PROJECT REPORT

Submitted by: Ankit Dhanghar
Branch: Mechanical
Enrollment: 23117023

Content

Project Overview

Accurate property valuation is a fundamental problem in real estate analytics and is traditionally based on structured tabular data such as property size, location, construction quality, and age. While these attributes are essential, they often fail to capture important environmental and neighborhood factors like greenery, road density, and overall visual appeal, which play a significant role in determining market value.

This project presents a **Satellite Imagery–Based Property Valuation System** using a **multimodal regression pipeline** that integrates tabular housing data with satellite images collected using geographic coordinates. By extracting visual features from satellite imagery and combining them with engineered numerical features, the model learns both structural and environmental patterns. This multimodal approach enables more accurate and context-aware property price predictions compared to traditional valuation methods.

Data Preprocessing

Dataset Loading

The datasets were stored on Google Drive and accessed using Google Colab. Two CSV files were loaded:

- **Training dataset (`train.csv`)**
- **Testing dataset (`test.csv`)**

The datasets were successfully read into Pandas DataFrames for analysis and preprocessing.

Initial Data Inspection

To understand the structure and content of the dataset:

- The first few rows of the dataset were examined
- The shape of the dataset was checked to identify the number of records and features
- Column names and data types were reviewed

This step helped in gaining a preliminary understanding of numerical and categorical attributes.

Handling Missing Values

A thorough check for missing values was performed across all features in the dataset.

Observation:

- No missing values were found in the dataset.

This indicates that the dataset is complete and does not require imputation or removal of records due to missing data.

Duplicate Data Analysis

Duplicate records can introduce bias in model training. Therefore, the dataset was checked for duplicate rows.

Observation:

- Few duplicate ids rows were found in the dataset.
- Later they were dropped

Statistical Summary

Descriptive statistics were generated for numerical features using summary functions. This provided insights into:

- Mean, median, minimum, and maximum values
- Standard deviation and data spread
- Presence of skewness or extreme values

These statistics helped identify potential outliers and scale differences among features.

Categorical Feature Analysis

Several categorical and ordinal features were identified, including:

- Number of bedrooms
- Number of floors
- Waterfront presence
- View quality
- Property condition
- Property grade

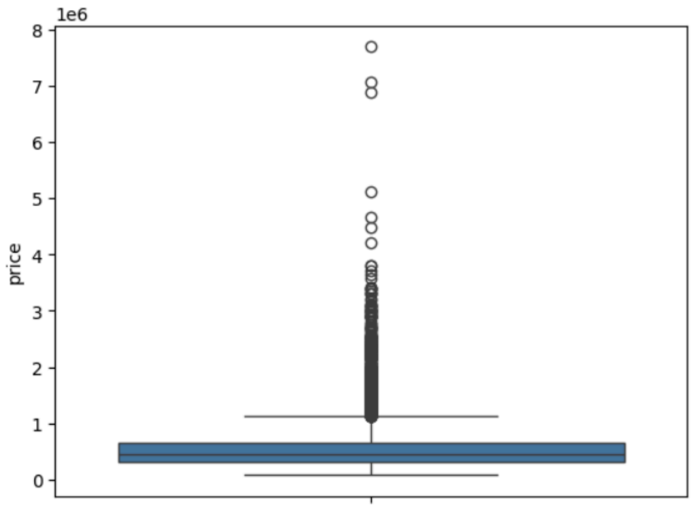
For each categorical feature:

- Frequency distributions were analyzed using value counts
- Class imbalance and dominant categories were identified

This analysis is important for understanding how categorical variables may influence the target variable.

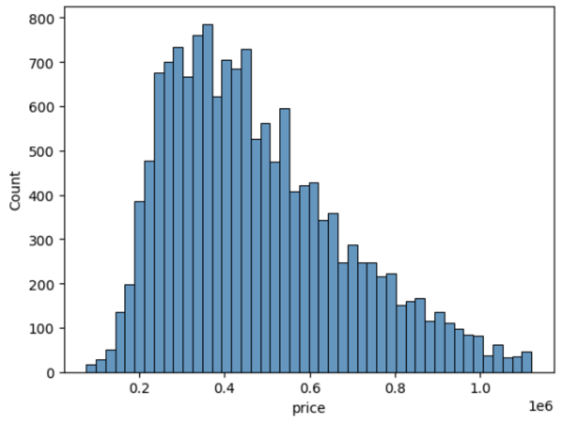
Outlier Detection

Outliers were examined in **price columns** using visual techniques such as box plots and distribution plots. This helped in identifying extreme values that could negatively impact model performance. They are removed by the IQR method.

