MBA Semester – IV

Research Project – Final Report

Name	Aditi Bagish
Project	231VMBR00159
Group	Data Science and Analytics
Date of Submission	18/05/2025

A study on "Precision Property Insights"

Research Project submitted to Jain Online (Deemed-to-be University)

In partial fulfillment of the requirements for the award of:

Master of Business Administration

Submitted by:

Aditi Bagish

USN:

231VMBR00159

Under the guidance of:

Mr. Sharath Srivatsa

Jain Online (Deemed-to-be University)

Bangalore

2023-25

DECLARATION

I, *Aditi Bagish,* hereby declare that the Research Project Report titled *"Precision Property Insights" has been* prepared by me under the guidance of the *Mr. Sharath Srivatsa.* I declare that this Project work is towards the partial fulfillment of the University Regulations for the award of the degree of Master of Business Administration by Jain University, Bengaluru. I have undergone a project for a period of Eight Weeks. I further declare that this Project is based on the original study undertaken by me and has not been submitted for the award of any degree/diploma from any other University / Institution.

Place: Bengaluru

Date: 18/05/2025

Aditi Bagish

Name of the Student

USN: 231VMBR00159

Table of Contents

1. Introduction

1.1 Problem Statement

Accurate property valuation has long posed a challenge within the real estate domain due to the complex interplay of factors influencing housing prices. Variables such as geographical location, architectural features, environmental surroundings, and socio-economic indicators contribute to the dynamic nature of real estate pricing. Traditional valuation methodologies, which often rely on manual appraisals and comparable sales analysis, are susceptible to human bias, regional inconsistencies, and a lack of adaptability in fluctuating market conditions.

This project seeks to address these limitations by applying a data-driven approach to predict property prices in specific U.S. regions. Utilizing a comprehensive dataset that includes variables like lot size, proximity to coastline, number of bedrooms and bathrooms, structural attributes, and quality indicators, the aim is to build a machine learning model capable of delivering consistent and scalable price estimations. The outcome is intended to support real estate professionals, investors, and buyers in making more informed and objective decisions.

1.2 Need for the Project

With data-driven decision-making becoming a cornerstone of modern industries, the real estate sector faces increasing pressure to adopt intelligent, automated systems for property valuation. Traditional appraisal methods often lack responsiveness to market trends and may fail to capture micro-level pricing signals that influence consumer behavior and investment strategies.

This project fulfills the need for an intelligent pricing solution by:

- Automating the valuation process using machine learning techniques;
- Extracting patterns and correlations not immediately evident through manual analysis;
- Providing stakeholders with real-time, reliable price forecasts;
- Minimizing subjective errors and enhancing confidence in market decisions.

By integrating exploratory data analysis with predictive modeling, the project delivers a transparent and reproducible framework that elevates the standard for property pricing in contemporary markets.

1.3 Scope of the Project: Business and Social Relevance

The U.S. residential real estate market surpasses $2.5 trillion in annual transactions, yet pricing inconsistencies remain prevalent. A key contributor to this inconsistency is the heavy reliance on subjective evaluation and outdated comparables. Through the use of machine learning and robust data preprocessing, this project aims to introduce a systematic, unbiased approach to property valuation.

1.3.1 Business Opportunity

- **Scalability**: A data-driven pricing model can be deployed across diverse geographical zones, property categories, and price ranges.
- **Operational Efficiency**: Reduces dependency on time-consuming appraisals and expedites sales cycles.
- **Decision Intelligence**: Enables real estate agencies, property technology (PropTech) firms, mortgage lenders, and institutional investors to make more accurate assessments.
- **Market Competitiveness**: Enhances customer engagement through personalized, data-backed recommendations.

According to industry data from Zillow, even a 5% deviation in price estimation can significantly impact purchase decisions and sales closure rates. Thus, improving prediction accuracy not only benefits businesses but also increases buyer confidence.

1.3.2 Social Impact

- **Transparency**: Provides clearer insights into pricing logic, benefiting first-time buyers and consumers in economically marginalized areas.
- **Fair Valuation**: Reduces risks of overvaluation or undervaluation, fostering equitable market access.
- **Adaptability to Change**: In light of evolving remote work dynamics and migration patterns post-2020, flexible and responsive pricing models are more essential than ever.

Ultimately, the project positions itself at the intersection of data science, real estate economics, and consumer empowerment—delivering solutions that are equitable, efficient, and adaptable to the modern property landscape.

- **Project Link**: PrecisionPropertyInsights.ipynb

2. Basic Data Decoding

This phase focuses on the essential preparation of raw data, including validation, cleansing, and feature selection. It ensures the data is logically structured and ready for deeper analysis.

2.1 Dataset Access and Loading

- **Environment:** Google Colab with Google Drive integration.
- **Methodology:** Mounted Drive and loaded Excel file into a DataFrame using `pandas.read_excel()`.

2.2 Initial Structural Assessment

To understand the data structure and identify potential issues, the following functions were used:

- *`df.head()`* – Previewed sample rows.
- *`df.columns`* – Retrieved column names.
- *`df.info()`* – Summarized:
  - Total records
  - Feature count
  - Data types
  - Non-null values
- *`df.describe()`* – Provided statistics such as:
  - Mean, standard deviation, min/max, and quartiles.

Feature	Action	Rationale
`cid`	Dropped	Unique identifier; no predictive value
`total_area`	Dropped	Derived from other features; risk of multicollinearity

Table 1: Elimination of Redundant Features

2.3 Handling Missing Data

- Total Records: 21,613
- Records with No Missing Values: 21,386
- Records with Missing Values and garbage/irrelevant data: 227 (approx. 1.05%)

Decision: Drop rows with missing values due to the low percentage.
Justification:

- Ensures compatibility with ML models that do not support nulls.
- Avoids introducing bias from imputation in a small subset.

2.4 Non-Numeric Value Detection and Correction

- Some features (e.g., `ceil`) contained invalid characters.
- A helper function identified non-numeric entries.
- `pd.to_numeric(errors='coerce')` converted values and introduced NaNs for invalid ones.
- These NaNs were subsequently dropped.

Post-cleaning checks:

- All relevant numeric columns had correct data types.
- Dataset integrity was preserved.

Step	Operation	Purpose
Data Access	Mounted Drive, loaded Excel file	Enabled access in Colab
Preliminary Inspection	Used `head()`, `info()`, `describe()`	Understood structure and statistics
Feature Elimination	Dropped `cid`, `total_area`	Removed redundancy
Missing Data Handling	Dropped ~1.05% rows	Ensured data quality
Type Correction	Converted string-formatted numbers to float	Ensured consistency for modeling

Table 2: Data Inspection Summary

2.5 Importance

- Established data cleanliness and structural consistency.
- Eliminated noise and redundant information.
- Handled missing and malformed entries appropriately.
- Prepared the dataset for meaningful transformations and machine learning.

3. Feature Engineering Part 1:

Attribute Understanding and Transformations

In this phase, each attribute was carefully examined for relevance, transformation opportunities, and encoding suitability for machine learning workflows.

