Quarterly Forecasting of Housing Prices in Portugal Using Transactional and Macroeconomic Data

José Pedro Martinho Oliveira

Master's Dissertation

Advisor at FEUP: Prof. Gonçalo Figueira
Advisor at the company: Mr. Diogo Almeida

U. PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Master's in Industrial Engineering and Management

2025-06-27

Resumo

Esta dissertação apresenta o desenvolvimento e implementação de uma ferramenta de previsão baseada em machine learning para estimar preços de imóveis residenciais em várias regiões em Portugal, com o objetivo principal de prever o preço médio por metro quadrado para trimestres futuros. Pretende-se apoiar os profissionais do setor imobiliário com avaliações mais rápidas, objetivas e informadas por dados, complementando os métodos tradicionais que se baseiam frequentemente na opinião de especialistas e em simples análises comparáveis.

No contexto atual do mercado, as empresas do setor imobiliário raramente dispõem de sistemas automatizados de previsão. As estimativas de preços são muitas vezes produzidas manualmente, o que limita a escalabilidade, a consistência e a capacidade de resposta a alterações no mercado. Para ultrapassar essa limitação, este trabalho propõe uma solução orientada por dados que combina registos de transações imobiliárias com indicadores macroeconómicos, através de um processo automatizado de aprendizagem automática.

A metodologia adotada seguiu o modelo CRISP-DM (Cross Industry Standard Process for Data Mining), evoluindo desde a compreensão do negócio até à implementação do modelo. Foi construído um conjunto de dados abrangente, integrando informações provenientes de plataformas transacionais e bases de dados macroeconómicas oficiais. Após um extenso pré-processamento, incluindo limpeza de dados, engenharia de atributos e transformações sensíveis ao tempo, foram avaliados vários modelos de aprendizagem automática. Entre eles, o modelo XGBoost (eXtreme Gradient Boosting) obteve o melhor desempenho, superando significativamente os modelos lineares e de redes neuronais. Foram ainda incorporadas técnicas de interpretabilidade baseadas em valores SHAP (SHapley Additive exPlanations), de forma a garantir a transparência das previsões.

O modelo final foi integrado num sistema de previsão automatizado, capaz de se atualizar com novos dados e de gerar previsões dinâmicas. Os resultados são apresentados através de uma plataforma interativa em Power BI, permitindo aos utilizadores filtrar estimativas por localização, características do imóvel e horizonte temporal, com acesso a tendências históricas e métricas de desempenho.

Este projeto demonstra como a aprendizagem automática e a automação podem melhorar o processo de avaliação imobiliária, oferecendo previsões escaláveis, interpretáveis e continuamente atualizáveis. Melhorias futuras poderão incluir a incorporação de variáveis mais granulares ao nível do imóvel e a adaptação da ferramenta a diferentes tipos de análise imobiliária. A solução constitui uma base sólida para uma tomada de decisão estratégica e orientada por dados no setor imobiliário.

Abstract

This dissertation presents the development and implementation of a forecasting tool based on machine learning to estimate residential property prices in various regions in Portugal, with the main objective of predicting the average price per square meter for future quarters. The aim is to support real estate professionals with faster, more objective and data-informed valuations, complementing traditional methods that are often based on expert opinion and simple comparable analyses.

In the current market landscape, real estate companies typically lack automated forecasting systems. Price estimations are often produced manually, limiting scalability, consistency, and responsiveness to market changes. To address this limitation, this work proposes a data-driven solution that combines property transaction records and macroeconomic indicators through an automated, end-to-end machine learning pipeline.

The methodology followed the CRISP-DM (Cross Industry Standard Process for Data Mining) framework, progressing from business understanding to model deployment. A comprehensive dataset was built by integrating information from transactional platforms and official macroeconomic databases. After extensive preprocessing, covering data cleaning, feature engineering, and time-aware transformations, multiple machine learning models were evaluated. Among them, XGBoost (eXtreme Gradient Boosting) achieved the best performance, significantly outperforming linear and neural models. Interpretability techniques based on SHAP (SHapley Additive exPlanations) values were incorporated to ensure transparency in predictions.

The final model was deployed in an automated forecasting system capable of updating with new data and generating dynamic predictions. Forecast results are presented via an interactive Power BI dashboard, allowing users to filter estimates by location, property characteristics, and forecast horizon, with access to historical trends and performance metrics.

This project demonstrates how machine learning and automation can enhance real estate valuation by delivering scalable, interpretable, and continuously updatable forecasts. Future improvements may include incorporating finer-grained property-level features and adapting the tool to different types of real estate analysis. The solution lays a solid foundation for data-driven strategic decision-making in the real estate sector.

Agradecimentos

Em primeiro lugar, agradeço ao Sr. Nuno Pinto por ter confiado no meu potencial para a realização deste trabalho e por me ter recebido de braços abertos na empresa. Agradeço, também, ao Sr. Diogo Almeida por todo o apoio e disponibilidade durante estes últimos meses. Senti-me muito confortável na empresa desde o primeiro dia, e muito se deve a estes dois senhores.

Agradeço ao Professor Gonçalo Figueira, meu orientador na Faculdade de Engenharia da Universidade do Porto, desde os conselhos na definição do projeto até às dicas de escrita de uma dissertação. Todo o apoio contribuiu imenso para esta fase final do meu ciclo de estudos.

Agradeço ao Leandro Silva, amigo de longa data, por todos os momentos, apoio e companhia nos últimos anos. Ser teu colega de casa e vizinho do lado durante todos estes anos facilitou muito a minha experiência universitária.

Agradeço a todos os meus amigos que me acompanharam neste percurso. Vocês inspiram-me e tornam-me uma pessoa melhor dia após dia.

Um agradecimento especial a toda a minha família, que deposita imensa confiança em mim e demonstra orgulho a cada passo que dou. Aos avós, tios e primos, um agradecimento do fundo do meu coração.

À minha irmã que, mesmo sem ter a noção disso, traz-me estabilidade e motivação. Assim como fizeste, vou acompanhar cada passo teu.

Agradeço também ao meu pai por todo o apoio. Obrigado pela oportunidade de estudar deslocado na melhor universidade de engenharia do país que, sem dúvida, me tornou mais competente para o meu futuro. Obrigado por constantemente me relembrares das minhas capacidades e que devo seguir o meu caminho naquilo que me fizer feliz. És uma grande inspiração.

Por fim, agradeço especialmente à minha mãe por ter sido o meu ombro amigo e maior zona de conforto durante estes anos. É inexplicável o contributo que tiveste no meu percurso. Obrigado por teres sempre paciência para me ouvir desabafar quando as coisas não correm bem, e por me ofereceres sempre todo o tipo de ajuda em qualquer situação. São os valores que me transmites que levo para a minha vida.

Acknowledgments

First and foremost, I would like to thank Mr. Nuno Pinto for believing in my potential to carry out this work and for welcoming me into the company with open arms. I also thank Mr. Diogo Almeida for all the support and availability over the past few months. I felt very comfortable in the company from the very first day, largely thanks to these two gentlemen.

I am grateful to Professor Gonçalo Figueira, my advisor at the Faculty of Engineering of the University of Porto, for his guidance from the initial project definition to the writing tips for this dissertation. All the support he provided greatly contributed to this final phase of my academic journey.

I would like to thank Leandro Silva, a long-time friend, for all the moments, support, and companionship over the past years. Being your housemate and next-door neighbor all these years has made my university experience much easier.

I thank all my friends who accompanied me on this journey. You inspire me and make me a better person every single day.

A special thanks to all my family, who place immense trust in me and show pride in every step I take. To my grandparents, uncles, aunts, and cousins, a heartfelt thank you.

To my sister, who brings me stability and motivation. Just as you have done for me, I will be there for you every step of the way.

I also thank my father for all his support. Thank you for giving me the opportunity to study away from home at the best engineering university in the country, which has undoubtedly made me more competent for my future. Thank you for constantly reminding me of my capabilities and encouraging me to pursue what makes me happy. You are a great inspiration.

Lastly, I especially thank my mother for being my confidante and greatest source of comfort throughout these years. Your contribution to my journey is beyond words. Thank you for always having the patience to listen when things didn't go well, and for always offering every kind of help in any situation. The values you have instilled in me are those I carry for life.

"Sometimes people make it seem like you have to have certain prerequisites or a crazy life story in order to be successful in this world. But the truth is, you really don't."

Wardell Stephen Curry II

Contents

Acronyms and Symbols

CI Confidencial Imobiliário

SIR Sistema de Informação Residencial (Residential Information System)

BP Banco de Portugal

BPstat Banco de Portugal's statistical portal

XGBoost eXtreme Gradient Boosting

EDA Exploratory Data Analysis

GDP Gross Domestic Product

GFCF Gross Fixed Capital Formation

DOM Document Object Notation

HTML HyperText Markup Language

JSON JavaScript Object Notation

XML eXtensible Markup Language

API Application Programming Interface

FFill Forward-Fill

BFill Backward-Fill

R^2 Coefficient of Determination

MAE Mean Absolute Error

RMSE Root Mean Squared Error

MAPE Mean Absolute Percentage Error

SMAPE Symetric Mean Absolute Percentage Error

SHAP SHapley Additive exPlanations

AI	Artificial Intelligence
XAI	eXplainable Artificial Intelligence
CRISP-DM	Cross Industry Standard Process for Data Mining
ID	Identification
CSV	Comma-Seperated Values
URL	Uniform Resource Locator

List of Figures

List of Tables

Chapter 1

Introduction

The real estate market has a major impact on a country's economy, often serving as an index of its economic conditions (Vaidynathan et al., 2023). Housing prices not only affect the supply and demand for real estate, they also influence household consumption, investment and financial stability. Therefore, the ability to predict the evolution of prices is of great interest to various stakeholders (real estate investors, financial institutions, political decision-makers and private individuals) as it guides investment strategies and public policies. As unexpected fluctuations in the value of real estate can have significant macroeconomic impacts, making an accurate price forecast is a vital contributor to better risk management and informed decision-making (Sharma et al., 2024).

In Portugal, there has been a strong growth trend in house prices over the last decade, after a relatively stagnant period during the 1990s until the financial crisis of 2007, reaching record values in several local markets (Rodrigues, 2022). This reinforces the need for accurate price forecasting tools to support more informed decision-making by the various players in the real estate sector. It should be noted that accurate forecasting has a great impact on investment decisions and business strategies (Rodrigues, 2022).

However, estimating real estate values can be difficult due to the market's complexity. Numerous elements impact property prices, from the property's location to specific aspects such as the type of home, its age, and general condition. Economic and social factors like job market conditions, interest rates, and population changes also play significant roles (Rodrigues, 2022). These variables have nonlinear interactions, which ultimately makes traditional valuation methods (often based solely on consultants' experience and direct comparative analyses) quite limited in terms of perceiving market complexity and difficult to scale due to the need to manually perform a large number of individual analyses. It is in this context that ML and AI approaches emerge as promising solutions. ML techniques can analyze extensive datasets and identify subtle trends, making predictions quicker and more accurate (Jafary et al., 2024). This development reflects a wider shift toward using technology and data-driven strategies in the real estate industry, including automated valuation methods and forecasting market behaviors (Conway, 2018).

It is important to highlight that access to detailed real estate market data today is much greater than it was a few years ago, creating new opportunities to apply advanced analytical methods.

The growing digitalization of this information has helped overcome previous barriers that made analysis difficult, enabling the development of more reliable predictive models (Sharma et al., 2024). Thanks to the wider access to detailed data and the improvement of analytical tools, it is now more practical to track and forecast price changes with better accuracy. For real estate companies, this shift opens the door to stronger positioning in the market, helping them deliver more reliable valuations and respond more quickly to changing trends.

This project was born from that very context, through a partnership with DILS. As an international company in the real estate market, DILS recognizes the growing importance of using data analysis and forecasting tools to strengthen its business practices and continuously improve its services. The need for more precise and objective property valuations, especially in a market that is evolving quickly and becoming more information-driven, was a major reason behind launching this dissertation work.

1.1 Company Overview and Project Objectives

DILS is a leading international real estate services group founded in Italy in 1971, with offices across Milan, Rome, Amsterdam, Lisbon, Porto, Algarve, Barcelona, Madrid and beyond (Dils, 2024). In May 2024, DILS entered the Portuguese market by acquiring Castelhana, a 25-year-old specialist in premium residential developments in Lisbon. This strategic move laid the foundation for a multi-million-euro investment plan to accelerate growth, introduce innovative residential practices, and diversify into retail, office and hospitality sectors in Portugal (Dils, 2024).

Despite its deep domain expertise and market intelligence, DILS recognized that traditional valuation methods, such as comparative market analysis using recent sales comparables, tend to be subjective, slow to adapt, and limited in predictive power. To modernize its toolkit, a 4.5-month dissertation-driven project was launched to build an automated ML forecasting tool for price per square meter in Portugal. This metric serves as a fundamental reference in real estate valuation, commonly used for benchmarking and comparing properties with varying areas and characteristics. The tool is expected to deliver forecasts as granular as possible for a near-future period to be defined, aligning with the operational needs of consultants and decision-makers.

This initiative is structured around three primary objectives:

- **Rapid Estimation of Future Property Values:** Generation of short-term forecasts of price per square meter across different segments and geographies, reducing analysis time from weeks to hours by training models on historical transaction and economic data. These forecasts aim to support stakeholders in multiple use cases, such as estimating the fair market value of a property, identifying investment opportunities, monitoring local price dynamics, and anticipating short-term market shifts.

- **Integration of Diverse Variables:** Enhance predictive accuracy by engineering features from detailed property attributes and macroeconomic indicators, allowing the model to capture both local physical conditions and broader economic trends that influence pricing.

- **Advancing the academic understanding of predictive modeling in real estate:** Beyond delivering a functional tool for the company, this dissertation aims to explore and compare different ML algorithms in terms of predictive performance, robustness, and suitability for time-aware forecasting. Particular emphasis is placed on model interpretability and on the construction of an automated, reusable pipeline. This objective supports the broader goal of contributing methodological insights to the application of ML in real estate markets.

Finally, all forecasts and historical fits will be presented via an interactive Power BI dashboard, enabling stakeholders to explore prediction intervals and model diagnostics in a user-friendly interface.

By uniting DILS's European expansion strategy with advanced analytics and automated workflows, the project eliminates subjective bias, accelerates decision cycles, and anchors market studies in transparent, empirically validated forecasts.

1.2 Methodology Used Throughout the Project

The methodological approach adopted in this project is inspired by the CRISP-DM framework, a robust and widely accepted model for structuring data mining and ML projects (Ncr et al., 1999). This structured approach ensures transparency, reproducibility, and rigor across the various phases of the study.

The CRISP-DM model is composed of six key phases (Figure A.1), and each of these stages was adapted to the context of real estate price forecasting:

- **Business Understanding:** The primary goal of the project was to develop a ML-based tool to forecast real estate prices per square meter. This objective, grounded in the operational needs of DILS, guided the entire analytical process from data acquisition to model deployment.

- **Data Understanding:** Initial data collection involved sourcing and consolidating property and macroeconomic data from available platforms. The focus in this phase was on evaluating the structure, completeness, and consistency of the data to identify early issues and opportunities for improvement. An EDA was also conducted to better understand the distribution of key variables, detect outliers and anomalies, and uncover temporal and spatial patterns relevant for subsequent modeling decisions.

- **Data Preparation:** Significant effort was directed toward cleaning, transforming, and enriching the data. This phase included selecting relevant features, handling missing values, and other steps to meet the technical requirements of ML algorithms.

- **Modeling:** Various ML models were explored and trained to forecast the price per square meter of properties in Portugal. Although the algorithms are specifically discussed in later sections, this phase was characterized by iterative testing and performance evaluation to identify the most suitable model for the task.

- **Evaluation:** The performance of the models was assessed against clearly defined success criteria, primarily predictive accuracy and robustness. The evaluation ensured that the model not only met statistical benchmarks but also aligned with the business objectives outlined at the project's inception.
- **Deployment:** Finally, the selected model was integrated into an automated pipeline and embedded within an interactive Power BI dashboard. This step ensured that the forecasting tool could be operationalized and regularly updated, making it a practical asset for ongoing market analysis.

By adhering to the CRISP-DM methodology, this project ensures methodological soundness while also delivering practical value through a data-driven decision support tool.

1.3 Dissertation Structure

The structure of this dissertation is designed to guide the reader through the rationale, development, and application of a real estate forecasting tool, following a logical progression from problem definition to results interpretation.

The next chapter provides a comprehensive review of the academic and technical literature relevant to property price forecasting. It introduces key modeling approaches, discusses the impact of various explanatory variables, and addresses prevalent challenges.

Then, the third chapter focuses on defining the business problem in detail. It outlines the limitations of the available data, the expectations of end-users, and the practical considerations that influenced the design of the forecasting system. This chapter serves as a bridge between theoretical concepts and practical implementation.

The fourth chapter is dedicated to the methodological approach. It explains the development process, including data handling, model training, and the use of visualization tools to deliver forecasts in an interactive and accessible format.

In the fifth chapter, the results are presented and analyzed. The performance of the models is evaluated using statistical error measures, the contribution of the different variables is analyzed, and the functionality of the final result is demonstrated through applied examples.

The final chapter draws conclusions, summarizing the achievements and acknowledging the limitations of the project. It also outlines opportunities for future enhancements and research extensions. A remark on the project's contribution to the sustainable development goals is also included in the final appendix of this dissertation (Appendix L).

Chapter 2

Literature Review

This chapter presents a structured literature review on the key concepts and methodologies relevant to real estate price forecasting. It begins by contrasting traditional econometric models with more recent machine learning approaches, highlighting their respective advantages and limitations. The review then addresses critical modeling considerations such as overfitting risks, the role of spatial and temporal granularity, and the integration of macroeconomic variables into predictive frameworks.

Subsequently, the chapter explores the practical steps involved in building robust forecasting models. These include data collection techniques, preprocessing tasks, and evaluation methods for model selection and validation. The chapter concludes with a focus on interpretability, introducing tools to explain model outputs and ensure transparency in real estate valuation contexts.

2.1 Real Estate Price Forecasting Models

Forecasting models for real estate prices can be broadly divided into classical econometric models and modern ML models. The traditional approach is the hedonic price model, where property price is explained by a linear or semi-logarithmic regression of observable characteristics such as area, number of rooms, location, and age (Wei et al., 2022). While these models are interpretable and grounded in theory, they assume linear additive relationships and full availability of relevant attributes (conditions rarely met in practice). This limitation often leads to bias and inability to capture complex interactions (Wei et al., 2022).

Nonlinear ML techniques have advanced significantly in recent decades, driven by greater data availability and computational power. Artificial Neural Networks (ANNs), for instance, have outperformed hedonic models in prediction accuracy. Abidoye and Chan (2018) found that an ANN achieved a MAPE of 15.94% compared to 38.23% with a hedonic model in Nigeria. In other words, the neural network proved about twice as accurate as the classic multiple regression model. Despite better performance, ANNs require large datasets, careful hyperparameter tuning, and have limited interpretability.

Ensemble methods such as Random Forest and Gradient Boosting have become widely used for price prediction. Random Forest builds multiple decision trees from bootstrap samples and averages their outputs, reducing overfitting. Gradient Boosting builds trees sequentially to correct previous errors. In a comparative study, Sharma et al. (2024) showed that XGBoost outperformed other models, including Random Forest and Neural Networks, in house price prediction. These methods offer high predictive power but at the cost of greater computational load and lower transparency.

Deep learning models have also been applied successfully. Xu and Zhang (2021), for example, used a recurrent neural network to forecast house price indices in 100 Chinese cities, achieving high accuracy (Xu and Zhang, 2021). However, deep networks are prone to overfitting, require significant processing resources, and demand dense datasets (conditions that may not be met in small-scale applications).

Over time, model usage has evolved. From the 1970s to 1990s, linear models dominated. In the 2000s, support vector machines and ensemble methods gained popularity. More recently, deep learning has become widespread. As noted by Al-Qawasmi (2022), recent literature (2017–2020) frequently applied Neural Networks, regression models, Random Forest, and boosting algorithms. This pattern indicates that no single universal best approach exists, as the choice depends on context and available data. Comparative studies generally find that boosting methods tend to outperform traditional linear models in terms of predictive error (Abidoye and Chan, 2018; Jafary et al., 2024), but the trade-offs involving interpretability, computational cost and explainability must be taken into account.

2.2 Understanding Overfitting

Overfitting is a common issue in predictive modeling that arises when a model learns not only the underlying patterns in the training data but also the noise and anomalies. This can occur when models become too specific to the historical data used during training, resulting in poor generalization to new or unseen data (Baldominos et al., 2018). As a result, the model may produce inaccurate predictions when applied to different market conditions. The real estate sector, characterized by high variability in property attributes and market dynamics, is particularly vulnerable to this problem (Baldominos et al., 2018).

Overfitting can severely affect real estate decision-making. If a model misinterprets market noise as significant patterns, it may incorrectly assess property values. This could lead to the misidentification of investment opportunities, particularly if the model flags properties as undervalued or overvalued based on non-representative data points. In a study using XGBoost for property valuation, it was observed that without appropriate regularization and parameter tuning, the model was susceptible to overfitting despite its strong predictive capabilities (Sharma et al., 2024).

Several mitigation strategies are essential to prevent overfitting in real estate models. Throughout this chapter, some overfitting prevention techniques will be mentioned, as they aim to ensure that the model learns generalizable patterns rather than specific anomalies. Addressing overfitting is critical for developing robust and reliable predictive models in real estate. Effective management of this issue ensures that ML tools contribute meaningful insights for property valuation and investment analysis across various market contexts.

2.3 Spatial and Temporal Granularity of Models

Granularity refers to the degree of detail with which information is structured (Figure 2.1), enabling the capture of fine variations in the occurrences of interest. In real estate, the spatial and temporal granularity of online property listings (namely geographic precision and update frequency) are essential for real-time monitoring of supply and demand. As shown by Loberto et al. (2020), such detail improves the reliability of price trend analyses and reduces bias caused by duplicate or outdated listings.

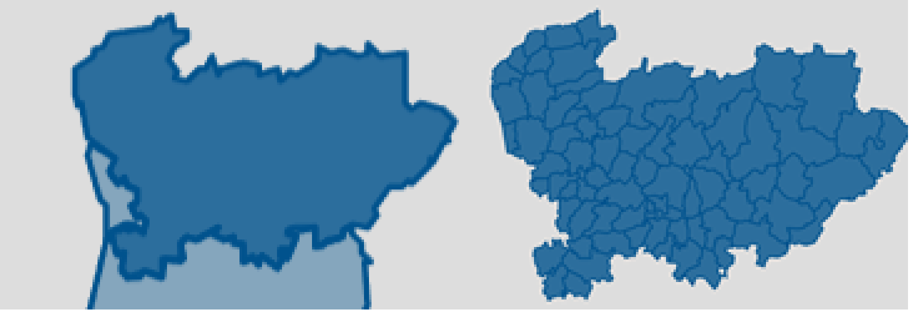


Figure 2.1: Lower (left) vs Higher (right) spatial granularity

In forecasting real estate prices with ML, the definition of spatial and temporal granularity is crucial. Spatial granularity determines geographic resolution (district, municipality, parish, 500×500 m grid), while temporal granularity defines observation frequency (monthly, quarterly, annual). These parameters must strike a balance between capturing local patterns and ensuring statistical robustness, avoiding both excessive noise and overfitting (Wentland et al., 2023).

Larger spatial units aggregate more transactions and stabilize estimates but may obscure important local variations (Jaroszewicz and Horynek, 2024). Finer scales reveal geographic disparities but often face data sparsity and reduced generalization. Additionally, spatial autocorrelation (especially in dense urban settings) can distort predictions if not accounted for, as nearby properties influence each other's values (Lo et al., 2022). Temporal granularity involves similar trade-offs. High-frequency data (for example, monthly) enables models to react quickly to short-term shocks and seasonal fluctuations (Rostami-Tabar and Mircetic, 2023), which is beneficial in volatile markets. However, it also increases noise and the likelihood of overfitting, especially where transaction volumes are low. Conversely, quarterly or annual data smooths out short-term volatility and improves performance in long-term forecasting scenarios (Varghese et al., 2023).

