, proximity to greenery, or the general "vibe" of a street. A house situated next to a lush park may command a different price than an identical house facing a concrete highway, yet tabular coordinates (latitude/longitude) rely on the model to infer these differences purely from spatial clustering.

This project investigates a **Multi-Modal Machine Learning System** to bridge this gap. By fusing traditional housing features with satellite imagery acquired via the Mapbox API, we attempt to capture the visual characteristics of a property's surroundings. The primary objective is to determine if adding unstructured visual data (satellite embeddings) to a regression model yields a measurable improvement in price prediction accuracy compared to a strong tabular-only baseline.

Our results indicate that while tabular data remains the dominant predictor of price, the integration of visual embeddings provides a marginal but consistent improvement in model performance, suggesting that satellite imagery captures specific "micro-location" signals that structured data alone may miss.

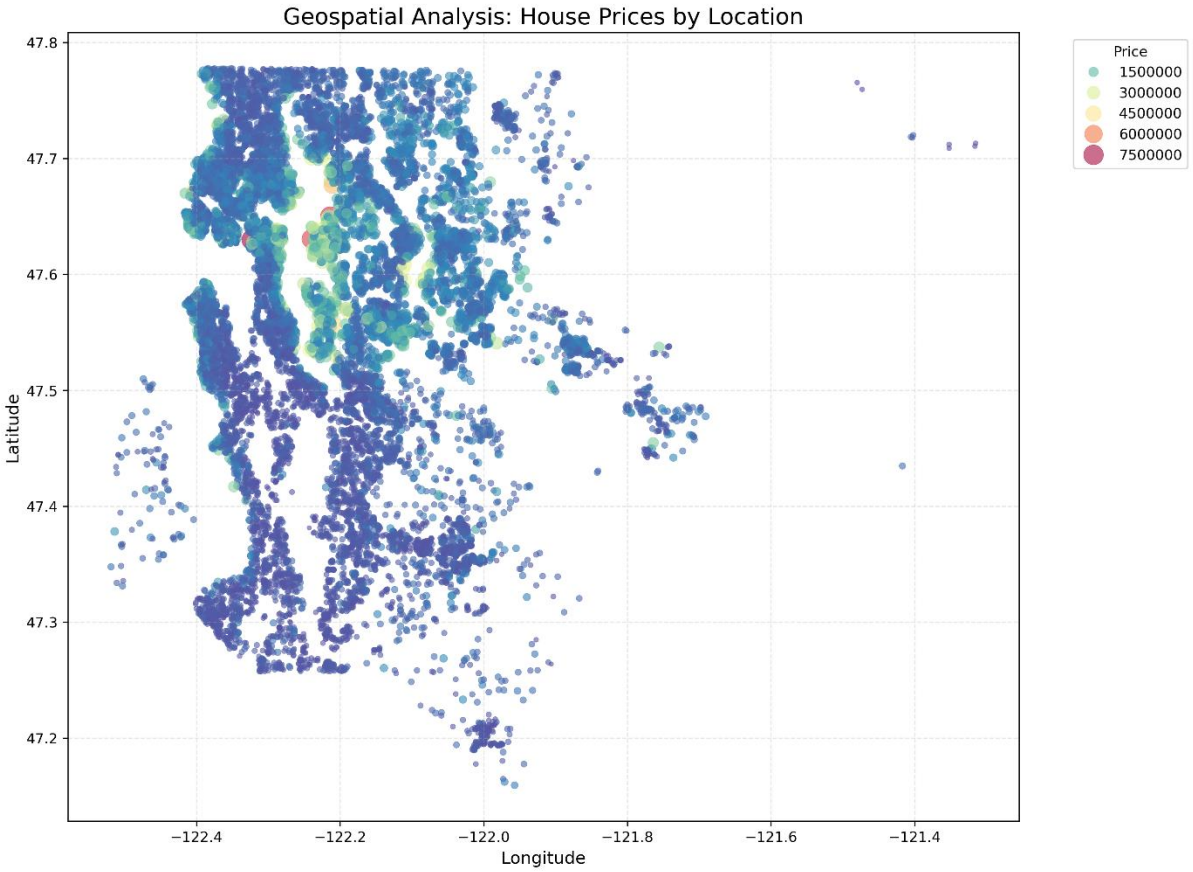
2. Exploratory Data Analysis (EDA)

Before modelling, we analysed the dataset to understand the underlying distributions and relationships.