Feature	Action / Transformation	Rationale	Attribute Type
`cid`	Dropped	No predictive power	Identifier
`dayhours`	Split into `sold_year`, `sold_month`, `sold_weekday`	Captures temporal patterns and seasonality	Temporal
`zipcode`	Reduced to `location_code`	Decreased cardinality while preserving locality	Geographic
`lat`, `long`	Retained	Valuable for geospatial modeling or mapping	Geographic
`lot_measure`, `living_measure`	Retained	Highly relevant to property valuation	Property Dimensions
`ceil_measure`	Dropped	Redundant; `ceil` retained	Property Dimensions

living_measure15, lot_measure15	Retained	Captures year 2015 property characteristics	Property Dimensions
room_bed, room_bath	Retained	Strong influence on property utility	Structural Features
basement	Transformed to is_basement (binary)	Easier interpretation	Structural Features
yr_built	Converted to age	More intuitive than construction year	Structural Features
yr_renovated	Transformed to is_renovated (binary)	Indicates recent upgrades	Structural Features
sight, condition, quality	Retained	Ordinal indicators of visual and structural quality	Quality Ratings
coast	Retained	Coastal proximity affects market value	Categorical (Location-based)
furnished	Retained	May influence property appeal and pricing	Categorical (Amenities)
price	Retained	Central prediction variable	Target Variable

Table 3: Transformed Features

Original Column	New Column / Transformation	Reason
`cid`	Dropped	Not predictive
`dayhours`	`sold_year`, `sold_month`, `sold_weekday`	Time-based insights
`basement`	`is_basement`	Binary format improves interpretability
`yr_built`	`age`	More meaningful than raw year
`yr_renovated`	`is_renovated`	Encodes renovation status
`zipcode`	`location_code`	Reduced cardinality
`ceil_measure`	Dropped	Redundant
Non-numeric columns	Converted to float	Numeric consistency
Rows with missing	Dropped (~1.05%)	Minimal data loss

Table 4: Feature Summary After Transformation

- **Improved Interpretability**: Clearer and more meaningful features.
- **Model Readiness**: Ensured numeric and binary formats suitable for ML models.
- **Reduction of Redundancy**: Eliminated unnecessary columns.
- **Encoding Preparedness**: Categorical features now ready for one-hot or label encoding.

.

4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in understanding the dataset's structure, distributions, and relationships between variables. This section covers the process of univariate and bivariate analysis, the handling of outliers, and feature engineering.

4.1 Univariate Analysis

Univariate analysis helps to understand the distribution of individual features, their central tendencies, spread, and outliers. In this process, continuous and categorical features were analyzed separately to derive insights.

- **Continuous Features**: Features such as `lot_measure15`, `living_measure15`, and `ceil_measure` were analyzed using histograms with kernel density estimates (KDE), providing insights into their distributions. For example, `lot_measure15` showed a skewed distribution with a few high-value outliers, prompting the need to remove extreme values. Box plots further highlighted the presence of outliers, and they were removed using the interquartile range (IQR) method. The removal of outliers in the `lot_measure15` feature resulted in an 8.87% data loss.

- **Categorical Features**: Features like `furnished` and `basement` were visualized using histograms to assess the distribution of categories. The `basement` feature, for example, revealed that 60% of records had a value of 0, indicating that a significant portion of properties had no basement. This feature was subsequently transformed into a binary variable, `is_basement`, with values indicating whether a basement was present or not.

- **Feature Transformation**: The `yr_renovated` feature was transformed into a binary variable `is_renovated`, which marked whether the property had undergone renovation. Similarly, the `yr_built` feature was converted to `age` by calculating the difference between the current year and the year of construction. This transformation enabled a more meaningful interpretation of the data, especially when considering the age of properties in relation to their price.

