AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING AND ROBOTICS WIMIR
MECHATRONIC ENGINEERING

PYTHON FOR MACHINE LEARNING AND DATA SCIENCE

Decoding Property DNA

Predicting Apartment Age From Market Features

Michał Domański, Anton-Matvii Kostiv, Ahmad Raoof

Kraków, November 2025

Table of Contents

1 Apartment Age Prediction as Asset Intelligence Tool

For urban energy modeling, urban planning, and climate resilience assessments, building age is essential because it indicates the construction materials, insulation standards, and mechanical systems likely present in a structure [1, 2]. Real estate professionals leverage building age for hedonic pricing models—statistical frameworks that estimate property values by analyzing how individual characteristics contribute to market prices.

1.1 The Data Gap Problem

Despite its critical importance, the construction year remains one of the most difficult and elusive property attributes to obtain at scale. This represents a significant market failure in real estate data infrastructure.

Research indicates that even in relatively well-developed property datasets, construction year data is frequently absent. Studies on geospatial databases found that less than 0.1% of buildings in some datasets contained information about the number of stories, and availability of construction year data was comparatively sparse. The Ordnance Survey's National Geographic Database explicitly acknowledges that building age is one of several attributes with significant data completeness challenges at the national scale [2].

1.2 Central Hypothesis

Based on the existing body of work—which predominantly focuses on price prediction—the scope of this research pivots to address the data scarcity issue identified in Section 1.1. We propose a paradigm shift from valuation (predicting price) to imputation (predicting missing attributes).

The hypothesis is posed as follows:
The age of an apartment in a Polish city can be predicted with statistically significant accuracy using a machine learning model trained on its relative market features.

Standard real estate modeling typically follows a hedonic pricing function defined as:

$$P = f(A, L, S) \tag{1}$$

where Price (P) is determined as a function of Age (A), Location (L), and Structural features (S). Our central hypothesis posits that this relationship is invertible. Assuming market efficiency, the observable characteristics of an apartment—including its price, location, and amenities—form a unique vector that points back to its era of construction. We formulate this inverse relationship as:

$$A = g(P, L, S) + \epsilon \tag{2}$$

In this formulation, the Age (A) is derived from the complex interactions between the property's valuation, spatial context, and physical attributes, where ϵ represents the residual error or unobserved variance.

1.3 Report Organization

This report is organized as follows:

- **Section 1** outlines the motivation for the project and the initial research hypothesis.
- **Section 2** provides details about the source, composition, and key statistics of dataset.
- **Section 3** covers Exploratory Data Analysis.
- **Section 4** describes the data preprocessing and machine learning models.

- **Section 5** summarizes the key conclusions based on results from Section 4.
- **Section 6** includes additional information related to project organization.

2 Dataset Description and Related Work

2.1 Dataset Overview and Provenance

The primary dataset used in this study is the “Apartment Prices in Poland” collection, curated by Krzysztof Jamroz and made available on Kaggle [3]. This dataset provides a comprehensive view of the Polish residential real estate market, specifically focusing on the 15 largest cities.

The data was acquired through a systematic web scraping methodology from local Polish real estate listing platforms. To enrich the property listings with valuable contextual information, each entry was augmented with geospatial data from OpenStreetMap (OSM) . This process involved calculating distances from each apartment to various Points of Interest (POIs), providing a richer representation of the neighborhood’s amenities.

The dataset is structured as a time series of monthly snapshots that cover the period from August 2023 to June 2024. It is divided into two distinct subsets:

- `apartments_pl_YYYY_MM.csv` contains listings for apartments for sale.
- `apartments_rent_pl_YYYY_MM.csv` contains listings for apartments for rent.

2.2 Data Fields and Statistical Information

The data set comprises a wide range of features for each property listing, which can be categorized as shown in Figure 1.

