Satellite Imagery Based Property Valuation

Name: Sameer Kumar
Enroll No: 24322024

1. Overview (Approach & Strategy)

This project aims to predict residential property prices using a multimodal learning approach combining structured tabular features with satellite imagery extracted using geographic coordinates (latitude and longitude). Traditional regression models are first trained using tabular data to establish baselines. Satellite images are then fetched using Sentinel Hub APIs, processed through a CNN to extract visual embeddings, and fused with tabular features for final prediction.

2. Dataset Description

- **Train rows:** 16,209
- **Test rows:** 5,404
- **Target:** “price”
- **Key Features:**
  - Structural (*bedrooms, bathrooms, sqft, etc.*)
  - Geographic (*lat, long*)
  - Visual (*satellite embeddings*)

3. Exploratory Data Analysis (EDA)

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

[2]:
```
print(os.getcwd())
```

```
C:\Users\Sameer Kumar\Documents\GitHub\satellite-imagery-property-valuation-clean\notebooks
```

[3]:
```
tab = pd.read_csv("../data/train.csv")
tab.head()
```

[3]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built	yr_reno
0	9117000170	20150505T000000	268643	4	2.25	1810	9240	2.0	0	0	...	7	1810	0	1961	
1	6700390210	20140708T000000	245000	3	2.50	1600	2788	2.0	0	0	...	7	1600	0	1992	
2	7212660540	20150115T000000	200000	4	2.50	1720	8638	2.0	0	0	...	8	1720	0	1994	
3	8562780200	20150427T000000	352499	2	2.25	1240	705	2.0	0	0	...	7	1150	90	2009	
4	7760400350	20141205T000000	232000	3	2.00	1280	13356	1.0	0	0	...	7	1280	0	1994	

5 rows × 21 columns

[4]:
```
tab.shape
```

[4]:
```
(16209, 21)
```

[5]:
```
tab.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16209 entries, 0 to 16208
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             16209 non-null  int64
 1   date           16209 non-null  object
 2   price          16209 non-null  int64
 3   bedrooms       16209 non-null  int64
 4   bathrooms      16209 non-null  float64
 5   sqft_living    16209 non-null  int64
 6   sqft_lot       16209 non-null  int64
 7   floors         16209 non-null  float64
 8   waterfront     16209 non-null  int64
 9   view           16209 non-null  int64
 10  condition      16209 non-null  int64
 11  grade          16209 non-null  int64
 12  sqft_above     16209 non-null  int64
 13  sqft_basement  16209 non-null  int64
 14  yr_built       16209 non-null  int64
 15  yr_renovated   16209 non-null  int64
 16  zipcode        16209 non-null  int64
 17  lat            16209 non-null  float64
 18  long           16209 non-null  float64
 19  sqft_living15  16209 non-null  int64
 20  sqft_lot15     16209 non-null  int64
dtypes: float64(4), int64(16), object(1)
memory usage: 2.6+ MB
```

[6]:
```
tab.describe()
```

[6]:

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sq
count	1.620900e+04	1.620900e+04	16209.00000	16209.000000	16209.000000	1.620900e+04	16209.000000	16209.000000	16209.000000	16209.000000	16209.000000	1620
mean	4.575771e+09	5.374703e+05	3.36782	2.113054	2073.274601	1.486767e+04	1.498828	0.006971	0.234253	3.407860	7.652971	178
std	2.874661e+09	3.603036e+05	0.93327	0.765242	907.009491	3.882570e+04	0.543032	0.083206	0.763152	0.651553	1.171050	82
min	1.000102e+06	7.500000e+04	0.00000	0.000000	290.000000	5.200000e+02	1.000000	0.000000	0.000000	1.000000	1.000000	29
25%	2.123049e+09	3.200000e+05	3.00000	1.500000	1430.000000	5.004000e+03	1.000000	0.000000	0.000000	3.000000	7.000000	120
50%	3.904950e+09	4.500000e+05	3.00000	2.250000	1910.000000	7.599000e+03	1.500000	0.000000	0.000000	3.000000	7.000000	156
75%	7.304301e+09	6.400000e+05	4.00000	2.500000	2550.000000	1.063100e+04	2.000000	0.000000	0.000000	4.000000	8.000000	220
max	9.900000e+09	7.700000e+06	33.00000	8.000000	12050.000000	1.164794e+06	3.500000	1.000000	4.000000	5.000000	13.000000	886