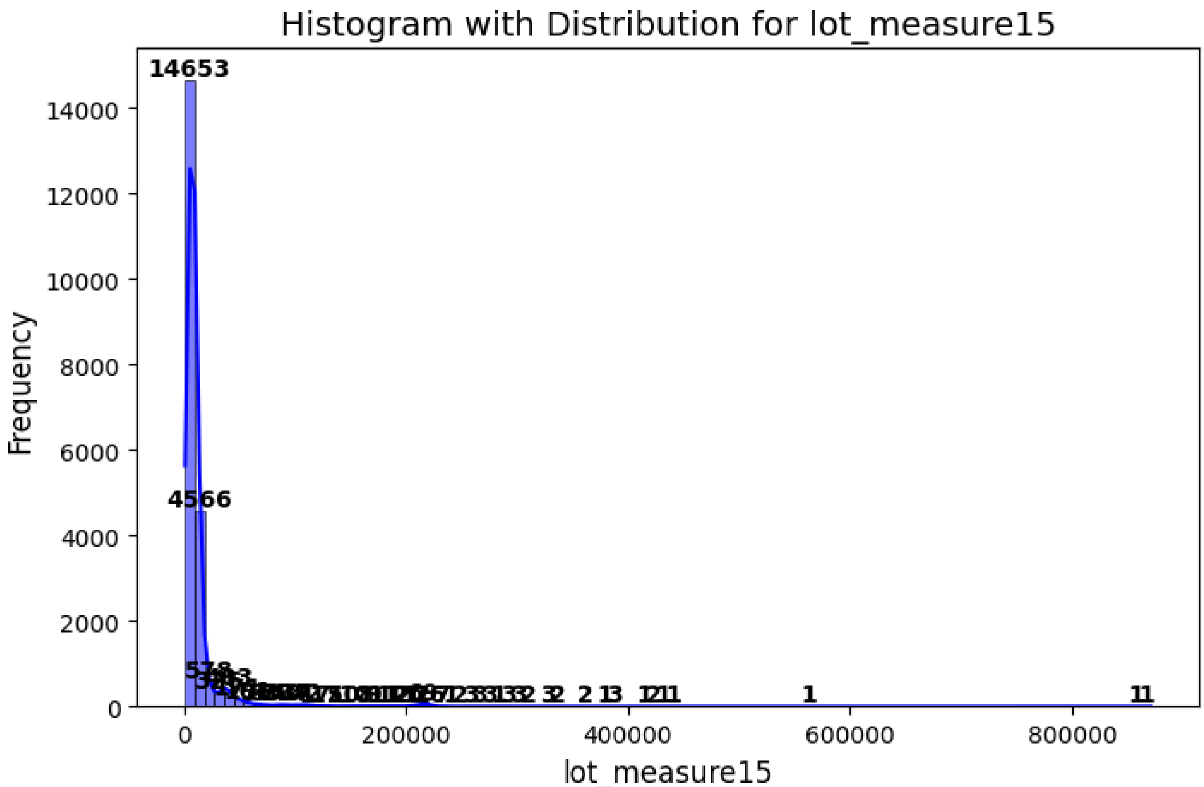

Figure 1: Histogram describing effect of Outliers

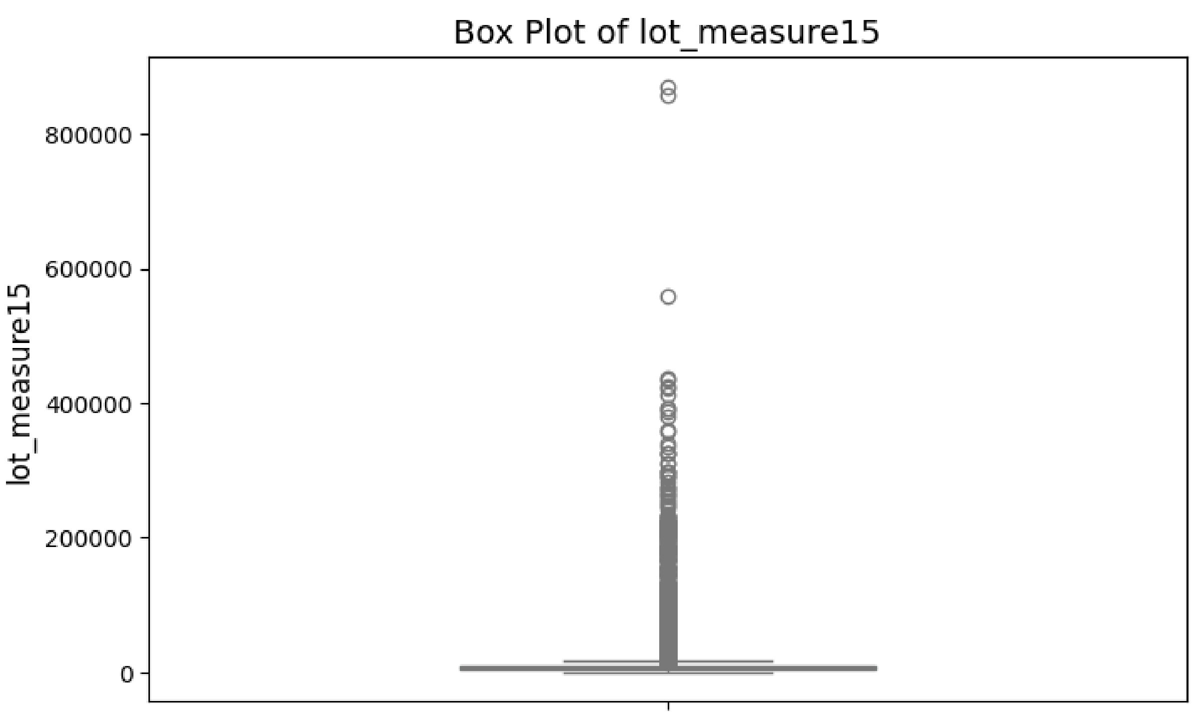

Figure 2: Box Plot describing effect of Outliers

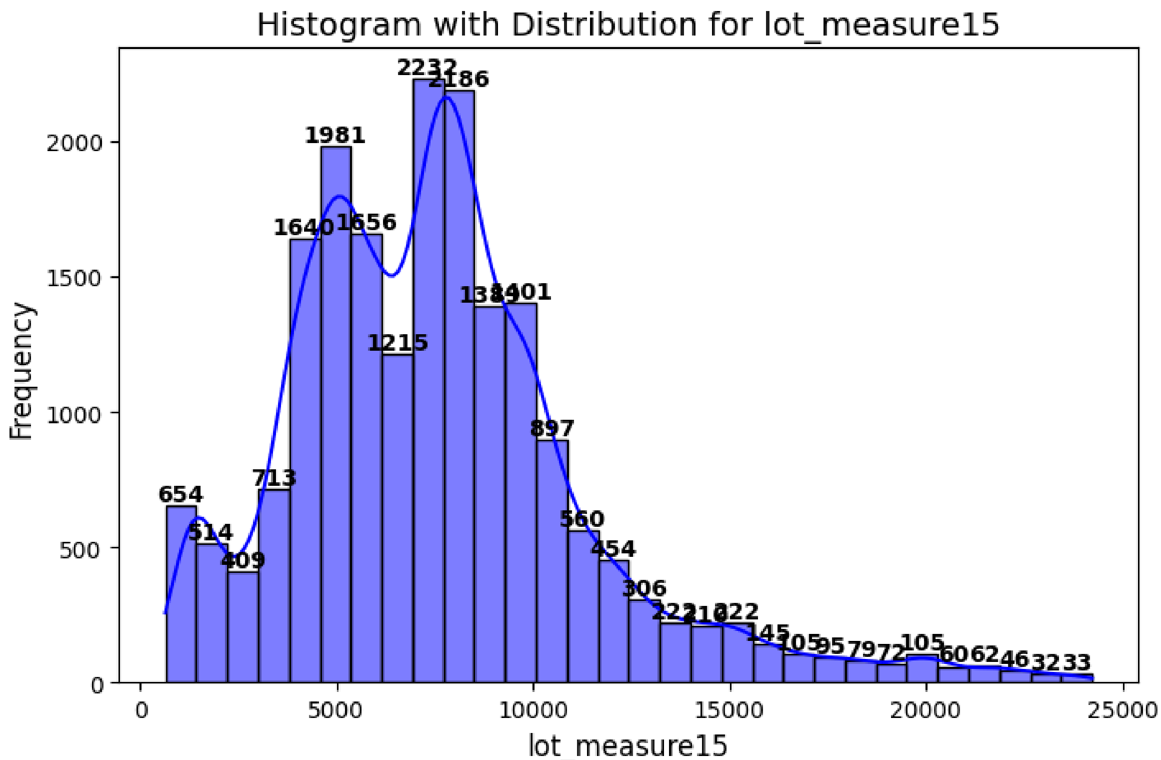

Figure 3: Histogram after Outlier Removal

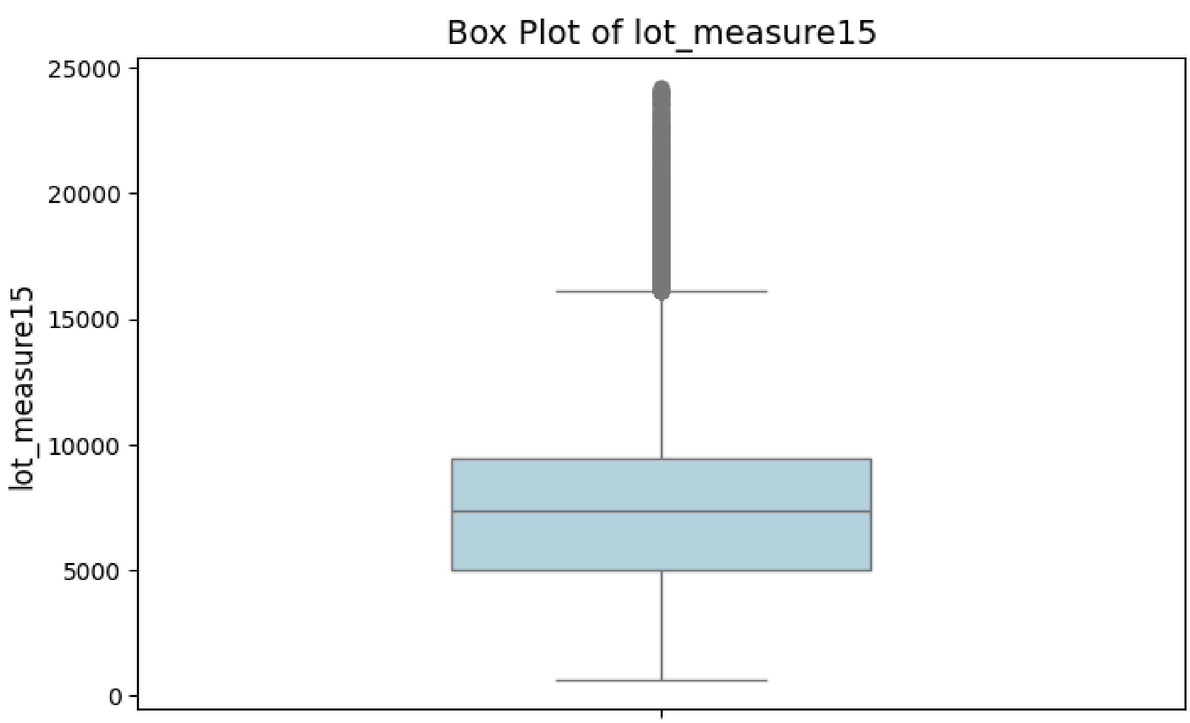

Figure 4: Boxplot after Outlier Removal

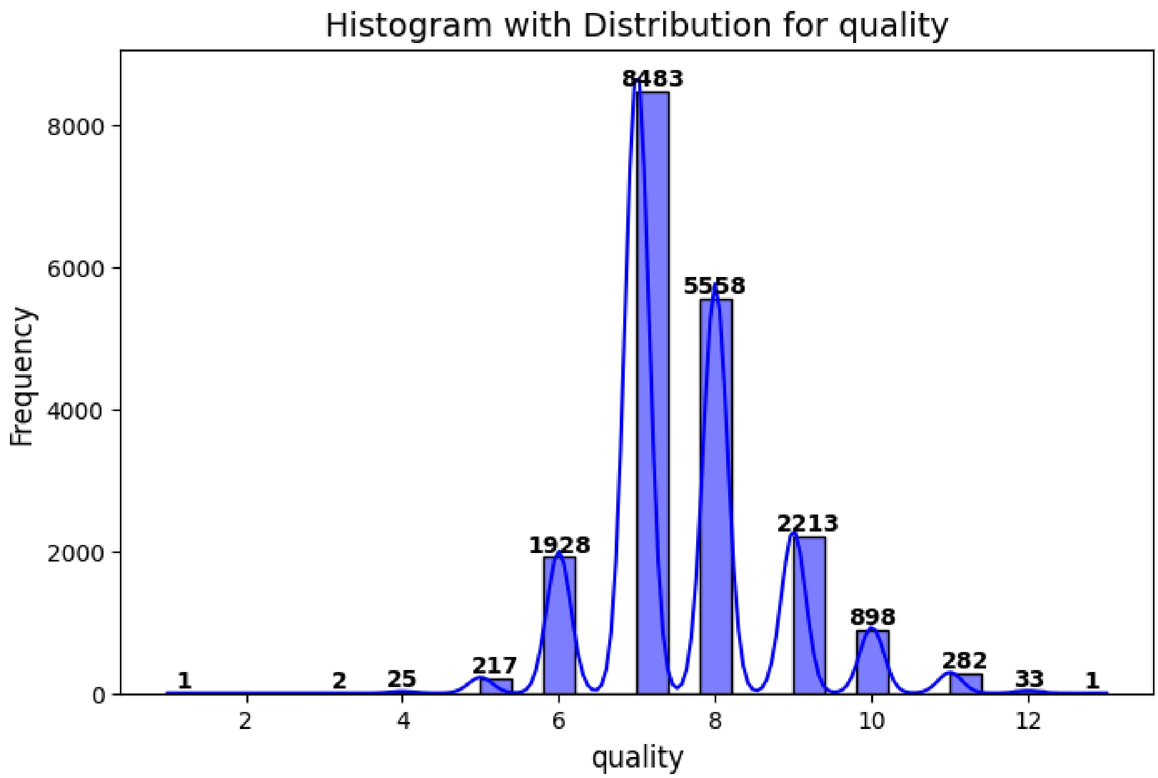

Figure 5: Categorical Features with discrete data

4.2 Bivariate Analysis

Bivariate analysis focuses on examining relationships between pairs of variables to uncover patterns and correlations.

- **Pairwise Relationships**: A pair plot was used to examine the relationships between continuous features such as `living_measure`, `lot_measure`, and `price`. This analysis provided a visual representation of how these features relate to each other. Strong positive correlations between `living_measure` and `price` were observed, which is intuitive, as larger living spaces typically result in higher property prices. Other continuous features, such as `ceil_measure`, also showed correlations with price but were less pronounced.
- **Correlation Matrix**: A correlation matrix was computed for all continuous features, providing quantitative insights into the strength and direction of relationships. For instance, `living_measure` and `price` showed a high positive correlation, indicating that larger properties tend to have higher prices. This information is vital for building predictive models, as these features would likely play a significant role in determining property prices.