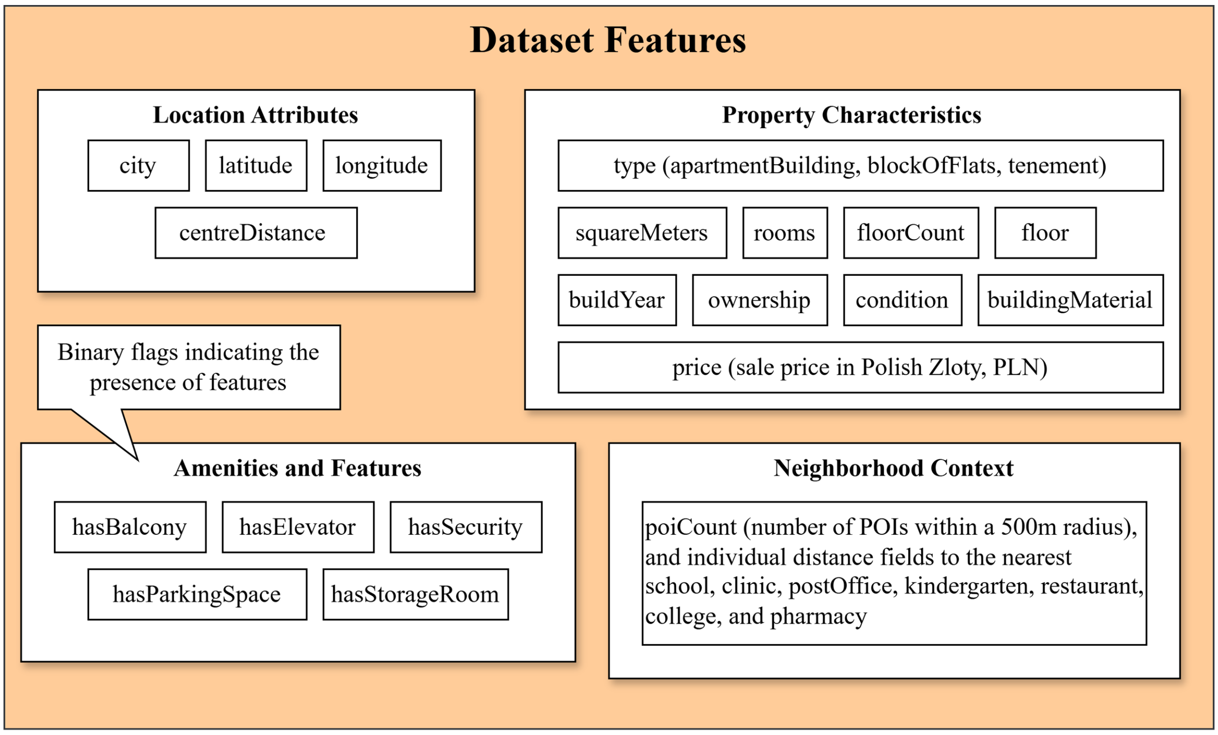

Dataset Features
Location Attributes
city
latitude
longitude
centreDistance
Property Characteristics
type (apartmentBuilding, blockOfFlats, tenement)
squareMeters
rooms
floorCount
floor
buildYear
ownership
condition
buildingMaterial
price (sale price in Polish Zloty, PLN)
Binary flags indicating the presence of features
Amenities and Features
hasBalcony
hasElevator
hasSecurity
hasParkingSpace
hasStorageRoom
Neighborhood Context
poiCount (number of POIs within a 500m radius), and individual distance fields to the nearest school, clinic, postOffice, kindergarten, restaurant, college, and pharmacy

Figure 1: Dataset Features Categorization

The initial, un-cleaned dataset for a single month (e.g., March 2024) contains approximately 16,000–17,000 samples with 28 features. As noted in the exploratory data analysis (EDA) by Jamroz [4], the data is not perfectly clean; several fields, such as `condition`, `buildingMaterial`, and `buildYear`, contain a significant number of missing values.

2.3 Related Work and Scope of Research

The dataset has already been the subject of several Kaggle analyzes.

Comprehensive Data Analysis
Jamroz [4] provides an extensive exploratory data analysis of the March 2024 snapshot. This work establishes key data patterns, such as the strong correlation between price and city location.

Deep Learning for Price Prediction
In a subsequent notebook, Jamroz [5] solves the problem of price prediction using a neural network. The work highlights the challenges of data quality, comparing a "cleanData" strategy with a "guessData" strategy . The "guessData" approach was found to be superior.

