## 1. Abstract

Accurate housing price prediction is a critical tool for real estate investors, urban planners, policy makers, and financial institutions (Chen et al., 2017; Glaeser and Nathanson, 2017). This project explores the use of machine learning models to forecast Q3 2024 housing prices across European Union (EU) regions, using a structured dataset comprising quarterly housing indices, macroeconomic indicators, and demographic attributes. The aim is to evaluate and compare various regression models to identify the most accurate and generalizable approach for predicting property values.

The dataset includes housing price indices from Q1 2020 to Q3 2024, along with relevant features such as GDP growth rate, average income, unemployment rate, population growth, and climate zone (Eurostat, 2023). After extensive exploratory data analysis (EDA), several engineered features were introduced to enhance predictive power, including interaction terms and ratio-based indicators. Six machine learning models were implemented: Linear Regression, Random Forest, Gradient Boosting, XGBoost, Support Vector Regression (SVR), and K-Nearest Neighbors (KNN) (Pedregosa et al., 2011; Chen and Guestrin, 2016).

Hyperparameter tuning and cross-validation were applied to optimize model performance, with metrics such as R², Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) used for evaluation. Results show that KNN and SVR outperformed traditional models, achieving R² scores of 0.9970 and 0.9932 respectively, with minimal prediction error. Feature importance analysis highlighted the impact of income, urbanization, and past housing trends on pricing (Lundberg and Lee, 2017).

The findings demonstrate that incorporating both economic indicators and historical price trends significantly improves model accuracy, and that non-linear models offer superior performance in this predictive context.

## 2. Introduction

The real estate sector plays a vital role in economic development, financial markets, and social well-being. Understanding and forecasting housing prices has become increasingly important in the context of market volatility, urban expansion, and post-pandemic economic shifts across the European Union (Gyourko et al., 2021; European Commission, 2022). Accurate prediction of housing prices supports informed decision-making for investors, developers, policy makers, and homebuyers alike. With the growing availability of structured data and the evolution of predictive modeling techniques, machine learning (ML) offers a powerful solution for addressing this complex challenge (Kaufmann and Steinmetz, 2020).

Traditional statistical models, such as linear regression, have been widely used in housing price prediction but often fall short in capturing non-linear relationships and interactions among features (Selim, 2009). Housing prices are influenced by a broad set of factors, including economic indicators (GDP growth, unemployment), demographic trends (urbanization, population growth), and regional attributes such as climate zone (Eurostat, 2023). Moreover, temporal trends embedded in historical housing prices provide additional predictive value, which can be overlooked by simplistic models.

This project leverages machine learning to forecast housing prices for Q3 2024 across EU cities and regions. The goal is not only to build predictive models but to compare a range of algorithms, assess their performance, and understand which features contribute most to price variability. By combining feature engineering, model tuning, and visual diagnostics, the project aims to identify a robust, generalizable solution to housing price forecasting in a dynamic, multi-dimensional environment.

The introduction of historical quarterly housing data as predictive features further enriches the model input space, offering a unique opportunity to capture temporal patterns. The ultimate objective is to deliver a practical, data-driven approach to housing price prediction that is both interpretable and accurate.

## 3. Business Understanding

### 3.1 Objectives

The primary objective of this project is to forecast housing prices across European Union (EU) regions for Q3 2024 using machine learning techniques applied to structured economic, demographic, and historical housing data. The project aims to:

* Evaluate and compare multiple regression models on predictive accuracy
* Identify the most suitable model for generalization and interpretability
* Analyze key factors influencing housing prices across different EU regions
* Use model explainability techniques to support data-driven insights

Success is defined as achieving a high predictive accuracy — ideally **R² ≥ 0.95** — alongside low MAE and RMSE scores, and generating interpretable outputs to inform both academic and applied audiences (Pedregosa et al., 2011; Lundberg and Lee, 2017).

### 3.2 Problem Definition

Housing affordability and regional price volatility have become critical issues in Europe, influenced by macroeconomic forces, urbanization, and supply-demand dynamics (European Commission, 2022). Traditional models, such as linear regression, lack the capacity to account for complex, non-linear patterns and temporal dependencies that are increasingly relevant in real estate forecasting (Selim, 2009).

The core problem addressed by this project is how to build accurate and interpretable predictive models for EU housing prices — accounting not only for cross-sectional features like income and GDP but also for **historical quarterly trends** in housing prices from **Q1 2020 to Q2 2024**.