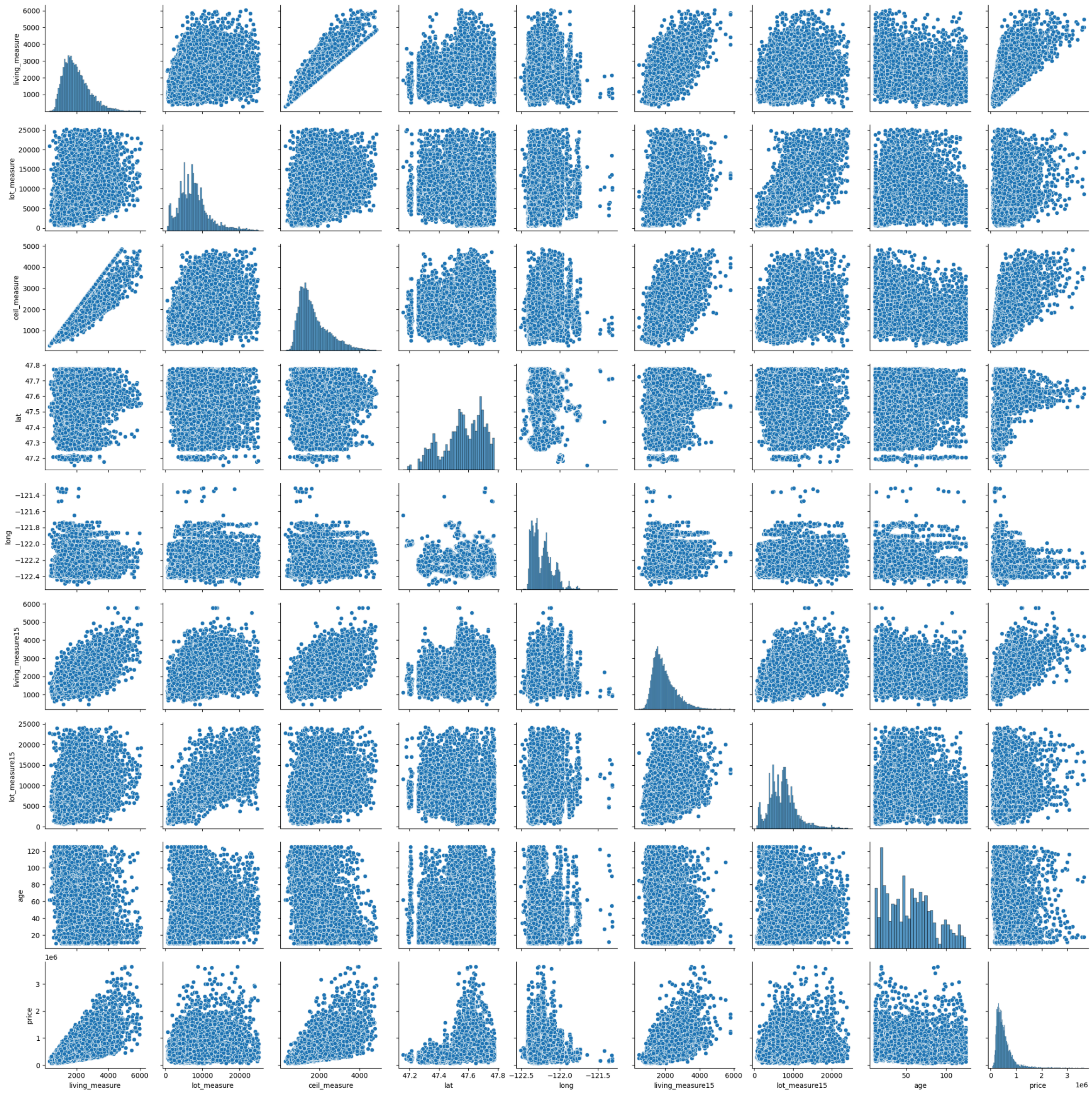

Figure 6: PairPlot of all relevant continuous numerical feature

4.3 Data Wrangling and Feature Optimization

Feature engineering is the process of transforming raw data into meaningful features that can improve the performance of machine learning models.

- **Categorical Feature Transformation**: The `zipcode` feature was recoded into broader regional categories by applying floor division to group similar zipcodes. This reduced the number of unique values and made the feature more manageable for machine learning algorithms. The new `location_code` feature represents larger geographic areas, making it a more generalized categorical variable while preserving essential regional information.

- **One-Hot Encoding**: Several categorical variables, including `location_code`, `sold_year`, `sold_quarter`, and `sold_weekday`, were transformed using one-hot encoding. This process creates binary variables for each category, which helps machine learning models understand categorical data in a numerical format. One-hot encoding of the `sold_month` feature was skipped as it was already replaced by `sold_quarter`.

- **Data Type Conversion**: Features like `room_bed`, `sight`, and `quality`, which were originally represented as categorical variables but encoded as integers, were explicitly cast to integer data types. This ensured consistency in data representation and optimized performance during modeling.

4.4 Insights and Conclusion

The EDA process yielded several important insights about the dataset:

- **Outliers**: Outliers were detected and removed from features like `lot_measure15`, leading to a cleaner dataset. This helped to improve the quality of the data and made it more suitable for analysis and modeling.
- **Feature Transformation**: The transformation of features such as `basement`, `yr_renovated`, and `yr_built` into binary and age-related variables added meaningful context to the dataset. For instance, the new `is_basement` feature allows for a clearer interpretation of whether a property has a basement, while `age` provides more direct insights into the property's age and potential wear-and-tear.
- **Geographic Information**: The transformation of the `zipcode` feature into the `location_code` category simplified the analysis while retaining valuable regional information. This step significantly reduced the dimensionality of the data, making it more manageable for modeling.
- **Correlations**: The correlation analysis revealed strong relationships between various features, such as `living_measure` and `price`. Understanding these relationships is crucial for developing predictive models that rely on these features.
- **Feature Encoding**: One-hot encoding of categorical variables, including `sold_year`, `sold_quarter`, and `sold_weekday`, allowed for effective integration of categorical data into machine learning models.

In conclusion, the EDA process provided a deeper understanding of the dataset and laid the groundwork for the next steps in analysis and model development. The data was cleaned, transformed, and preprocessed to enhance its quality, making it ready for predictive modeling.

5. Feature Engineering Part 2: Post EDA

Following the extensive exploratory data analysis, a refined feature selection process was conducted to improve the quality of the dataset and ensure that only relevant features with strong predictive value were carried forward.

5.1 Feature Selection Criteria

Two major rules were defined to guide the filtering process:

- **Relevance to Target Variable**:
  - Features with a Pearson correlation coefficient > 0.05 or < -0.05 with respect to `price` were considered meaningful.
  - This helps exclude features that do not influence the house price in any direction.
- **Inter-feature Redundancy**:
  - When two features exhibited a high mutual correlation (> 0.85), one of them was discarded.
  - This reduces multicollinearity and prevents model overfitting due to duplicate information.

5.2 Continuous Feature Refinement

A correlation heatmap revealed key insights into the relationships among continuous variables and their predictive power:

- **Dropped Features**:
  - `long`: Had negligible correlation with the target variable.
  - `age`: Despite its intuitive appeal, it showed low correlation in this dataset.
  - `ceil_measure`: Highly correlated with `lot_measure`, making it redundant.

- **Retained Continuous Features**:

- `living_measure`: Captures the livable floor area and is positively correlated with price.
- `lot_measure`: Reflects land size and shows moderate correlation.
- `lat`: Geographic latitude which showed significant regional pricing patterns.
- `living_measure15` and `lot_measure15`: measurements at year 2015

- **Additional Notes**:
  - Emphasis was placed on preserving variables that balance both individual property details and neighborhood-level metrics.
  - The final set avoids duplication while maintaining representational diversity.