Additionally, Soltani et al. (2022) report that integrating spatio-temporal lag variables further enhances forecasting accuracy by accounting for dependencies across both space and time, especially across heterogeneous geographic contexts.

In sum, effective ML-based forecasting of housing prices depends on context-specific calibration of spatial and temporal granularity. Poorly chosen granularity settings can lead to models that either miss critical localized dynamics or are overwhelmed by noise, ultimately reducing their practical value in real estate applications.

2.4 Macroeconomic Variables Relevant to the Real Estate Market

The integration of macroeconomic variables into ML models for real estate price forecasting is increasingly emphasized due to their ability to capture systemic economic forces influencing housing demand and supply. Variables such as GDP, inflation, unemployment, and government expenditures are consistently associated with shifts in market expectations and buyer behavior (Vaidynathan et al., 2023). For example, real GDP growth tends to boost income and employment, stimulating housing demand (Vaidynathan et al., 2023; Iacoviello and Neri, 2010), whereas high unemployment suppresses affordability and transaction volumes.

Inflation presents a dual effect. It can erode purchasing power, reducing affordability, but also enhance housing's appeal as a store of value during uncertain periods (Vaidynathan et al., 2023). Government consumption has a direct and sustained impact on real house prices by increasing aggregate demand and disposable income, as demonstrated by Khan and Reza (2017), with results robust across methods and timeframes. Moreover, the housing market interacts with the broader economy, not only responding to but also influencing consumption and output through wealth effects and credit channels (Iacoviello and Neri, 2010).

Additional indicators, such as private consumption, GFCF, exports, and imports, signal broader economic trends relevant to housing. GFCF often accompanies infrastructure development and urban expansion (Vaidynathan et al., 2023), while trade variables influence GDP and employment in tradable sectors, especially in open economies. Incorporating these indicators aligns with recent ML-based forecasting studies, where models such as decision trees, random forests, and boosting techniques have proven effective in capturing their nonlinear and interacting effects (Vaidynathan et al., 2023).

Ultimately, including macroeconomic variables enhances predictive accuracy and embeds economic realism into ML models. While the relative influence of each variable may vary by context, their inclusion ensures responsiveness to both structural and cyclical changes in the economy (Iacoviello and Neri, 2010; Vaidynathan et al., 2023; Khan and Reza, 2017).

2.5 Database Creation

In modern forecasting workflows, assembling a unified database from diverse online sources is a foundational step that directly impacts model accuracy and robustness. Programmatic ingestion methods enable researchers to gather both high-frequency recent data and extensive historical records within a coherent, time-stamped repository. This layered strategy ensures comprehensive coverage and structural consistency, allowing subsequent cleaning, normalization, and enrichment processes to produce the high-granularity datasets that state-of-the-art forecasting models require (Ferrara et al., 2014).

2.5.1 Web Scraping Techniques

Modern web scraping ecosystems now span a spectrum of approaches to match the nature of target pages. When dealing with JavaScript-driven sites, headless browser tools such as Selenium emulate full user interactions, loading scripts, firing events, and exposing a rendered DOM tree for extraction. In contrast, for predominantly static HTML, lightweight parsers like Beautiful Soup and Selenium provide high-speed, low-overhead parsing ideal for large-scale, low-latency pipelines. On the cutting edge, many advanced scrapers bypass HTML parsing altogether by reverse-engineering JSON or XML endpoints exposed by the site, which both accelerates data retrieval and reduces brittleness (Khder, 2021). Yet without careful safeguards, naïve scraping can introduce sampling bias, whether from personalized content feeds or rapidly shifting page layouts, so it's critical to integrate systematic data cleaning and validation steps to detect and correct such distortions (Khder, 2021).

2.5.2 API-Based and Structured Data Access

APIs offer a stable, documented interface for data retrieval, often including authentication, rate-limiting, and versioning features that safeguard both providers and consumers. Well-designed APIs abstract away presentation details, exposing only the necessary endpoints and fields, which streamlines downstream storage and processing (Foerderer, 2023). Moreover, APIs enable selective data queries (such as date-range filtering or field selection) reducing bandwidth and storage overhead. However, APIs may not expose every data point of interest; combining API calls with targeted scraping can fill those gaps.

2.6 Data Preprocessing

Data preprocessing is an indispensable step in real-estate price forecasting, ensuring that the inputs are of high quality and ready for modeling. In the era of big data, uneven data quality can significantly degrade appraisal accuracy, making preprocessing a standard prerequisite in modern pipelines (Wei et al., 2022).

2.6.1 Data Cleaning

2.6.1.1 Missing Value Imputation in Temporal Real-Estate Data

Missing values are a common problem in statistical datasets, often stemming from non-responses, dropouts, sensor malfunctions, or data integration issues. If unaddressed, these gaps can introduce significant bias into analyses, particularly in time-dependent datasets like real estate price series (Ribeiro and de Castro, 2021).

Several basic imputation techniques are frequently employed. Mean substitution replaces missing entries with the average of observed values, preserving the overall mean but reducing variance and correlation metrics. Median imputation, more robust against outliers, is advantageous for skewed distributions. Mode imputation is suited to categorical data, filling missing entries with the most frequent category but disregards inter-variable dependencies (Emmanuel et al., 2021).

For sequential data such as real estate time series, temporal imputation strategies offer more relevance. FFill imputes missing prices by carrying forward the last observed value, maintaining chronological order and avoiding future information leakage. Conversely, BFill uses the next available value, which can lead to look-ahead bias if applied prematurely (Ribeiro and de Castro, 2021). Empirical studies confirm that FFill and BFill outperform simpler methods in scenarios with high autocorrelation, with FFill often preferred in predictive modeling to ensure causality is respected (Kamalov and Sulieman, 2021). Moreover, FFill is particularly effective for preserving trends in positively correlated time series data, whereas BFill is better suited for exploratory analyses rather than predictive tasks (Kamalov and Sulieman, 2021).

2.6.1.2 Outlier Handling in Real Estate

An outlier is an observation in a dataset that deviates significantly from other observations, potentially indicating variability in measurement, experimental error, or a novel phenomenon. It can distort statistical analyses and model predictions if not properly addressed (Páez, 2009).

In terms of real estate prices, outliers frequently reflect real market heterogeneity, such as the presence of high-end districts or luxury submarkets, rather than merely constituting data anomalies. Consequently, indiscriminate removal of such data points risks under-representing affluent areas and potentially biases predictive models towards mid-range transactions (Páez, 2009). This under-representation can systematically lead to the under-prediction of values in high-value regions, as emphasized in spatial econometric studies that demonstrate the significance of recognizing market segmentation and spatial autocorrelation in property valuation models (Páez, 2009).

2.6.2 Data Normalization

Data normalization rescales features to a common scale, preventing variables with different magnitudes from dominating model training and improving numerical stability in real-estate forecasting pipelines (de Amorim et al., 2023). Among the most commonly employed techniques, *Z-score* normalization and Min-Max scaling are widely used.

Z-score normalization transforms features to have zero mean and unit variance, making it particularly effective when features follow a Gaussian distribution or when preserving outlier influence is acceptable (de Amorim et al., 2023). This is done by subtracting the set average from the value and dividing it by the standard deviation. Conversely, *Min-Max* scaling rescales features to a predefined range, typically [0,1], which is beneficial for algorithms sensitive to the absolute magnitude of data, such as Neural Networks (de Amorim et al., 2023).

2.6.3 Feature Engineering

Feature engineering is another pivotal phase in the development of predictive models, particularly within real estate forecasting, where data often encompasses diverse temporal, spatial, and socioeconomic dimensions (Zhao et al., 2022). It refers to the process of transforming raw variables into meaningful inputs that enhance a model's capacity to detect patterns and make accurate predictions. This can involve decomposing composite variables, encoding categorical data, or designing new features that capture trends or weights based on domain knowledge (Lee et al., 2023).

In practice, even modest adjustments can markedly improve a model's performance. These enhancements help ensure that the model captures both persistent and evolving trends in the data (Lee et al., 2023). Ultimately, thoughtful feature engineering bridges the gap between raw information and predictive insight, serving as a foundation for building reliable and interpretable models (Zhao et al., 2022).

2.7 Methodologies for Evaluating, Validating and Adjusting Models

2.7.1 Model Evaluation

A rigorous evaluation of predictive models is essential in regression tasks such as forecasting property prices. Performance metrics must balance statistical robustness with interpretability, especially in real estate, where model transparency affects investment decisions and public trust.

In this context, the coefficient of determination (R^2) stands out as the most informative and appropriate metric. It quantifies the proportion of variance in the dependent variable explained by the independent variables, offering a clear, intuitive interpretation of model performance. Being scale-independent, R^2 allows comparison across datasets with differing units and ranges, which is particularly relevant in real estate markets marked by high price variability (Chicco et al., 2021).

Chicco et al. (2021) argue that R^2 outperforms other common metrics like MAE, MAPE, RMSE, and SMAPE in informativeness and comparability. These alternatives, while useful for expressing prediction errors in concrete terms, have drawbacks such as unbounded upper limits and scale dependency. For example, the same RMSE can reflect different levels of performance depending on the dataset, whereas an R^2 value near 1 consistently indicates strong predictive accuracy.

Nonetheless, MAE and RMSE remain important. Their main advantage is interpretability in real-world units. MAE reflects the average magnitude of errors, while RMSE penalizes larger deviations, useful in financial applications such as property valuation. A MAE of 500 directly communicates the typical error to stakeholders like investors and clients. RMSE, due to its sensitivity to large errors, is especially valuable in risk-sensitive scenarios (Chicco et al., 2021).

These insights are supported by empirical work. In Hernandez et al. (2024), Linear Regression, Lasso, and XGBoost were applied to house prices in Frisco and Plano, Texas, using a range of metrics. R^2 captured the overall explanatory power of the models and highlighted key predictors like area, location, and property age. Meanwhile, MAE and RMSE provided practical context on monetary error, aiding stakeholder understanding of real-world impacts (Hernandez et al., 2024).

In sum, while R^2 is recommended as the primary evaluation metric for its clarity, scalability, and robustness, it should be complemented by MAE and RMSE. A multimetric strategy ensures both statistical soundness and practical relevance in real estate modeling, supporting balanced, evidence-based decision-making.

2.7.2 Model Validation

The temporal arrangement of data imposes a strict chronological order on model training and evaluation. In predictive modeling contexts, particularly time series forecasting, this temporal structure demands validation methods that respect the natural ordering of data while preventing information leakage from the future into the past.

Temporal holdout validation splits the dataset along this timeline in such a way that the training set contains only earlier observations and the test set consists solely of subsequent ones. This method offers a straightforward yet effective approach to estimate out-of-sample performance, preserving the integrity of the evaluation by avoiding contamination from future information (Tashman, 2000). It mirrors a real-world forecasting scenario where predictions rely exclusively on past data.

Expanding on this, walk-forward validation (also referred to as forward chaining or rolling origin) provides a more dynamic and robust mechanism. In this method, the model is retrained iteratively so that with each step, the most recent observations are added to the training set, and testing is performed on the next time segment. This process enables the model to adjust to structural changes in the data such as market shifts or policy transitions, which helps reduce bias and enhances the realism of performance evaluation. Empirical studies indicate that rolling-origin validation effectively captures temporal variability in forecast accuracy and promotes better generalization (Tashman, 2000).

Scikit-learn's *TimeSeriesSplit* is a robust and widely used implementation of this methodology. It generates a series of sequential train-test partitions with expanding training windows and separate test periods, always preserving the order of time (Figure B.1). This structure prevents future data from influencing model training and is particularly appropriate for models that incorporate time-dependent behavior (Arlot and Celisse, 2010).

In contrast with traditional *k-fold* cross-validation, which assumes sample independence and identical distribution, time series cross-validation accommodates the sequential dependence inherent in temporal data. Applying standard *k-fold* to time series can lead to inflated performance assessments because of unintended inclusion of future information in training. This underscores the importance of aligning validation strategies with the structure of the data (Arlot and Celisse, 2010).

Furthermore, the literature emphasizes that validation methods should not only minimize overfitting but also promote robustness through exposure to diverse temporal patterns and anomalies. This includes strategies such as employing multiple test periods, recalibrating model coefficients regularly, and using rolling training windows. These practices help ensure that evaluation procedures are both resilient and representative of real-world application scenarios (Tashman, 2000).

2.7.3 Adjusting Hyperparameters

Hyperparameters are configuration settings defined prior to training a ML model, governing aspects like learning rate, model complexity, and training duration. Hyperparameters have a significant influence on the model's performance and generalization, so tuning them correctly is essential for obtaining the best results (Arnold et al., 2024).

Automated techniques are widely employed for hyperparameter tuning in ML. A traditional approach is *GridSearchCV* provided by Scikit-learn, which exhaustively tests combinations in a predefined grid. This method is effective when the parameter space is small but becomes computationally infeasible for complex models due to the combinatorial explosion of possible parameter combinations (Scikit-learn developers, 2024a). For instance, if there are three hyperparameters with 10 possible values each, *GridSearchCV* it would evaluate 1,000 combinations. This exhaustive nature ensures that the best combination within the grid is found but at the cost of significant computational resources (Scikit-learn developers, 2024a).

An alternative is *RandomizedSearchCV*, also available in scikit-learn, which addresses the computational inefficiency of grid search by sampling a fixed number of parameter settings from specified distributions. Instead of evaluating all possible combinations, it selects a random subset, allowing for a more efficient exploration of the hyperparameter space (Scikit-learn developers, 2024b). This approach is particularly beneficial when some hyperparameters have a more significant impact on model performance than others, as it can discover good combinations without exhaustively searching the entire space (Scikit-learn developers, 2024b). Moreover, *RandomizedSearchCV* allows for the specification of distributions for continuous hyperparameters, enabling a more nuanced search. A visual representation of *GridSearch* and *RandomSearch* can be seen in Figure B.2.

In summary, the final model is generally obtained through internal validation combined with automated search techniques like the ones presented, aiming to maximize predictive accuracy metrics without overfitting. The choice among these methods depends on the size of the hyperparameter space, computational resources, and the specific requirements of the task at hand. Employing robust hyperparameter tuning libraries, along with thorough model evaluation and validation strategies, ensures that the chosen model is both accurate and reliable when deployed in real-world scenarios.

2.8 Interpretable and Justifiable Approaches in ML Models

In the real estate domain, where credit and investment decisions involve large sums, model interpretability is crucial. Interpretable approaches typically rely on “white box” models with transparent mechanics, such as linear regressions and simple decision trees (Management Solutions, 2023). These models allow predictions to be justified using clear mathematical or logical terms, which helps build trust among users like investors, banks, and clients. However, this interpretability often comes at the cost of lower predictive performance compared to more complex “black box” models.

Advanced ML models are inherently opaque. In these cases, *ex post hoc* explanation techniques are used to extract justifications after the model has been trained. This creates a trade-off between model accuracy and the ability to justify predictions, leading many projects to adopt complex models supported by explanation tools (Management Solutions, 2023).

The field of XAI also promotes best practices for model transparency. Beyond selecting commercially meaningful variables and performing exploratory analysis, emphasis is placed on reproducibility and governance, such as version control and proper documentation. In short, even when using sophisticated models, results must be understandable by domain experts and explainable to stakeholders, especially in regulated or financial settings.

2.8.1 Interpretability Techniques in ML

Several techniques have been developed to support interpretability in real estate models, with a growing emphasis on making complex AI systems more transparent and trustworthy. A notable example is SHAP, which is part of a broader set of techniques within the field of XAI, and aims to make ML and AI models comprehensible to humans and thus align with regulatory and ethical standards (Management Solutions, 2023).

SHAP provides a unified framework grounded in cooperative game theory. It calculates the average marginal contribution of each feature across all possible feature combinations, thereby delivering consistent and fair attributions of feature importance. This allows SHAP to offer both global explanations of model behavior and localized insights into how specific features affect individual predictions. For instance, in a real estate model, SHAP can indicate that a variable such as GDP contributed positively by a quantifiable portion to the predicted property price (Management Solutions, 2023).

Concluding, SHAP is a part of a growing set of *post-hoc* interpretability methods, which are critical for industries like real estate where model decisions must be transparent and explainable to meet regulatory demands and maintain user trust. The application of such techniques is essential not only for enhancing model transparency but also for detecting bias, validating outputs, and enabling meaningful human oversight over ML or AI-driven decisions (Management Solutions, 2023).

Chapter 3

Problem Definition

This chapter will provide a more comprehensive description of the problem, making reference to resources already used by the company, as well as the inputs needed to develop a reliable forecasting tool. A more streamlined version of the project's mapping is presented through a SIPOC diagram in Appendix C.

3.1 Problem Characterization

DILS is a real estate consultancy that currently lacks a data-driven tool for forecasting residential property prices in Portugal. Price estimates (€/m²) are manually produced by consultants based on historical transaction data from CI's SIR extension (discussed later) and rely heavily on professional judgment. This traditional approach, grounded in comparable sales, lacks standardization and objectivity, leading to inconsistencies and inefficiencies. A predictive model could enhance transparency and improve the accuracy of market assessments.

Accurately anticipating housing market trends is essential, as real estate prices impact investment decisions, portfolio strategies, and policy planning. Forecasts help companies, investors, and homebuyers make better decisions, especially in volatile markets. Without a forecasting system, DILS is limited in its ability to provide timely, data-based advice.

This project aligns with a broader industry shift toward data-driven valuation. Conventional methods based on expert opinion and recent sales are prone to bias and struggle to scale with large, dynamic datasets. In contrast, automated valuation models powered by ML standardize and continuously update price estimates using big data. These models are increasingly adopted by major firms and supported in academic literature. Implementing such a system at DILS would modernize its operations, reduce reliance on subjective input, and enable evidence-based forecasting aligned with current market conditions.

The goal is to develop an integrated forecasting tool capable of projecting future prices per square meter, conditioned on geographic and temporal identifiers, property attributes (typology, condition) and macroeconomic indicators. Automated routines will support data ingestion, temporal and spatial alignment, model transformation, and recalibration to ensure timely, high-resolution forecasts.

This work aims to balance the constraints and opportunities of available data (detailed in upcoming subchapters) and to optimize forecast granularity and performance. Choices related to geographic levels, macroeconomic variables, and modeling techniques will be justified in detail later.

The objective is not to replace human expertise, but to complement it, as consultants remain responsible for interpreting results and applying domain knowledge where it adds the most value. Ultimately, this thesis responds to the absence of an accurate, automated platform for forecasting residential real estate prices in Portugal, addressing DILS's analytical gap and reflecting the broader shift toward ML-based decision support in the real estate sector.

3.2 Data Gathering

The aim of the project is therefore to create a tool capable of estimating the price per square meter of a property type in Portugal, taking into account geographical, physical, temporal and economic variables. The effectiveness and level of detail of the tool is highly dependent on the data available for analysis, which will be discussed below.

3.2.1 SIR for Real Estate Transactional Data

DILS consultants do their analysis using the CI's platform, which provides current and historical statistics on the real estate market. This platform has an extension, SIR, which allows various filters to be applied to enable more targeted analysis.

3.2.1.1 Territory Covered

To begin the explanation of the filters for the analyses, it is essential to mention that SIR provides information at an aggregate level and not on a per-property basis. This means that each result reflects the broader market trend for the selected area, taking into account all the other filters chosen. Individual transactions or specific addresses are never exposed; instead, SIR presents a picture of how prices and other key metrics are evolving across an entire zone.

When it comes to defining that zone, SIR offers various levels of geographical granularity. At the coarsest level, mainland Portugal is divided into just six major regions, ideal for detecting broad inter-regional patterns. From there, users can go down to the municipality level to capture localized dynamics, and even further to the parish scale for hyper-local analysis. For a visual example, Figure D.1 shows all the municipalities in the northern major region on the left, and all the parishes in the municipality of *Guimarães* on the right.

This tiered zoning structure allows the evaluation model to balance the statistical robustness of larger aggregates with the detailed insights that arise when comparing neighboring parishes (thus ensuring that trends are both reliable and context-sensitive).

3.2.1.2 Variable Filters

SIR provides several metrics for analysis (for example, the number of properties sold, the average sale price per square meter and the average price per transaction) based on any combination of filters. Together, these indicators offer a comprehensive view of the local market, capturing not only transaction volume but also property size, value, and composition.

Given that the project's main goal is to forecast future price per square meter, this metric is the most relevant. Focusing the model on it allows the projection of unit-level price dynamics that reflect broader market trends. Although SIR also offers detailed distribution statistics (quartiles and percentiles), such breakdowns are often unreliable at the parish level due to limited transaction volume. As a result, while quartile data can support broader regional analysis, the evaluation engine prioritizes the more stable average price per square meter series to ensure consistency across all areas.

3.2.1.3 Time Frame Filters

For the temporal dimension of the tool, SIR's users will have the flexibility to examine property valuations on a quarterly, half-yearly or annual basis, spanning from the first quarter of 2007 (the earliest period for which comprehensive transaction data is available) through the first quarter of 2025, the most recent quarter at the time this paper was written. This ensures that the model captures key inflection points in Portugal's property market, including the pre-crisis boom, the global financial and debt crises, subsequent recovery phases and the very latest market developments. By allowing selection of any time span within this interval, users can perform precise comparative analyses and observe how valuation trends evolve over time.

3.2.1.4 Typology Filters

The next filter provided by the SIR concerns the typology of the property, allowing users to restrict their search according to the number of bedrooms and the type of dwelling. Currently, the individual typology options include apartments with one bedroom or less (T1 or lower), apartments with two bedrooms (T2), apartments with three bedrooms (T3) and apartments with four bedrooms or more (T4 or higher). In the case of detached houses, the tool distinguishes between houses with up to three bedrooms (T3 or less) and houses with four bedrooms or more (T4 or more).

In addition to these detailed categories, aggregate filters are available for all apartments, all detached houses and the total of all property types. While totals by dwelling type can offer useful insights into supply or market share, the overall total often adds little analytical value, as it merges distinct property segments and may obscure relevant differences.

3.2.1.5 State of the Property

The SIR platform classifies properties as "new" if they have never been occupied, identifying first-sale transactions directly from developers rather than relying on construction dates—an approach required due to the zone-level aggregation explained in 3.2.1.1. This designation typically implies intact finishes, compliance with current regulations, and active guarantees. "Used" properties, by contrast, refer to all resales and reflect wear, renovations, or layout changes from previous occupants.

While the "Total" option aggregates new and used units, relying on this combined filter risks masking differences in price premiums and risk profiles between first-occupancy and resale properties, thereby reducing the granularity and accuracy of price per square meter estimates.

3.2.1.6 Data Integration

The SIR platform does not have APIs, so automated extraction integrated into a model has to be done using web scraping techniques. The output of each SIR search is made available in Excel, so web scraping will need to operate in order to get the downloads needed to gather the relevant data. The files will then be processed (as will be explained in the next chapter) so that they can become valuable inputs for the project.

3.2.2 BPstat for Macroeconomic Data

BPstat, the BP's online statistical portal, offers free and open access to a vast repository of over 200 000 time series covering the Portuguese and Euro Area economies. Users can retrieve both extensive historical data (spanning from the mid-19th century) to the present and official annual forecasts for key macroeconomic variables for the next two years. Access is available via an intuitive web interface as well as a programmatic API, ensuring seamless integration into both academic research and professional databases.

3.2.2.1 Historical Data

With regard to the indicators recommended in the literature, BPstat provides time series with multiple periodicities (monthly, quarterly and annual) and in various formats (year-on-year growth rates, current value series or chained volume indices), depending on the indicator. This flexibility allows the researcher to select the analytical framework best suited to each objective. Each series goes back several decades, guaranteeing the historical depth required for long-term analysis, and can be exported directly as CSV or Excel files via the web interface for any manual processing required (example on Figure E.1)

In addition to the interactive interface, the BPstat API provides programmatic access to all series and their metadata without the need for authentication, and the data can be retrieved very easily. This automation speeds up the assembly of a macroeconomic database and guarantees the reproducibility of the data collection process, thus facilitating integration into econometric models or information systems. The specific choices of format and frequency for each variable will be discussed and justified in Chapter 4, where the methodological criteria underlying these decisions are presented.

3.2.2.2 Economic Forecasts

Regarding economic forecasts, BP also has a portal dedicated to forecasts for Portugal and the Euro Area, where official economic projections for the next two years are published, as well as the rest of the outlook for the current year and the final result for the previous year (example in Figure E.2).