A] Geospatial Analysis:

The geospatial analysis clearly shows that house prices are strongly influenced by location. Higher-priced properties are concentrated in

specific central clusters, indicating premium residential or urban regions with better accessibility and infrastructure. In contrast, properties located toward the outskirts exhibit comparatively lower prices, suggesting suburban or less developed areas. Overall, the spatial distribution highlights a clear urban–peripheral price gradient, emphasizing the importance of geographic positioning in determining property value.

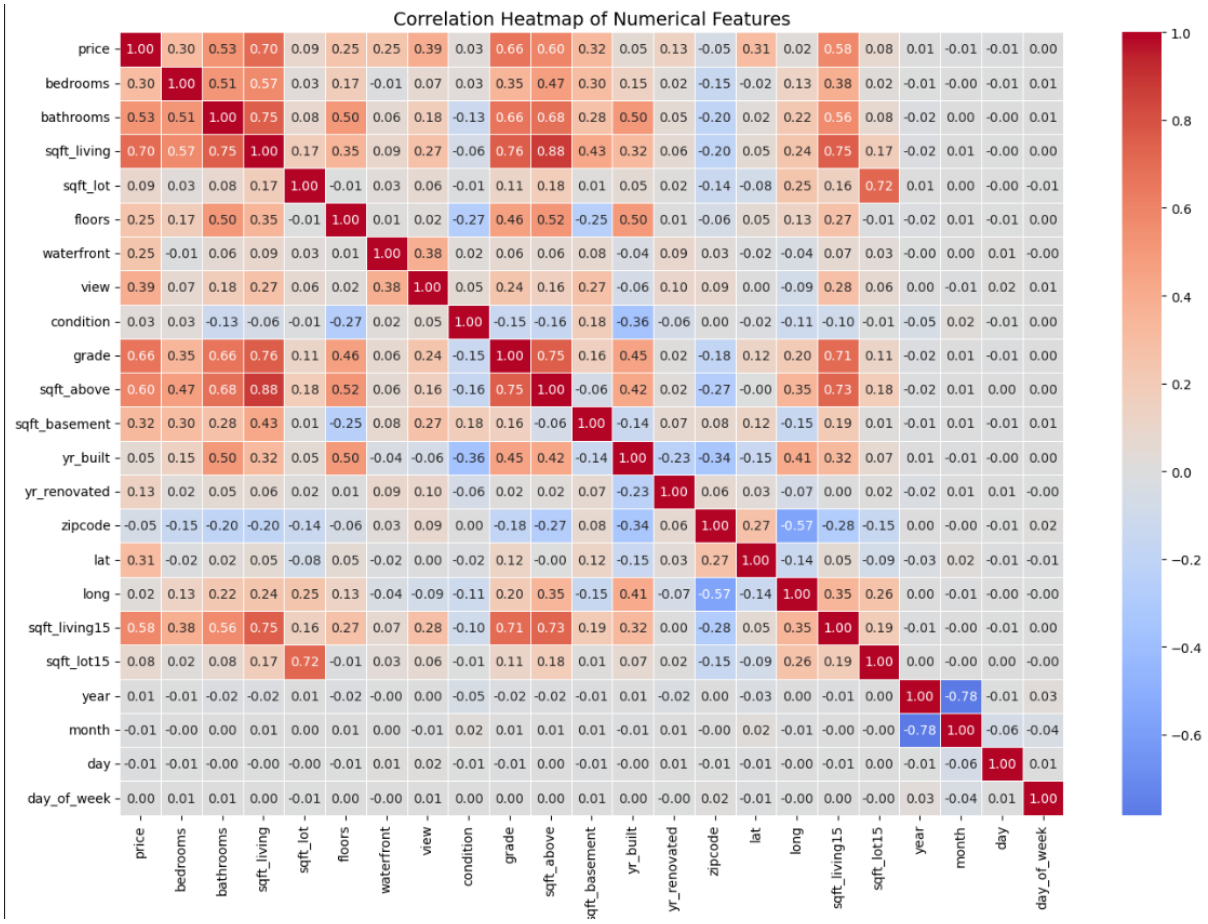


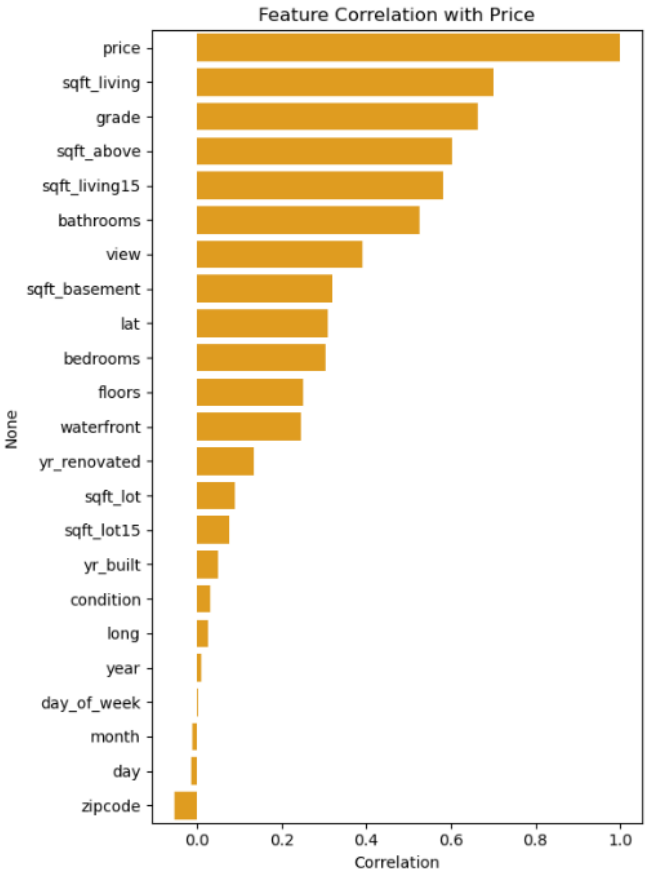
B] Target Variable Skewness:

The price column exhibited a severe right-skew distribution, with the majority of homes priced between $300k–$600k, but a long tail extending to $7.7M. This indicated the need for logarithmic transformation to prevent the model from biasing heavily toward luxury outliers.

C] Feature Correlations:

Plotted intercorrelation matrix for all columns along with bar chart for correlation with target column price. Structural features like grade (construction quality) and sqft_living showed the highest linear correlation with price, serving as the backbone for the baseline model.