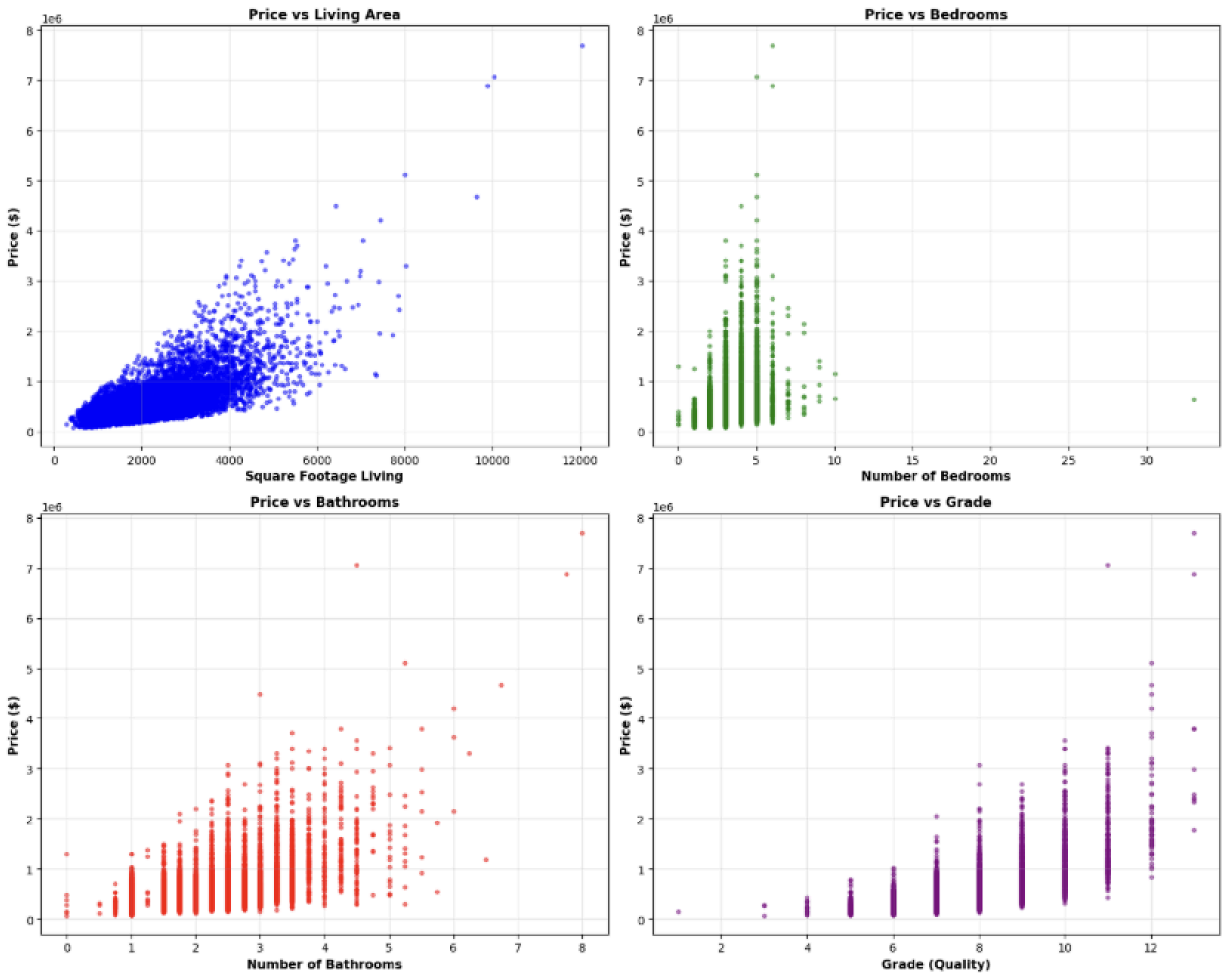


Skewness

Some skewness is also present in “price” columns. It is handled by log transformation.