These projections cover many variables recommended by the literature (GDP, inflation, unemployment rate, GFCF, exports, imports, private consumption and public consumption) and are expressed on an annual basis, without the breakdown into quarterly or monthly figures that characterizes historical time series. The forecasting portal is updated at predefined intervals, typically coinciding with BP's regular statistical releases in the first quarter of each year. Since no API endpoint is provided for these forward-looking series, automatic retrieval requires web-scraping techniques, which nevertheless allow the official biennial projections to be seamlessly incorporated into econometric tables or database systems.

3.3 Data Handling

To feed a reliable forecasting engine, it is essential to assemble a coherent, well-structured database that brings together transaction-level real estate prices and macroeconomic indicators, so that both sets of information can be jointly explored in later stages of the analysis.

3.3.1 Variable Selection

The main variable of interest from the SIR platform is the price per square meter, as it directly aligns with the forecasting target. To capture broader demand and financing conditions, it is essential to enrich the model with macroeconomic context from BPstat. While the literature helps identify relevant indicators, the final selection must also consider data availability, update frequency, and conceptual alignment. Given that BPstat contains over 200,000 time series, it is crucial to narrow the focus to a small set of high-value indicators. This targeted approach improves computational efficiency and minimizes overfitting by emphasizing variables with strong theoretical and empirical justification.

3.3.2 Granularity Alignment

Spatial and temporal granularity are key considerations when integrating SIR and BPstat data. SIR provides historical price per square meter at the parish level with quarterly aggregation, setting a clear baseline for spatial and temporal resolution. In contrast, most BPstat indicators are reported only at the national level, with varying frequencies (monthly, quarterly, or annual). Reconciling these mismatches requires data transformations, such as aggregating monthly series to quarters, interpolating or replicating annual data across quarters, and clearly documenting each estimate's origin. These operations inevitably entail some granularity loss, as national values are attributed to local units and high-frequency data is averaged.

Data availability timelines also differ. SIR typically publishes each quarter's data in the following month, enabling a consistent one-month lag for registration. BPstat, however, follows no uniform release schedule, as publication delays vary by variable, ranging from weeks to several months. Managing these asynchronies requires aligning data based on actual availability dates and addressing any resulting gaps.

To ensure coherence, it is essential to define transformation strategies that harmonize property prices with macroeconomic indicators at a common resolution. This must preserve as much detail as possible while ensuring consistency across the integrated dataset.

3.3.3 Scope Decisions

Based on an in-depth analysis of data heterogeneity presented in the previous subsections, the core decisions for the forecasting tool's development can be defined:

- **Spatial Granularity:** Decision on whether the tool should be made at the level of the parish, municipality or region, also remembering that a finer geographical resolution improves local relevance, but increases data dispersion and the computational burden;
- **Temporal Resolution:** Definition of the temporal aggregation that the model will adopt (monthly, quarterly or annually), both for collecting inputs and for presenting outputs;
- **Macroeconomic Indicators:** The selection of the set of indicators that will enter the model, which must be a much reduced part of the series offered by BPstat, combining theoretical importance with stable and frequent publication;
- **Extension of the Forecast Horizon:** It is necessary to define the time window that should be explored, as well as the way in which it is divided (the latter particularly depends on the time resolution decision).

These decisions are crucial because they establish the project's starting point. Each one has implications for the database structure, model complexity and evaluation protocols. The detailed rationale and methodology underlying each of these choices, as well as any others not yet discussed, will be documented in Chapter 4, ensuring that every parameter reflects a balance between analytical rigor and practical feasibility.

3.4 ML Pipeline Development

Once the above mentioned key parameters have been established, a systematic evaluation of various ML algorithms is required to determine which approach best captures the patterns present in the data and produces the most accurate predictions. Based on the literature review, a list of strong candidates will be compiled, including Neural Networks, Linear Regression, Random Forest and two boosting methods (Gradient Boosting and XGBoost).

3.4.1 Model Requirements

3.4.1.1 Database Structure

According to the CRISP-DM framework, once business objectives are defined and the data is explored, the process advances to the preparation phase. Here, raw data must be transformed into a structured format suitable for modeling, ensuring a stable and consistent foundation for the algorithms.

Key tasks such as data cleaning, handling missing values, and feature engineering should be centralized within a standardized preprocessing pipeline. This ensures uniformity across model inputs, improves reproducibility, and reduces maintenance over time. The resulting dataset must serve as the single source of truth for training, tuning, and evaluation.

At this stage, the data must fully meet the needs of the forecasting model. Each entry should represent an observation of average property price per square meter, paired with clearly defined attributes. No field may remain empty, so the missing values must be removed or imputed using a defined strategy. Categorical variables like typology, condition, or region must be encoded into numerical formats appropriate for predictive modeling.

The final dataset should be a clean, fully numerical matrix, where each row corresponds to one observation and each column to a distinct feature. Only with this foundation can training, validation, and model comparison be conducted reliably.

3.4.1.2 Specific Model Requests

Each algorithm imposes specific requirements on variable format and distribution.

Linear Regression assumes a linear and additive relationship between predictors and target, performing best with scaled and uncorrelated features. Its accuracy drops when facing nonlinear patterns, which are common in real estate data.

Tree-based models like Random Forest, Gradient Boosting and XGBoost are more flexible. They tolerate mixed distributions and do not require feature scaling, but still rely on consistent imputation of missing values and proper encoding of categorical variables. Gradient Boosting benefits from clean data, and while scaling is rarely essential, it can improve training stability. XGBoost adds native support for sparse matrices and handles missing values automatically, simplifying preprocessing. However, high-cardinality categorical variables still require efficient encoding. Hyperparameters such as learning rate, number of estimators or tree depth must be tuned to optimize performance.

Neural Networks are the most demanding, as inputs must be normalized or standardized, and categorical variables transformed into embeddings or one-hot vectors to ensure stable gradient propagation.

In all cases, model performance depends not only on preprocessing but also on careful hyperparameter tuning. Combinations of models, preprocessing strategies, and parameter sets must be tested to identify the best-performing pipeline.

3.4.2 Model Selection

To determine whether a new model iteration outperforms its predecessor, an evaluation metric must be chosen. This metric then serves as the common standard for comparing top configurations across different algorithms. The literature review recommends the R^2 as the primary comparison metric because it measures the proportion of variance in the observed data that is explained by the model. A value of R^2 close to the number 1 indicates that the model accounts for almost all of the variability in the target variable; a value close to 0 signals minimal explanatory power; and a value less than zero means that the model explains less than a simple average. By focusing on R^2, model selection emphasizes overall goodness of fit and ensures that comparisons remain both interpretable and independent of the scale of the data.

3.4.3 Model Interpretability

Interpretability is essential for transforming a complex prediction engine ("black box") into a transparent decision support tool. By revealing how input variables drive each forecast, interpretability builds trust among stakeholders, enables domain experts to validate model logic, and uncovers potential biases or unintended correlations. It also supports regulatory compliance, since financial and advisory applications increasingly demand explanations of algorithmic outputs.

As described in the literature review, *post hoc* methods can deliver these insights without degrading model performance. For instance, SHAP provides a principled way to decompose each prediction into contributions from individual features. Using SHAP allows clear identification of which of the selected features carry the greatest weight in driving the target variable forecasts, guiding both model refinement and practical decision making.

3.5 Power BI Dashboard

Consultants require rapid access to forecast results to support their analyses, so forecasts generated by the tool must be delivered through an interactive Power BI dashboard. The dashboard should offer filters for relevant dimensions such as geographic area, forecast horizon, property typology and property condition. Filters for economic indicators are not necessary because those variables apply uniformly across all scenarios and cannot be meaningfully subdivided.

Visual elements should include some features like time series charts of predicted price per square meter and summary tables of forecast values for the selected filters. Automatic data refresh must ensure that each quarterly update of transaction and economic data is reflected without delay. These interactive capabilities enable consultants to drill into local trends, compare multiple forecast horizons and export customized reports, all without any programming expertise.

Chapter 4

Methodological Approach

This chapter outlines the methodological approach adopted for building the forecasting tool. It details the scope definition, database assembly, data preparation, model training, and result visualization processes, ensuring a structured and reproducible development aligned with the project's objectives.

4.1 Scope Definition

4.1.1 Granularity Decisions

In order to capture spatial and temporal variations in property prices with as much detail as possible, the modeling was carried out using the highest feasible granularity: on a quarterly basis and at the parish level. This fine level of disaggregation not only improves the model's ability to reflect local market dynamics but also enables bottom-up analyses. It is intended to provide a flexible exploration of the data through customized filters, such as typology, state, year, quarter and parish, with the ability to aggregate results at the municipality level if necessary.

From a temporal perspective, quarterly granularity was selected as it offers the most detailed frequency available that remains both analytically robust and compatible with other relevant data sources. Most macroeconomic indicators are reported quarterly or exhibit greater reliability at that frequency, and the SIR real estate transaction data is also available on a quarterly basis, ensuring direct alignment between datasets. Moreover, quarterly analysis provides an effective balance between capturing temporal dynamics and reducing short-term volatility, supporting the generation of more stable and interpretable forecasts.

Geographically, the selection of the parish level reflects the intention to capture local market dynamics. Property characteristics and prices can vary substantially from one parish to another, even within the same municipality, so a model at the most disaggregated level captures these local heterogeneities. Importantly, this granularity enables the bottom-up modeling approach, where detailed forecasts at the parish level can be aggregated to the whole municipality if needed. The reverse process (disaggregating higher-level forecasts) would require additional assumptions and would be less reliable. Therefore, starting from the most granular level was considered not only more robust for the study objectives but also more aligned with the goal of providing flexible, user-driven analysis tools.

Although working at such fine granularity implies dealing with potential data scarcity in less active parishes and increased computational complexity, the benefits in terms of predictive accuracy, usefulness, and analytical flexibility (due to greater local specificity and bottom-up structure) clearly outweigh these difficulties.

4.1.2 Variable Selection

4.1.2.1 SIR

As presented in the previous chapter, the SIR will be used as the source of real estate transaction data. The extracted and updated data will include the region, municipality, parish, typology, condition, price per square meter, and the date of the information. Each row is interpreted, for example, as follows: "a new T2 apartment in the parish of Caldelas, located in the municipality of Guimarães and the Northern region, had an average price of 1,500 euros per square meter in the third quarter of 2022".

4.1.2.2 BPstat Data

BPstat was therefore used as the source of explanatory macroeconomic data for the model. The selection of variables is closely related to the topics discussed in Subsection 2.4 of this thesis and to the management of challenges described throughout Chapter 3.

GDP and consumption reflect the level of economic activity, inflation represents purchasing power and real interest rates, GFCF captures investment (including new construction), imports and exports reflect openness to international markets, and unemployment indicates labor market conditions. As previously noted, studies suggest a relationship between these indicators and real estate prices.

Accordingly, the year-on-year variation rate was chosen for the following variables: GDP, inflation, private consumption, public consumption, imports, exports, and GFCF. In addition, the national unemployment rate, expressed as an absolute percentage, was also included.

The inflation rate (year-on-year variation of the Consumer Price Index) is usually reported monthly, so it was necessary to align it with the model's quarterly granularity. The chosen approach was to convert monthly inflation into a quarterly figure by taking the simple arithmetic mean of the monthly values within each quarter. In other words, quarterly inflation was calculated as the average of the three corresponding monthly year-on-year inflation rates (for example, the inflation for the first quarter of a given year is the average of the inflation rates for january, february and march of that year). This method provides a representative average value of inflation over the quarter, smoothing out one-off monthly fluctuations and aligning the timing with the other indicators that are already on a quarterly basis (Stock and Watson, 2008).

Alternatively, it was possible to consider using the inflation rate for the last month of each quarter or calculating the change in the price index for the last month of the quarter compared to the same month of the previous year. However, the quarterly average was considered more robust and more informative of the overall trend in consumer prices during the period, avoiding the risk of overestimating a single potentially atypical month. This ensured that the "quarterly inflation" variable reflected the average behavior of prices during the quarter, consistent with the interpretation of the other macroeconomic variables included.

4.1.3 Future Data Incorporation

In order to forecast prices for future periods, at a certain point it is necessary to obtain some information for those periods. Since it is not possible to access exact future data, the best approximation comes from official projections, also available through BPstat.

All the economic variables selected above have official forecasts for the next two years (and for the rest of the current year), provided on an annual basis. In order to incorporate these annual projections into the quarterly model, the annual rates of change were distributed evenly over the four quarters of the corresponding year. In practice, it was assumed that the projected year-on-year change for each year would also occur in each of its four quarters. For example, if the GDP projection indicated growth of +4% in year $t+1$ compared to year t, then each quarter of $t+1$ (first, second, third and fourth) was modeled with a year-on-year growth rate of +4% compared to the same quarter of the previous year.

This assumption of a flat intra-annual profile simplifies the incorporation of projections and ensures consistency between the annual aggregate and the underlying quarterly figures (the average of the quarterly changes will correspond to the projected annual change). Naturally, this approach ignores potential seasonal patterns or intra-annual differences, but in the absence of more detailed quarterly data in the projections, it was considered a reasonable compromise. The uniform distribution of annual forecasts across quarters is a common practice in macroeconomic projection scenarios when only annual figures are available, as it provides estimated quarterly series that preserve the total or annual average in line with the official forecast, without introducing an arbitrary bias in any specific quarter (Chen and Andrews, 2008).

4.2 Database Construction and Updating

The database used in this study originated from a manual extraction process and was later maintained through automated updates via a Python script.

The initial manual extraction from SIR compiled all available quarterly series up to 2007, which goes far beyond the time horizon actually used for forecasting. However, at the company's request, this extended range was included so the database could also serve as a data repository. The corresponding time series were then added to these records, meaning each row of the dataset can be interpreted as follows: "In the second quarter of 2024, a T4 or larger house located in the parish of Alcabideche, which belongs to the municipality of Cascais and the Lisbon Metropolitan Area region, had an average price per square meter of 4,022€; in that same quarter, the unemployment rate was 6.1%, the year-on-year GDP growth rate was 1.5%, and so on".

In this way, transactional and macroeconomic data are combined per quarter in a single table. A few rows and columns from the table, including the example just described, can be seen in Figure J. For instance, inflation is labeled as *ihpc*, GDP as *pib*, the unemployment rate as *tx_desemp*, GFCF as *fbcf*, public and private consumption as *cons_pub* and *cons_priv*, and exports and imports as *export* and *import*, respectively.

With this initial database consolidated, the focus shifted to developing a Python script capable of extracting, verifying, and adding new transactional and macroeconomic data whenever available, in order to ensure the long-term continuity of the model. This approach eliminates the dependency on periodic manual extractions, which are time-consuming and prone to errors.

4.2.1 SIR Web Scraping

As previously noted, the collection of new data comes from two different platforms, each offering distinct extraction solutions. In the case of SIR, which does not provide APIs, a web scraping python script was developed. This script is capable of logging into the website, navigating to the desired page, and selecting the most recent quarterly data available for extraction. The extracted data is then filtered by removing variables that are irrelevant to the model, and the date is checked to ensure that no duplicate entries are added to the database.

To implement this, Python libraries such as BeautifulSoup and Selenium were used for parsing the content and locating relevant elements. More detailed information about the extraction code can be found in Appendix F.

4.2.2 BPstat Extraction

As for the automatic updating of BPstat data, the presence of APIs makes the process significantly easier and faster. With series identifiers for each desired variable, a Python script was developed to send requests to the APIs and retrieve the required data, also described in more detail through Appendix F. Note that in the case of inflation, since it is aggregated monthly, the three most recent available values. If the month corresponding to the latest value is march, june, september, or december (which mark the end of the first, second, third, and fourth quarters), then the average of the last three values is considered. Otherwise, the information is not deemed valid for insertion. For the remaining variables, since they are all aggregated quarterly, only the latest available value is used. The values for each variable are then allocated to the database, with correct assignment by year and quarter using the already updated *data* (date) column as a reference.

Since release dates for new data vary across indicators, it is possible that some recent entries will contain missing values for certain variables. These cases are handled with special treatment, as will be explained in more detail later.

4.3 Data Cleaning and Feature Engineering

The dataset, which can now be updated periodically, will always have extra information, as it is essentially an information repository. Thus, with each forecast, cleaning, selection and transformation processes are carried out in order to prepare the data in the best way to be inserted into the model. A more direct explanation of the steps involved in the data preparation python script can be found in Appendix G (as well as the modeling and forecasting part, since it refers to the entire pipeline), with actual python functions used. Also, a portion of the result of the whole cleaning and feature engineering operation can be seen in Appendix H, where the relevant variables and some example lines are shown, already in the correct format.

4.3.1 Exploratory Data Analysis

A comprehensive EDA was conducted to better understand the behavior of the data and the structure of the variables that feed the forecasting model. This step is important to notice some patterns or inconsistencies in the data, which leads to better preprocessing and modeling decisions. It will also be possible to assess whether the trends in the final model follow the same patterns and remain consistent.

4.3.1.1 Notable Trends and Observations

Figure I.1 presents the average price per square meter over time. Following a decline between 2007 and 2014 (likely driven by the aftermath of the financial crisis), prices began to rise steadily from 2015, an upward trend that became particularly pronounced after 2020 and continues up to the date of this study. With this analysis, it can be seen that time positioning plays an important role in price.

Figure I.2 shows a scatter plot of price per square meter across typologies, further segmented by the property's condition. Several patterns emerge, as new properties consistently command higher prices across all typologies, and price dispersion within each category is substantial. This reflects the intrinsic heterogeneity of the real estate market and indicates that the typology and status variables may be among the most important for the model to consider.

Subsequently, Figure I.3 shows how the average price per square meter was distributed among the municipalities of the Lisbon Metropolitan Area in the first quarter of 2025. This reduced example shows the differences in prices between them, reflecting the importance of geography in the definition of prices.

Finally, with regard to the economic variables, only the unemployment rate was analyzed, since it is the only one that is not represented as a year-on-year rate of change. Figure I.4 shows an inverse trend in which the price rises as the unemployment rate falls, and vice versa (from 2015 to 2025). Accordingly, the same behavior is expected in the forecasts.

4.3.1.2 Outlier Treatment

Although several price observations may appear extreme on the EDA charts (Appendix I), none stand out far enough from the overall distribution to justify removal based solely on statistical thresholds. Instead, the notion of an “outlier” in real estate must be contextualized. Consequently, potential outliers were evaluated not in isolation, but in conjunction with other features such as location, typology, and condition. This multidimensional interpretation revealed that high prices were often legitimate in specific combinations (for example, new properties in high-demand urban areas). Moreover, filtering based on price alone risks removing meaningful edge cases and reducing the model’s ability to learn from the full scope of market variation.

Another factor influencing the decision not to apply global outlier removal is that the distribution of values changes over time, as the database is constantly updated. What is considered extreme in earlier years may become typical in more recent periods. Outlier detection must therefore be relative and context-aware.

In short, no systematic outlier trimming was applied. Instead, the modeling approach was chosen to be robust against skewed distributions and capable of adapting to high-variance data, ensuring that no relevant information is discarded and that the model reflects the true behavior of the market.

4.3.2 Redundant Rows

The first cleanup concerns redundant information that doesn’t add value to the model. This applies, for instance, to rows representing the overall totals for all typologies, the total count of property states, as well as rows that aggregate data at the municipality-cluster level. As discussed in the previous chapter, both the total across typologies and the total across property states are considered redundant, and including them may introduce noise or confusion into the model’s learning process.

However, the dataset does retain the total number of houses and the total number of apartments across their respective typologies. These aggregates are treated as distinct typologies in their own right and are included in the final analyses. To reduce redundancy while preserving their analytical usefulness, a new boolean column (*typ_boolean*) is added to identify whether each row refers to a house or an apartment (with 0 representing apartment and 1 representing house). This approach avoids inflating the dataset with multiple similar typologies while still allowing aggregated housing trends to be captured in a controlled and interpretable manner.

Regarding municipality clusters, the primary focus of the analysis is at the parish level. The tool is designed to support detailed insights at this finer geographical resolution, and any municipality interpretation can be derived from the combined data of the constituent parishes. It’s important to note that the SIR system only publishes data when a minimum number of transactions is reached to ensure statistical reliability. Therefore, there are cases where municipality-level data is available even though no parish-level data is shown. This happens when the municipality meets the transaction threshold, but none of its parishes do individually.

As the goal is to prioritize data that is granular, interpretable, and based on concrete observations, all rows corresponding to aggregated totals (total typologies, total property states, and municipality-level clusters) are excluded. This ensures that the final dataset is anchored at the parish level, where the most detailed and actionable insights can be extracted.

4.3.3 Useless Columns

Within the initial dataset (shown in Appendix J), two additional unnecessary columns are also removed in “agregacao” (aggregation) and “var” (variable). These columns contained only the repeated values “Trimestral” (quarterly) and “Preço de Venda / m2” (sale price per square meter), respectively, across all rows. They had been included as checks when inserting new data into the dataset to ensure data integrity; since they no longer add value to the analysis, they are removed at this point.

4.3.4 Seasonality Detection and Organization

The next step, though simple, proved highly useful. It consisted in splitting the *data* (date) column into the separate features *ano* (year) and *trimestre* (quarter). This separation improves data readability and allows the model to capture seasonal trends, since it includes a dedicated column identifying the numerical quarter of each observation. Although this step may seem trivial, it ensures better temporal organization of the data, which is essential for the consistent application of time series modeling techniques.

4.3.5 Historical Timeframe Definitions

4.3.5.1 Time Span Selection

In terms of selecting the historical horizon used for training, it was defined that the model would use the last 12 years of available data (48 historical quarters) to learn the patterns of relationship between macroeconomic variables and real estate prices. This decision was based on both practical and theoretical considerations.

On the one hand, a 12-year time span offers a sufficiently broad historical window to train ML algorithms in a robust and stable manner, capturing key macroeconomic fluctuations such as recessions, recoveries, and extraordinary events like the COVID-19 pandemic. The decision to extend the horizon to 12 years, rather than using a shorter period, was driven by the need to ensure a meaningful volume of data, especially considering that the SIR provides average values aggregated over transactions (rather than individual-level records). As a result, the number of available observations is significantly lower than the actual number of transactions, making it necessary to expand the time frame in order to achieve statistical reliability.

On the other hand, this 12-year limit still avoids incorporating excessively outdated data, which may no longer be representative due to structural shifts in the housing market, macroeconomic environment, or the physical and demographic evolution of the parishes themselves. For example, price relationships and patterns that existed in the early 2000s, particularly around the time of the 2007–2008 financial crisis, may no longer hold true today. Thus, the 12-year horizon was chosen as a balanced compromise between ensuring enough data density and maintaining contextual relevance for current market conditions.

4.3.5.2 Temporal Weighting

To emphasize recent data while still incorporating historical information, a grouped exponential time-weighting scheme was applied, inspired by the group seasonal indices of Ouwehand et al. (2007).

First of all, a new column named *tempo_continuo* (continuous time) was added to the dataset, which encodes each observation's position in time as a sequential integer. This value starts at 0 for the earliest quarter in the time horizon and increases by 1 for each subsequent quarter, resulting in a range from 0 to 47 across the 48 quarters considered in the model.

Then, using this continuous time index, the temporal distance from the most recent quarter in the dataset was computed. Observations were then grouped into blocks of four quarters, so that each group represents a full year of data. The most recent group receives full importance, with a weight of 100%. Older groups are down-weighted using an exponential decay function, as described in equation 4.1:

$$w = e^{-\lambda\Delta}, \lambda > 0 \tag{4.1}$$

where Δ is the group number, corresponding to the complete years separating the observation from the most recent quarter, and λ the coefficient that quantifies weight loss.

This approach ensures that each of the four most recent quarters have observations that contribute with maximum relevance. For example, if the latest available data point corresponds to the first quarter of 2025, then 2025 Q1, as well as 2024 Q4, Q3, and Q2, will each have a weight of 100%. The first reduction in weight occurs with 2024 Q1, the second with 2023 Q1, and continues to decay for earlier groups. It was decided internally to give a weight of around 20% to the oldest observations, so the λ value is 0.15 in order to give 100% weight to the four most recent quarters, falling exponentially to approximately 20% in the four oldest quarters. In doing so, the model is guided to focus on recent seasonal patterns while still retaining the structure and influence of older observations over a longer horizon.