[7]:
```
tab.isnull().sum()
```

[7]:
```
id              0
date            0
price           0
bedrooms        0
bathrooms       0
sqft_living     0
sqft_lot        0
floors          0
waterfront      0
view            0
condition       0
grade           0
sqft_above      0
sqft_basement   0
yr_built        0
yr_renovated    0
zipcode         0
lat             0
long            0
sqft_living15   0
sqft_lot15      0
dtype: int64
```

[8]:
```
tab['zipcode'] = tab['zipcode'].astype(str)
tab['date'] = pd.to_datetime(tab['date'])
```

Plots for visualizing different features:

[9]:
```
plt.figure(figsize=(8,5))
plt.hist(tab['price'], bins=50)
plt.title("Price Distribution")
plt.xlabel("Price")
plt.ylabel("Frequency")
plt.show()
```

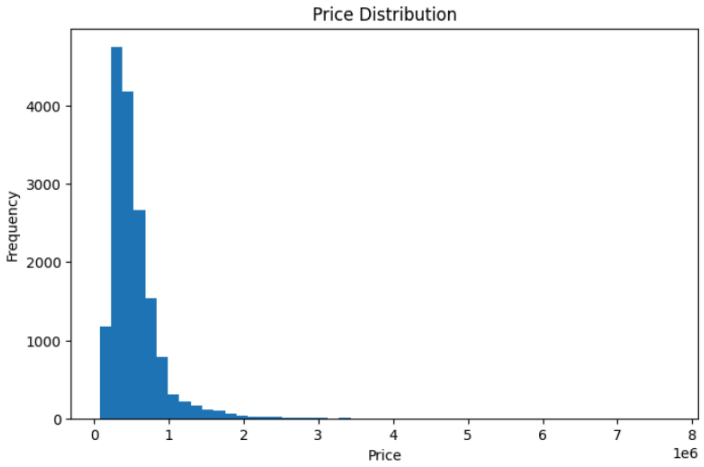


Figure 1: Distribution of house prices showing right-skewed behavior.

[10]:
```
plt.figure(figsize=(8,5))
plt.scatter(tab['sqft_living'], tab['price'], alpha=0.3)
plt.title("Price vs Sqft Living")
plt.xlabel("Sqft Living")
plt.ylabel("Price")
plt.show()
```

Figure 2: sqft_living showing a strong positive correlation with price

[11]:
```
plt.figure(figsize=(10,5))
sns.boxplot(x=tab['bedrooms'], y=tab['price'])
plt.title("Price vs Bedroom Count")
plt.show()
```

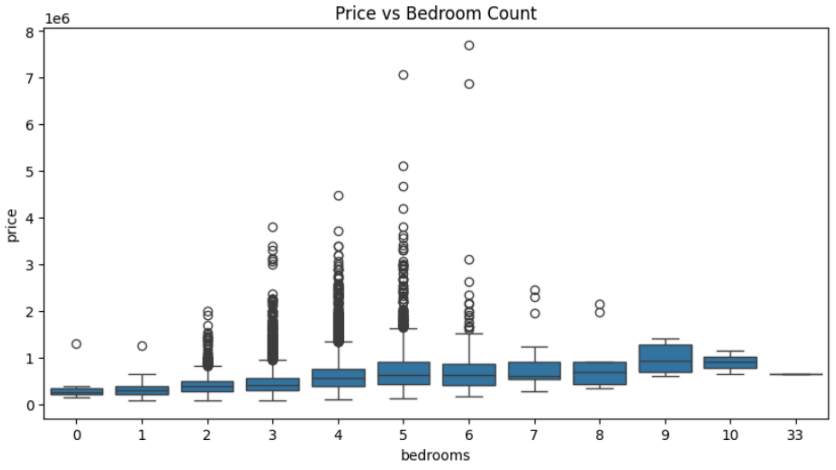


Figure 3: Plot showing that adding bedrooms doesn't always increase price

```
[12]: plt.figure(figsize=(10,5))
      sns.boxplot(x=tab['bathrooms'], y=tab['price'])
      plt.title("Price vs Bathroom Count")
      plt.show()
```

