TÉCNICO LISBOA

Real Estate Market Analysis For Investment and Buying Cycles Prediction

Maria Pereira Marques Quaresma Leitão

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Pedro Alexandre Simões dos Santos

Examination Committee

Chairperson: Prof. Nuno João Neves Mamede
Supervisor: Prof. Pedro Alexandre Simões dos Santos
Members of the Committee: Prof. José Alberto Rodrigues Pereira Sardinha

November 2021

Acknowledgments

First and foremost, I must express my gratitude to my parents, Sandra Leitão and Paulo Leitão, and my family, especially my aunt, Susana Leitão, my godmother, Elsa Ferreira, and my grandparents, Maria da Conceição, Margarida Rosa, and Raul Manuel, for always pushing and supporting me to be a better version of myself. Their love for me and their patience to accept my absences contributed extraordinarily to the successful completion of this project.

Also, I would like to thank my supervisors, Prof./Dr. Pedro Santos and Eng. Carlos Mesquita for giving me the opportunity to work on this project. Their knowledge and guidance throughout this last year were of most value not only for this work but also for my education.

At last but not in least, I am really grateful to my college friends, especially Gonçalo de Melo, Rui Nóbrega, and Carlos Sequeira. I would never have finished this project without their esteem and companionship. I will forever keep them in my heart. Thanks to Maria Osório, for always having a word of lightness when times were dark; to Miguel Ramos and João Lopes for knowing when I needed a break from work; to Lara Ramos for always being next door when I needed; and to Matilde Pereira and Ricardo Bandeira for being in this journey with me since day one. Also, my special thanks to my best friends, Ana Ventaneira, Ana Serrano, and Inês Fole, who kept sending their love and encouragement, even far away, and have been doing so, for as long as I remember to exist.

With all my heart, thank you all.

Abstract

The traditional methods used to estimate Real Estate prices are sometimes too subjective and lack accuracy. The most common approaches to calculate a property's price are the Market Approach, the Income Approach, and the Cost Approach. Artificial Intelligence is applied to house price prediction, and Machine Learning models are developed and tested to research the best algorithm to achieve better accuracy results to overcome the subjectivity carried by these methods. This project provides some background on how the Real Estate market functions and how some State-of-the-Art solutions address the industry's requirement of Artificial Intelligence. It also describes a few experiments on several algorithms to understand how adequate they are in the scope of the problem, either by trying to achieve a precise duplicate detection model or by aiming to develop a model capable of computing Real Estate values. In those experiments it was found that it is of value to separate the dataset by location, creating subsets of data. Also, from the several algorithms tested, the majority of subsets achieved better results with Random Forest and Gradient Boosting.

Keywords: Real Estate, Artificial Intelligence, Machine Learning

Resumo

Os métodos tradicionais que são utilizados para calcular preços de imóveis são por vezes demasiado subjectivos e pouco precisos. Normalmente, as abordagens que são mais comuns para calcular o preço de um imóvel são o Método Comparativo de Mercado, o Método do Rendimento e o Método do Custo. A Inteligência Artificial é aplicada à previsão dos preços dos imóveis, e são desenvolvidos e testados modelos de Machine Learning para encontrar o algoritmo mais adequado que alcance melhores resultados e que elimine a subjectividade associada a estes métodos. Este projecto começa por fornecer algum conhecimento sobre o funcionamento do mercado imobiliário e como algumas soluções de Estado da Arte respondem aos requisitos da indústria de Inteligência Artificial. Descreve também algumas experiências em vários algoritmos para compreender quão adequados são no âmbito do problema, quer tentando alcançar um modelo de detecção de duplicados, quer visando desenvolver um modelo que seja capaz de calcular valores imobiliários. Nessas experiências verificou-se que vale a pena separar os dados por localidade, criando subconjuntos de dados. Além disso, dos vários algoritmos usados, a maioria dos modelos obteve melhores resultados com Random Forest e Gradient Boosting.