### 3.3 Scope

This project focuses exclusively on structured tabular data covering EU cities and regions. The dataset includes economic indicators such as GDP growth, unemployment rate, average income, population growth, and urbanization, along with housing price indices for each quarter from 2020 to 2024 (Eurostat, 2023). Climate zones and region classifications are included as categorical variables.

The predictive target is **Q3 2024 housing prices**. The scope excludes external unstructured data sources (e.g., satellite imagery, sentiment analysis, or web scraping). Models are limited to supervised learning techniques using Python-based tools.

### 3.4 Project Plan

The project follows a structured machine learning workflow inspired by CRISP-DM:

1. **Data Collection & Understanding**: Load and explore the EU housing dataset, perform statistical summaries, and visualize trends.
2. **Data Preparation**: Clean data, encode categorical features, scale numeric values, and engineer new features (e.g., ratios and interactions).
3. **Modeling**: Train and tune six models — Linear Regression, Random Forest, Gradient Boosting, XGBoost, SVR, and KNN — using GridSearchCV for optimization.
4. **Evaluation**: Use R², MAE, and RMSE to assess performance. Visualize predictions and residuals.
5. **Comparison & Insights**: Determine the best-performing model and identify key predictive features driving housing prices.
6. **Conclusion**: Summarize findings and propose future work, including time-series modeling or integration of external data sources.

By combining time-aware feature engineering with non-linear models and explainable AI techniques, this project aims to deliver a robust, interpretable solution for housing price forecasting in the EU (Gyourko et al., 2021; Kaufmann and Steinmetz, 2020).

## 4. Technologies & Tools Used

### 4.1 Models and Machine Learning Algorithms

This project applied a diverse set of supervised regression algorithms to predict housing prices for Q3 2024. The models were selected to balance performance, interpretability, and algorithmic variety, ensuring that linear, non-linear, and instance-based approaches were fairly represented.

* **Linear Regression** was used as a baseline model due to its simplicity and interpretability, but its performance was limited by its linear assumptions (Selim, 2009).
* **Random Forest Regressor** and **Gradient Boosting Regressor** were employed as ensemble tree-based models capable of capturing non-linear relationships and feature interactions without requiring feature scaling. These models are known for their robustness and strong performance on structured datasets (Pedregosa et al., 2011).
* **XGBoost Regressor** (Extreme Gradient Boosting) was included for its efficiency and scalability in handling complex feature interactions and sparse data (Chen and Guestrin, 2016).
* **Support Vector Regression (SVR)** was used as a kernel-based model with the radial basis function (RBF) kernel to capture high-dimensional patterns in the data (Kaufmann and Steinmetz, 2020).
* **K-Nearest Neighbors (KNN) Regressor** served as a non-parametric baseline model based on similarity between observations in the feature space.

Each model was trained and evaluated using a consistent workflow, including train/test splitting, hyperparameter tuning via GridSearchCV, and performance assessment using R², MAE, and RMSE metrics.

### 4.2 Libraries and Tools

The implementation was carried out in Python using a variety of open-source libraries commonly used in data science and machine learning:

* **Data manipulation and preprocessing**:
  + pandas for structured data handling
  + numpy for numerical operations
  + sklearn.preprocessing for scaling and encoding features
* **Machine learning modeling**:
  + scikit-learn for regression algorithms, pipeline workflows, cross-validation, and performance metrics (Pedregosa et al., 2011)
  + xgboost for the XGBoost regressor (Chen and Guestrin, 2016)
* **Hyperparameter tuning and validation**:
  + GridSearchCV from sklearn.model\_selection to perform exhaustive grid search over model parameters
* **Visualization and interpretability**:
  + matplotlib and seaborn for visual analytics such as correlation matrices, boxplots, and residual plots
* **Development environment**:
  + All analysis and modeling were conducted in **Jupyter Notebook**, which provided an interactive and modular development environment ideal for data experimentation, visualization, and reproducible workflows.

This combination of models and tools enabled a robust and flexible approach to data exploration, model development, and result interpretation.

## 5. Data Understanding & Preparation

### 5.1 Dataset Overview

The dataset used in this project was obtained from publicly available **Eurostat** sources and structured into a tabular format representing regional housing markets across the European Union (Eurostat, 2023). It includes **310 records** (observations) across **29 features**, covering the period from **Q1 2020 to Q3 2024**. The dataset contains quarterly housing price indices for each city or region, along with associated economic and demographic features.