In addition to enabling the assignment of time-based weights, the *tempo_continuo* variable also serves as a more granular temporal indicator for modeling purposes. While macroeconomic variables provide useful context, they may fail to capture certain shifts, such as sharp increases or declines in price levels over time, which follow internal temporal trends rather than external economic drivers. By introducing a continuous representation of time, the model is given an additional mechanism to detect and incorporate such endogenous temporal dynamics more effectively.

The *ano* (year) column is eliminated because the combination of the time weighting scheme and the *tempo_continuo* (continuous time) variable already captures the positioning and relative importance of each observation over time, making a separate year indicator redundant. Unlike *ano*, which repeats the same value for four consecutive quarters and therefore contributes limited new information, *tempo_continuo* provides a unique and sequential identifier for each quarter, improving the model's understanding of temporal progression. The inclusion of both variables could introduce collinearity or dilute the effect of more informative features. The *trimestre* (quarter) column, however, is retained to capture possible seasonality, allowing the model to identify and learn recurring patterns specific to each quarter of the year.

4.3.6 Year-on-year Rate Adjustments

One of the major transformations performed involved creating cumulative variables from the chained year-on-year trajectories of the macroeconomic indicators. As previously noted, among the selected variables, only the unemployment rate is expressed as an absolute percentage; the others are growth rates.

It was deemed beneficial to also have a representation of the level or accumulated evolution of these indicators over time as model inputs. To achieve this, an accumulated index was constructed for each indicator by chaining the year-on-year variations sequentially. In practical terms, a base value of one unit was defined for each indicator's index in the training dataset; from that point onward, for each quarter, the index value is multiplied by the growth factor corresponding to that quarter's year-on-year rate.

For example, suppose the model is to be used in the first quarter of 2025, with the time window starting in the first quarter of 2013. Using GDP as an example, there already exists a column for the year-on-year growth rate, *pib*, and a new feature *pib_h* is created to store its trajectory. If in the first quarter of 2013 the year-on-year GDP rate was +2%, then *pib_h* for that quarter is $1 \times 1.02 = 1.02$. Later, if in the first quarter of 2014 the *pib* column shows +3%, the *pib_h* feature becomes $1.02 \times 1.03 = 1.0506$, and it continues in this manner.

Recall that a year-on-year rate applies to a specific quarter, so each macroeconomic variable will have four separate trajectories within its "*_h*" feature, one for each quarter. What happens in this GDP example also applies to the other variables expressed as year-on-year rates, namely inflation, public consumption, private consumption, imports, exports, and GFCF (all suffixed with "*_h*" after the original variable name), with only the unemployment rate remaining unchanged. Appendix H already shows the succession of the new features created (and some example trajectories), and also the assignment of weights described earlier (labeled *peso_temporal*), as well as the new *tempo_continuo* feature and the house-apartment boolean (*typ_boolean).*

These cumulative index variables are useful because they convey, in a comparative manner, the relative level of each macroeconomic variable in each period (versus a base period), rather than only the percentage change. In the model context, this means the algorithm can understand not only whether the variable is increasing or decreasing in a given quarter, but also at what level it stands relative to the past (for example, whether it has reached a maximum over the years considered). This accumulated context can improve predictive performance in scenarios where the absolute level of a variable influences real estate prices beyond its point-in-time growth rate.

4.3.7 Encoding of Categorical Data

ML models ultimately operate on numeric values, but not all information needs to be converted into pure calculation numbers. Variables originally in text format are assigned integer codes while retaining the category type. This approach ensures that the algorithm treats these fields as class identifiers rather than continuous quantities with ordinal relationships. By using the category type, the model interprets each code as a distinct label, without assuming that one code is inherently greater or better than another.

In this context, the *tipologia* (typology) and *estado* (state) columns are always converted to category. For example, in the typology variable, the option "Apt. T1 ou inf" (Apartment T1 or lower) is mapped to 0, "Apt. T2" (Apartment T2) is mapped to 1, "Apt. T3" to 2, and so on. As a result, every occurrence of "Apt. T3" receives code 2, but the model will not perform arithmetic operations on these codes, each integer serves solely as a label. A CSV file is generated to record the mappings for future reference in the dashboard.

4.3.8 Feature Selection

4.3.8.1 Macroeconomic Variables

After constructing the cumulative index variables mentioned in the subsection 4.3.4, keeping the original rate-based variables often becomes redundant and can introduce multicollinearity into the model. It was decided to eliminate the originals from the models afterwards, simplifying the data set, reducing noise and ensuring that each predictor makes a unique contribution of information. In practice, this simplifies the calculation, improves interpretability and prevents the model from giving too much importance to closely related inputs. Consequently, removing the original variables once their derived counterparts are in place helps maintain a simple and efficient feature set that focuses on the most informative representations of the underlying data.

Thus, of the macroeconomic indicators, the model will only receive *tx_desemp* without modification, and then all the transformations applied to the others: *pib_h*, *ihpc_h*, *fbcf_h*, *cons_priv_h*, *cons_pub_h*, *export_h* and *import_h*.

4.3.8.2 Geographical Data

In addition to the municipality totals mentioned and removed, SIR contains other redundant geographical fields. As can be seen in Appendix J, *regiao* (region), *concelho* (municipality) and *freguesia* (parish) appear twice, once as a name and once as an ID. Since ML models only read numerical values, it was decided that the repetition would be eliminated by deleting the text versions of these variables. Thus, the geographical components are only represented by their ID.

Between the remaining IDs, *regiao_ID* is dropped because it is overly broad and does not contribute precise local context. By contrast, *concelho_ID* is retained to capture nearby information from other parishes from the same municipality, whenever some parish-level data is sparse or unavailable. Consequently, each record will include only *freguesia_ID* and *concelho_ID*, ensuring a lean feature set that maintains geographic relevance without redundancy. The initial combinations of name and ID are saved so that it can be possible to search for the information on the dashboard by the name of the municipality and parish.

4.3.8.3 Final Selection

Taking all this into account, the final training dataset includes the municipality ID, parish ID, typology and respective boolean, state, continuous time, quarter, unemployment rate, and cumulative variables for inflation, GDP, GFCF, private consumption, public consumption, imports, and exports. Finally, it logically includes the target variable of the study, which is the average sale price per square meter.

The temporal weights of the observations, although shown in Appendix H, are dropped from the file before training. They are previously extracted to be passed to the model via a *sample_weight* parameter, adjusting the contribution of each training instance, allowing the model to prioritize more recent observations during training. A summary of the final selection can be seen in table 4.1, with each variable, its category and a brief description.

Table 4.1: Description of the variables used in the dataset

Variable	**Category**	**Description**
concelho_id	Spacial identifier	Municipality of the observation
freguesia_id	Spacial identifier	Parish of the observation
tipologia	Property Characteristics	Typology of the dwellings in the observation
typ_boolean	Property Characteristics	Apartment/House indication
estado	Property Characteristics	State of the dwellings in the observation
tempo_continuo	Temporal	Temporal evolution
trimestre	Temporal	Quarter of the observation
tx_desemp	Macroeconomic	Unemployment rate on the observation date
ihpc_h	Cumulative Macroeconomic	Inflation on the observation date
pib_h	Cumulative Macroeconomic	GDP on the observation date
fbcf_h	Cumulative Macroeconomic	GFCF on the observation date
cons_priv_h	Cumulative Macroeconomic	Private consumption on the observation date
cons_pub_h	Cumulative Macroeconomic	Public consumption on the observation date
import_h	Cumulative Macroeconomic	Imports on the observation date
export_h	Cumulative Macroeconomic	Exports on the observation date
preco_m2	Target	Average price (€/m^2) of the observation

4.3.9 Addressing Missing Values

Due to the nature of the database created and the automatic updating processes, there is only one scenario in which missing values can occur. This is exclusively the insertion of macroeconomic data from BPstat, which does not follow fixed update schedules for all the variables. As a result, during a given database update, some macroeconomic indicators may already contain values for the most recent quarter of SIR transactions, while others may not.

Since this is a time series model, it is essential to preserve the chronological order of the information. Therefore, the FFill method was adopted to handle missing values. That is, any missing data (always occurring in the most recent periods) will be imputed using the most recent available value. In the case of the unemployment rate, which is an absolute percentage measured quarterly, any missing value will be filled using the unemployment rate from the previous quarter.

For cumulative variables, however, which are constructed as independent trajectories per quarter, the FFill approach is adapted accordingly. In these cases, missing values are filled using the value from the same quarter of the previous year, in order to respect the distinct seasonal paths and year-over-year dynamics of each trajectory.

4.4 Machine Learning Models

Here, it begins the model section, with final preparations, hyperparameter tuning, model validation and evaluation, interpretability and finally forecasts. It should be noted that a more direct explanation of this overflow together with the processes in Subchapter 4.3 can be found in Appendix G, containing actual functions used in python.

4.4.1 Final Data Preparation

The dataset is first split into a feature matrix containing transformed macroeconomic indicators, parish identifiers and the other explanatory variables, and a target vector only with the price-per-square-meter values.

Linear Regression and Neural Networks are sensitive to feature scale and therefore require normalization, whereas tree-based methods such as XGBoost, Gradient Boosting and Random Forests are inherently scale-invariant and do not. Consequently, before fitting Linear Regression and Neural Network models, *z-score* normalization is applied. Lastly, it is also important to note that Neural Network models do not support the existence of a sample weight, so the weight is not assigned when dealt with in this project.

4.4.2 Hyperparameter Tuning

Each of the five algorithms is wrapped in Scikit-Learn's *RandomizedSearchCV*. The search spaces for all hyperparameters follow the configurations recommended on the official Scikit-Learn website. A fixed number of random combinations are sampled and evaluated using the same time-aware cross-validation folds described below, and the parameter set that maximizes the chosen validation metric is selected. This procedure guarantees that every model benefits from systematic, documented tuning under identical evaluation conditions, thereby yielding better performance.

4.4.3 Model Validation

Given the inherently sequential structure of quarterly property-price data, where observations are interdependent, subject to evolving trends, seasonality and occasional regime shifts, a validation strategy that respects chronology is essential. Scikit-Learn's *TimeSeriesSplit* is leveraged (with ten expanding-window folds) so that each train–test split mirrors the real-world forecasting process. The model is always trained on all data up to a given fold (applying data normalization when needed and temporal weights when supported) and then tested on the immediately following period (following fold). By advancing the cutoff point one fold at a time, this rolling-origin scheme prevents any future information from contaminating model fitting, exposes the algorithm to a variety of market conditions over the full history, and highlights its stability as the forecast horizon shifts further away from the training window. For this to happen reliably and without errors, the dataset is sorted in ascending order of the date of the information.

4.4.4 Evaluation Methods

Once all ten folds have been evaluated, the function aggregates the out-of-sample results by taking the arithmetic mean of the per-fold R², MAE and RMSE values, yielding a single summary metric for each model. A high average R² demonstrates consistent explanatory power across different market regimes; a low average MAE indicates that the model's typical prediction error (in euros per square meter) remains small and interpretable for stakeholders; and the average RMSE reflects the extent to which large deviations are penalized, offering insight into the model's robustness against occasional extreme forecasting errors. As recommended by the literature review, the model chosen will be the one with the highest R², keeping the MAE and RMSE for stakeholder analysis and interpretation of the results.

4.4.5 Feature Significance Analysis and Interpretability

Following the selection of the optimal forecasting model, a post-hoc SHAP analysis is conducted to quantify how each feature influences the predicted price per square meter. In accordance with the framework outlined earlier, global feature importance will be assessed by averaging absolute SHAP values across the dataset to identify the most impactful predictors, while local explanations will decompose individual forecasts into additive contributions, indicating whether and by how much each feature shifts the target relative to a baseline. To visualize these results, two complementary SHAP plots are used:

- **Summary plot:** Provides an overview of global feature importance by ranking variables according to the average magnitude of their SHAP values across all predictions.

- **Dependence plots:** Useful for a detailed view of how an individual feature affects model predictions at the observation level. Each point represents a single data instance, with the horizontal axis displaying the actual value of a given feature, and the vertical axis showing its corresponding SHAP value, indicating whether the feature raised or lowered the predicted price. Additionally, each point is colored based on the value of a second variable, allowing for the visualization of interaction effects. This second variable is chosen automatically by SHAP, and corresponds to the one that has the greatest interaction with the main variable.

These visual tools are particularly valuable in understanding the model's internal logic and supporting the interpretation of the results discussed in Subsection 5.3.

4.4.6 Predictions

4.4.6.1 Building the Forecast Dataset

Once the training stage and selection of the best-performing model have been completed, the focus turned to using this model to effectively project future prices, thus generating a complete predictive scenario for the eight quarters following the last historical data. To do this, it was necessary to put together a forecast dataset that included all the relevant combinations of location and type of property in the future periods and to fill in the respective explanatory variables with the predicted or assumed values for these horizons.

The automated process generates a new forecast database with each run of the code, always using the date of the last available quarter as a reference. The last observed quarter is identified by consulting the existing time series, then a skeleton dataset is built by combining each parish and its municipality with each typology and state pairing and assigning each of the following quarters. If the actual data extends to the second quarter of 2025, the skeleton will cover the third quarter of 2025 to the second quarter of 2027. This produces $P \times S \times T \times N$ rows, where P, S, T and N refer to the number of parishes, states, typologies and number of future quarters. This will result in a very dense volume of forecasts, which will have to be inserted into a Power BI dashboard for easier and more interactive navigation by the consultants.

All the model inputs, except the target price per square meter variable, are filled in automatically according to deterministic rules that extend the training framework. The *typ_boolean* binary signal receives 0 or 1 based on the typology category and the continuous time index *continuous_time* restarts at the next integer after the last training quarter (the training covers quarters 0 to 47, so the forecast starts at 48 and continues until 55).

As for the macroeconomic variables, it is necessary to assume BPstat's annual projections for each quarter of the year in question, as explained in 4.1.3. The unemployment rate *tx_desemp* comes as an absolute percentage, as in the training dataset, so it is just added without transformation. All the other macroeconomic variables are originally reported as annual growth rates, so they follow the cumulative trajectories carried forward from their last observed quarterly values. Thus, even with the same growth forecast for all quarters, there are still four different cumulative paths for each variable (one for each quarter), which follow the old training values, keeping the path consistent.

Official BPstat projections are extracted automatically on each run by a web scraping routine (fast enough to not incur any operational delay), returning a table of projected values by year and variable. Finally, these forecasts are merged into the skeleton dataset, taking into account the year to which each information relates. An overview of an example dataset to be used for forecasting can be seen in Appendix K (the year is re-inserted for easier navigation in the Power BI dashboard). Rerunning the code at any time recalibrates the forecast horizon and refreshes all input features without manual intervention, producing a dataset that matches exactly its execution date and the most recent public economic outlook.

4.4.6.2 Saving and Uploading Predictions

Finally, the best performing model is trained on the training dataset and generates predictions for price per square meter on the forecast dataset. Then, actual figures for the last two years are added to the forecast data. This way, consultants can compare their forecasts with recent figures to better base their decisions, allowing them to quickly compare recent real-world values with the model's results. If, in the future, a consultant wants to insert even more (or less) historical information, the code change is quite practical and straightforward, simply changing the *n_years* parameter that is passed to the *add_comparison_data* function (visible on the bottom of Figure G.1 in Appendix G). The final result is saved in a specific folder, along with all the supporting tables needed to create an interactive and dynamic Power BI interface. An additional link table has also been added to this folder to connect the districts to the municipalities, making it easier to navigate from top to bottom.

4.5 Power BI Visualization

With all the necessary information already organized into separate tables, a Power BI dashboard was created to provide consultants with a quick and intuitive overview of the data. The dashboard includes the following tables:

- **Predictions:** Retains the structure used in the predictive model and includes key fields for navigation such as *Year*, *Quarter*, *typology_id*, *state_id*, *parish_id*, and *municipality_id* (it also holds the other explanatory variables used in the model, though these are not directly relevant for dashboard filtering);

- **Districts:** Lists each district with its respective ID and name;

- **Municipalities:** Provides the ID and name of each municipality along with the corresponding district ID;

- **Parishes:** Lists parishes with their IDs, names, and associated municipality IDs;

- **State:** Includes the different property states with their IDs and labels;
- **Typology:** Includes property typologies with their respective IDs and descriptions;
- **Update Date:** Displays the date of the last data refresh, used solely for informational purposes rather than filtering;
- **Metrics:** Stores the performance indicators of the forecasting model, namely the RMSE and MAE values, allowing consultants to assess the model's accuracy when evaluating predictions.

As mentioned earlier, the forecast table includes forecasts for the next eight quarters, as well as historical data for the eight quarters prior to the first forecast period. This configuration allows consultants to quickly compare recent real-world values with the model's results.

The UML diagram in Figure 4.1 illustrates the relationships between the tables. One-to-many relationships are established from *Districts* to Municipalities, from Municipalities to *Parishes*, and from *Parishes* to the *Predictions* table, allowing for hierarchical geographic filtering. Additionally, the *Predictions* table links to *Typology* and *State* for further semantic filtering. Both the *Metrics* and *Update Date* tables remain disconnected from the relational schema, as they serve only display and reference purposes.

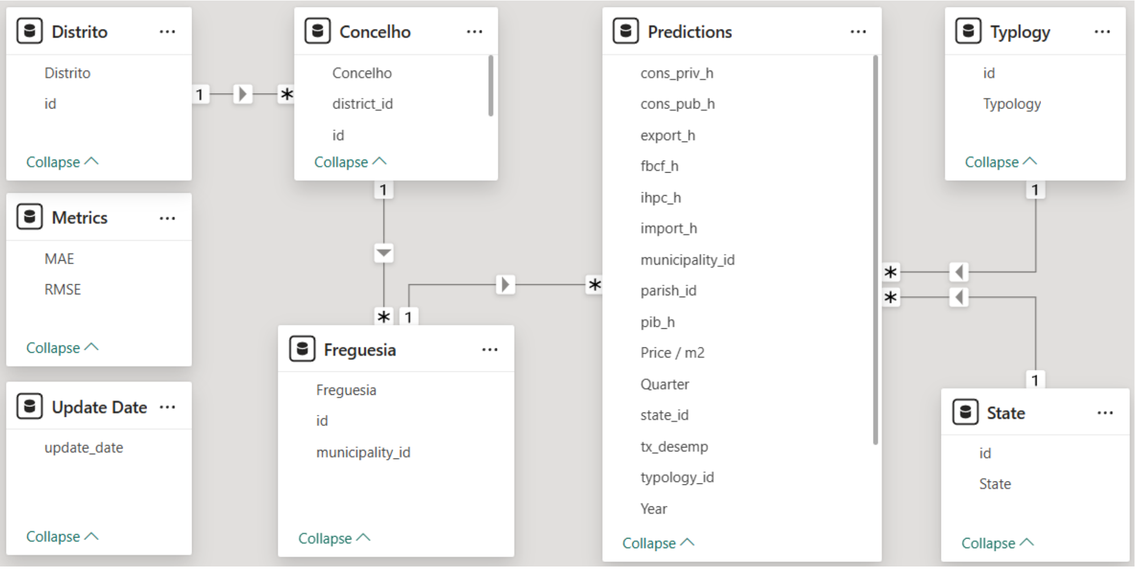


Figure 4.1: UML Diagram

With each run of the script, the tables are replaced in the folder used to upload the data. Thus, simply using Power BI's "refresh" option updates the content of the tables and refreshes the information on the dashboard to the latest version.

Chapter 5

Results

When analyzing results, it makes sense to start by evaluating and choosing of the predictive models. This is followed by interpreting the values of the metrics, as well as understanding SHAP's analysis of the model, which will address the importance of each feature and how each one affects the results. Finally, the interactive analysis interface that is provided to the consultants will be shown. It is important to note that the results presented refer to a run of the script, the latest official information of which is for the first quarter of 2025. Therefore, the data horizon for this analysis begins in the second quarter of 2013.

5.1 Model Performance Comparison

In line with the methodology established earlier, R² was treated as the primary metric for model selection due to its effectiveness in summarizing the explained variance and overall fit. This choice was grounded in literature emphasizing that while R² is crucial for understanding a model's explanatory power, error metrics like MAE and RMSE offer essential insight into the practical magnitude of prediction errors. Accordingly, the model with the highest R² on the test set was chosen as the best predictive model, and the MAE and RMSE values were examined to contextualize the prediction accuracy in meaningful, monetary terms for stakeholders. Table 5.1 summarizes the performance of each model in terms of R², RMSE, and MAE.

Table 5.1: Model evaluation metrics (R², RMSE and MAE)

Model	**R^2**	**RMSE**	**MAE**
Random Forest	0.7346	593.52	403.54
XGBoost	0.8785	399.41	256.36
Gradient Boosting	0.8482	449.77	297.39
Neural Networks	0.5521	712.61	526.33
Linear Regression	-0.2043	1172.21	919.86

XGBoost emerged as the top performer with an R² of 0.8785, meaning it explains approximately 87.85% of the variance in the property price per square meter on the test data. This was the highest R² among the models, thus marking XGBoost as the best model by the primary metric. Notably, this finding is consistent with studies that have also reported XGBoost to be the most accurate algorithm for house price prediction, discussed in the literature review.

Gradient Boosting had the next best performance, with an R² of 84.82%, only slightly lower than XGBoost. As for Random Forest, it obtained the third most acceptable result, but already well below the previous two, with an R² of 73.46%. These ensemble tree-based models benefited from their ability to capture nonlinear relationships and complex feature interactions, which probably contributed to their good results.

The Neural Network model delivered a more modest R² (significantly lower than the ensemble methods), indicating weaker predictive performance. This comparatively lower R² for the neural network can be attributed to the relatively small training dataset available, as Neural Networks typically require substantial data to learn effectively; and practical constraints such as the inability to incorporate sample weighting in its training. In this case, unlike the tree-based models, the neural network could not easily use the *sample_weight* technique to emphasize recent observations, potentially limiting its ability to learn the market's temporal trends. These factors align with observations in the literature that inadequate data volume can degrade neural network performance.

Finally, the Linear Regression model underperformed by a large margin, even yielding a negative R² on the test set. A negative R² indicates that the model's predictions were worse than simply using the mean price as a prediction. This poor result highlights the inability of a simple linear model to capture the inherently nonlinear and complex relationships that govern real estate prices. In other words, the linear hedonic approach, with its strict additive linear form, failed to fit the data, whereas more flexible nonlinear models were able to learn the patterns much more effectively.

XGBoost was then selected as the final model for deployment given its highest R² and overall robust performance, while the other models ranged from strong (Gradient Boosting) to moderate (Neural Network, Random Forest) to unacceptable (Linear Regression) in predictive accuracy. For the XGBoost model with the data window considered, the best combination of hyperparameters was: *subsample* of 1, *reg_lambda* of 5, *reg_alpha* of 1, 1000 *n_estimators*, *min_child_weight* of 1, *max_depth* of 5, *learning_rate* of 0.1, gamma of 1, and *colsample_bytree* of 0.8.

Due to the divergences in the results of the metrics, it is believed that XGBoost will remain the best performing model for at least the near future. Only the competition from Gradient Boosting stands out, as it is the most likely to present better evaluation metrics in another future analysis period. If this is the case, the change in the pipeline only involves a small substitution in the *best_model* variable that goes into the *save_predictions* function, explained in Appendix G.

5.2 Error Metrics Interpretation

While R² drove the model selection, the error metrics provide critical insight into the model's prediction accuracy in practical terms. The chosen XGBoost model attained a MAE of 256.36 €/m² and an RMSE of 399.41 €/m² on the test set. These figures mean that, on average, the absolute difference between the predicted price and the actual price is about 256 euros per square meter, and typically (in a root-mean-square sense) the error is around 399 euros per square meter.

In the context of real estate, where property prices per square meter in the studied market can range broadly (often on the order of thousands of €/m²), an error of a few hundred euros per square meter is reasonably moderate. For example, if a particular parish has an actual average price of 4,000 €/m², the model's RMSE suggests the prediction could be roughly 10% higher or lower (within about ±400 €/m²) in a one-standard-error range. The MAE of 256.36 €/m² indicates that on average, the model's estimate will deviate from the true value by about that amount; therefore, a consultant might expect the model's pricing advice to be off by about 250 euros per square meter on average.