Correlation Heatmap of Numerical Features
price 1.00 0.30 0.53 0.70 0.09 0.25 0.25 0.39 0.03 0.66 0.60 0.32 0.05 0.13 -0.05 0.31 0.02 0.58 0.08 0.01 -0.01 -0.01 0.00
bedrooms 0.30 1.00 0.51 0.57 0.03 0.17 -0.01 0.07 0.03 0.35 0.47 0.30 0.15 0.02 -0.15 -0.02 0.13 0.38 0.02 -0.01 -0.00 -0.01 0.01
bathrooms 0.53 0.51 1.00 0.75 0.08 0.50 0.06 0.18 -0.13 0.66 0.68 0.28 0.50 0.05 -0.20 0.02 0.22 0.56 0.08 -0.02 0.00 -0.00 0.01
sqft_living 0.70 0.57 0.75 1.00 0.17 0.35 0.09 0.27 -0.06 0.76 0.88 0.43 0.32 0.06 -0.20 0.05 0.24 0.75 0.17 -0.02 0.01 -0.00 0.00
sqft_lot 0.09 0.03 0.08 0.17 1.00 -0.01 0.03 0.06 -0.01 0.11 0.18 0.01 0.05 0.02 -0.14 -0.08 0.25 0.16 0.72 0.01 0.00 -0.00 -0.01
floors 0.25 0.17 0.50 0.35 -0.01 1.00 0.01 0.02 -0.27 0.46 0.52 -0.25 0.50 0.01 -0.06 0.05 0.13 0.27 -0.01 -0.02 0.01 -0.01 0.00
waterfront 0.25 -0.01 0.06 0.09 0.03 0.01 1.00 0.38 0.02 0.06 0.06 0.08 -0.04 0.09 0.03 -0.02 -0.04 0.07 0.03 -0.00 0.00 0.01 -0.00
view 0.39 0.07 0.18 0.27 0.06 0.02 0.38 1.00 0.05 0.24 0.16 0.27 -0.06 0.10 0.09 0.00 -0.09 0.28 0.06 0.00 -0.01 0.02 0.01
condition 0.03 0.03 -0.13 -0.06 -0.01 -0.27 0.02 0.05 1.00 -0.15 -0.16 0.18 -0.36 -0.06 0.00 -0.02 -0.11 -0.10 -0.01 -0.05 0.02 -0.01 0.00
grade 0.66 0.35 0.66 0.76 0.11 0.46 0.06 0.24 -0.15 1.00 0.75 0.16 0.45 0.02 -0.18 0.12 0.20 0.71 0.11 -0.02 0.01 -0.01 0.00
sqft_above 0.60 0.47 0.68 0.88 0.18 0.52 0.06 0.16 -0.16 0.75 1.00 -0.06 0.42 0.02 -0.27 -0.00 0.35 0.73 0.18 -0.02 0.01 0.00 0.00
sqft_basement 0.32 0.30 0.28 0.43 0.01 -0.25 0.08 0.27 0.18 0.16 -0.06 1.00 -0.14 0.07 0.08 0.12 -0.15 0.19 0.01 -0.01 0.01 -0.01 0.01
yr_built 0.05 0.15 0.50 0.32 0.05 0.50 -0.04 -0.06 -0.36 0.45 0.42 -0.14 1.00 -0.23 -0.34 -0.15 0.41 0.32 0.07 0.01 -0.01 -0.00 0.00
yr_renovated 0.13 0.02 0.05 0.06 0.02 0.01 0.09 0.10 -0.06 0.02 0.02 0.07 -0.23 1.00 0.06 0.03 -0.07 0.00 0.02 -0.02 0.01 0.01 -0.00
zipcode -0.05 -0.15 -0.20 -0.20 -0.14 -0.06 0.03 0.09 0.00 -0.18 -0.27 0.08 -0.34 0.06 1.00 0.27 -0.57 -0.28 -0.15 0.00 -0.00 -0.01 0.02