The key variable to be predicted is the **Q3 2024 housing price index**, treated as a continuous numerical target for regression modeling. Input features include:

* **Macroeconomic indicators**: GDP growth rate, unemployment rate, average income, interest rate
* **Demographic attributes**: population growth rate, urbanization rate
* **Geographic/categorical fields**: country, city, region, climate zone
* **Time-series fields**: quarterly housing prices from Q1 2020 to Q2 2024

The structured format of the dataset enabled feature-level manipulation, model-ready preprocessing, and integration of domain-driven engineered features.

### 5.2 Initial Exploration and Descriptive Statistics

Initial data inspection revealed that the dataset was **clean, complete, and well-structured**, with no missing values in the primary features. Descriptive statistics were generated using pandas, summarizing central tendencies and ranges for numerical attributes such as GDP growth, unemployment rate, and average income.

Visual inspection using seaborn pairplots and histograms indicated that some features exhibited skewness or outliers, particularly in the GDP growth and unemployment rate variables. Boxplots confirmed regional differences in housing prices, especially when grouped by climate zone or region classification.

A **correlation matrix** was generated to identify relationships between input features and the target variable. Predictors such as average income, unemployment rate, and certain historical quarters (e.g., Q2\_2024) showed strong correlation with Q3\_2024 prices.

### 5.3 Feature Engineering

To improve predictive power and capture interaction effects, two derived features were engineered:

* **GDP\_Urban\_Interaction**: a multiplicative interaction between GDP growth rate and urbanization rate, intended to capture the compounded effect of economic vitality in urban centers
* **Income\_Unemployment\_Ratio**: average income divided by (unemployment rate + 1), designed to reflect real purchasing power and economic accessibility

Additionally, **categorical variables** such as Region and Climate Zone were **one-hot encoded** to convert them into machine-readable numerical formats. This ensured compatibility with linear and distance-based models, which require all features to be numerical (Pedregosa et al., 2011).

The final feature set included both original and engineered variables, creating a rich, multidimensional dataset for model training.

### 5.4 Data Cleansing and Formatting

Data cleansing involved minimal intervention due to the pre-cleaned nature of the dataset. The following actions were taken:

* Dropped non-predictive identifiers (Country, City)
* Converted all categorical variables into dummy variables using one-hot encoding
* Standardized the dataset using consistent data types (float64 for continuous variables)

All numeric columns were checked for outliers and unusual distributions. Some minor skewness remained in features like GDP growth, but no transformation was applied as tree-based and kernel models are generally robust to non-normality (Kaufmann and Steinmetz, 2020).

### 5.5 Data Preparation for Modeling

To prepare for modeling, the dataset was split into input features (**X**) and the target variable (**y = Q3\_2024**) using scikit-learn. A traditional **80/20 train-test split** was applied, ensuring that performance metrics reflect out-of-sample generalization.

Crucially, all **historical quarterly housing prices (Q1\_2020 to Q2\_2024)** were preserved as features. This choice enabled models to learn from temporal patterns, improving accuracy compared to previous versions that excluded these columns.

Feature scaling was applied using **StandardScaler** from sklearn.preprocessing, but only to the numeric columns where distance-based algorithms like **KNN** and **SVR** would be sensitive to feature magnitude. Tree-based models were trained on unscaled data.

This careful data preparation ensured compatibility across all models and preserved the temporal signal needed for effective forecasting.

### 5.6 Visualizations and Insights

A range of visualizations were used to guide understanding and interpretation:

* **Correlation Heatmap**: Identified strong relationships between past housing quarters and Q3 2024 pricing
* **Boxplots**: Showed variation in housing prices across climate zones and regions
* **Pairplots**: Visualized relationships between macroeconomic predictors (income, GDP, unemployment)
* **Residual Plots and Actual vs Predicted**: Evaluated model fit across different algorithms
* **Feature Importance (Tree Models)**: Visualized the most impactful predictors for Random Forest, Gradient Boosting, and XGBoost

These visual tools enhanced both model development and result interpretation, supporting deeper insight into the complex dynamics of EU housing prices.

## 6. Modeling & Evaluation

### 6.1 Modeling Strategy

The predictive goal of this project is to estimate Q3 2024 housing prices across European Union regions using supervised machine learning regression techniques. Regression was selected over classification because the target variable, housing price index, is continuous rather than categorical. The modeling strategy focused on building, tuning, and comparing a variety of regression algorithms to evaluate their ability to capture complex relationships within the data.

