

Predicting Real Estate Prices Using Tabular Data and Satellite Image Embeddings

1. Introduction

Accurate real estate price prediction is a critical problem in urban planning, real estate analytics, and financial decision-making. Traditional property valuation models rely mainly on **structured (tabular) features** such as number of bedrooms, square footage, and location-based numerical attributes. However, these features often fail to capture **visual and environmental factors** such as neighborhood layout, greenery, road connectivity, and surrounding infrastructure.

With the advancement of deep learning and computer vision, **satellite imagery** provides a powerful complementary data source. Satellite images implicitly encode neighborhood quality, urban density, and land usage patterns—factors that strongly influence property prices but are difficult to quantify numerically.

This project proposes a **multimodal machine learning approach** that combines:

- **Tabular property features**
- **Satellite image embeddings extracted using a pre-trained CNN (ResNet-18)**

The goal is to demonstrate that integrating visual information with traditional tabular data significantly improves property price prediction accuracy.

2. Problem Statement

Given:

- A dataset containing **property-level tabular features**
- Geographic coordinates (latitude and longitude) for each property

Predict:

- The **market price** of properties as accurately as possible.

We compare two models:

1. **Baseline model** using only tabular features
 2. **Enhanced multimodal model** using tabular features + satellite image embeddings
-

3. Dataset Description

3.1 Tabular Features

The dataset contains structured attributes commonly used in real estate valuation:

- bedrooms
- bathrooms
- sqft_living
- sqft_lot
- floors
- waterfront
- view
- condition
- grade

The **target variable** is:

- price (continuous)
-

3.2 Satellite Images

- Satellite images are fetched using the **Mapbox Static Satellite API**
- Each image is centered at the property's latitude and longitude
- Image resolution: **224 × 224**
- Zoom level chosen to capture neighborhood-level details

Images are stored locally in:

images/train/

images/test/

4. Data Preprocessing

4.1 Tabular Data Cleaning

- Missing values were checked and removed
 - Only relevant numerical features were retained
 - Features were standardized using **StandardScaler**
-

4.2 Satellite Image Downloading

A custom Python script (data_fetcher.py) downloads satellite images using latitude and longitude coordinates.

Key steps:

- Construct API request URL
- Download image
- Convert to RGB
- Save locally with consistent naming

This step ensures reproducibility and decouples image fetching from model training.

5. Feature Engineering

5.1 Image Embedding Extraction

Instead of training a CNN from scratch, we use **transfer learning**:

- **ResNet-18** pre-trained on ImageNet
- The final classification layer is removed
- Output: a **512-dimensional embedding vector** per image

Why ResNet-18?

- Lightweight
- Strong generalization
- Efficient for medium-sized datasets

Each satellite image is transformed into a numerical representation capturing spatial and visual patterns.

5.2 Multimodal Feature Construction

For each property:

- Tabular features \rightarrow vector of size n
- Image embedding \rightarrow vector of size 512

These are concatenated horizontally to form a **combined feature vector**:

Final Feature Vector = [Tabular Features | Image Embedding]

6. Model Architecture

6.1 Regression Model

We use **Ridge Regression** as the final predictive model.

Reasons:

- Handles high-dimensional features well
- Regularization prevents overfitting
- Interpretable and stable

6.2 Train–Validation Split

- **80% training**
- **20% validation**
- Split performed randomly from the training dataset

This allows fair evaluation using unseen data while avoiding data leakage.

7. Evaluation Metrics

We use two standard regression metrics:

Root Mean Squared Error (RMSE)

Measures average prediction error in currency units.

Lower RMSE → better model accuracy.

R² Score (Coefficient of Determination)

Measures how much variance in price is explained by the model.

- $R^2 = 1$ → perfect prediction
- $R^2 = 0$ → no predictive power

8. Experimental Results

8.1 Tabular-Only Model

- Input: Only structured numerical features
 - Performance:
 - **Lower R^2**
 - Higher RMSE
 - Limitation:
 - Cannot capture neighborhood quality visually
-

8.2 Multimodal Model (Tabular + Images)

Final Validation Results:

Validation RMSE: 65,335.97

Validation R^2 : 0.9695

Key Observations

- Extremely high R^2 (~97%)
 - Significant reduction in RMSE
 - Satellite images provide strong contextual information
 - Visual environment plays a major role in property valuation
-

9. Result Interpretation

The dramatic improvement in performance demonstrates that:

- Satellite imagery encodes valuable spatial signals
- CNN embeddings successfully translate visual patterns into numeric features
- Combining heterogeneous data sources leads to superior predictions

This validates the hypothesis that **multimodal learning outperforms unimodal approaches** in real estate valuation.

10. Applications

- Real estate price estimation platforms
- Urban development analysis

- Investment risk assessment
 - Automated valuation models (AVMs)
 - Smart city planning
-

11. Limitations

- Satellite image API rate limits
 - Images reflect static snapshots (not temporal changes)
 - Model interpretability for image features is limited
 - Requires significant storage for image data
-

12. Future Work

- Use **temporal satellite imagery**
 - Apply **attention-based multimodal networks**
 - Experiment with deeper CNNs (ResNet-50, EfficientNet)
 - Integrate street-view imagery
 - Deploy as a web-based prediction service
-

13. Conclusion

This project successfully demonstrates that integrating satellite image embeddings with traditional tabular features significantly improves real estate price prediction accuracy. The multimodal approach captures both **quantitative property characteristics** and **qualitative neighborhood context**, resulting in a highly accurate and robust predictive model.

The achieved **R^2 of 0.97** confirms the effectiveness of deep learning-based feature extraction combined with classical regression models.