This level of accuracy can be translated into practical guidance for DILS consultants. In practice, the forecasted price can be presented not as a single point estimate but as a range (for example, "4,000 ± 400 €/m²"), conveying a confidence interval that reflects the model's error bounds. By doing so, consultants can give clients a sense of the uncertainty and a realistic span of probable prices, thus managing expectations and building trust in the tool's outputs.

The inclusion of error metrics alongside point predictions is crucial for stakeholders, as it contextualizes the risk of error in monetary terms. In essence, knowing that the typical error is on the order of a few hundred euros per square meter allows DILS to gauge how much leeway to consider when making pricing decisions. If needed, the firm can adjust its internal guidelines, for instance, requiring manual review or caution for predictions in areas where an additional ±400 €/m² could significantly influence a deal.

Overall, the magnitude of the MAE and RMSE indicates that the model achieves a respectable level of precision for real estate forecasting, and the errors are small enough to be useful for decision-making, yet they are also a reminder that predictions are not exact appraisals and should be used in conjunction with expert domain knowledge.

5.3 Feature Importance and Effects

After selecting XGBoost as the final model, an in-depth interpretability analysis was performed using SHAP values. This post-hoc analysis sheds light on how each feature contributes to the model's predictions of price per square meter.

5.3.1 Global Importance

After selecting XGBoost as the final model, an in-depth interpretability analysis was performed using SHAP values. This post-hoc analysis sheds light on how each feature contributes to the model's predictions of price per square meter. A global feature importance summary (Figure 5.1) was generated by computing the mean absolute SHAP value of each feature across the dataset. The resulting bar chart highlights the most influential predictors in the model. According to this analysis, the seven most important features (in descending order of influence) are *concelho_id*, *tempo_continuo*, *freguesia_id*, *estado*, *ihpc_h*, *tx_desemp*, and *tipologia*.

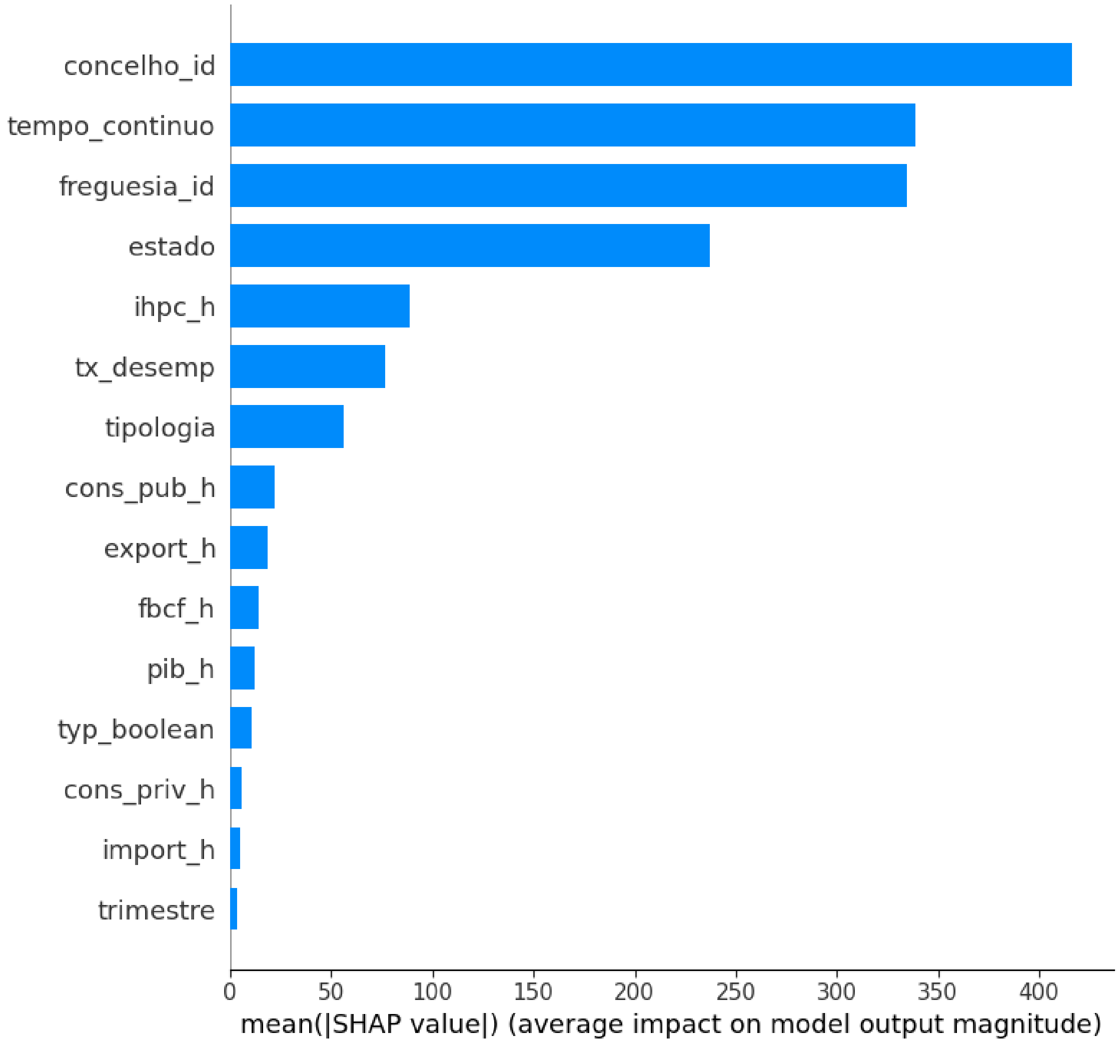


Figure 5.1: SHAP Summary Plot

The regional component emerges as one of the most influential factors, along with the constructed temporal evolution component. The property's state also proves highly important, with more moderate contributions coming from inflation trends and the unemployment rate. Property typology completes the group of top features, contributing at a relatively moderate level.

The remaining features continue to contribute, although their weights are much smaller. The feature corresponding to the quarter exhibits the lowest importance. Nevertheless, its inclusion remains essential in order to split the forecasts into individual quarters.

5.3.2 Effects on Predictions

To interpret the influence of individual features on the model's predictions, SHAP dependence plots were examined for the seven most important variables, previously identified in the summary plot. These were grouped into four thematic dimensions: spatial influence, temporal influence, property characteristics and macroeconomic environment.

5.3.2.1 Spatial Influence

The characteristics "municipality_id" (municipality) and "parish_id" (parish) capture the geographical heterogeneity in price formation. The dependency graphs (Figures 5.2 and 5.3) show that different municipalities and parishes consistently exert upward or downward pressure on forecast prices, confirming their strong base effect. Some spatial agglomerations have high positive SHAP values (above 1,000), while others significantly depress forecasts (below -1,000).

In addition, at municipality level, the *tempo_continuo* coloring indicates a hybrid trend, in that the more positively classified geographical areas register slight price increases over time, as opposed to the less classified areas, which register slight decreases. As for the parish, the most interactive variable is the status of the property (*estado*), and the discrepancy between new and used properties is clearly visible given the strong presence of red dots ("new") on top of blue dots ("used") in the dependency graph.

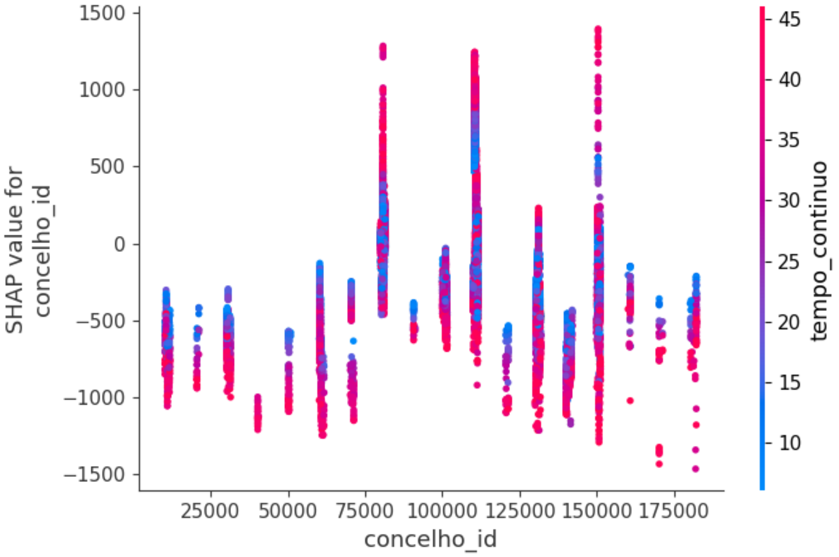


Figure 5.2: SHAP Dependence Plot for *concelho_id*

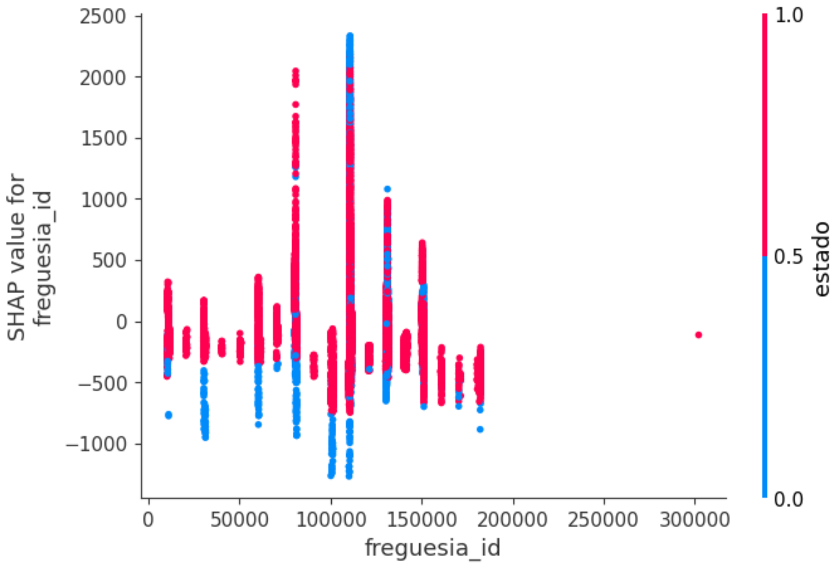


Figure 5.3: SHAP Dependence Plot for *freguesia_id*

5.3.2.2 Temporal Influence

The *tempo_continuo* variable therefore serves as an indicator of the passage of time, increasing by one unit per quarter. Its dependency graph (Figure 5.4) shows a clear upward trajectory in SHAP values as time progresses. The initial *tempo_continuo* values (the previous quarters) correspond to negative SHAP values, while the most recent observations contribute positively to the price forecasts. This confirms that the model effectively captures the underlying trend of price growth present in the data set. The variable that interacts the most is also property status, once again highlighting the importance of a property being new for price increases.

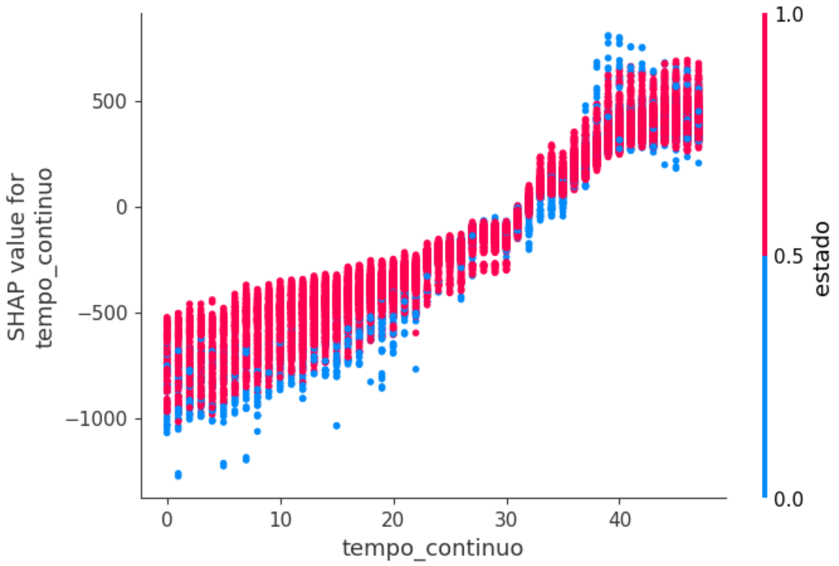


Figure 5.4: SHAP Dependence Plot for *tempo_continuo*

5.3.2.3 Property Characteristics

It is worth reiterating that the influence of the state of the property on the price is quite noticeable, given the analysis of the previous plots. As for the effect of typology, this is shown in Figure 5.5, and is more dispersed. Although some typologies are weakly associated with price increases, the overall pattern is less linear and seems more dependent on context. This suggests that typology alone is a weaker predictor than the state of the property, and probably interacts with location or market segment in more complex ways.

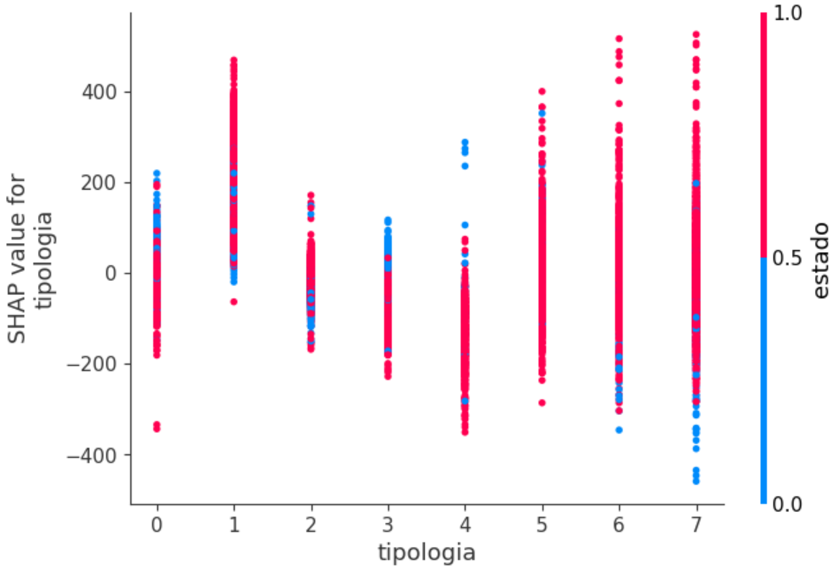


Figure 5.5: SHAP Dependence Plot for *tipologia*

5.3.2.4 Macroeconomic Environment

Finally, inflation (*ihpc_h*) and unemployment (*tx_desemp*) reflect broader economic conditions. The graph for *ihpc_h* (Figure 5.6) shows that higher inflation values tend to be associated with higher contributions to SHAP, indicating that nominal price forecasts increase in inflationary contexts, which is a consistent macroeconomic relationship. Conversely, the graph for *tx_desemp* (Figure 5.7) reveals a downward pattern, in which higher unemployment levels are associated with negative SHAP values. This suggests that the deterioration in labor market conditions reduces the expected pressure on prices from the demand side.

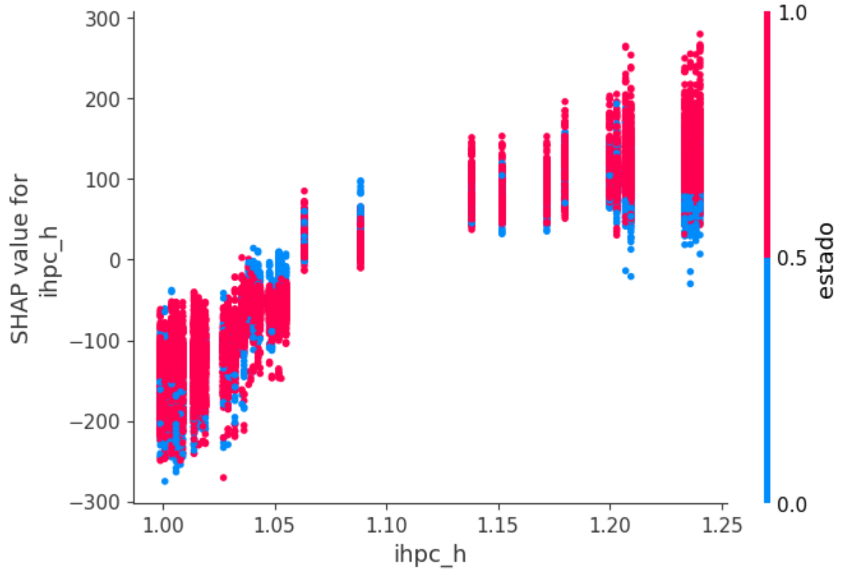


Figure 5.6: SHAP Dependence Plot for *ihpc_h*

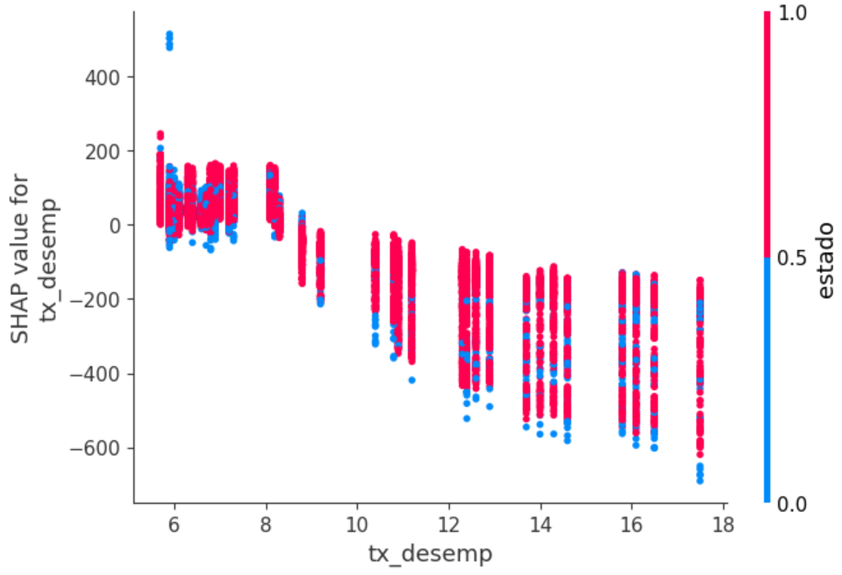


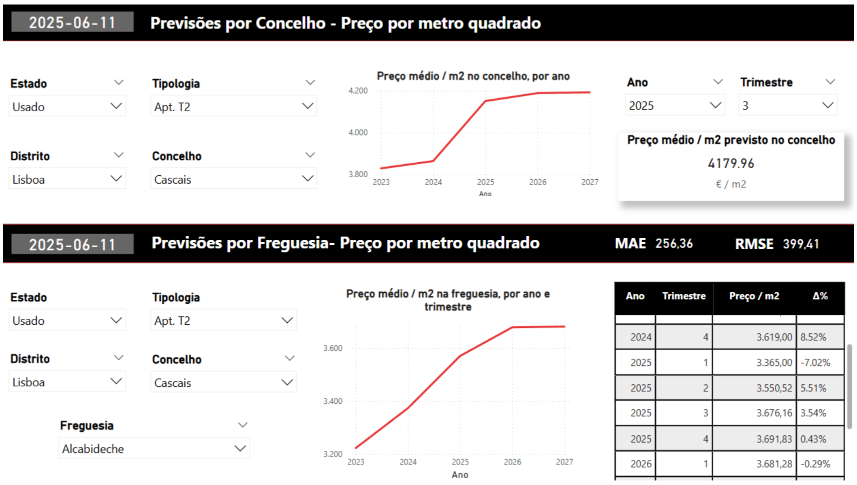
Figure 5.7: SHAP Dependence Plot for *tx_desemp*

5.3.3 Summary of Variable Interpretation

In summary, the SHAP analysis provided a transparent and structured view of how the XGBoost model interprets different input features in generating price predictions. The summary plot identified spatial identifiers, temporal progression, property condition, and macroeconomic indicators as the most influential variables. The dependence plots further clarified the nature of these effects: spatial variables exert a a stable and significant influence; time contributes positively through a clear upward trend; property condition shows a strong monotonic impact; and inflation and unemployment behave in line with expected macroeconomic theory. Together, these insights not only validate the internal logic of the model but also reinforce its interpretability for practical deployment. It should also be noted that consistency with the trends and observations described in the EDA (subsection 4.3.1.1) has been maintained.

5.4 Power BI Deployment

Finally, the forecasts generated by the XGBoost model are presented through a Power BI dashboard, providing consultants with quick and dynamic access via customizable filters such as location, property type, condition, and time period. As shown in Figure 5.8, the dashboard is divided into two sections. The upper section offers a general overview at the municipality level, while the lower section focuses on detailed analysis at the parish level.



2025-06-11 **Previsões por Concelho - Preço por metro quadrado**

Estado: Usado
Tipologia: Apt. T2
Distrito: Lisboa
Concelho: Cascais
Ano: 2025
Trimestre: 3

Preço médio / m2 previsto no concelho
4179.96
€ / m2

2025-06-11 **Previsões por Freguesia- Preço por metro quadrado** MAE 256,36 RMSE 399,41

Estado: Usado
Tipologia: Apt. T2
Distrito: Lisboa
Concelho: Cascais
Freguesia: Alcabideche

Ano	Trimestre	Preço / m2	Δ%
2024	4	3.619,00	8.52%
2025	1	3.365,00	-7.02%
2025	2	3.550,52	5.51%
2025	3	3.676,16	3.54%
2025	4	3.691,83	0.43%
2026	1	3.681,28	-0.29%

Preço médio / m2 no concelho, por ano
4.200
4.000
3.800
2023
2024
2025
2026
2027
Ano
Preço médio / m2 na freguesia, por ano e trimestre
3.600
3.400
3.200
2023
2024
2025
2026
2027
Ano

Figure 5.8: Power BI Dashboard

In the example illustrated in Figure 5.8, the upper section displays data filtered by "used" and "T2 apartment" within the municipality "Cascais", that belongs to the "Lisboa" district. The system automatically generates a trend graph over time (including the previous two years) which, in this case, shows a slight increase from 2025 to 2027. On the right-hand side, users can also select specific years and quarters to view the average price per square meter for the chosen municipality during that period. For instance, Cascais is forecasted to reach an average of 4,179.96 €/m² in the third quarter of 2025.

In the lower section of the dashboard, the same filters are applied, with the addition of a specific parish for analysis in Alcabideche. Similarly, a trend graph is displayed which, consistent with the overall trend of the municipality, shows a slight increase continuing through 2027. The table on the right presents historical data from the past eight quarters alongside forecasts for the next eight quarters, allowing consultants to assess local trends more effectively. According to the table, the model forecasts an average price of 3,676.16 €/m² for Alcabideche in the third quarter of 2025, representing a 3.54% increase compared to the second quarter of that year.

Additionally, the dashboard displays the MAE and RMSE values, providing consultants with an understanding of the model's predictive error. These metrics help establish a confidence interval of ±RMSE, while the MAE represents the average deviation from actual values in model validation.

Chapter 6

Conclusion and Future Work

This dissertation set out to design and implement a forecasting tool capable of estimating real estate prices per square meter in Portugal using ML techniques. Developed in collaboration with DILS, the project addressed a key gap in the company's valuation process, namely the absence of a robust, data-driven predictive system. By integrating macroeconomic indicators, granular geographical data, and property attributes into a unified ML pipeline, the study aimed to enhance both the speed and accuracy of property appraisals.

The work followed a structured methodology inspired by the CRISP-DM framework, encompassing data gathering, preprocessing, modeling, and deployment. Historical property transaction data from the SIR platform and macroeconomic series from BPstat were consolidated into a single dataset, then cleaned and enriched through advanced feature engineering. After evaluating the nature of the available data and operational objectives, it was decided that the forecasting model would operate at the granular level of parish and quarter, maximizing local specificity while preserving statistical robustness and allowing bottom-up aggregations.