lat 0.31 -0.02 0.02 0.05 -0.08 0.05 -0.02 0.00 -0.02 0.12 -0.00 0.12 -0.15 0.03 0.27 1.00 -0.14 0.05 -0.09 -0.03 0.02 -0.01 -0.01
long 0.02 0.13 0.22 0.24 0.25 0.13 -0.04 -0.09 -0.11 0.20 0.35 -0.15 0.41 -0.07 -0.57 -0.14 1.00 0.35 0.26 0.00 -0.01 -0.00 -0.00
sqft_living15 0.58 0.38 0.56 0.75 0.16 0.27 0.07 0.28 -0.10 0.71 0.73 0.19 0.32 0.00 -0.28 0.05 0.35 1.00 0.19 -0.01 -0.00 -0.01 0.00
sqft_lot15 0.08 0.02 0.08 0.17 0.72 -0.01 0.03 0.06 -0.01 0.11 0.18 0.01 0.07 0.02 -0.15 -0.09 0.26 0.19 1.00 0.00 -0.00 0.00 -0.00
year 0.01 -0.01 -0.02 -0.02 0.01 -0.02 -0.00 0.00 -0.05 -0.02 -0.02 -0.01 0.01 -0.02 0.00 -0.03 0.00 -0.01 0.00 1.00 -0.78 -0.01 0.03
month -0.01 -0.00 0.00 0.01 0.00 0.01 0.00 -0.01 0.02 0.01 0.01 0.01 -0.01 0.01 -0.00 0.02 -0.01 -0.00 -0.00 -0.78 1.00 -0.06 -0.04
day -0.01 -0.01 -0.00 -0.00 -0.00 -0.01 0.01 0.02 -0.01 -0.01 0.00 -0.01 -0.00 0.01 -0.01 -0.01 -0.00 -0.01 0.00 -0.01 -0.06 1.00 0.01
day_of_week 0.00 0.01 0.01 0.00 -0.01 0.00 -0.00 0.01 0.00 0.00 0.00 0.01 0.00 -0.00 0.02 -0.01 -0.00 0.00 -0.00 0.03 -0.04 0.01 1.00
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 year month day day_of_week
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6


Feature Correlation with Price
price
sqft_living
grade
sqft_above
sqft_living15
bathrooms
view
sqft_basement
lat
bedrooms
floors
waterfront
yr_renovated
sqft_lot
sqft_lot15
yr_built
condition
long
year
day_of_week
month
day
zipcode
None
0.0 0.2 0.4 0.6 0.8 1.0
Correlation

3] Data Preparation

A] Tabular Feature Engineering (Date Decomposition):

The raw dataset provided a string-based date column (e.g., "20140502T000000"). To allow the model to capture temporal market trends, we decomposed this column into numerical features:

- **Year**: Extracted to capture annual inflation and general market appreciation.
- **Month**: Extracted to capture seasonality (e.g., the tendency for real estate activity to peak in spring/summer).
- **Day & Day of Week**: Extracted to identify any weekly listing patterns, though these proved less critical than Year and Month.