Columns Relationship with the price



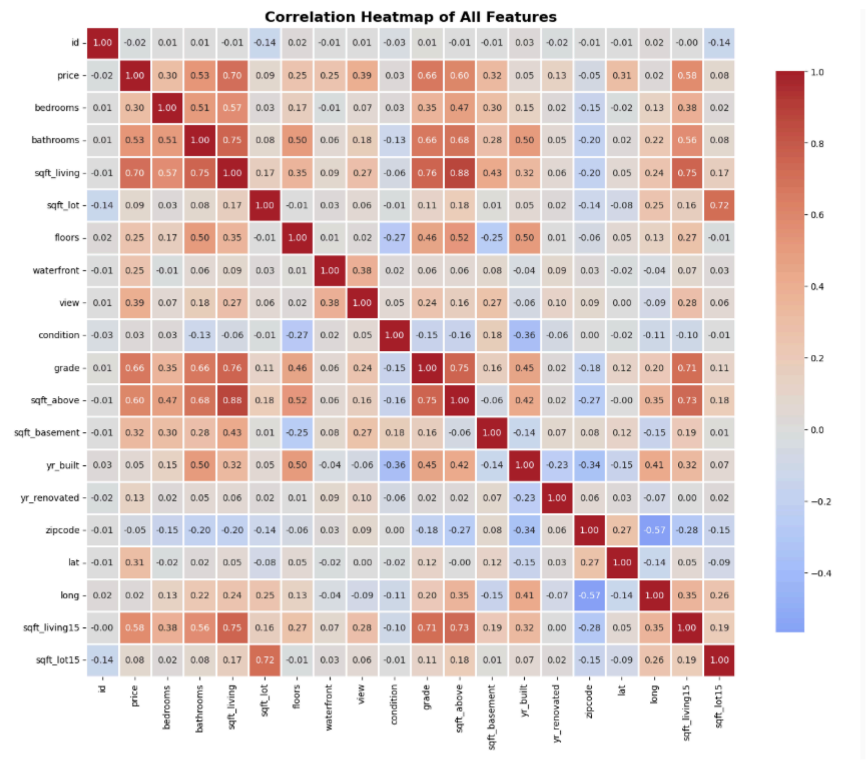
Square Footage Living Area:
House price increases significantly with an increase in living area, showing a strong positive correlation. Larger houses also exhibit greater price variation, indicating the influence of additional factors such as location and quality.

Number of Bedrooms:
The relationship between bedrooms and price is weak and non-linear. Houses with similar bedroom counts often have widely varying prices, suggesting that bedroom count alone is not a strong predictor.

Number of Bathrooms:
Price generally increases with the number of bathrooms. Compared to bedrooms, bathrooms show a clearer and more consistent positive impact on house prices.

Grade (Quality):
A strong positive trend is observed between grade and price. Higher-grade properties consistently command higher prices, highlighting construction quality as a key determinant of house value.

Correlation Heatmap



TOP FEATURES CORRELATED WITH PRICE

price	1.000000
sqft_living	0.700933
grade	0.664266
sqft_above	0.602648
sqft_living15	0.581781
bathrooms	0.525487
view	0.390534
sqft_basement	0.320301
lat	0.310008
bedrooms	0.304454
floors	0.251428
waterfront	0.245221

yr_renovated	0.133075
sqft_lot	0.088526

Feature Engineering

Total Rooms

The total number of rooms was calculated by summing the number of bedrooms and bathrooms. This feature provides a more complete representation of the usable interior space compared to considering bedrooms and bathrooms separately.

Living Area Relative to Neighbors

This feature measures how large a property is compared to surrounding houses by dividing the living area by the average living area of nearby properties. It captures neighborhood context and relative property scale, which can significantly influence market value.

Built-Up Area Indicator

A built-up area indicator was derived from satellite imagery by inverting the green ratio. Higher values represent denser construction and reduced green cover around the property.

House Age

House age was computed by subtracting the year of construction from the current year. This representation is more intuitive and informative than using the raw construction year.

Renovation Indicator

A binary renovation feature was created to indicate whether a property has undergone renovation. This helps differentiate older houses that have been upgraded from those that have not.

Green-Adjusted Living Area

The living area was adjusted using the surrounding green ratio obtained from satellite imagery. Properties with similar sizes but greener surroundings are assigned higher effective values.

Urban Clutter Score

This feature combines the living-area comparison with edge density from images to quantify urban congestion and visual complexity around the property.