Five predictive models were developed and compared: Linear Regression, Neural Networks, Random Forest, Gradient Boosting, and XGBoost. In the most recent analysis at the time of the dissertation, the XGBoost model outperformed all others, achieving an R^2 of 0.8785, along with the lowest RMSE and MAE values. This indicated not only high explanatory power but also reliable prediction accuracy in euros per square meter. Error metrics showed that predictions were on average within 250 euros of actual prices, which is an acceptable margin for operational real estate decision-making. SHAP-based interpretability confirmed that the model's predictions aligned with domain knowledge, with key influences being location (municipality and parish), temporal recency, property condition, typology, and macroeconomic variables like unemployment and inflation. The resulting tool provides consultants with forecasts that are both quantitative and explainable. This not only accelerates advisory tasks but introduces standardization and transparency in valuation. In practice, the tool can support price setting and help consultants reach informed conclusions based on its outputs.

Despite its strengths, the study has several limitations. Data sparsity in some parishes can undermine local prediction reliability. Aggregation in the SIR dataset may obscure transaction-level nuances. Moreover, the model was developed based on parish-level average values, as individual transaction-level data was not accessible. This constraint limits the model's ability to capture the full heterogeneity of the housing market. Additionally, the model does not account for various potentially relevant property-level features not present in the dataset. These could include elements such as whether a property has a swimming pool, balconies, proximity to transport or services, and other housing attributes that influence market value.

Future research could address these limitations by expanding the underlying datasets to include more detailed, property-level information. Access to transactional-level data would enable finer-grained modeling and improve the model's ability to distinguish between properties within the same parish. In addition, incorporating geospatial analytics and external location-based variables, such as accessibility to transport or proximity to amenities, could further enhance model accuracy. Another promising direction would be the exploration of time series-specific models that explicitly capture temporal dependencies and seasonal effects, possibly outperforming current approaches for longer-term forecasting if the appropriate data requirements are met. From a methodological standpoint, combining ML algorithms with domain-specific time series techniques may offer new insights and improve forecast stability under different market regimes.

From a scientific perspective, this dissertation contributes to the growing literature on data-driven property valuation by demonstrating that combining transactional, temporal, spatial, and macroeconomic data within an ML framework can yield high-performing and interpretable models. In contrast with traditional econometric approaches, this work emphasizes scalability, bottom-up structure, and explainability, features that are increasingly valued in both research and practice. Future studies could explore model generalization across different national contexts, benchmark various interpretability methods, or assess the long-term performance of forecasting systems in dynamic housing markets.

In conclusion, this study demonstrates the feasibility and value of applying ML to real estate forecasting. The XGBoost model developed here not only improves predictive performance over traditional approaches but also delivers consistent and structured outputs that can be interpreted and contextualized. For DILS, it provides a practical, scalable tool that enhances data-driven decision-making. More broadly, it contributes to a shift toward more transparent, consistent, and analytically rigorous practices in real estate valuation.

Bibliography

Abidoye, R. B. and Chan, A. P. (2018). Improving property valuation accuracy: A comparison of hedonic pricing model and artificial neural network. *Pacific Rim Property Research Journal*, 24(1):71–83.

Al-Qawasmi, J. (2022). Machine learning applications in real estate: critical review of recent development. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 231–249. Springer.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection.

Arnold, C., Biedebach, L., Küpfer, A., and Neunhoeffer, M. (2024). The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*, 12(4):841–848.

Baldominos, A., Blanco, I., Moreno, A. J., Iturrarte, R., Bernárdez, Ó., and Afonso, C. (2018). Identifying real estate opportunities using machine learning. *Applied sciences*, 8(11):2321.

Chen, B. and Andrews, S. H. (2008). An empirical review of methods for temporal distribution and interpolation in the national accounts. *Survey of Current Business*, 88(5):31–37.

Chicco, D., Warrens, M. J., and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623.

Conway, J. J. E. (2018). Artificial intelligence and machine learning: current applications in real estate.

de Amorim, L. B., Cavalcanti, G. D., and Cruz, R. M. (2023). The choice of scaling technique matters for classification performance. *Applied Soft Computing*, 133:109924.

Dils (2024). Dils enters the portuguese market with castelhana, leader in new residential developments.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., and Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big data*, 8.

Ferrara, E., De Meo, P., Fiumara, G., and Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-based systems*, 70:301–323.

Foerderer, J. (2023). Should we trust web-scraped data? *arXiv preprint arXiv:2308.02231*.

Hernandez, J., Chang, D., Gutierrez, S., and Huggins, P. (2024). Predictive analysis of local house prices: Leveraging machine learning for real estate valuation. *SMU Data Science Review*, 8(1):12.

Iacoviello, M. and Neri, S. (2010). Housing market spillovers: evidence from an estimated dsge model. *American economic journal: macroeconomics*, 2(2):125–164.

Jafary, P., Shojaei, D., Rajabifard, A., and Ngo, T. (2024). Automated land valuation models: A comparative study of four machine learning and deep learning methods based on a comprehensive range of influential factors. *Cities*, 151:105115.

Jaroszewicz, J. and Horynek, H. (2024). Aggregated housing price predictions with no information about structural attributes—hedonic models: Linear regression and a machine learning approach. *Land*, 13(11):1881.

Kamalov, F. and Sulieman, H. (2021). Time series signal recovery methods: comparative study. In *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–5. IEEE.

Khan, H. and Reza, A. (2017). House prices and government spending shocks. *Journal of Money, Credit and Banking*, 49(6):1247–1271.

Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).

Lee, H., Jeong, H., Lee, B., Lee, K. D., and Choo, J. (2023). St-rap: A spatio-temporal framework for real estate appraisal. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4053–4058.

Lo, D., Chau, K. W., Wong, S. K., McCord, M., and Haran, M. (2022). Factors affecting spatial autocorrelation in residential property prices. *Land*, 11(6):931.

Loberto, M., Luciani, A., and Pangallo, M. (2020). What do online listings tell us about the housing market? *arXiv preprint arXiv:2004.02706*.

Management Solutions (2023). Explainable Artificial Intelligence (XAI): Desafios na interpretabilidade de modelos.

Ncr, P. C., Clinton, J., Ncr, R. K., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. (1999). Crisp-dm 1.0.

Ouwehand, P., Hyndman, R. J., de Kok, T., and van Donselaar, K. H. (2007). A state space model for exponential smoothing with group seasonality. *Monash University Working Paper*.

Páez, A. (2009). Recent research in spatial real estate hedonic analysis. *Journal of Geographical systems*, 11(4):311–316.

Ribeiro, S. M. and de Castro, C. L. (2021). Missing data in time series: A review of imputation methods and case study. *Learning and Nonlinear Models-Revista Da Sociedade Brasileira De Redes Neurais-Special Issue: Time Series Analysis and Forecasting Using Computational Intelligence*, 19(2).

Rodrigues, P. M. (2022). *The real estate market in Portugal: prices, rents, tourism and accessibility*. Fundação Francisco Manuel dos Santos.

Rostami-Tabar, B. and Mircetic, D. (2023). Exploring the association between time series features and forecasting by temporal aggregation using machine learning. *Neurocomputing*, 548:126376.

Scikit-learn developers (2024a). 3.2.1. exhaustive grid search. Accessed: 2025-03-11.

Scikit-learn developers (2024b). 3.2.2. randomized parameter optimization (randomizedsearchcv). Accessed: 2025-03-11.

Sharma, H., Harsora, H., and Ogunleye, B. (2024). An optimal house price prediction algorithm: Xgboost. *Analytics*, 3(1):30–45.

Soltani, A., Heydari, M., Aghaei, F., and Pettit, C. J. (2022). Housing price prediction incorporating spatio-temporal dependency into machine learning algorithms. *Cities*, 131:103941.

Stock, J. H. and Watson, M. W. (2008). Phillips curve inflation forecasts.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, 16(4):437–450.

Vaidynathan, D., Kayal, P., and Maiti, M. (2023). Exploring the influence of economic factors on median list and selling prices in the us housing market.

Varghese, K., Mahdaviabbasabad, S., Gentile, G., Eldafrawi, M., et al. (2023). Effect of spatio-temporal granularity on demand prediction for deep learning models. *Transport and Telecommunication*, 24(1):22–32.

Wei, C., Fu, M., Wang, L., Yang, H., Tang, F., and Xiong, Y. (2022). The research development of hedonic price model-based real estate appraisal in the era of big data. *Land*, 11(3):334.

Wentland, S., Cornwall, G., and Moulton, J. G. (2023). For what it's worth: Measuring land value in the era of big data and machine learning. Technical report, Bureau of Economic Analysis.

Xu, X. and Zhang, Y. (2021). House price forecasting with neural networks. *Intelligent Systems with Applications*, 12:200052.

Zhao, Y., Ravi, R., Shi, S., Wang, Z., Lam, E. Y., and Zhao, J. (2022). Pate: Property, amenities, traffic and emotions coming together for real estate price prediction. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE.

Appendix A

CRISP-DM Methodology

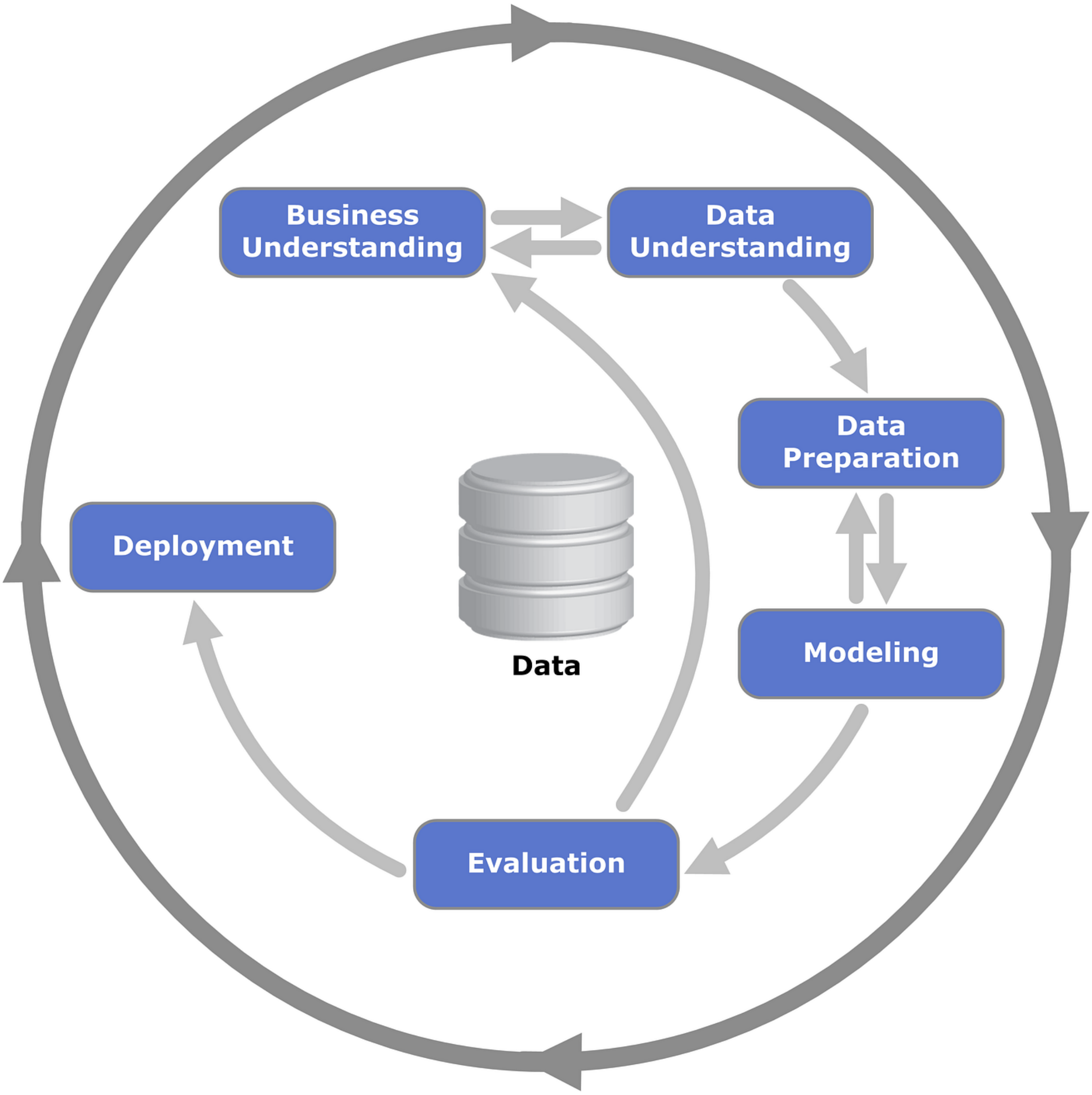


Figure A.1: CRISP-DM Methodology

Appendix B

Model Tuning and Validation

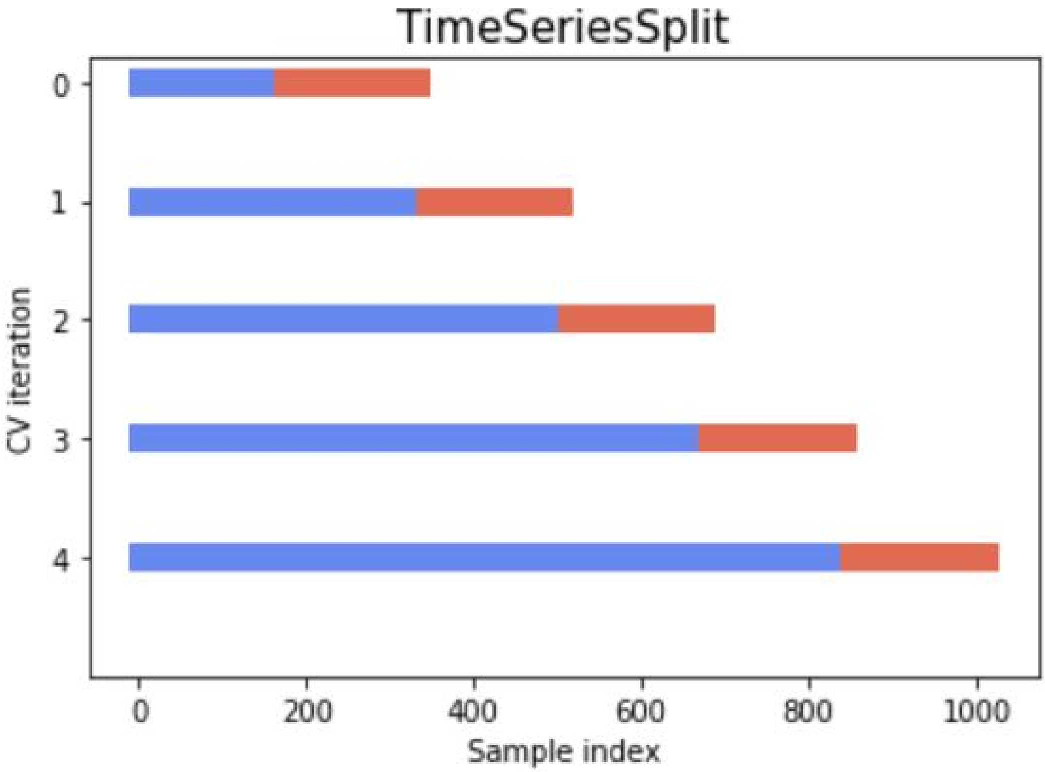


Figure B.1: TimeSeriesSplit sequential train-test partitions

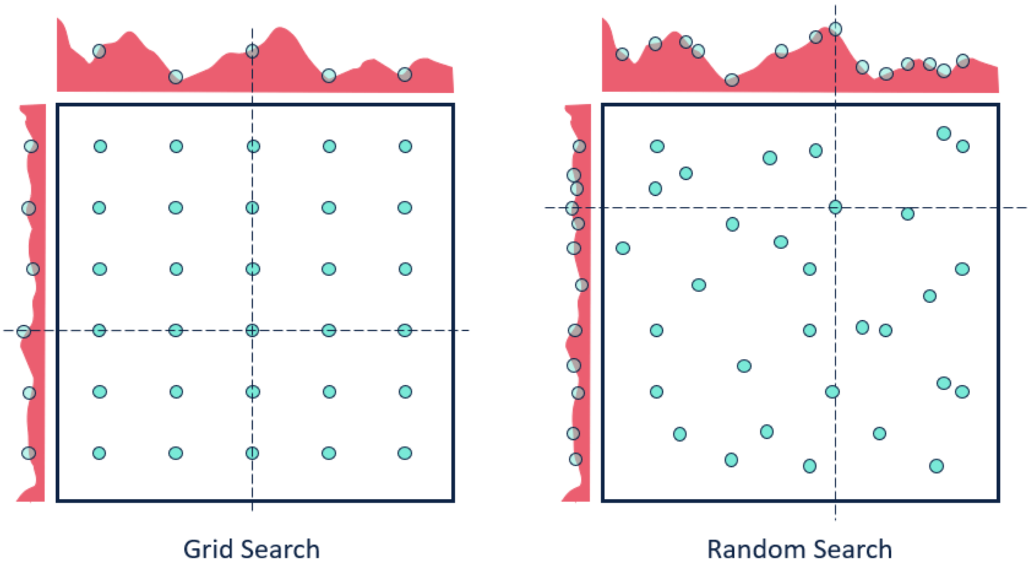
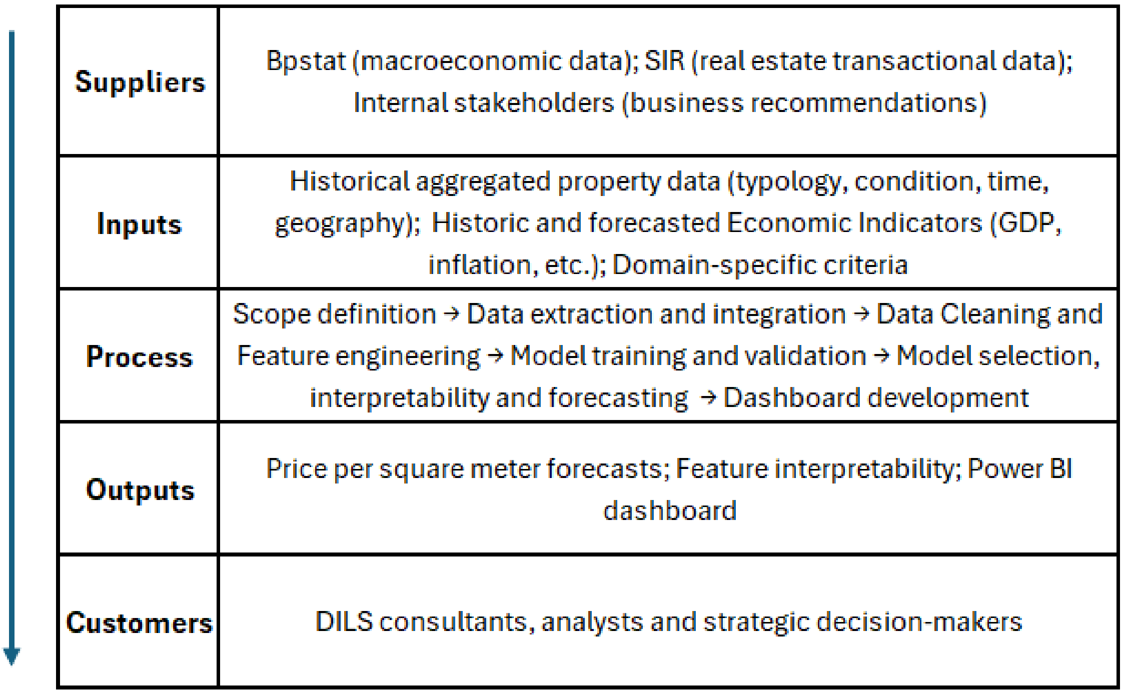


Figure B.2: GridSearch vs RandomSearch representation

Appendix C

Framework’s SIPOC diagram

Table C.1: Framework’s SIPOC diagram



Suppliers	Bpstat (macroeconomic data); SIR (real estate transactional data); Internal stakeholders (business recommendations)
Inputs	Historical aggregated property data (typology, condition, time, geography); Historic and forecasted Economic Indicators (GDP, inflation, etc.); Domain-specific criteria
Process	Scope definition → Data extraction and integration → Data Cleaning and Feature engineering → Model training and validation → Model selection, interpretability and forecasting → Dashboard development
Outputs	Price per square meter forecasts; Feature interpretability; Power BI dashboard
Customers	DILS consultants, analysts and strategic decision-makers

Appendix D

Examples of SIR Spatial Options

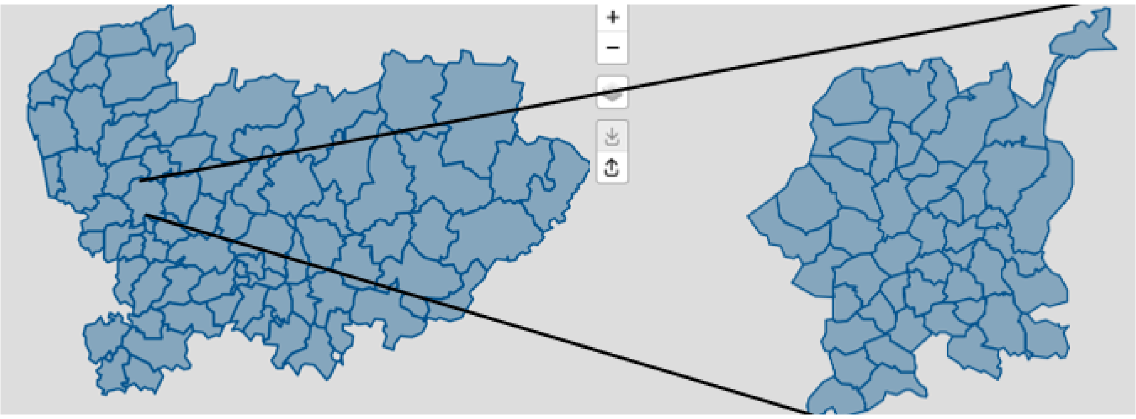
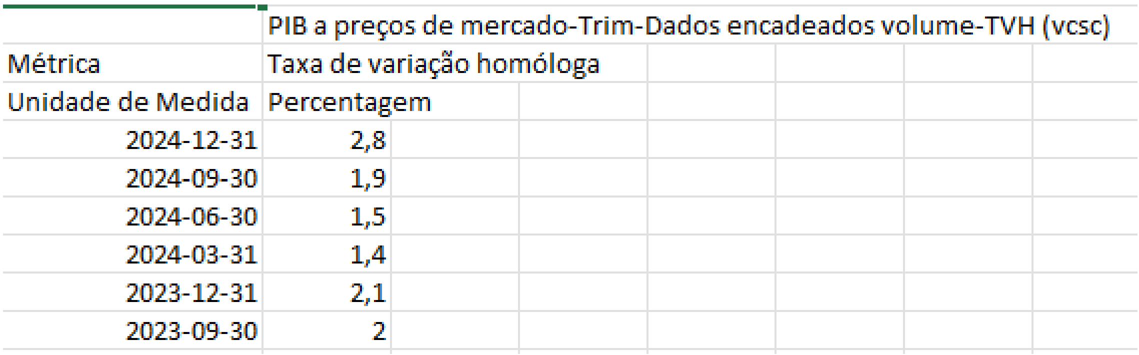


Figure D.1: Examples of SIR Spatial Options

- **On the left:** All the municipalities in the northern region;
- **On the right:** All the parishes in the municipality of Guimarães, which is located in the northern region.