Rental Market Focus and Geospatial Analysis
Stencel [6] narrows the focus to the rental market in Warsaw, performing a dedicated EDA and creating interactive visualizations in Tableau.

Prediction of Apartment Prices in Major Polish Cities
Wiktoria Galarowicz [7] developed and compared Random Forest and XGBoost on the data. XGBoost was the superior model, achieving high accuracy with a Mean Absolute Percentage Error (MAPE) of 9% .

2.4 Scope of this Research

Based on this existing body of work, the scope of this research is to develop, validate and interpret a robust predictive model for the age of apartments. We will:

1. Systematically address the data quality issues identified in [4, 5].
2. Build upon Machine Learning modeling approaches.

3 Initial and Exploratory Data Analysis

Before making any hypotheses or drawing conclusions we must build understanding of the data, look for patterns and relations.

3.1 Correlations

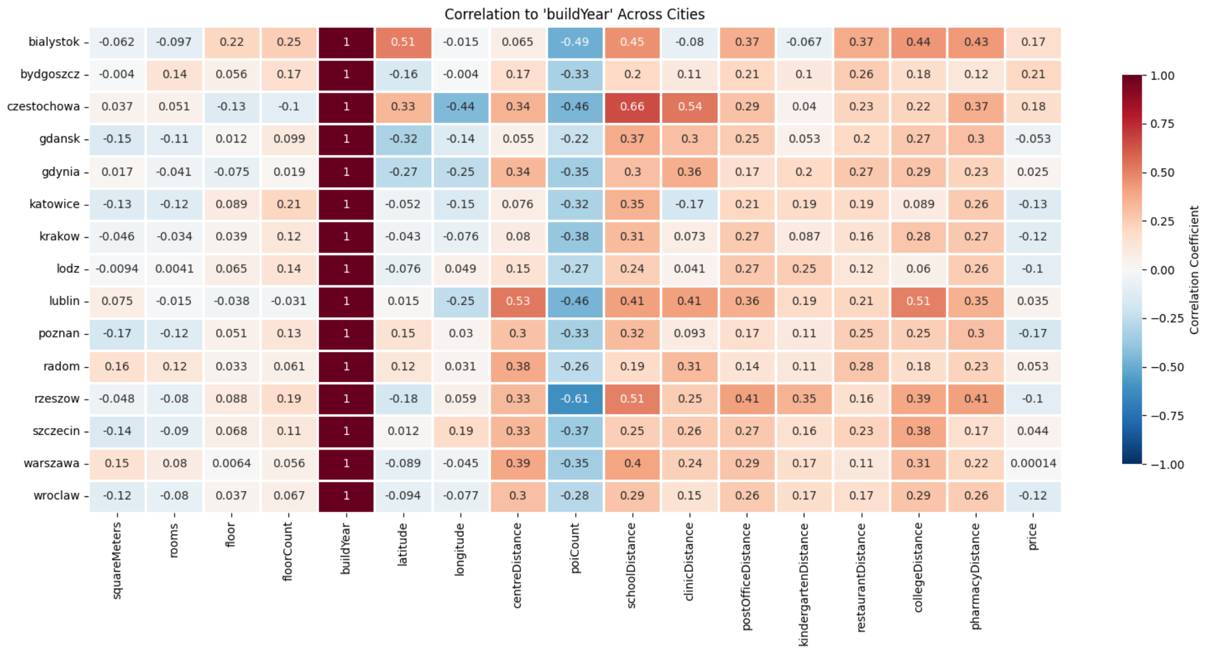
Correlation analysis plays a crucial role in understanding the relationships between variables and validating the assumptions that will drive further work. In simple terms, correlation testing helps identify whether and how strongly variables are related to each other, providing information on patterns or trends that could suggest causal relationships.

The correlation test was performed using the **Pearson correlation coefficient**, which is a widely used statistical method to measure the linear relationship between two continuous variables. The Pearson correlation coefficient (r) ranges from -1 to +1, where:

- +1 indicates a perfect positive linear relationship,
- -1 indicates a perfect negative linear relationship,
- 0 indicates that there is no linear relationship between the variables.