Natural Amenity Score

A composite score was created by combining green ratio and water ratio, representing the availability of natural environmental features near the property.

Neighborhood Harmony Index

Neighborhood harmony was calculated by comparing property size with surrounding built-up density. Higher values indicate better integration of the property within its neighborhood.

Luxury Environment Score

A weighted luxury score was engineered using normalized property grade, view quality, and surrounding greenery. This feature captures high-end environmental and structural characteristics.

Image Data Collection Using Mapbox API

To enrich the dataset with visual information, satellite images corresponding to each property location were collected using the **Mapbox Static Images API**. This step enables the integration of image data with tabular features for further analysis and modeling.

API Token Configuration

The Mapbox API access token was securely retrieved from **Google Colab Secrets** to ensure safe authentication. A validation check was performed to confirm successful loading of the token before initiating any data downloads.

Image Collector Initialization

A custom image collection module was used to manage the downloading process. The image collector was initialized with:

- Image resolution of **256 × 256 pixels**
- Zoom level set to **18** for high-detail satellite imagery
- Latitude and longitude values extracted from the cleaned dataset

Batch-Wise Image Downloading

To efficiently handle large volumes of data and avoid API rate limits, images were downloaded in **batches of 1000 properties**. Each batch processed a subset of the dataset sequentially.

For every property:

- Latitude and longitude coordinates were passed to the Mapbox API
- Images were saved using a unique property identifier
- Previously downloaded images were skipped to avoid redundancy
- A short delay was added between requests to comply with API usage limits

Progress Tracking and Error Handling

During the download process:

- Successful and failed downloads were counted
- Real-time progress updates were displayed for monitoring
- Batch-level execution time was recorded to analyze performance

This ensured transparency, robustness, and recovery from interruptions.

Image–Tabular Data Mapping

After completing the image downloads, an **image-to-property mapping file** was generated. This mapping links each property ID to its corresponding image file path, enabling seamless integration of image data with tabular features during model training.

The same steps are done to download both train and test images.

Image Feature Extraction Using Convolutional Neural Networks

To extract meaningful visual features from the satellite images, a **deep learning–based feature extraction pipeline** was implemented using a pre-trained **ResNet-50** convolutional neural network (CNN). This approach allows high-level visual patterns to be captured without training a CNN from scratch.

Model Selection and Configuration

The **ResNet-50** model, pre-trained on the ImageNet dataset, was used as a fixed feature extractor. The final classification layers were removed, and **global average pooling** was applied to obtain compact feature embeddings for each image. All images were resized to **224 × 224 pixels**, which is the standard input size for ResNet-50.

Batch-Wise Image Processing

Images were processed in **batches of 32** to optimize memory usage and computational efficiency. A batch generator was used to sequentially load images from disk, ensuring scalability for large datasets.

Checkpointing for Fault Tolerance

To handle long execution times and avoid reprocessing already analyzed images, a **checkpoint mechanism** was implemented. After each batch:

- Extracted features and metadata were saved to disk
- Previously processed images were skipped during subsequent runs
- Processing could be resumed seamlessly in case of interruptions

This significantly improved robustness and reliability.

CNN Feature Extraction

Each image was passed through the ResNet-50 network to obtain a **high-dimensional feature embedding** representing visual characteristics such as texture, structure, and spatial patterns. From these embeddings, statistical summaries were computed:

- Mean
- Standard deviation
- Maximum value
- Minimum value

These statistics provide compact numerical representations of complex image features.

Hand-Crafted Visual Features

In addition to CNN embeddings, simple domain-specific visual features were extracted directly from raw images:

- **Green ratio:** Indicates vegetation density
- **Water ratio:** Highlights presence of water bodies
- **Edge density:** Represents structural complexity in the image

These features complement CNN representations by capturing interpretable visual cues.

Feature Storage and Export

All extracted features were organized into a structured **tabular format**, where each row corresponds to a property image. The final image feature dataset was saved as a **CSV file**, enabling easy integration with tabular property data for multimodal modeling.