A multi-model approach was adopted to assess the performance of diverse algorithm types — including linear, tree-based, kernel-based, and instance-based models. All models were trained using the same feature set and evaluated on a consistent 80/20 train-test split to ensure fairness in comparison. The target variable was the housing price index for Q3 2024, while the input features included both macroeconomic indicators and historical housing price data from Q1 2020 to Q2 2024.

Each model was evaluated using R², Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) metrics. These metrics were chosen for their ability to quantify model accuracy, average prediction error, and sensitivity to outliers, respectively (Pedregosa et al., 2011).

### 6.2 Implemented Machine Learning Models

Six different regression models were implemented and evaluated:

* **Linear Regression**: Used as a baseline model due to its simplicity and interpretability. It assumes linear relationships between features and the target, which may limit its accuracy when non-linear patterns are present (Selim, 2009).
* **Random Forest Regressor**: An ensemble of decision trees that improves accuracy through bootstrapped aggregation (bagging). It handles feature interactions and non-linearity well and is resistant to overfitting.
* **Gradient Boosting Regressor**: A boosting technique that sequentially trains weak learners, minimizing errors at each step. It generally provides strong predictive performance, though it can be slower to train.
* **XGBoost Regressor**: An optimized version of gradient boosting known for its speed, regularization techniques, and scalability (Chen and Guestrin, 2016).
* **Support Vector Regression (SVR)**: A kernel-based model that maps input data into higher-dimensional space using an RBF kernel. It is effective in handling non-linear patterns, though it can be sensitive to scaling and hyperparameters (Kaufmann and Steinmetz, 2020).
* **K-Nearest Neighbors (KNN) Regressor**: A non-parametric model that predicts based on the average of the nearest neighbors in the feature space. It is intuitive and often effective when the data is well-distributed.

Each model brought different strengths to the task, enabling a rich comparison across modeling philosophies.

### 6.3 Hyperparameter Tuning and Cross-Validation

Hyperparameter tuning was conducted using **GridSearchCV** from scikit-learn, applying **5-fold cross-validation** for each model. The objective was to maximize R² score on the validation set while minimizing overfitting. Tuning was particularly important for tree-based and kernel-based models.

**Tuned parameters included**:

* **Random Forest**: n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf
* **Gradient Boosting**: n\_estimators, learning\_rate, max\_depth
* **XGBoost**: learning\_rate, max\_depth, subsample, colsample\_bytree
* **SVR**: C, epsilon, and kernel (RBF)
* **KNN**: n\_neighbors, weights (uniform or distance-based)

Linear Regression was left untuned, as it contains no hyperparameters in its standard form.

This tuning process resulted in significantly improved performance across most models, especially Random Forest and SVR.

### 6.4 Model Evaluation Metrics

Three primary metrics were used to assess model performance:

* **R² (Coefficient of Determination)**: Measures the proportion of variance in the target variable explained by the model. A value closer to 1 indicates better fit.
* **Mean Absolute Error (MAE)**: Represents the average magnitude of error, offering a simple and interpretable measure of prediction accuracy.
* **Root Mean Squared Error (RMSE)**: Penalizes larger errors more than MAE, making it useful for identifying models that minimize large deviations.

These metrics were selected due to their widespread use in regression problems and their ability to provide a well-rounded evaluation of performance (Gareth et al., 2013).

Note: Precision and Recall were not used, as these are classification metrics and not applicable to continuous prediction tasks.

### 6.5 Comparative Results and Analysis

The performance of all six models is summarized in the table below:

| **Model** | **R² Score** | **MAE** | **RMSE** |
| --- | --- | --- | --- |
| Linear Regression | 0.9551 | 3.41 | 7.98 |
| Random Forest | 0.9880 | 1.67 | 4.12 |
| Gradient Boosting | 0.9811 | 2.34 | 5.18 |
| XGBoost | 0.9759 | 2.32 | 5.85 |
| SVR | 0.9932 | 2.36 | 3.11 |
| KNN Regressor | 0.9970 | 1.02 | 2.05 |

**KNN Regressor** outperformed all other models, achieving the highest R² (0.9970) and the lowest MAE and RMSE. This suggests that instance-based learning was particularly effective in this dataset, likely due to well-behaved feature scaling and the inclusion of historical housing price trends.

**SVR** also performed well, confirming its strength in capturing non-linear patterns. **Random Forest** offered a strong balance of performance and interpretability, particularly when combined with feature importance visualization. Ensemble models like **Gradient Boosting** and **XGBoost** performed well but were marginally less accurate in this case.