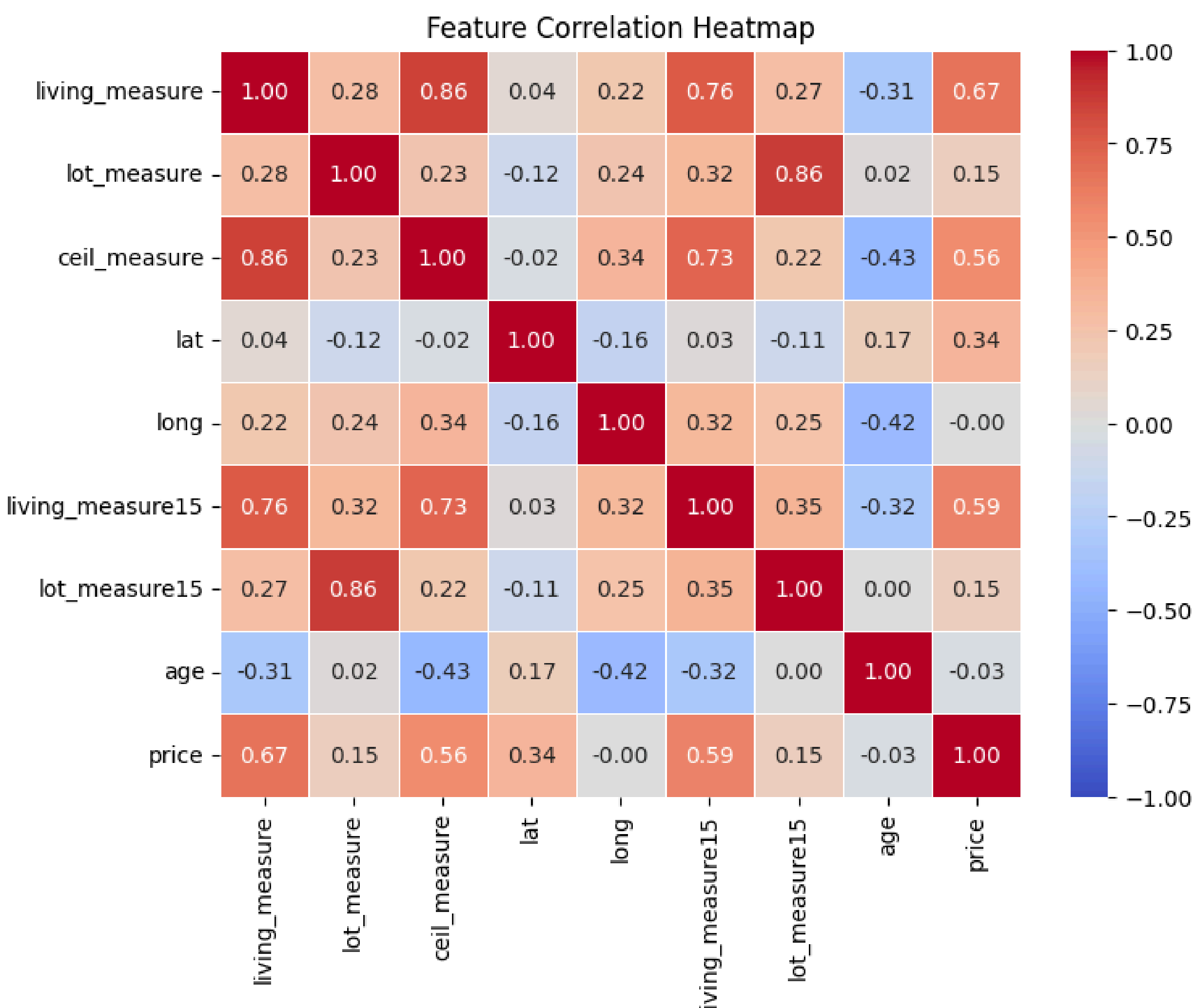

Figure 7: Correlation Heatmap for Continuous Features

5.3 Discrete Feature Refinement

A similar correlation-based selection was performed on discrete and categorical features:

- **Selection Logic**:
    - Only features with sufficient correlation strength to `price` were retained.
    - Binary and ordinal encodings from previous transformations were evaluated.

- **Considerations**:
    - Features like `coast`, `furnished`, `room_bed`, `room_bath`, `condition`, and `quality` were maintained due to their meaningful variance in pricing.
    - Engineered features such as `is_renovated`, `is_basement`, and temporal variables (e.g., `sold_quarter`, `sold_weekday`) added value to seasonal and structural analysis.
    - Encoded location identifiers (`location_code`) offered spatial granularity without the complexity of raw zipcodes.

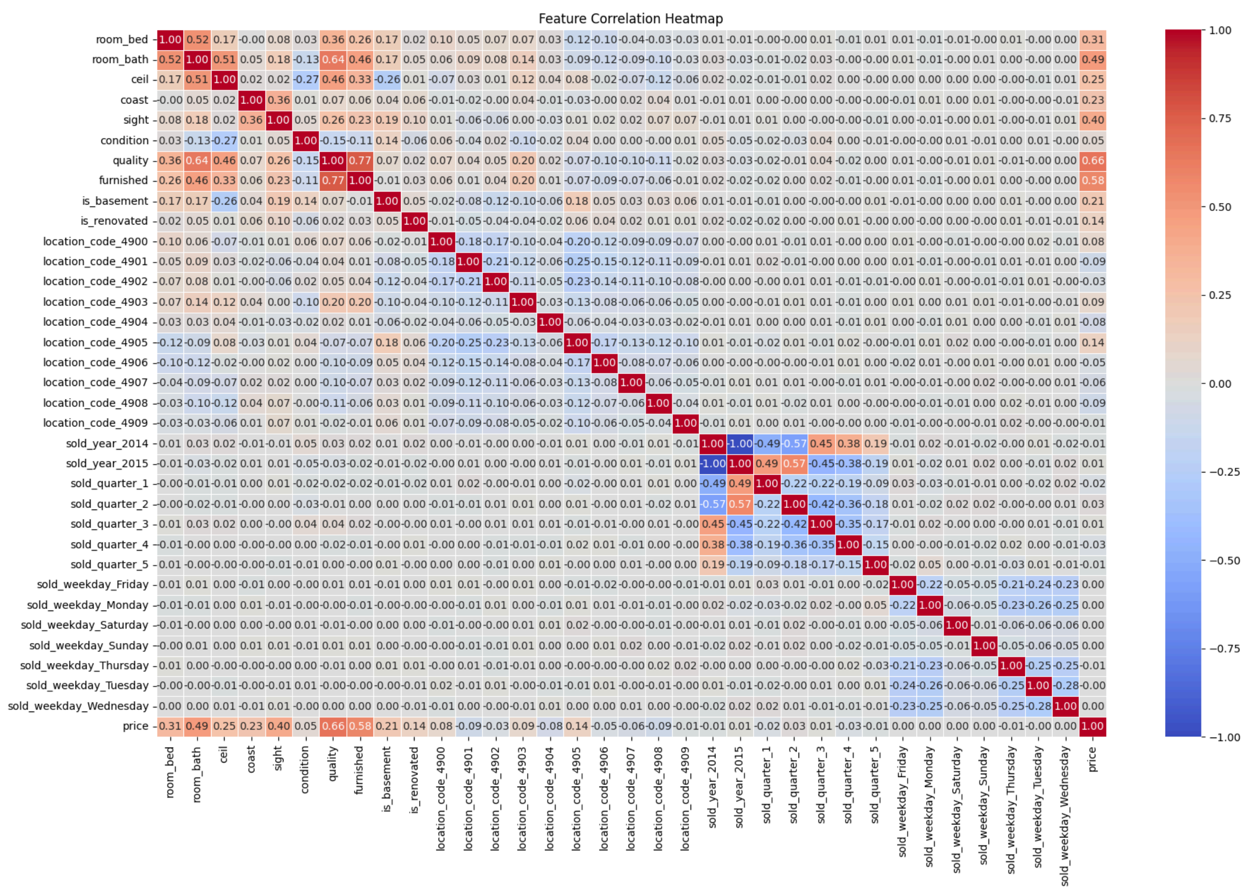

Figure 8: Correlation Heatmap for Discrete Features

5.4 Final Feature Set

After filtering, the selected features consisted of a robust blend of structural, geographic, and temporal variables:

- **Continuous Variables**:
  - `living_measure`, `lot_measure`, `lat`, `living_measure15`, `lot_measure15`
- **Key Discrete/Categorical Variables**:
  - `room_bed`, `room_bath`, `quality`, `condition`, `sight`, `coast`, `furnished`
  - `is_renovated`, `is_basement`, `location_code`, `sold_quarter`, `sold_weekday`, `sold_year`
- **Summary**:
  - The selected features maintained data integrity, reduced noise, and increased model interpretability.
  - This refined dataset is now optimized for efficient and effective model training in the subsequent phase.

6. Model Training

This section outlines the performance of various regression models applied to the dataset. The goal was to predict housing prices using a range of algorithms and evaluate them using consistent performance metrics.

6.1 Evaluation Metrics

The models were assessed based on the following criteria:

- **R^2 Score**: Represents the proportion of variance in the target variable that is predictable from the input features. Higher values indicate better performance.
- **Mean Absolute Percentage Error (MAPE)**: Represents the average absolute percentage difference between the actual and predicted values. Lower values indicate better performance.

6.2 Model Performance Overview

The following models were trained and evaluated:

Linear Regression

- R^2 Score: 0.7019
- MAPE: 23.77%

  A basic benchmark model. It exhibited moderate performance, but struggled to capture non-linear relationships in the data.

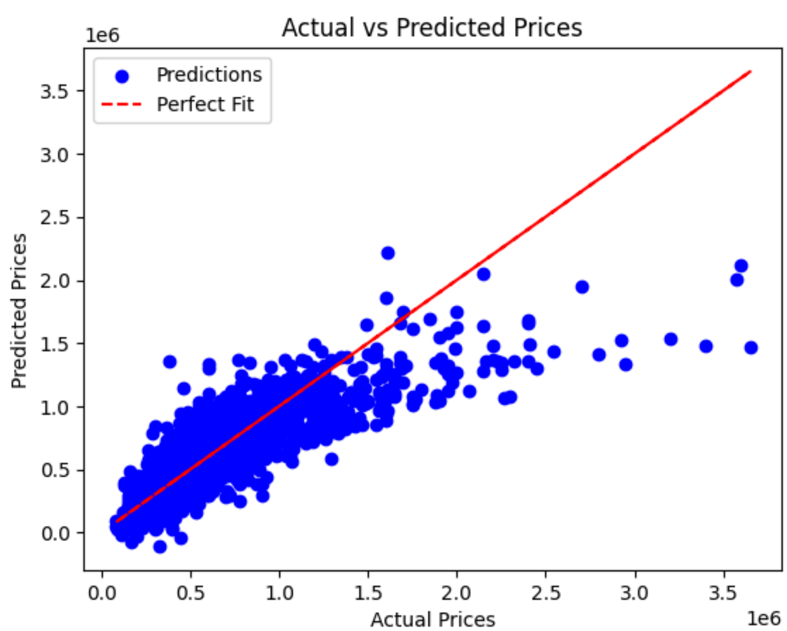

Figure 9: Base Model: Linear Regression results

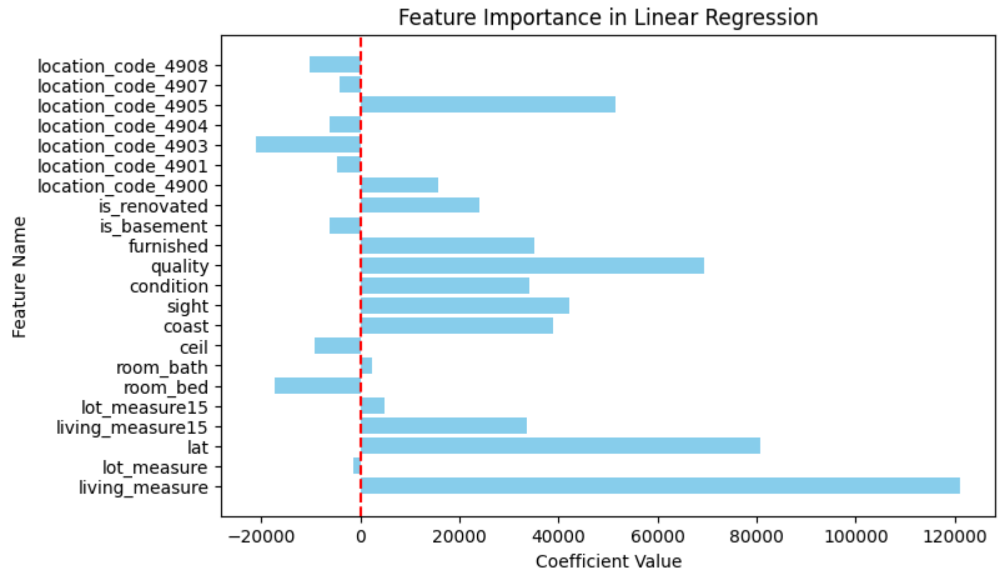

Figure 10: Feature Importance on basis of Linear Regression coefficients

Ridge Regression

- R^2 Score: 0.7019
- MAPE: 23.77%

Introduces L2 regularization to address potential multicollinearity. Performance remained identical to linear regression.

Lasso Regression

- R^2 Score: 0.7019
- MAPE: 23.77%

Utilizes L1 regularization which can reduce the number of features by shrinking some coefficients to zero. Performance was marginally lower than Ridge Regression.

Polynomial Regression

- R^2 Score: 0.8067
- MAPE: 18.82%

A non-linear extension of linear regression. Demonstrated a substantial improvement in model fit, capturing more complex patterns.

Support Vector Regressor (SVR)

- R^2 Score: -0.0700
- MAPE: 41.29%

Failed to generalize on the dataset. Poor performance likely due to high sensitivity to parameter tuning and scale in this context.

Decision Tree Regressor

- R^2 Score: 0.7312

- MAPE: 18.86%
  A non-parametric model that captured non-linear relationships, but prone to overfitting.

Random Forest Regressor

- R^2 Score: 0.8461
- MAPE: 15.10%
  An ensemble of decision trees, which significantly improved prediction accuracy compared to single tree models.