After initial analysis of the data, it was observed that correlation of price to buildyear was negligible for both renting and saling data. We will focus on other features and their relations which will provide more accurate results. This allows us to combine the data for further analysis and discard price feature. After combining all the data available the dataset size was consisting of 266415 unique entries with 27 features (excluding price).

The figure below shows the correlation of all features with build year in different cities. We can see that the problem is not so simple: depending on a city there are different relations in data and the problem is not simple enough to be generalized for all cities at once.



	squareMeters	rooms	floor	floorCount	buildYear	latitude	longitude	centreDistance	poiCount	schoolDistance	clinicDistance	postOfficeDistance	kindergartenDistance	restaurantDistance	collegeDistance	pharmacyDistance	price
bialystok	-0.062	-0.097	0.22	0.25	1	0.51	-0.015	0.065	-0.49	0.45	-0.08	0.37	-0.067	0.37	0.44	0.43	0.17
bydgoszcz	-0.004	0.14	0.056	0.17	1	-0.16	-0.004	0.17	-0.33	0.2	0.11	0.21	0.1	0.26	0.18	0.12	0.21
czestochowa	0.037	0.051	-0.13	-0.1	1	0.33	-0.44	0.34	-0.46	0.66	0.54	0.29	0.04	0.23	0.22	0.37	0.18
gdansk	-0.15	-0.11	0.012	0.099	1	-0.32	-0.14	0.055	-0.22	0.37	0.3	0.25	0.053	0.2	0.27	0.3	-0.053
gdynia	0.017	-0.041	-0.075	0.019	1	-0.27	-0.25	0.34	-0.35	0.3	0.36	0.17	0.2	0.27	0.29	0.23	0.025
katowice	-0.13	-0.12	0.089	0.21	1	-0.052	-0.15	0.076	-0.32	0.35	-0.17	0.21	0.19	0.19	0.089	0.26	-0.13
krakow	-0.046	-0.034	0.039	0.12	1	-0.043	-0.076	0.08	-0.38	0.31	0.073	0.27	0.087	0.16	0.28	0.27	-0.12
lodz	-0.0094	0.0041	0.065	0.14	1	-0.076	0.049	0.15	-0.27	0.24	0.041	0.27	0.25	0.12	0.06	0.26	-0.1
lublin	0.075	-0.015	-0.038	-0.031	1	0.015	-0.25	0.53	-0.46	0.41	0.41	0.36	0.19	0.21	0.51	0.35	0.035
poznan	-0.17	-0.12	0.051	0.13	1	0.15	0.03	0.3	-0.33	0.32	0.093	0.17	0.11	0.25	0.25	0.3	-0.17
radom	0.16	0.12	0.033	0.061	1	0.12	0.031	0.38	-0.26	0.19	0.31	0.14	0.11	0.28	0.18	0.23	0.053
rzeszow	-0.048	-0.08	0.088	0.19	1	-0.18	0.059	0.33	-0.61	0.51	0.25	0.41	0.35	0.16	0.39	0.41	-0.1
szczecin	-0.14	-0.09	0.068	0.11	1	0.012	0.19	0.33	-0.37	0.25	0.26	0.27	0.16	0.23	0.38	0.17	0.044
warszawa	0.15	0.08	0.0064	0.056	1	-0.089	-0.045	0.39	-0.35	0.4	0.24	0.29	0.17	0.11	0.31	0.22	0.00014
wroclaw	-0.12	-0.08	0.037	0.067	1	-0.094	-0.077	0.3	-0.28	0.29	0.15	0.26	0.17	0.17	0.29	0.26	-0.12

Figure 2: Corelation map - all features

3.2 Data Preparation and Cleaning

The raw dataset, comprising monthly snapshots of real estate listings, required significant processing to address missing values, outliers, and irrelevant features.

3.2.1 Scope Reduction and Filtering

To isolate specific market behaviors and eliminate inter-city variance during this phase of the experiment, the dataset was filtered to focus exclusively on Kraków, Warsaw, Wroclaw, Gdansk. While the original dataset covers 15 cities, reducing the scope allows for a more controlled analysis of local features, such as distance to the city center and specific Points of Interest (POIs). This reduced dataset size to 41687.