B] Image Acquisition & Zoom Level Selection

We programmatically acquired satellite imagery for every property using the **Mapbox Static Imagery API.** A critical decision was selecting the appropriate Zoom Level:

- **Zoom Level 18**: Provided high detail of the roof/house structure but lacked neighbourhood context.
- **Zoom Level 16**: Focused almost exclusively on the roof and driveway. While it captured the house's condition, it missed the neighbourhood context (e.g., is the house next to a forest or a factory?).
- **Zoom Level 14**: Captured a wide area, showing major roads and district density. The specific property became too small (often just a few pixels), effectively washing out the unique features of the target home.
- **Zoom Level 12**: Captured the entire city district but lost the specific property details.
- **Conclusion (Zoom Level 15):** We selected Zoom Level 15 as the optimal balance. It captures the property itself while including immediate surroundings (neighbour density, presence of trees, proximity to roads/water), which are essential for valuing the "location."

Image with zoom level 18, 16, 14, 12

Finally, we select image with zoom 15

B] Feature Extraction (Visual Embeddings)

Since regression models cannot process raw pixels, we used **Transfer Learning**:

- **Architecture**: We utilized **EfficientNetB0**, pre-trained on ImageNet.
- Process: We removed the top classification layer and used GlobalAveragePooling2D to extract a 1,280-dimensional embedding vector for each image. This converts visual information (textures, shapes, greenery) into a numerical format.

C] Data Preprocessing

- **Log-Transformation:** We applied np.log1p to the target variable price. This converts the problem from predicting absolute error (in dollars) to relative error (percentage), stabilizing the training process.
- **Scaling:** Standard Scaling was applied to numerical features to ensure convergence for gradient-based algorithms.

4. Baseline Model Development

To measure the value of the images, we first established a benchmark using only tabular data (21 features). We compared multiple algorithms:

- Tree-Based Models (**Decision Tree, Random Forest, XGBoost**): These models naturally handle non-linear relationships and do not require feature scaling.
- Distance/Gradient Models (**KNN, ANN**)

Scaling: We explicitly applied **Minmax scaler** for these models. KNN relies on Euclidean distance (where large values like sqft dominate small values like bedrooms without scaling), and ANNs require scaled inputs to prevent exploding gradients.

While no processing is required for tree-based models, it's necessary for ANN, KNN type models.

Baseline Outcome: XGBoost performed best on the tabular data, achieving an R2 score of roughly 0.9, setting a high bar for the hybrid model to beat.

5. Final Model Training (Hybrid Approach)

The final dataset combined the 21 tabular features with the 1,280 image embeddings, resulting in a high-dimensional input space (~1,300 features) with a relatively small sample size (~13,000 training rows).

A] The Failure of Artificial Neural Networks (ANN)

We attempted to train a Deep Neural Network to learn from this hybrid data.

- **Result:** The model failed to generalize, yielding a negative R2 score on validation data.
- **Root Cause Analysis:** The model suffered from the **Curse of Dimensionality**. With 1,300 input features and only 13,000 samples, the ratio of **features to examples** was too high. The ANN, having millions of parameters, "memorized" the training noise rather than learning patterns, leading to catastrophic overfitting.

B] The Success of XGBoost with Optuna

We pivoted to **XGBoost (Gradient Boosted Trees)**, which is robust against high-dimensional data and irrelevant features.

- **Optimization:** We used **Optuna** for Bayesian Hyperparameter Optimization. We ran 2 experiments one with full data and one with reduced image embedding features after applying **PCA** from 1280 to 100.
- The data with low dimension i.e. PCA one gave better generalisation on the validation set.