Keywords: Mercado Imobiliário, Inteligência Artificial, Machine Learning

Contents

List of Tables xi

List of Figures xiii

1 Introduction 1
- 1.1 Understanding Real Estate Market . . . 1
- 1.2 Real Estate and Artificial Intelligence . . . 3
- 1.3 Problem Statement . . . 4
- 1.4 Goals . . . 4
- 1.5 Document Organisation . . . 4

2 Background 5
- 2.1 Traditional Methods for Real Estate Appraisal . . . 5
- 2.2 Real Estate Market Segmentation . . . 6
- 2.3 Real Estate Appraisal using Machine Learning . . . 7
- 2.4 Data Collection and Pre-processing . . . 7
  - 2.4.1 One-hot Encoding and Ordinal Encoding . . . 7
  - 2.4.2 Data Normalisation . . . 8
  - 2.4.3 Dimensionality Reduction . . . 8
- 2.5 Finding Duplicates . . . 8
  - 2.5.1 Plagiarism Detection Techniques . . . 9
  - 2.5.2 Vector Space Models . . . 9
- 2.6 Machine Learning Concepts . . . 10
  - 2.6.1 K-fold Cross Validation . . . 10
  - 2.6.2 Regression . . . 11
  - 2.6.3 Artificial Neural Networks . . . 11
  - 2.6.4 Decision Trees and Random Forests . . . 12
  - 2.6.5 Adaptive Boosting and Gradient Boosting . . . 12
  - 2.6.6 Support Vector Machine . . . 13
  - 2.6.7 Grid Search . . . 13
  - 2.6.8 Evaluation Metrics . . . 14

3 State of The Art 15
- 3.1 Real Estate Market Segmentation . . . 15
- 3.2 Data . . . 15
  - 3.2.1 Data Collection . . . 16
  - 3.2.2 Data Exploration . . . 16
  - 3.2.3 Data Preprocessing . . . 16

Missing Values . . . 16
Outliers . . . 16
Data Normalisation . . . 17
Variable Dummification . . . 17
3.3 Finding Duplicates . . . 17
Plagiarism Detection Techniques . . . 18
3.4 Experiments With Algorithms Used In This Project . . . 18
3.4.1 Linear Regression . . . 18
3.4.2 Artificial Neural Networks . . . 18
3.4.3 Random Forest . . . 19
3.4.4 Adaptive Boosting and Gradient Boosting . . . 19
3.4.5 Support Vector Machines . . . 19
3.4.6 Comparing Regression Models with Artificial Neural Networks . . . 19
3.4.7 Results and Discussion . . . 22

4 Development **25**
4.1 Data . . . 25
4.1.1 Data Collection . . . 25
4.1.2 Variables . . . 25
4.1.3 Data Exploration . . . 27
4.2 Methodology . . . 28
4.2.1 Data Preprocessing . . . 28
Missing Values . . . 28
Outliers . . . 29
Dimensionality Reduction . . . 31
Duplicates . . . 31
Ordinal and One-hot Encoding . . . 33
Data Normalisation . . . 34
4.2.2 Training . . . 34
Linear Regression . . . 35
Artificial Neural Networks . . . 35
Random Forest . . . 35
Boosting Algorithms . . . 36
Support Vector Machines . . . 36

5 Results **37**
5.1 Duplicates . . . 37
5.2 Linear Regression . . . 37
5.3 Artificial Neural Networks . . . 38
5.4 Random Forest . . . 39
5.5 Adaptive Boosting . . . 40
5.6 Gradient Boosting . . . 41
5.7 Extreme Gradient Boosting . . . 42
5.8 Support Vector Regression . . . 43
5.9 Evaluation . . . 44

6 Conclusion **47**
6.1 Contributions 47
6.2 Future Work 48

Bibliography **49**

A Data Exploration **55**

List of Tables

3.1 Predictive ability of the Hedonic Pricing Model and the Artificial Neural Network. 20
3.2 Valuation accuracy of the HPM and the ANN models prediction. 21
3.3 Evaluation metrics' results from the two best Networks. 21
3.4 Evaluation metrics' results from the Regression Pricing Model. 22