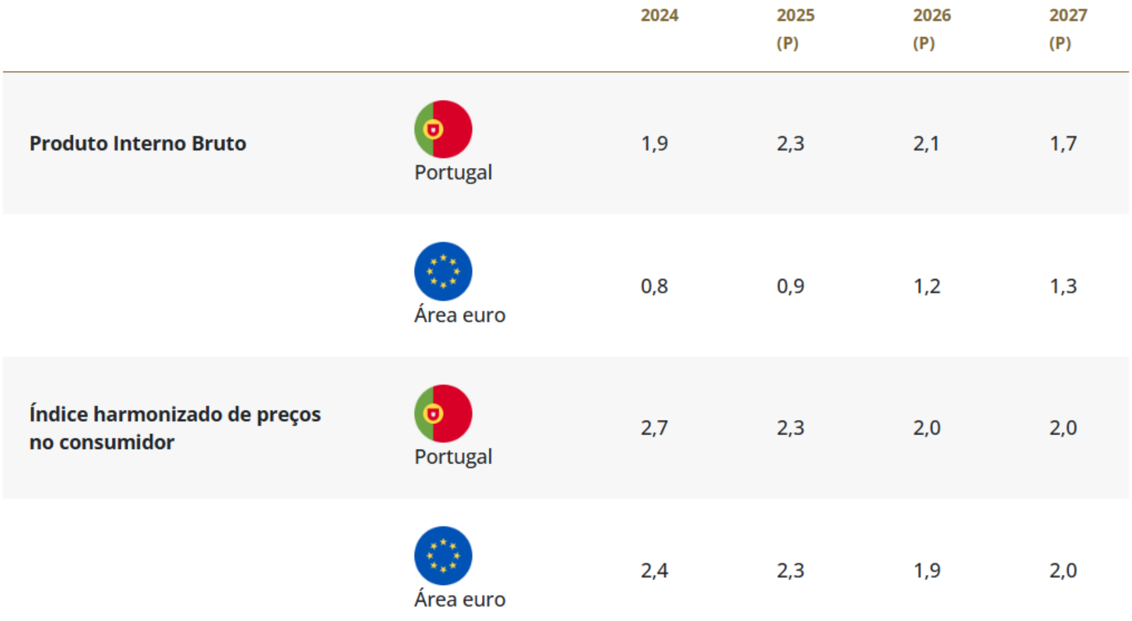
Appendix E

BPstat Content



	PIB a preços de mercado-Trim-Dados encadeados volume-TVH (vcsc)
Métrica	Taxa de variação homóloga
Unidade de Medida	Percentagem
2024-12-31	2,8
2024-09-30	1,9
2024-06-30	1,5
2024-03-31	1,4
2023-12-31	2,1
2023-09-30	2

Figure E.1: Output in Excel of the year-on-year rate of change in GDP at market prices, quarterly, chain-linked volume data



		2024	2025 (P)	2026 (P)	2027 (P)
Produto Interno Bruto	Portugal	1,9	2,3	2,1	1,7
	Área euro	0,8	0,9	1,2	1,3
Índice harmonizado de preços no consumidor	Portugal	2,7	2,3	2,0	2,0
	Área euro	2,4	2,3	1,9	2,0

Figure E.2: BPstat forecasts for GDP and Inflation in Portugal and Euro Area

Appendix F

Database Update Script

The layout of the entire code could be exhaustive, so it was decided to explain, part by part, what happens in the script's main function. The *main()* function (Figure F.1) coordinates the entire automation process that updates the central forecasting database.

This update is composed of two core stages: extracting new quarterly residential transaction data from the SIR platform and enriching the resulting dataset with macroeconomic indicators from BPstat. Together, they ensure that the forecasting model always uses the latest available information. Below is a step-by-step explanation of this process, with references to specific functions and variables.

The script begins by storing the current time in the variable (*tempo_inicial*) to measure execution duration. It defines the URL for the data source (*wanted_page*), the path to save the extracted Excel files (*pasta_dados*), and the file path to the full forecasting database (*database_path*).

It is important to note that the extraction must be performed in six iterations, one for each major region, since it is not feasible to extract all parish-level data at once due to the large data volume. So, the script iterates over the six major mainland Portuguese regions defined in the list (*zonas*): *Norte*, *AM Lisboa*, *AM Porto*, *Centro*, *Alentejo* and *Algarve*. For each region (*zona*), the following sequence is executed:

- The function *login* starts an authenticated session with the SIR platform using the credentials stored in environment variables (*email*, *password*);

- Next, *get_driver* initializes a Selenium WebDriver with cookies from the authenticated session, allowing automated navigation of the platform while remaining logged in;

- Then, *load_main_page* navigates to the data request form and waits until a key interface element is fully loaded;

- Afterwards, *selecionar_trimestral* sets the time granularity of the data to quarterly (ensuring compatibility with the rest of the forecasting workflow), choosing the last available quarter;

- The function *select_checkboxes* then activates filters for typology, property condition, variables to extract and geographical disaggregation (then, a short delay (*time.sleep(2)*) ensures that the interface has completed any dynamic rendering required);

- After that, *select_zone* selects the current region and the function *include_concelhos* expands and selects all municipalities within that region (as geographical disaggregation is already selected, the output includes all parishes within the municipalities);

- The function *finalize* confirms all selections, triggering the generation of the data output;

- Then, *download_excel* waits for the output, downloads the resulting Excel file, renames it using the current region and timestamp, and moves it into the designated folder *pasta_dados*;

- Finally, the browser is closed via *driver.quit* to free system resources.

Once all regions have been processed and the respective Excel files saved, the function *filtrar_e_juntar_ficheiros_excel* merges the individual files into a single dataset. Then, the function filters the data to retain only entries for the variable “Preço de Venda / m²” (sale price per square meter), producing an appropriate CSV file for insertion named using the variable *filename*.

To ensure the integrity of this file and avoid errors, the function *check_variable* checks that only the expected variable is present, and the function *check_date* verifies that the time reference is quarterly and that the same data does not already exist in the main database. These two validation steps return boolean flags (*valid_var* and *valid_date*), which are then evaluated by the function *check_file* to determine whether the file can be safely added (*validate*).

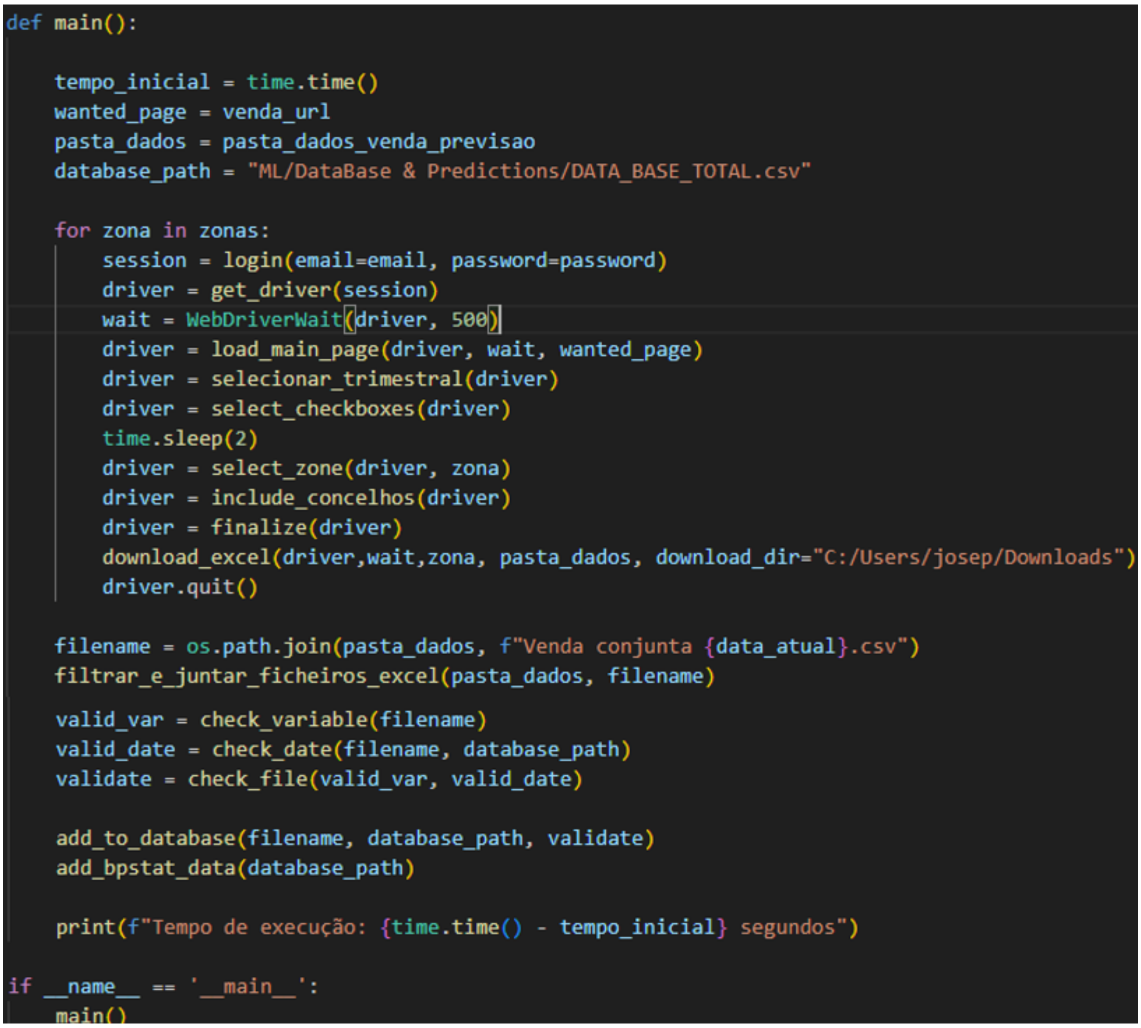
If validation succeeds (*validate == 1*), the function *add_to_database* appends the newly cleaned dataset (*filename*) to the full forecasting dataset located at *database_path*. This ensures that the model has access to the most up-to-date transactional records from the real estate market.

After updating the property transaction data, the script proceeds to update macroeconomic indicators by calling the function *add_bpstat_data*, which is the last visible function in Figure F.1. Initially, this function calls internally another one called *get_bpstat_data*, which interacts with the BPstat API and downloads the most recent values for the eight key indicators: inflation, GDP, unemployment rate, private and public consumption, GFCF, exports and imports. Still in *get_BPstat_data*, data is retrieved only if the latest available date corresponds to the end of a quarter (only if the month is march, june, september, or december), to maintain consistency with the forecasting model’s quarterly granularity (applying the monthly to quarterly transformation for inflation described in subchapter 4.2.2).

Each of these series is written to a separate CSV file in a folder. Then, within *add_bpstat_data*, these values are loaded and added to the corresponding quarter in the dataset by matching rows on the "data" column. This enrichment allows the model to link real estate prices with relevant macroeconomic conditions.

Finally, the updated database is saved back to the original CSV path (*database_path*). The script concludes by printing the total execution time, calculated as the difference between the current time and the earlier value stored in *tempo_inicial*.

This integrated process guarantees that the forecasting model is continuously fueled with recent, validated, and context-rich data, combining local property market behavior with national economic signals, all through a single automated routine.



```
def main():

    tempo_inicial = time.time()
    wanted_page = venda_url
    pasta_dados = pasta_dados_venda_previsao
    database_path = "ML/DataBase & Predictions/DATA_BASE_TOTAL.csv"

    for zona in zonas:
        session = login(email=email, password=password)
        driver = get_driver(session)
        wait = WebDriverWait(driver, 500)
        driver = load_main_page(driver, wait, wanted_page)
        driver = selecionar_trimestral(driver)
        driver = select_checkboxes(driver)
        time.sleep(2)
        driver = select_zone(driver, zona)
        driver = include_concelhos(driver)
        driver = finalize(driver)
        download_excel(driver,wait,zona, pasta_dados, download_dir="C:/Users/josep/Downloads")
        driver.quit()

    filename = os.path.join(pasta_dados, f"Venda conjunta {data_atual}.csv")
    filtrar_e_juntar_ficheiros_excel(pasta_dados, filename)

    valid_var = check_variable(filename)
    valid_date = check_date(filename, database_path)
    validate = check_file(valid_var, valid_date)

    add_to_database(filename, database_path, validate)
    add_bpstat_data(database_path)

    print(f"Tempo de execução: {time.time() - tempo_inicial} segundos")

if __name__ == '__main__':
    main()
```

Figure F.1: Scraper's Main Code

Appendix G

Pipeline Script

As in the previous appendix, it was decided to explain what happens in the main function, avoiding an exhaustive explanation of the entire code.

The *main()* function presented in Figure G.1 corresponds to the final version of the forecasting routine delivered to the company. This consolidated version was designed to be fully operational, automated, and interpretable by internal users.

It is important to note that this version uses exclusively the *XGBoost* model as the predictive engine, as it demonstrated the best performance during the validation phase. Consequently, unlike earlier experimental versions, the *main()* function does not invoke any normalization routine, since such preprocessing is not required for tree-based models. Nevertheless, normalization functions such as *StandardScaler* (for *z-score* normalization) and *MinMaxScaler* (for *Min-Max* normalization) are included in the codebase and can be used if other algorithms, such as Neural Networks, are to be tested in the future.

Similarly, although the *XGB* function is explicitly called in this version, equivalent functions exist for other models such as *RandomForest*, *GradientBoosting*, *LinearRegression*, and *MLPRegressor* (for Neural Networks). These alternatives were omitted from the main routine for the sake of clarity and focus, as the objective was to deliver a robust and optimized solution based on the best-performing model.

This being clear, the script begins by setting the path to the updated real estate historical database (*dataset_path*) and the output directory for the Power BI dashboard (*PBI_dir*). The function *read_dataset* loads the database into memory. Then, *separar_ano_trimestre* splits the original *data* column into separate *ano* (year) and *trimestre* (quarter) columns for temporal processing.

To focus the analysis on recent data, the function *update_to_newest* filters the database to retain only the last 12 years of quarterly data. For later usage, it also identifies the earliest and latest year-quarter combinations available (stored in *primeiro_ano*, *primeiro_trimestre*, *maior_ano*, *maior_trimestre*).

Next, *time_and_weights* creates the new time index variable (*tempo_continuo*) and assigns an exponentially decaying weight (*peso_temporal*) to each observation, emphasizing more recent data during model training.

The function *clean_transform_dataset* performs a complete cleaning and transformation of the dataset:

- Drops the irrelevant columns and filters out the redundant or aggregate rows;

- Sorts the dataset in ascending date order;

- Creates a boolean column *typ_boolean* to distinguish between apartment and house typlogies;

- Encodes categorical string fields using another defined function called *convert_string_to_class*, saving their mappings for dashboard integration;

- Constructs cumulative historical paths for each macroeconomic variable via the function *feature_creation_1*, erasing the original ones right after;

- Imputes missing values using the FFill's logic described, invoking the function *deal_w_nulls*;

After all this, the cleaned dataset is saved locally as *data_cleaned.csv*

The next step is to create a forecast dataset. This begins with the function *extract_economic_forecasts*, which scrapes the most recent macroeconomic projections for the next two years from BPstat's official portal. The function *predict_table_creator* then uses this data, together with the structure of the historical dataset, to generate a forecasting-ready dataset saved as *To_predict.csv*.

As it was used in previous functions, only now is the column *ano* dropped. Then, the function *split_data* is used to separate the features (X) from the target variable (y, average price per square meter).

After that, the model training and selection phase begins:

- The function *XGB* tunes a *XGBRegressor* using *RandomizedSearchCV* with 10-fold time series cross-validation (via *TimeSeriesSplit*), previously extracting *temporal_weight* to be used as a parameter and not as a feature;

- The selected model is then validated using the *time_series_validation* function, which performs a walk-forward validation and calculates the average RMSE, MAE, and R²;

- The function *explain_with_shap* generates interpretability outputs, as a SHAP summary plot (showing the most influential features) and a series of dependence plots (for individual variables).

The best-performing model is defined as *best_model* and trained on the full dataset using the function *train_model*, which incorporates sample weights.

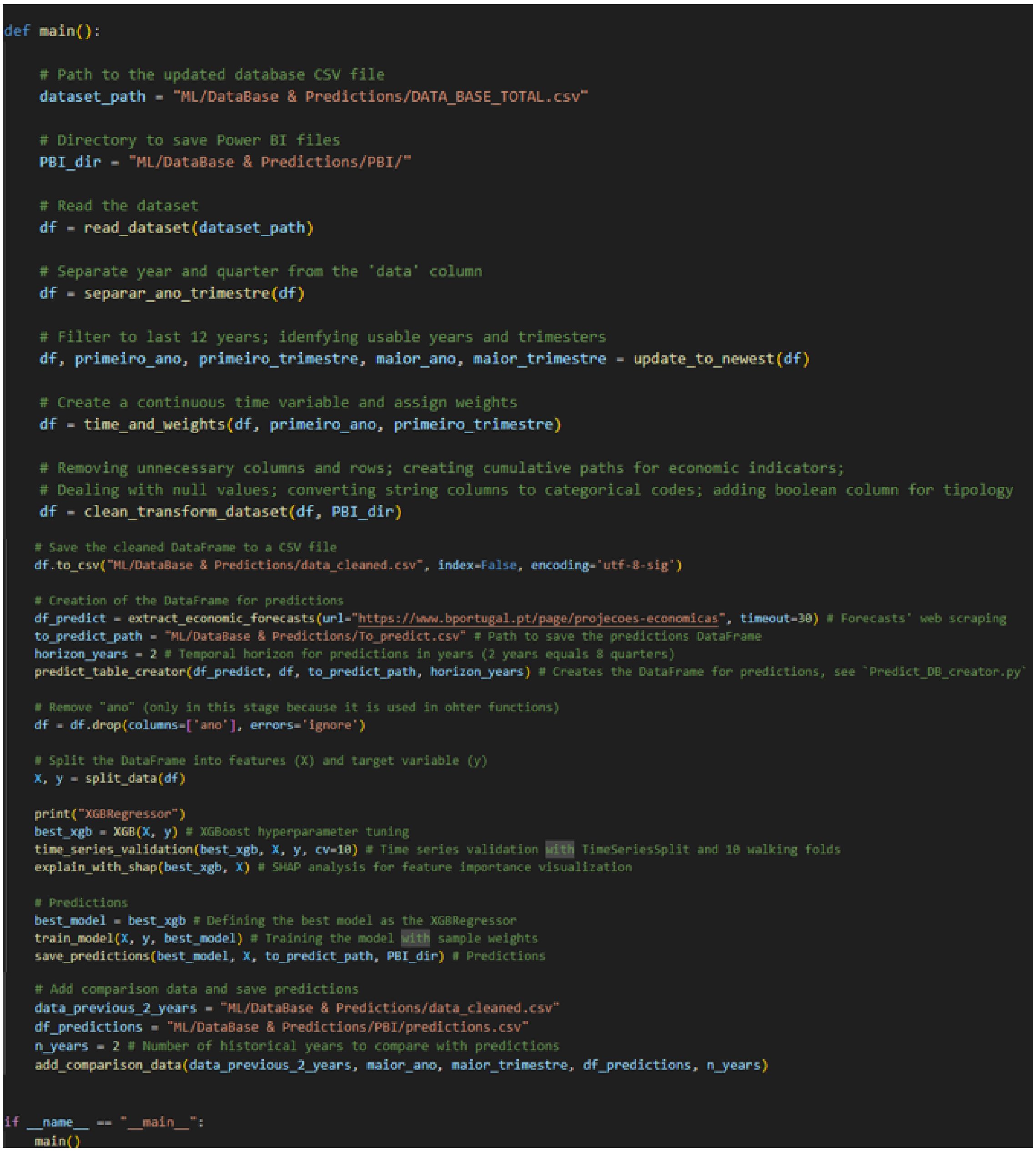
The trained model is then used to generate predictions for future quarters using *save_predictions*, which:

- Reads the prediction dataset from *To_predict.csv*;

- Applies the selected best model to compute forecasts;

- Saves predictions in *predictions.csv* and records the current date in *update_date.csv*.

To support historical comparison in the Power BI dashboard, the function *add_comparison_data* appends data from the previous *n_years* (two in the case described) to the forecast file. This allows users to visually assess how future prices compare with recent real trends.

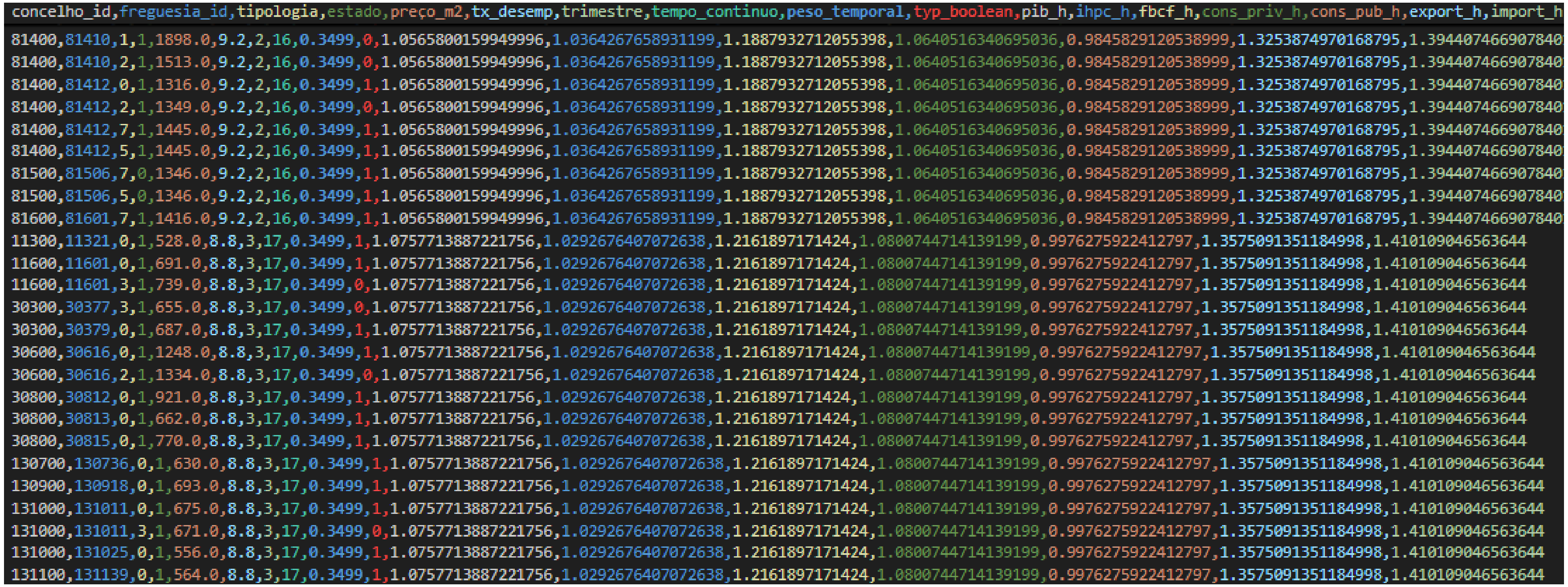


```
def main():

    # Path to the updated database CSV file
    dataset_path = "ML/DataBase & Predictions/DATA_BASE_TOTAL.csv"

    # Directory to save Power BI files
    PBI_dir = "ML/DataBase & Predictions/PBI/"

    # Read the dataset
    df = read_dataset(dataset_path)

    # Separate year and quarter from the 'data' column
    df = separar_ano_trimestre(df)

    # Filter to last 12 years; idenfying usable years and trimesters
    df, primeiro_ano, primeiro_trimestre, maior_ano, maior_trimestre = update_to_newest(df)

    # Create a continuous time variable and assign weights
    df = time_and_weights(df, primeiro_ano, primeiro_trimestre)

    # Removing unnecessary columns and rows; creating cumulative paths for economic indicators;
    # Dealing with null values; converting string columns to categorical codes; adding boolean column for tipology
    df = clean_transform_dataset(df, PBI_dir)

    # Save the cleaned DataFrame to a CSV file
    df.to_csv("ML/DataBase & Predictions/data_cleaned.csv", index=False, encoding='utf-8-sig')

    # Creation of the DataFrame for predictions
    df_predict = extract_economic_forecasts(url="https://www.bportugal.pt/page/projecoes-economicas", timeout=30) # Forecasts' web scraping
    to_predict_path = "ML/DataBase & Predictions/To_predict.csv" # Path to save the predictions DataFrame
    horizon_years = 2 # Temporal horizon for predictions in years (2 years equals 8 quarters)
    predict_table_creator(df_predict, df, to_predict_path, horizon_years) # Creates the DataFrame for predictions, see `Predict_DB_creator.py`

    # Remove "ano" (only in this stage because it is used in ohter functions)
    df = df.drop(columns=['ano'], errors='ignore')

    # Split the DataFrame into features (X) and target variable (y)
    X, y = split_data(df)

    print("XGBRegressor")
    best_xgb = XGB(X, y) # XGBoost hyperparameter tuning
    time_series_validation(best_xgb, X, y, cv=10) # Time series validation with TimeSeriesSplit and 10 walking folds
    explain_with_shap(best_xgb, X) # SHAP analysis for feature importance visualization

    # Predictions
    best_model = best_xgb # Defining the best model as the XGBRegressor
    train_model(X, y, best_model) # Training the model with sample weights
    save_predictions(best_model, X, to_predict_path, PBI_dir) # Predictions

    # Add comparison data and save predictions
    data_previous_2_years = "ML/DataBase & Predictions/data_cleaned.csv"
    df_predictions = "ML/DataBase & Predictions/PBI/predictions.csv"
    n_years = 2 # Number of historical years to compare with predictions
    add_comparison_data(data_previous_2_years, maior_ano, maior_trimestre, df_predictions, n_years)


if __name__ == "__main__":
    main()
```