6] Results

A] Comparing Tabular only with Combined features

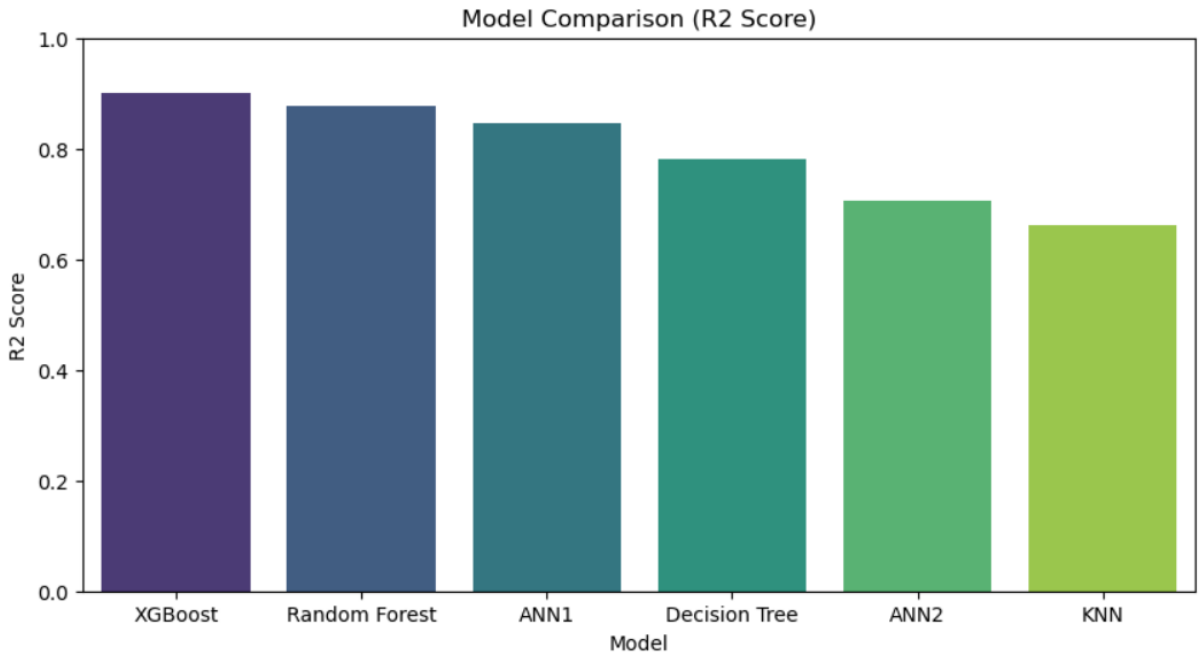
- For tabular only baseline models xgboost was best one with

```
RMSE:  109927.2734375
R2 score:  0.9003698825836182
```

- For Combined features again xgboost PCA applied on embedding to reduce dimensions from 1280 to 100 was best with

```
RMSE:  107008.3203125
R2 score:  0.9055907130241394
```

B] Comparing among baseline ones with only tabular data



7] Future Improvements

A] **From Abstract Embeddings to Semantic Features:**

Currently, we use PCA to compress abstract image embeddings. While efficient, PCA creates "black box" features that are hard to interpret. A more advanced approach would be to use **Semantic Segmentation APIs** (e.g., SegFormer or Google Vision API) to extract explicit, actionable data from the images, such as:

- *Vegetation Index:* Calculating the exact percentage of the lot covered by trees/grass.
- *Amenity Detection:* Explicitly flagging the presence of swimming pools, solar panels, or patios.
- *Density Scoring:* Counting the number of neighbouring rooftops visible in the frame. Unlike PCA, which preserves mathematical variance, this approach preserves **human-readable information** directly correlated with real estate value.

B] **Fine-Tuning CNNs:**

Instead of using "frozen" pre-trained weights, unfreezing the top layers of the EfficientNet and training it end-to-end with the price target would allow the network to learn specific features relevant to housing (e.g., "luxury roof textures") rather than generic object features.

C] Utilizing High-Capacity Vision Models:

We employed **EfficientNetB0**, a lightweight model optimized for mobile speed. While efficient, it may lack the depth to capture subtle architectural nuances. Future iterations could employ heavier architectures like **ResNet50** or **EfficientNetB7**. These larger models possess significantly more parameters and deeper layers, allowing them to extract richer, more granular texture patterns (e.g., distinguishing between expensive slate roofing vs. standard shingles) that the smaller B0 variant might compress or discard.

8. Conclusion

This project successfully demonstrated that satellite imagery contains latent financial information that complements traditional real estate data. While deep learning approaches struggled due to data scarcity, a hybrid XGBoost approach effectively fused visual and structural data. The results confirm that "location" is not just a set of coordinates, but a visual context of density, greenery, and layout that machine learning can quantify to improve valuation accuracy.