4.1 Description and type of values of each variable in the data. 26
4.2 Number of instances by municipality. 34

5.1 Linear Regression Results. 38
5.2 Artificial Neural Networks Grid Search Results. 39
5.3 Random Forest Grid Search Results. 40
5.4 Adaptive Boosting Grid Search Results. 41
5.5 Gradient Boosting Grid Search Results. 42
5.6 Extreme Gradient Boosting Grid Search Results. 43
5.7 Support Vector Regression Grid Search Results. 44
5.8 Overview of MAPE (%) values for every algorithm tested. 45

List of Figures

1.1 Phases of the Real Estate Cycle [1]. 2

2.1 BERT input representation [2]. 10
2.2 Example of 4-fold cross validation being applied to a dataset of people with or without a disease (black and grey icons, respectively) [3]. 10
2.3 A model designed to separate data into two categories as in image A, might be overfitted to the data (B) or underfitted (C) [3]. 11
2.4 On the left, a single neuron, and, on the right, the architecture of an Artificial Neural Network with one input layer, three hidden layers, and one output layer [4, 5]. 12

3.1 Estimated property prices [6]. 22

4.1 Price by area per municipality. 27
4.2 Price by area per municipality. 28
4.3 Merging data workflow. 29
4.4 Relation between Area, Price and Property Type before (a)(b) and after (c)(d) removing outliers. 30
4.5 Relation between Area, Price and Number of Rooms (depicted by color) after removing outliers. 31
4.6 Correlation Matrix. 32
4.7 Correlation Matrix of fundamental features. 32
4.8 Data pre-processing workflow. 34
4.9 Work flow model. 35

A.1 Area-Price property type before removing outliers. 56
A.2 Area-Price property type after removing outliers. 56
A.3 Area-Price number of rooms before removing outliers. 57
A.4 Area-Price number of rooms after removing outliers. 57
A.5 Area-Price number of bathrooms before removing outliers. 58
A.6 Area-Price number of bathrooms after removing outliers. 58
A.7 Area-Price energy certificate before removing outliers. 59
A.8 Area-Price energy certificate after removing outliers. 59
A.9 Area-Price condition before removing outliers. 60
A.10 Area-Price condition after removing outliers. 60

Chapter 1

Introduction

The term Real Estate denotes real, or physical, property. It refers to the property, land, buildings, air rights above the land, and underground rights below the land [7].

Real Estate is closely intertwined with human well-being. Since one of the basic human needs is shelter, we need protection from blazing sun, freezing temperatures, wind, and rain. Without this protection human skin and organs are damaged from extreme temperatures [8]. Therefore, houses are seen as the goods that satisfy that human necessity. In other words, the Real Estate Market arises as the industry responsible for selling and buying of those goods. Thus, people will always need a house to live, and that is why the Real Estate will always be a valued market.

The Real Estate Market is focused on pricing and a cyclic market. When it comes to investing in a property, one must be aware of the prices practiced for that type of property, as well as in which phase of the cycle the market is at the moment. However, it is not always straightforward where we are exactly in the cycle at any given time.

1.1 Understanding Real Estate Market

The four types of Real Estate encompass Residential Real Estate, Commercial Real Estate, Industrial Real Estate, and Land. Residential Real Estate is about properties used for residential purposes, such as single-family homes, condos, cooperatives, duplexes, townhouses, and multifamily residences with fewer than five individual units. Commercial Real Estate includes Real Estate used for commerce and service supply, such as shopping centres and strip malls, medical and educational buildings, hotels and offices. Industrial Real Estate includes buildings that are used for research, production, storage, and distribution of goods, such as manufacturing buildings and property, and warehouses. Nonetheless, some buildings that distribute goods are considered Commercial Real Estate. It matters to classify each type of Real Estate because the zoning, construction, and sales are handled differently. Lastly, we have Land, which refers to vacant land, working farms, and ranches.