3.2.2 Handling Missing Values

As identified in the dataset description, the target variable `buildYear` suffered from incomplete data. Since this is the target variable for our unsupervised learning model, imputation was deemed risky; therefore, rows missing the construction year were removed. Dataset size shrinked almost twice containing 21165 examples.

After removing duplicates and rows with missing critical features, the dataset size was reduced to 12.969 samples.

3.2.3 Outlier Removal

To prevent extreme anomalies from skewing the model—such as typos in price, luxury properties inconsistent with the general market, or unrealistic distances—a percentile-based clipping method was applied. While the Interquartile Range (IQR) is a common standard, strict percentile capping offers more direct control over the specific proportion of data retained.

For every numerical feature in the dataset, lower and upper bounds were calculated at the 1st and 99th percentiles, respectively. Let $P_1(f)$ and $P_{99}(f)$ denote the percentiles for a feature f. A data point x was retained only if it satisfied the condition:

$$P_1(f) \leq x_f \leq P_{99}(f) \quad \forall f \in \text{Features} \tag{3}$$

The filtering process was applied sequentially across all numeric columns. This implies an intersection of validity: if a property listing was an outlier in *any* single dimension (e.g., normal price but extreme distance to the city center), it was removed entirely. This rigorous approach prioritizes data quality and density over quantity, ensuring the model learns from the core market distribution rather than edge cases.

3.2.4 Feature Encoding

Machine learning models require numerical input for calculation. Consequently, the categorical feature `city` was transformed using One-Hot Encoding. This process creates binary columns for each city (e.g., `city_Warsaw`, `city_Wroclaw`). To prevent perfect multicollinearity (known as the "dummy variable trap"), one category was dropped (`drop_first=True`), resulting in $N-1$ binary features representing the locations.

3.3 Final Feature Selection and Splitting

Based on the correlation analysis and data availability, the final feature set selected for training included following 14 features:

- buildYear — Year in which the building was constructed,
- collegeDistance — Distance from the property to the nearest college or university,
- squareMeters — Total floor area of the property in square meters,
- restaurantDistance — Distance from the property to the nearest restaurant,
- centreDistance — Distance from the property to the city center,
- kindergartenDistance — Distance from the property to the nearest kindergarten,
- schoolDistance — Distance from the property to the nearest school,
- clinicDistance — Distance from the property to the nearest medical clinic,
- pharmacyDistance — Distance from the property to the nearest pharmacy,
- postOfficeDistance — Distance from the property to the nearest post office,
- *Encoded City Indicators* (derived from `city`)
- poiCount — Number of nearby points of interest within a defined radius,
- geocluster — Geographic cluster label assigned to the property location.

Interestingly, the price feature had very little Distance from the property to the nearest medical clinic tle correlation to the age of buildings.

The data was normalized using MinMaxScaler to scale all numerical features to a bounded range, specifically between 0 and 1. This preprocessing step is vital for the subsequent unsupervised learning phase, as algorithms relying on distance calculations (such as clustering) are sensitive to the magnitude of variables. Without normalization, features with large values (e.g., `centreDistance` in meters) would disproportionately influence the model compared to features with smaller ranges (e.g., `poiCount`). The transformation for a feature x is calculated as:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

where x_{min} and x_{max} represent the minimum and maximum values of the feature, respectively.

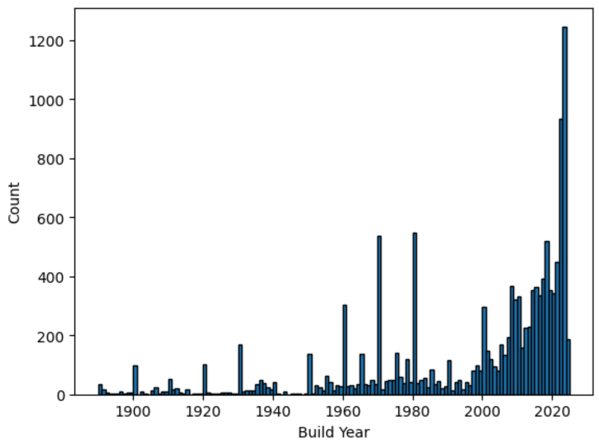


Figure 3: Histogram of data in final dataset over years

The dataset was split into three subsets to ensure robust validation:

1. **Training Set (60%):** Used to learn the model parameters.
2. **Validation Set (20%):** Used for hyperparameter tuning and interim evaluation.
3. **Test Set (20%):** Held out for the final performance assessment.