Random Forest with RandomizedSearchCV

- Best Parameters: {'n_estimators': 100, 'min_samples_split': 10, 'min_samples_leaf': 2, 'max_depth': 15}
- R^2 Score: 0.8592
- MAPE: 14.22%
  Hyperparameter tuning yielded further improvements in predictive performance.

XGBoost Regressor

- R^2 Score: 0.8671
- MAPE: 14.12%
  A gradient boosting framework optimized for speed and accuracy. Outperformed Random Forest in both metrics.

XGBoost with GridSearchCV

- Best Parameters: {'colsample_bytree': 0.7, 'learning_rate': 0.05, 'max_depth': 7, 'min_child_weight': 5, 'n_estimators': 500, 'subsample': 0.8}
- R^2 Score: 0.8829
- MAPE: 13.40%
  Provided the best overall performance after extensive hyperparameter tuning.

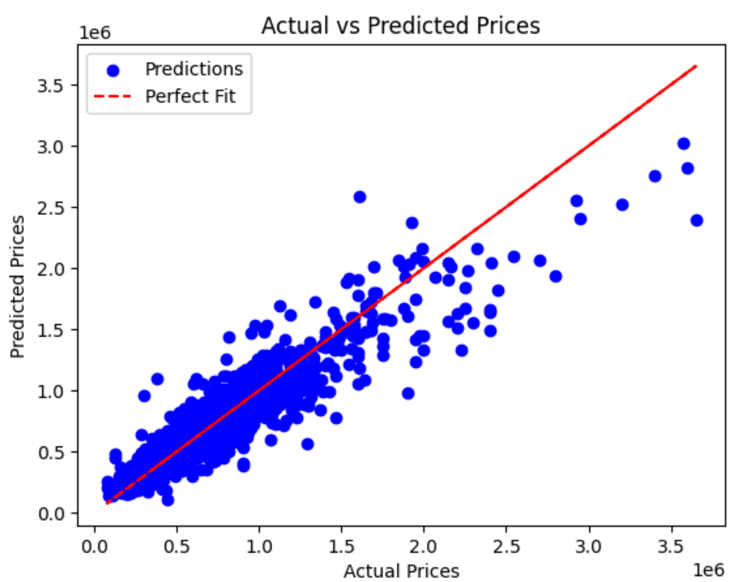

Figure 11: XGBoost model Best Results

LightGBM with RandomizedSearchCV

- Best Parameters: {'subsample': 0.8, 'num_leaves': 31, 'n_estimators': 300, 'max_depth': 7, 'learning_rate': 0.05, 'colsample_bytree': 0.7}
- R^2 Score: 0.8829
- MAPE: 13.40%
  Matched the performance of tuned XGBoost, with advantages in computational efficiency.

Deep Learning Model

- The model is a feedforward neural network for regression tasks. It consists of an input layer, two Dense layers with 128 units and ReLU activation, each followed by Dropout (0.2), a third Dense layer with 64 units, and a final output layer with 1 unit for continuous prediction. This structure helps capture complex patterns while reducing overfitting.
- R^2 Score: 0.7825
- MAPE: 18.28%
  A neural network-based regressor. While promising, it did not surpass tree-based ensemble methods in accuracy after certain sets of hyperparameter tuning, layer updates, adding dropouts, and optimizer changes

Figure 12: Deep Learning Model Results

6.3 Comparative Performance Summary

Model	R^2 Score	MAPE (%)	Remarks
Linear Regression	0.7019	23.77	Baseline model
Ridge Regression	0.7019	23.77	Regularized linear model
Lasso Regression	0.7019	23.77	L1 regularization, no significant gain
Polynomial Regression	0.8067	18.82	Captures non-linear relationships
Support Vector Regressor	-0.0700	41.29	Underperformed significantly
Decision Tree Regressor	0.7312	18.86	Simple, interpretable tree-based model
Random Forest	0.8461	15.10	Strong ensemble model
Random Forest (Tuned)	0.8592	14.22	Improved via hyperparameter tuning
XGBoost	0.8671	14.12	High-performing gradient boosting model
XGBoost (Tuned)	0.8829	13.40	Best performance after tuning
LightGBM (Tuned)	0.8829	13.40	Matched XGBoost; computationally efficient
Deep Learning	0.7825	18.28	Moderate performance

Table 5: Comparative Model Performance Summary

6.4 Summary

Among all tested models, XGBoost and LightGBM with hyperparameter tuning demonstrated the highest predictive performance with R^2 scores of 0.8829 and the lowest MAPE of 13.40%. Random Forest with tuning also performed well, although slightly behind XGBoost and LightGBM. Simpler models such as linear, ridge, and lasso regression exhibited notably lower performance, and SVR failed to generalize effectively.

7. Conclusion

This study presents a comprehensive approach to real estate price prediction through the application of modern machine learning techniques. By systematically addressing each stage of the data science pipeline—from data preprocessing and exploratory analysis to model development and evaluation—this work demonstrates the efficacy of ensemble learning algorithms in capturing the complex, non-linear relationships inherent in housing data.

Among the various models tested, XGBoost and LightGBM emerged as the most performant, with the best model achieving an R^2 score of 0.8829 and a mean absolute percentage error (MAPE) of 13.40%. These results underscore the capacity of gradient boosting methods to generalize well on real-world datasets, especially when appropriately tuned and validated using cross-validation techniques.

In addition to highlighting the predictive power of machine learning in the context of real estate, this research contributes to the broader academic discourse on data-driven valuation methodologies. It also opens avenues for future investigation, including the integration of geospatial analytics, time-series modeling, and socioeconomic indicators to further enhance model robustness and applicability.

Overall, this work provides a solid foundation for subsequent academic exploration in predictive modeling, urban economics, and decision support systems, and underscores the growing relevance of artificial intelligence in empirical research across interdisciplinary domains.

References

Wang, C., & Wu, H. (2018). *A new machine learning approach to house price estimation. BISKA Bilisim Technology, 168, 27649–27656.* *https://www.researchgate.net/publication/330001343_A_new_machine_learning_approach_to_house_price_estimation*

Maloku, F., Maloku, B., & Dinesh Kumar, A. (2024). *House Price Prediction Using Machine Learning and Artificial Intelligence. Journal of Artificial Intelligence & Cloud Computing, 3(4), 1–10.* *https://www.researchgate.net/publication/383057414_House_Price_Prediction_Using_Machine_Learning_and_Artificial_Intelligence*

Aniobi, D. E., Ochuba, C. O., & Nguideen, S. B. (2023). *House Price Prediction: Comparative Analysis of Regression-Based Machine Learning Algorithms. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 11(X).* *https://www.researchgate.net/publication/374977155_House_Price_Prediction_Comparative_Analysis_of_Regression-Based_Machine_Learning_Algorithms*

Maloku, F., Maloku, B., & Dinesh Kumar, A. (2024). *House Price Prediction Using Machine Learning and Artificial Intelligence. Journal of Artificial Intelligence & Cloud Computing, 3(4), 1–10.* *https://www.researchgate.net/publication/383057414_House_Price_Prediction_Using_Machine_Learning_and_Artificial_Intelligence*