Figure G.1: Pipeline's Main Code

Appendix H

Features Selected for Modeling



```
concelho_id,freguesia_id,tipologia,estado,preço_m2,tx_desemp,trimestre,tempo_continuo,peso_temporal,typ_boolean,pib_h,ihpc_h,fbcf_h,cons_priv_h,cons_pub_h,export_h,import_h
81400,81410,1,1,1898.0,9.2,2,16,0.3499,0,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81400,81410,2,1,1513.0,9.2,2,16,0.3499,0,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81400,81412,0,1,1316.0,9.2,2,16,0.3499,1,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81400,81412,2,1,1349.0,9.2,2,16,0.3499,0,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81400,81412,7,1,1445.0,9.2,2,16,0.3499,1,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81400,81412,5,1,1445.0,9.2,2,16,0.3499,1,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81500,81506,7,0,1346.0,9.2,2,16,0.3499,1,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81500,81506,5,0,1346.0,9.2,2,16,0.3499,1,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
81600,81601,7,1,1416.0,9.2,2,16,0.3499,1,1.0565800159949996,1.0364267658931199,1.1887932712055398,1.0640516340695036,0.9845829120538999,1.3253874970168795,1.394407466907840
11300,11321,0,1,528.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
11600,11601,0,1,691.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
11600,11601,3,1,739.0,8.8,3,17,0.3499,0,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30300,30377,3,1,655.0,8.8,3,17,0.3499,0,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30300,30379,0,1,687.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30600,30616,0,1,1248.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30600,30616,2,1,1334.0,8.8,3,17,0.3499,0,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30800,30812,0,1,921.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30800,30813,0,1,662.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
30800,30815,0,1,770.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
130700,130736,0,1,630.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
130900,130918,0,1,693.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
131000,131011,0,1,675.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
131000,131011,3,1,671.0,8.8,3,17,0.3499,0,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
131000,131025,0,1,556.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
131100,131139,0,1,564.0,8.8,3,17,0.3499,1,1.0757713887221756,1.0292676407072638,1.2161897171424,1.0800744714139199,0.9976275922412797,1.3575091351184998,1.410109046563644
```

Appendix I

Exploratory Data Analysis

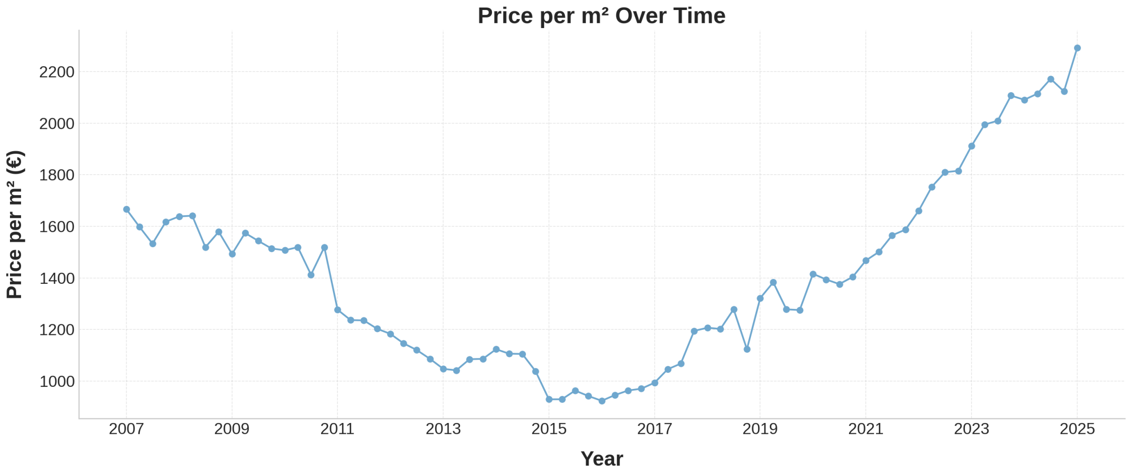


Figure I.1: Price per Square Meter Over Time

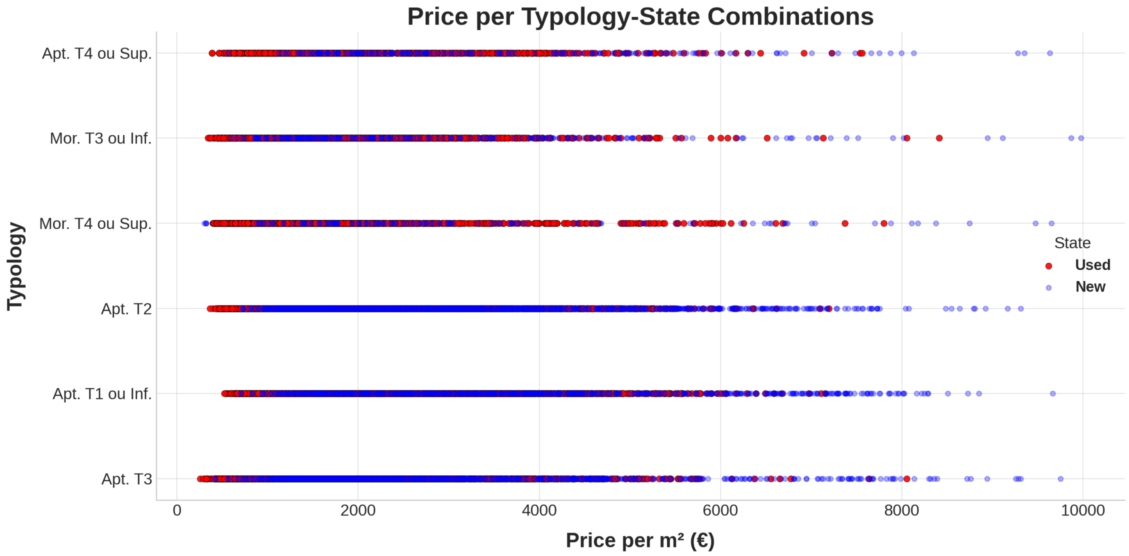


Figure I.2: Price per Square Meter Across Typology-State Combinations

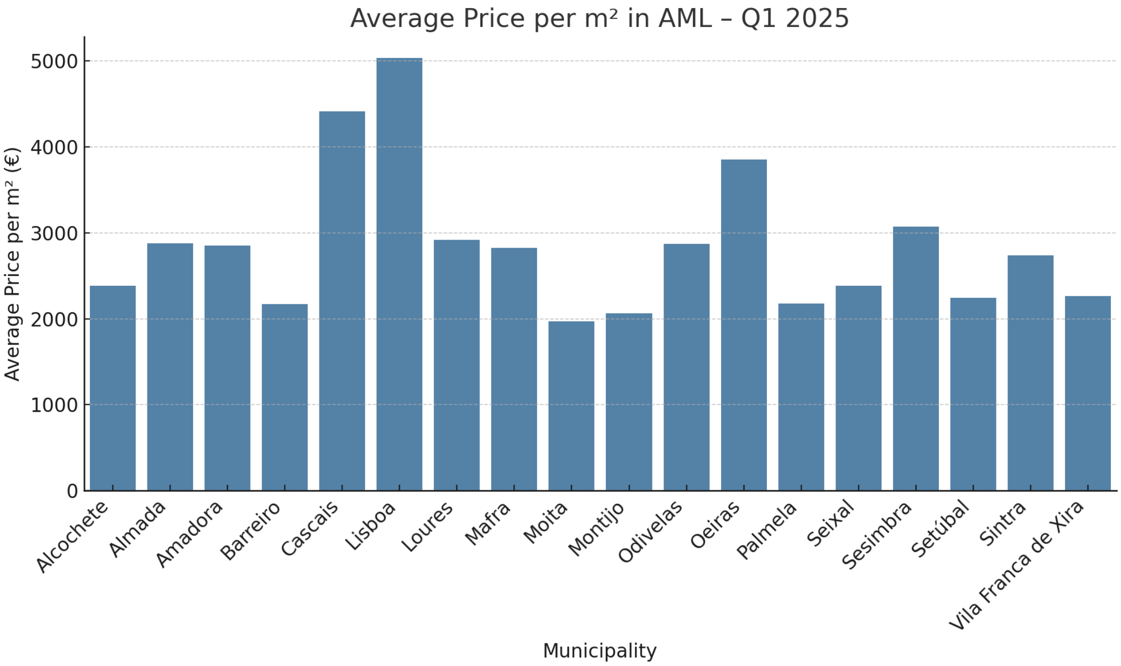


Figure I.3: Price per Square Meter Across Lisbon Metropolitan Area's Municipalities (2025 Q1)

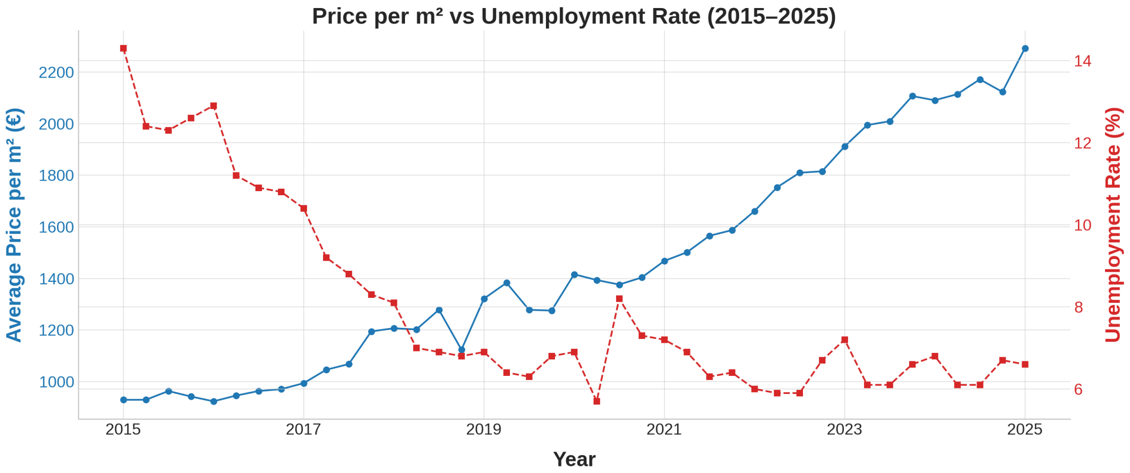
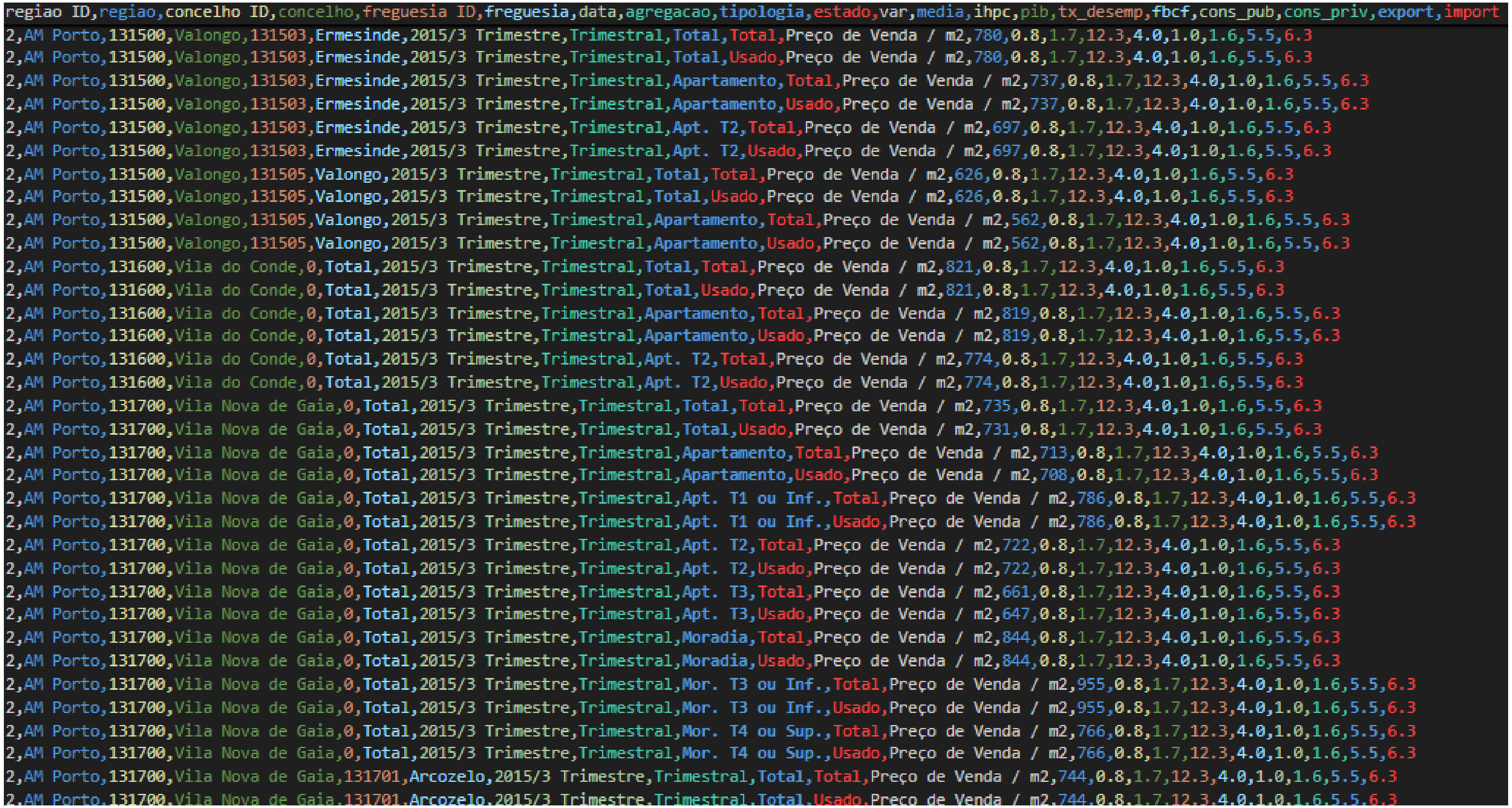


Figure I.4: Price per Square Meter Over Time

Appendix J

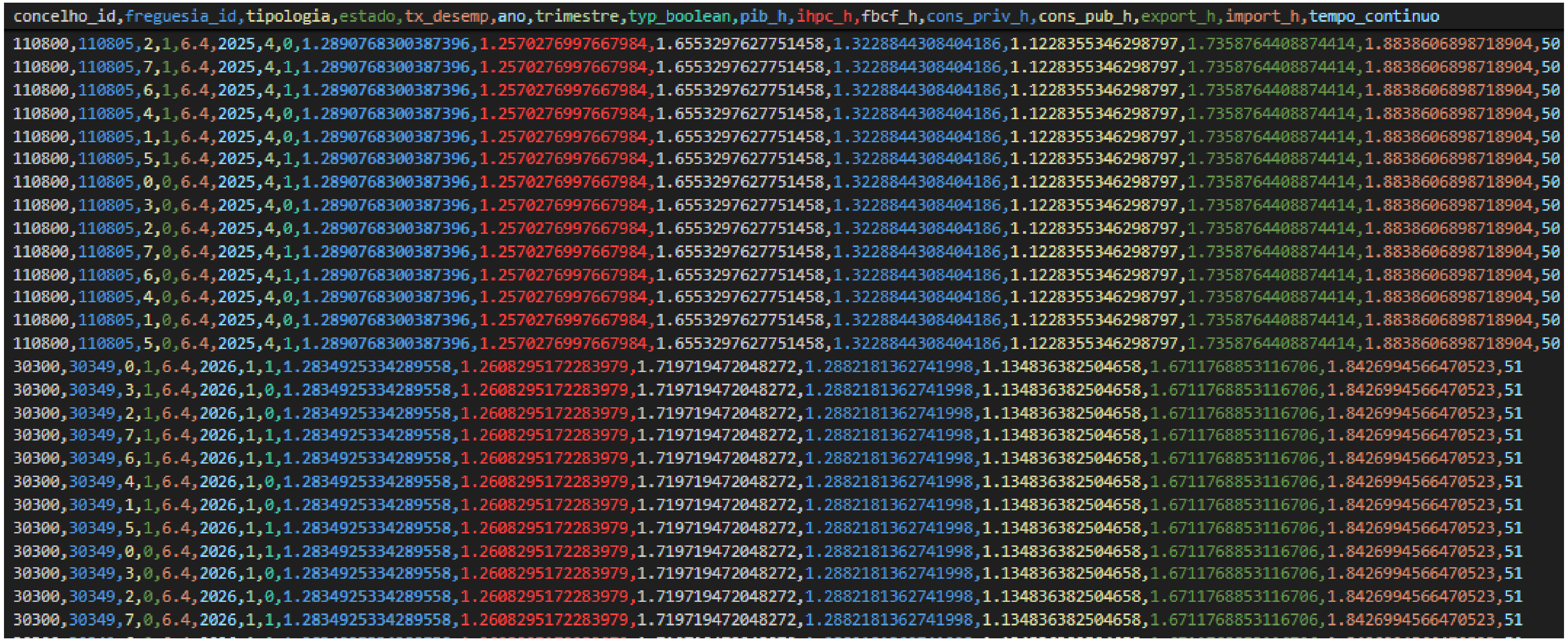
Data Repository Overview



```
regiao ID,regiao,concelho ID,concelho,freguesia ID,freguesia,data,agregacao,tipologia,estado,var,media,ihpc,pib,tx_desemp,fbcf,cons_pub,cons_priv,export,import
2,AM Porto,131500,Valongo,131503,Ermesinde,2015/3 Trimestre,Trimestral,Total,Total,Preço de Venda / m2,780,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131503,Ermesinde,2015/3 Trimestre,Trimestral,Total,Usado,Preço de Venda / m2,780,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131503,Ermesinde,2015/3 Trimestre,Trimestral,Apartamento,Total,Preço de Venda / m2,737,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131503,Ermesinde,2015/3 Trimestre,Trimestral,Apartamento,Usado,Preço de Venda / m2,737,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131503,Ermesinde,2015/3 Trimestre,Trimestral,Apt. T2,Total,Preço de Venda / m2,697,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131503,Ermesinde,2015/3 Trimestre,Trimestral,Apt. T2,Usado,Preço de Venda / m2,697,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131505,Valongo,2015/3 Trimestre,Trimestral,Total,Total,Preço de Venda / m2,626,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131505,Valongo,2015/3 Trimestre,Trimestral,Total,Usado,Preço de Venda / m2,626,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131505,Valongo,2015/3 Trimestre,Trimestral,Apartamento,Total,Preço de Venda / m2,562,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131500,Valongo,131505,Valongo,2015/3 Trimestre,Trimestral,Apartamento,Usado,Preço de Venda / m2,562,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131600,Vila do Conde,0,Total,2015/3 Trimestre,Trimestral,Total,Total,Preço de Venda / m2,821,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131600,Vila do Conde,0,Total,2015/3 Trimestre,Trimestral,Total,Usado,Preço de Venda / m2,821,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131600,Vila do Conde,0,Total,2015/3 Trimestre,Trimestral,Apartamento,Total,Preço de Venda / m2,819,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131600,Vila do Conde,0,Total,2015/3 Trimestre,Trimestral,Apartamento,Usado,Preço de Venda / m2,819,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131600,Vila do Conde,0,Total,2015/3 Trimestre,Trimestral,Apt. T2,Total,Preço de Venda / m2,774,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131600,Vila do Conde,0,Total,2015/3 Trimestre,Trimestral,Apt. T2,Usado,Preço de Venda / m2,774,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Total,Total,Preço de Venda / m2,735,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Total,Usado,Preço de Venda / m2,731,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apartamento,Total,Preço de Venda / m2,713,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apartamento,Usado,Preço de Venda / m2,708,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apt. T1 ou Inf.,Total,Preço de Venda / m2,786,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apt. T1 ou Inf.,Usado,Preço de Venda / m2,786,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apt. T2,Total,Preço de Venda / m2,722,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apt. T2,Usado,Preço de Venda / m2,722,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apt. T3,Total,Preço de Venda / m2,661,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Apt. T3,Usado,Preço de Venda / m2,647,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Moradia,Total,Preço de Venda / m2,844,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Moradia,Usado,Preço de Venda / m2,844,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Mor. T3 ou Inf.,Total,Preço de Venda / m2,955,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Mor. T3 ou Inf.,Usado,Preço de Venda / m2,955,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Mor. T4 ou Sup.,Total,Preço de Venda / m2,766,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,0,Total,2015/3 Trimestre,Trimestral,Mor. T4 ou Sup.,Usado,Preço de Venda / m2,766,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,131701,Arcozelo,2015/3 Trimestre,Trimestral,Total,Total,Preço de Venda / m2,744,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
2,AM Porto,131700,Vila Nova de Gaia,131701,Arcozelo,2015/3 Trimestre,Trimestral,Total,Usado,Preço de Venda / m2,744,0.8,1.7,12.3,4.0,1.0,1.6,5.5,6.3
```

Appendix K

Predictions Database Example



```
concelho_id,freguesia_id,tipologia,estado,tx_desemp,ano,trimestre,typ_boolean,pib_h,ihpc_h,fbcf_h,cons_priv_h,cons_pub_h,export_h,import_h,tempo_continuo
110800,110805,2,1,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,7,1,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,6,1,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,4,1,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,1,1,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,5,1,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,0,0,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,3,0,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,2,0,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,7,0,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,6,0,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,4,0,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,1,0,6.4,2025,4,0,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
110800,110805,5,0,6.4,2025,4,1,1.2890768300387396,1.2570276997667984,1.6553297627751458,1.3228844308404186,1.1228355346298797,1.7358764408874414,1.8838606898718904,50
30300,30349,0,1,6.4,2026,1,1,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,3,1,6.4,2026,1,0,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,2,1,6.4,2026,1,0,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,7,1,6.4,2026,1,1,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,6,1,6.4,2026,1,1,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,4,1,6.4,2026,1,0,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,1,1,6.4,2026,1,0,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,5,1,6.4,2026,1,1,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,0,0,6.4,2026,1,1,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,3,0,6.4,2026,1,0,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,2,0,6.4,2026,1,0,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
30300,30349,7,0,6.4,2026,1,1,1.2834925334289558,1.2608295172283979,1.719719472048272,1.2882181362741998,1.134836382504658,1.6711768853116706,1.8426994566470523,51
```

Appendix L

Contribution to the Sustainable Development Goals

This dissertation project aligns with the principles and objectives of three Sustainable Development Goals (SDGs) identified by UNESCO as particularly relevant to engineering practice: SDG 9 (Industry, Innovation and Infrastructure), SDG 11 (Sustainable Cities and Communities), and SDG 8 (Decent Work and Economic Growth). These goals reflect the essence of the project, which merges data-driven innovation, technological integration, and decision support within the real estate domain.

The project contributes to SDG 9 by embedding advanced machine learning algorithms and automated data pipelines into the valuation process, modernizing a traditionally manual and experience-based system. By integrating transactional and macroeconomic data sources into a unified, scalable forecasting tool, the project enhances the quality and resilience of analytical infrastructure, in line with UNESCO's call for innovation-driven engineering.

In relation to SDG 11, the project promotes more transparent and informed housing market analyses by providing localized, quarterly predictions of price per square meter. These forecasts support better planning and accessibility assessments at the municipal and parish levels, which is crucial for developing inclusive housing policies and sustainable urban growth strategies.

Finally, the initiative supports SDG 8 by reducing the manual effort required from real estate consultants, who often rely on time-consuming and subjective methods. By automating valuation tasks and integrating results into an intuitive Power BI dashboard, the project empowers professionals to focus on higher-value insights and client engagement, contributing to a more efficient, rewarding and knowledge-based working environment.