Visual diagnostics, including residual plots and actual vs. predicted scatter plots, showed that the top models achieved excellent predictive consistency. Summary plots confirmed that features like average income, Q2\_2024 housing price, and urbanization rate had the highest global impact on the prediction output (Lundberg and Lee, 2017).

These results underscore the value of using multiple models and incorporating both temporal and socio-economic features to build a robust housing price forecasting framework.

## 7. Results & Insights

The evaluation results across six machine learning models provided key insights into the effectiveness of different regression strategies for housing price prediction in the EU. The performance comparison clearly showed that incorporating both economic indicators and historical quarterly housing prices significantly improves predictive accuracy. The inclusion of temporal data (Q1 2020 to Q2 2024) was instrumental in allowing the models to capture lag-based trends and regional seasonality.

The **K-Nearest Neighbors (KNN) Regressor** delivered the best overall performance, achieving an R² score of **0.9970**, MAE of **1.02**, and RMSE of **2.05**. Its non-parametric, distance-based nature made it highly effective in this well-distributed, structured dataset. The success of KNN also reflected the usefulness of the feature scaling process and the relevance of input features, particularly those derived from past housing prices and socio-economic ratios.

**Support Vector Regression (SVR)** was another high performer, with an R² of **0.9932** and RMSE of **3.11**. SVR effectively modeled complex, non-linear relationships in the data, particularly through the use of the RBF kernel. Its sensitivity to scaling was mitigated by careful preprocessing using StandardScaler.

**Random Forest** achieved an excellent balance between predictive accuracy and interpretability. With an R² of **0.9880** and RMSE of **4.12**, it consistently ranked among the top three models and was especially useful for analyzing feature importance. Ensemble-based methods like **Gradient Boosting** and **XGBoost** also performed competitively, achieving R² values above 0.97, but were marginally outperformed by KNN and SVR.

Feature importance plots from Random Forest and Gradient Boosting identified **average income**, **Q2\_2024 housing prices**, and **urbanization rate** as the most influential predictors. This finding aligns with domain knowledge, where income levels and recent market trends are well-established drivers of housing demand (Gyourko et al., 2021; European Commission, 2022).

Overall, the project demonstrated that models capable of capturing non-linear relationships and temporal dependencies — such as KNN, SVR, and Random Forest — consistently outperform linear and purely additive models. The inclusion of past housing price data and relevant macroeconomic indicators was instrumental in improving accuracy and capturing real-world price dynamics.

## 8. Challenges & Mitigation

Throughout the course of this project, several technical and practical challenges were encountered. These challenges related primarily to data preparation, model tuning, performance balancing, and evaluation consistency. However, each was addressed through iterative experimentation, diagnostic analysis, and the application of appropriate machine learning practices.

One of the first challenges involved deciding how to treat the **historical housing price data** from Q1 2020 to Q2 2024. In earlier versions of the notebook, these quarterly values were excluded from the input features, leading to lower model performance and poor temporal awareness. Upon including these historical quarters as predictors, model accuracy improved significantly. The key mitigation strategy here was to preserve the full time series and reformat it to support tabular regression — a simple but effective enhancement.

Another challenge involved **feature scaling**, which was critical for distance-based and kernel-based models like **SVR** and **KNN**. Without appropriate scaling, these models produced suboptimal results or became sensitive to irrelevant feature magnitudes. This was mitigated by using StandardScaler on numerical features while leaving tree-based models unaffected. This dual-preprocessing pipeline ensured compatibility across all algorithms.

**Hyperparameter tuning** also posed a time and computational cost. Exhaustive grid search using GridSearchCV with five-fold cross-validation was essential but expensive, especially for XGBoost and SVR. This was managed by carefully constraining the search space to realistic and literature-supported parameter values (Pedregosa et al., 2011; Chen and Guestrin, 2016).

**Model evaluation and comparison** required consistent metric application and visualization to avoid subjective interpretation. In early stages, the models were evaluated on R² only, but later expanded to include **MAE** and **RMSE**, allowing a more nuanced comparison. This challenge was mitigated by implementing a central comparison table and standardizing evaluation code blocks across all models.

Finally, maintaining **code modularity and interpretability** while integrating six different models, tuning pipelines, and diagnostic plots required clear commenting, clean notebook organization, and markdown-based narrative explanations. Consistent section headers and reusable code patterns were key to resolving complexity and ensuring reproducibility.

Overall, while the project faced several common challenges in real-world machine learning workflows — including data integration, model calibration, and performance evaluation — each was addressed using transparent, methodical, and repeatable solutions.