For evaluation of baseline models described in the next section tha data was split into two subsets:

1. **Training Set (80%):** Used to learn the model parameters.
2. **Test Set (20%):** Held out for the final performance assessment.

4 Implementation and Configuration of Models

4.1 Evaluation Metrics

To evaluate the model's performance comprehensively, we utilized different metrics of prediction accuracy: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), the Coefficient of Determination (R^2), and the Standard Deviation of residuals (σ).

MAE provides the most intuitive measure for this dataset, representing the average absolute difference between the predicted and actual construction years. MAPE expresses this error relative to the magnitude of the target variable. They are calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{true} - y_{pred}| \tag{5}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_{true} - y_{pred}}{y_{true}}\right| \tag{6}$$

Additionally, R^2 (R-squared) was used to measure the proportion of variance in the target variable explained by the model, while the standard deviation of the residuals highlights the consistency of the model's errors.

4.2 Model Selection Strategy

The implementation phase began with a comparative analysis of four distinct regression algorithms to establish a performance baseline. The objective was to identify which class of models best captured the non-linear relationships between geospatial features and the apartment construction year.

4.2.1 Linear Regression

Linear Regression served as the baseline model to establish a minimum performance threshold. It assumes that the construction year (y) shares a linear relationship with the input features (X). The model aims to minimize the Residual Sum of Squares (RSS) between the observed and predicted values:

$$\min_{w} ||Xw - y||_2^2 \tag{7}$$

While computationally efficient, this model cannot capture the complex, concentric, and often irregular growth patterns of Polish cities.

4.2.2 Polynomial Regression

To address the limitations of the linear baseline, Polynomial Regression (Degree 3) was implemented. This method expands the feature space by creating non-linear interactions between variables (e.g., x^2, x^3, and $x_i \cdot x_j$). This allows the model to fit curved decision surfaces, theoretically better suited for modeling distances from a city center.

4.2.3 Random Forest Regressor

Random Forest is an ensemble learning method that employs *bagging* (bootstrap aggregating). It constructs a multitude of decision trees at training time and outputs the average prediction of the individual trees. By averaging noisy but unbiased models (deep decision trees), it reduces variance and resists overfitting. It naturally handles non-linear data and interactions between features without requiring explicit feature engineering (like polynomial expansion).

4.2.4 XGBoost

Extreme Gradient Boosting (XGBoost) represents the state-of-the-art in gradient boosting algorithms. Unlike Random Forest, which builds trees independently, XGBoost builds trees *sequentially*. Each new tree attempts to correct the errors (residuals) made by the previous trees. It optimizes a regularized objective function that balances predictive accuracy with model complexity:

$$\mathcal{O}\lfloor|(\theta) = \mathcal{L}(\theta) + \Omega(\theta) \tag{8}$$

XGBoost was selected for its execution speed, ability to handle sparse data (generated by One-Hot Encoding), and superior performance in tabular data competitions.

4.2.5 Baseline Performance Evaluation

The models were evaluated on a held-out test set (20% of data). As shown in Table 1, the tree-based ensemble methods significantly outperformed the linear approaches.

Baseline Model Performance (Test Set)

Model	**MAE (Years)**	**RMSE**	**R^2 Score**
Linear Regression	20.61	26.88	0.204
Polynomial Regression (Deg 3)	16.63	23.39	0.397
Random Forest (Default)	**9.81**	**16.85**	**0.687**
XGBoost (Default)	10.99	17.36	0.668

Initial results indicated that the **Random Forest** model achieved the highest accuracy with an R^2 of 0.687, followed closely by XGBoost. The linear models proved insufficient for this task, likely due to the complex, non-linear nature of urban development patterns. Consequently, further optimization focused exclusively on the two tree-based ensembles.