Features explanation

cnn_mean: The average value of all CNN-extracted features. It represents the overall strength or presence of learned visual patterns in the image.

cnn_std: Measures how much the CNN features vary across the image. Higher values indicate more visual diversity and complex patterns.

cnn_max: The maximum value among CNN features. It captures the strongest activated visual pattern detected by the CNN**.**

cnn_min: The minimum value among CNN features. It represents the weakest or least activated visual response in the image.

green_ratio: The proportion of pixels dominated by green color. It is commonly used to estimate vegetation or plant cover in the image.

water_ratio: The proportion of pixels identified as water-like based on color characteristics. It helps detect rivers, lakes, or other water bodies.

edge_density: The ratio of edge pixels to total pixels in the image. Higher edge density indicates more texture, boundaries, or structural complexity.

This was done to make learning easier and more reliable by keeping only what matters most. By compressing and summarizing, repetition and noise are reduced while important patterns remain. With less to process, training becomes faster, results generalize better, and instability is avoided**.**

Baseline Model Training Using Tabular Data Only

To establish a baseline performance, multiple regression models were trained using **only tabular features**, without incorporating image-based information. This helps in understanding how well traditional machine learning models perform before applying multimodal techniques.

Models Implemented

Three regression models with increasing complexity were trained and evaluated:

- **Linear Regression:** Used as a simple baseline to capture linear relationships between features and target variables.
- **Random Forest Regressor:** An ensemble-based model capable of learning non-linear feature interactions.
- **XGBoost Regressor:** A gradient boosting model optimized for high predictive accuracy and robustness.

All models were trained on the same training dataset to ensure a fair comparison.

Model Evaluation Metrics

Performance was evaluated using standard regression metrics:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- R² Score

Results and Comparison:

Model	MAE	RMSE	R2-score
Linear Regression	0.1894	0.2446	0.7019
Random Forest	0.1228	0.1759	0.8458
XGBoost	0.1162	0.1650	0.8643

Multimodel Training

To evaluate the impact of combining image-based features with tabular data, a **feature-level fusion approach** was adopted. Image-derived features were integrated with structured tabular features, and the resulting dataset was used to train predictive models.

Feature Normalization

Selected image features—**green ratio, water ratio, edge density, and built-up area**—were normalized using **z-score standardization**. This ensured that all features had zero mean and unit variance, preventing any single feature from dominating the learning process.

Data Alignment and Fusion

To maintain correct correspondence between tabular and image data:

- Image features were aligned using the **image identifier**
- Tabular features were aligned using the **property ID**
- Both datasets were sorted and concatenated column-wise

This resulted in a unified **multimodal feature set** representing both numerical property attributes and visual characteristics.

XGBoost on Fused Features

An **XGBoost regressor** was trained on the fused dataset with carefully tuned hyperparameters to capture non-linear interactions across tabular and image features. The model demonstrated strong predictive performance.

- R² Score (XGBoost – Fused Features): **0.8843(Best_model)**

Deep Learning Hybrid Fusion Model

A deep learning–based hybrid fusion model was also trained using combined tabular and image representations. However, this model showed significantly lower predictive performance compared to XGBoost.

- R² Score (Deep Learning Hybrid Model): **0.3388**

Performance Comparison and Analysis

The comparison indicates that:

- **XGBoost significantly outperformed** the deep learning hybrid model
- Tree-based ensemble methods are more effective for structured tabular data
- The deep learning model likely requires more data, better architecture tuning, or end-to-end image learning to perform competitively

Conclusion

The results demonstrate that **classical machine learning models with feature-level fusion** can outperform deep learning hybrids in structured datasets. Consequently, **XGBoost was selected as the final model** due to its superior accuracy and robustness.

Final Prediction on Test Data

After completing data preprocessing, feature engineering, and model training, the finalized pipeline was applied to the **test dataset** to generate house price predictions. To ensure consistency and reliability, the **same preprocessing and feature transformation steps used during training** were strictly followed for the test data.

Replication of Training Preprocessing Steps

The test data underwent identical operations as the training data, including:

- Normalization of image-derived features
- Alignment of tabular and image datasets using property identifiers
- Generation of engineered features such as house age, renovation indicator, environmental scores, and neighborhood metrics

- Feature selection and ordering to match the trained model input structure

This ensured that the trained model received data in a consistent format, preventing data leakage or feature mismatch issues.

Model Inference

The optimized **XGBoost regression model**(multimodel) selected as the final model based on superior performance, was used to predict house prices on the processed test dataset. The model generated continuous price predictions for each property record.

Result Storage

The predicted values were combined with their corresponding property identifiers and stored in a structured format. The final predictions were exported as a **CSV file**, enabling easy submission, evaluation, and further analysis.