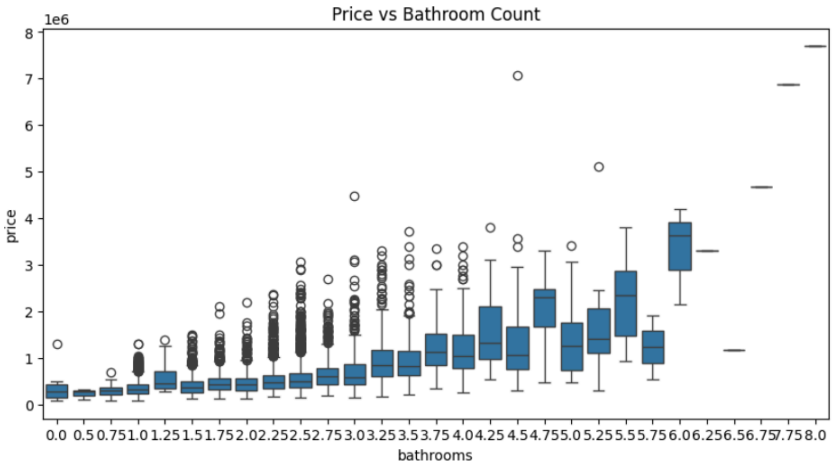


Figure 4: Plot showing that bathrooms correlate better than bedrooms

```
[13]: plt.figure(figsize=(12,8))
      corr = tab.corr()
      sns.heatmap(corr, cmap="coolwarm", annot=False)
      plt.title("Feature Correlation Heatmap")
      plt.show()
```

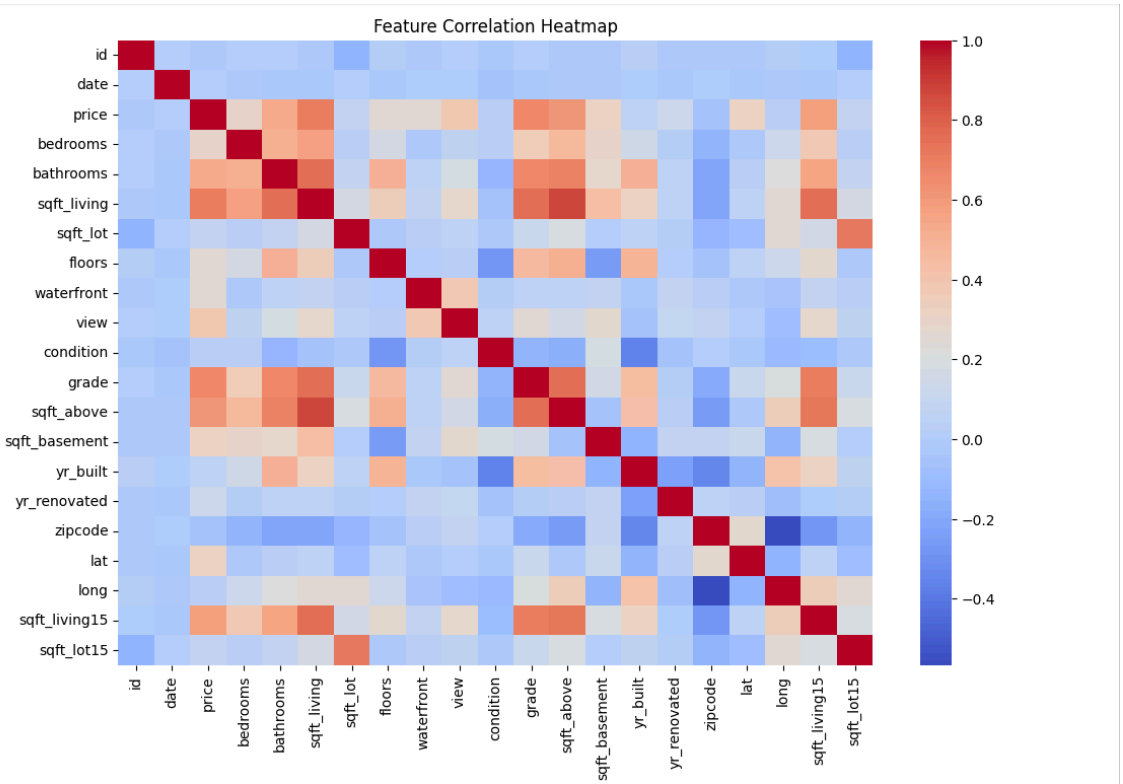


Figure 5: Feature Correlation Heatmap showing that grade, sqft_above, sqft_living, sqft_living15 show strong relationships with price

```
[14]: plt.figure(figsize=(8,6))
plt.scatter(tab['long'], tab['lat'], c=tab['price'], cmap='coolwarm', alpha=0.5)
plt.colorbar(label="Price")
plt.title("Geospatial Price Distribution")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()
```