4.3 Hyperparameter Optimization

To improve predictive performance, we employed `RandomizedSearchCV` (5-fold cross-validation) to explore the hyperparameter space. This method was chosen over Grid Search to efficiently sample a wide range of configurations given the computational cost of ensemble models.

4.3.1 Random Forest Tuning

The Random Forest tuning focused on n_estimators (100–1000), max_depth, and split criteria. However, experiments revealed that the Random Forest model was difficult to improve beyond its baseline. The best tuned configuration achieved an R^2 of approximately 0.54 and an MAE of 9.81 years, which was surprisingly lower than the default configuration. This suggested that the default parameters were already near-optimal for this specific feature set, or that the model was prone to overfitting when depth constraints were relaxed.

Best RF parameters:

- **max_depth:** 16
- **min_samples_leaf:** 4
- **n_estimators:** 642

4.3.2 XGBoost Tuning and Final Model Selection

Conversely, the XGBoost model demonstrated significant responsiveness to hyperparameter tuning. Several experiments were conducted, iteratively refining the search space for `learning_rate`, `max_depth`, and regularization terms (`reg_alpha`, `reg_lambda`).

The optimization process yielded a superior configuration that overtook the initial Random Forest baseline. The best performing model utilized the `lossguide` growth policy with a deeper tree structure.

Thus, the final model selected was the XGBoost Regressor. The optimized hyperparameters that achieved the best balance of bias and variance (Test $R^2 \approx 0.71$, MAE ≈ 7.8 years) are detailed below:

- **n_estimators:** 1,769 (High number of boosting rounds)
- **max_depth:** 14 (allowing capture of complex interaction effects)
- **learning_rate:** ≈ 0.01 (Slow learning to prevent overfitting)
- **colsample_bytree:** ≈ 0.77
- **Regularization:** `gamma` ≈ 3.6, `reg_alpha` ≈ 0.73

Results obtained on test set from tuned model:

- **MAE:** 7.8 years
- **RMSE:** 13.7
- **STD of prediction errors:** 13.7

5 Summary and Conclusions

5.1 Verification of Research Hypothesis

Based on the experimental results, this hypothesis is **confirmed**. The final XGBoost model achieved a Coefficient of Determination (R^2) of approximately **0.71** on the test set. In the context of real estate valuation—where data is often noisy, subjective, or incomplete—explaining over 70% of the variance in construction year through indirect features (location, price, amenities) represents a statistically significant result.

5.2 Interpretation of Results in Historical Context

The performance of the model must be interpreted against the backdrop of the dataset's temporal scope. The data comprises buildings constructed between **1875 and 2024**, a range of nearly 150 years that covers vastly different architectural eras:

- 19th-century tenement houses (*Kamienice*),
- Inter-war modernist architecture,
- Post-war socialist prefabricated blocks (*Wielka Płyta*),
- Modern developer projects (*Deweloperka*).

The achieved Mean Absolute Error (MAE) of **7.8 years** is highly promising. Given the distinct technological and material standards associated with these eras, an error margin of less than a decade allows the model to reliably categorize a building into its correct historical and technological period. For example, the model can effectively distinguish between a building from the 1970s (likely uninsulated concrete slab) and one from the 2000s (modern masonry and insulation), providing critical inputs for energy modeling and urban planning.

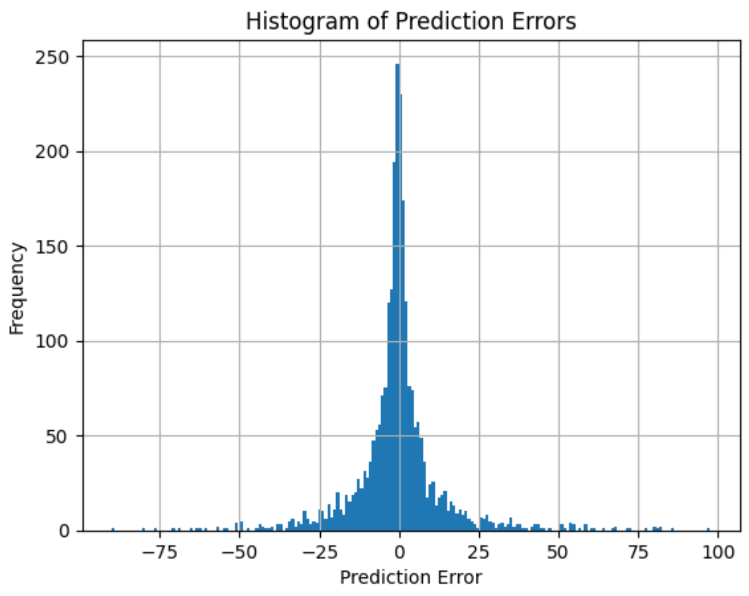


Figure 4: Error distribution

5.3 Closing Remarks

This research highlights the viability of using machine learning as an "Asset Intelligence" tool to bridge the data gap in real estate infrastructure. By leveraging publicly available listing data and open-source geospatial metrics (OpenStreetMap), we successfully reconstructed missing property attributes without costly physical inspections.

While the Random Forest model provided a strong baseline, the XGBoost approach proved superior in capturing the complex, non-linear interactions between a property's location and its age. The low MAE suggest that this methodology is robust enough for deployment in urban analysis platforms.

6 Additional Information

6.1 List of Abbreviations

- **EDA**: Exploratory Data Analysis
- **IQR**: Interquartile Range
- **MAE**: Mean Absolute Error
- **MAPE**: Mean Absolute Percentage Error
- **OSM**: OpenStreetMap
- **POI**: Point of Interest
- **RMSE**: Root Mean Squared Error
- **XGB**: Extreme Gradient Boosting

6.2 Documentation and Visualization Tools

- **Overleaf**: Report document creation.
- **Gemini**: Grammar review.
- **Perplexity AI**: Research.
- **draw.io**: Diagram creation.

6.3 Contributions of Team Members

- **Hypothesis formulation**: A-M. Kostiv, Michał Domański, Ahmad Raoof
- **Relevant work research**: A-M. Kostiv
- **Exploratory Data analysis**: Michał Domański, Ahmad Raoof
- **Data preparation**: Ahmad Raoof, Michał Domański
- **Model implementation and configuration**: Michał Domański
- **Model testing**: Michał Domański
- **Report writing**: A-M. Kostiv
- **Report review**: Michał Domański, Ahmad Raoof
- **Project organization**: Michał Domański, A-M. Kostiv, Ahmad Raoof
- **Presentation of project results**: A-M. Kostiv, Michał Domański, Ahmad Raoof

6.4 Source Code

The source code and all data used in this project is available at:
`https://github.com/firiusz123/PropertyDNA`
`https://colab.research.google.com/drive/1XxvgsGOn1FJOm5BkB1juezjQNidjXSez?usp=sharing`

References

[1] J. Rosser *et al.*, "Predicting residential building age from map data," *Computers, Environment and Urban Systems*, 2019.

[2] M. Biljecki *et al.*, "Estimating building age with 3d gis," in *12th 3D Geoinfo Conference*, (Melbourne, Australia), 2017.

[3] K. Jamroz, "Apartment prices in poland." Kaggle Dataset, 2024. Available: `https://www.kaggle.com/datasets/krzysztofjamroz/apartment-prices-in-poland`.

[4] K. Jamroz, "Apartment for sale in poland - data analysis." Kaggle Notebook, 2024. Available: `https://www.kaggle.com/code/krzysztofjamroz/apartment-for-sale-in-poland-data-analysis`.

[5] K. Jamroz, "Apartment for sale in poland - deep learning model." Kaggle Notebook, 2024. Available: `https://www.kaggle.com/code/krzysztofjamroz/apartment-for-sale-in-poland-deep-learning-model`.

[6] J. Stencel, "Apartments for rent in warsaw - eda." Kaggle Notebook, 2024. Available: `https://www.kaggle.com/code/juliastencel/apartments-for-rent-in-warsaw-eda`.

[7] W. Galarowicz, "Prediction of apartment prices in major polish cities using machine learning," 2024.