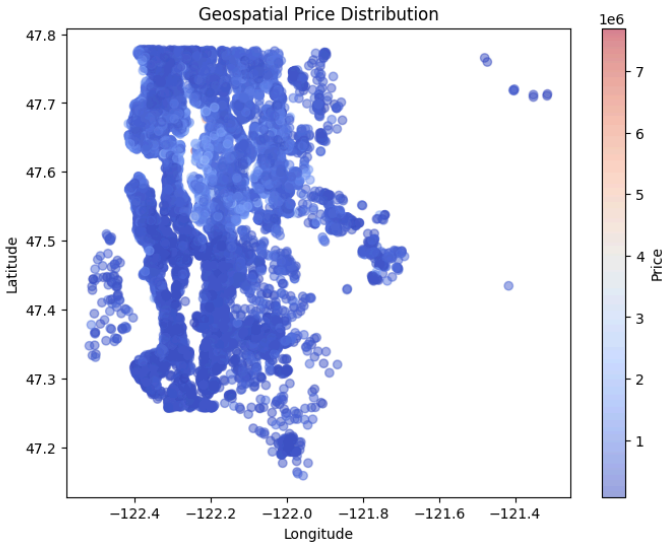


Figure 6: Plot showing that geographic location affects price (price varies with lat & long)

4. Satellite Imagery Pipeline

1. Motivation

Satellite imagery provides valuable contextual information such as green cover, road connectivity, urban density, and proximity to water bodies, which are not captured by traditional tabular property features. To incorporate this visual context, satellite images were used as an additional modality in the prediction pipeline.

2. Image Source & API

Satellite images were fetched using the Sentinel Hub API with Sentinel-2 L1C data. OAuth-based authentication was used to securely access the API. True Color (RGB) imagery was selected to capture realistic visual representations of the property surroundings.

3. Geographic Processing

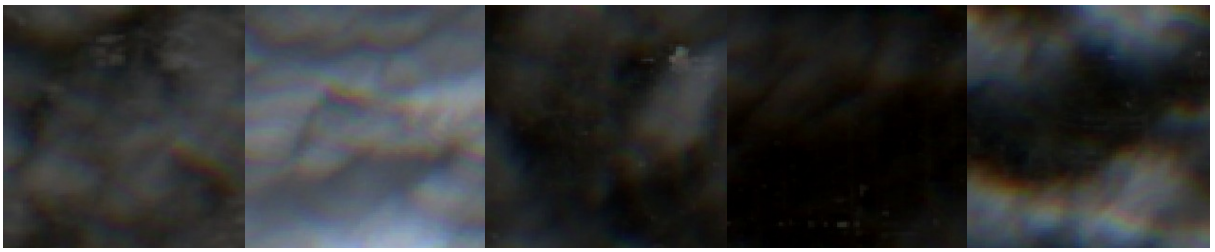
For each property, a geographic bounding box was constructed around the latitude and longitude coordinates. A fixed spatial window was used to ensure consistent context across

properties. Coordinates were transformed from WGS84 to Web Mercator projection before requesting the imagery.

4. Download & Storage Strategy

Each downloaded image was stored in a directory named after the corresponding property ID to maintain traceability between tabular records and images. Due to API rate limits and processing unit constraints, satellite images could be successfully downloaded for a subset of the dataset. Properties without images were excluded from multimodal training but retained for tabular-only evaluation.

Sample satellite images fetched using Sentinel-2 for some property locations:



5. Image Preprocessing

All satellite images were resized to a fixed resolution and normalized to match the input requirements of the convolutional neural network. Standard image preprocessing techniques such as scaling pixel values and channel normalization were applied.

6. Feature Extraction

A pretrained ResNet-50 convolutional neural network was used to extract visual features from the satellite images. The final classification layer was removed, and the output from the global average pooling layer (2048-dimensional embedding) was used as a compact representation of each image.

7. Output of This Pipeline

The extracted image embeddings were stored as a serialized dictionary mapping property IDs to their corresponding feature vectors. These embeddings were later concatenated with scaled tabular features to form the final multimodal feature matrix.

Due to API request limits, some test images could not be fetched; however, the training pipeline and multimodal fusion were fully implemented and evaluated.

5. Models Used

Model	Input Type	RMSE	R2
Linear Regression	Tabular	189,900.02	0.708440
Random Forest	Tabular	121,494.08	0.880659
XGBoost	Tabular	108,089.42	0.905541
XGBoost	Tabular + Image	122,478.75	0.878717

XGBoost achieved the best tabular-only performance and was selected for final test prediction generation.

7. Results & Comparison

Model performance was evaluated using **RMSE** and **R2 score**. Multiple tabular baseline models were trained, including Linear Regression, Random Forest, and XGBoost. Among these, **XGBoost achieved the best tabular-only performance** with an RMSE of approximately **108,089**, and was selected as the baseline model.

A multimodal model was then trained by fusing standardized tabular features with satellite image embeddings extracted using a pretrained ResNet-50 network. The multimodal XGBoost model achieved an RMSE of approximately **122,479**. While this did not outperform the tabular baseline, it successfully demonstrated the integration of satellite imagery into the prediction pipeline.

The performance difference can be attributed to partial satellite image availability, fixed spatial context, and the use of frozen CNN embeddings. Despite this, the multimodal approach highlights the potential of incorporating visual neighborhood information and establishes a foundation for future improvements with higher-quality imagery and enhanced model tuning.

8. Architecture Diagram

1. System Architecture Overview

The proposed system follows a **multimodal learning architecture** that integrates structured tabular data with satellite imagery to predict property prices. The pipeline consists of two parallel feature extraction branches: one for tabular data and one for satellite images, followed by feature fusion and regression.

2. Architecture Flow

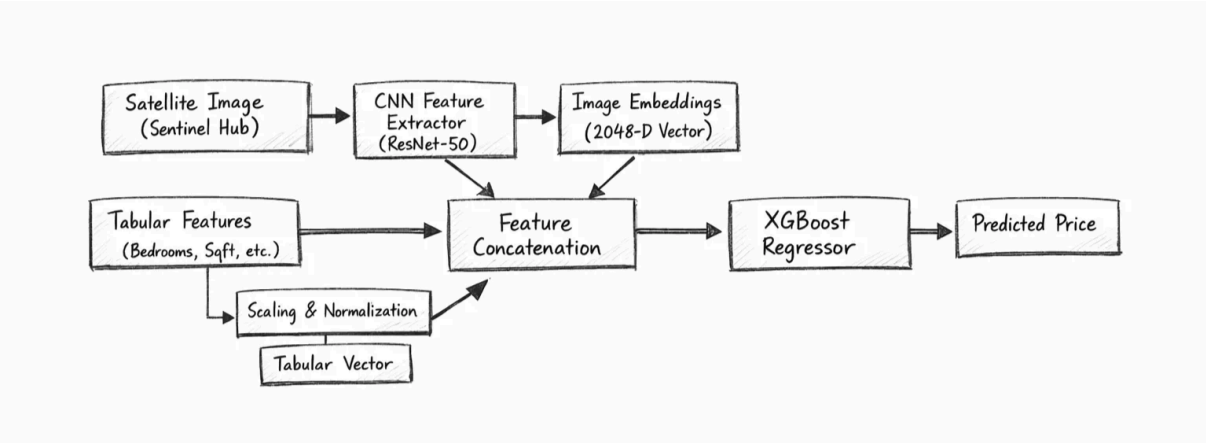


Figure: Multimodal architecture combining CNN-based image embeddings with tabular features for regression

3. Component Description

- **Tabular Branch:** Structured property attributes are cleaned and standardized to ensure consistent scaling before model training.
- **Image Branch:** Satellite images corresponding to property locations are processed through a pretrained **ResNet-50** CNN. The classification head is removed, and deep visual embeddings are extracted to represent neighborhood-level visual context.
- **Feature Fusion:** The scaled tabular feature vector is concatenated with the image embedding to form a unified multimodal representation.
- **Regression Model:** An **XGBoost regressor** is trained on the fused features to predict the final property price.

4. Design Rationale

This architecture allows the model to leverage both **explicit numerical features** and **implicit visual cues from satellite imagery**. The modular design enables independent improvement of each branch and supports scalability to higher-resolution images or alternative deep learning backbones in future work.

9. Conclusion

This project demonstrates an end-to-end multimodal house price prediction pipeline combining tabular property features with satellite imagery. Traditional tabular models, particularly XGBoost, achieved strong predictive performance and served as a reliable baseline. Satellite images were successfully integrated using CNN-based feature extraction, showcasing the feasibility of incorporating visual context such as neighborhood characteristics into property valuation. Although the multimodal model did not outperform the tabular baseline due to limited image coverage and resolution constraints, the framework establishes a scalable foundation for future improvements using higher-quality imagery and advanced fusion techniques.

Thank you for your time and consideration