There are three types of Real Estate Markets [9] – a buyer's market, a seller's market and a balanced market –, and four phases in the Real Estate Cycle [10, 11, 12] – Recovery, Expansion, Hyper Supply and Recession. On the one hand, a buyer's market gives advantage to buyers, because the housing supply will outweigh the demand, giving buyers more negotiating power when making a purchase. On the other hand, a seller's market will be more convenient for sellers, as there will be less properties avail-

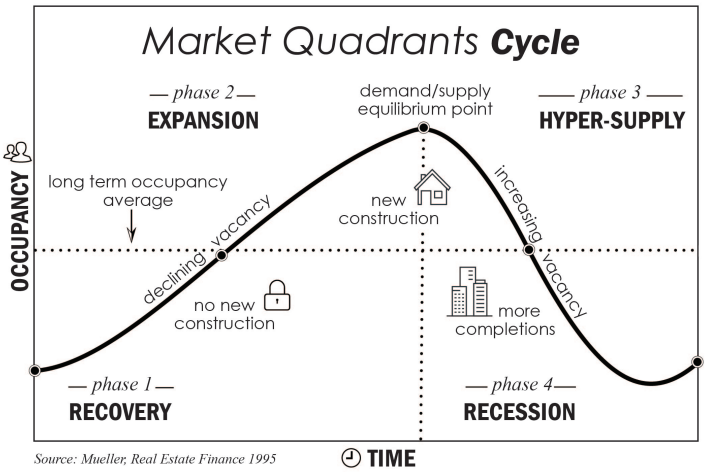


Figure 1.1: Phases of the Real Estate Cycle [1].

able to sell, and buyers will be competing amongst themselves. A balanced market will have a demand and supply, as the number of homes to sell will be keeping up with demand. Concerning the phases in the Real Estate Cycle, the Recovery phase happens right after a Recession and it is when there is a slowly growing demand for housing and high (but decreasing) vacancy rates, making the real estate price to be slowly rising. Next, the Expansion stage will take place, when the market is recovering, and housing has a higher demand due to the economy. Both Recovery and Expansion phases will be more advantageous to sellers, making the market at these times a Seller's Market. Hyper Supply will then take over and the balanced demand and supply will be replaced by an oversupply, where there will be more houses for sale than the market demand, making the prices lower. When Recession comes it will be the result of an over-inflated growth, and it will entail a decline in prices, since housing demand will be reduced and supply increased. At this time Real Estate price will be at its lowest, making this the best time to invest. This way, Hyper Supply and Recession phases enable a Buyer's Market. The balanced market usually takes place for a short period of time at the turning point between a seller's and a buyer's market, which will normally be between Expansion and Hyper Supply and then between Recession and Recovery.

There are several factors responsible for the variations in the Real Estate Market. The economic factors that affect the Real Estate investment strategies include macroeconomics, microeconomics, business and local factors, economic development cycles, foreign economic activity, economic globalization and national economic policy factors.

The influence of macroeconomics factors like the Gross National Product (GNP), can be observed through indicators such as incomes, either by rents or direct employment, investments, space for living and producing, housing demand and supply [13]. The construction sector and real estate dealings also have a large share of the Gross Domestic Product (GDP) of a country, and they can directly and indirectly affect each other. For example, in 2018, the Real Estate transactions represented 12% of the Portuguese GDP, contributing €24.1 billion to our nation's economic output [14].

Along with economic factors there are political and social factors, environmental and scientific factors, which can also entail variations in the Real Estate market [15]. For example, the demographics of certain regions might influence demand. Changes in income or children growing older and moving out may cause that population to want to relocate [16].

1.2 Real Estate and Artificial Intelligence

Like many others, the Real Estate sector is adapting to a data-driven world by defining use cases for Artificial Intelligence. Purchasing a house involves a huge investment and therefore a huge concern. The classical methods to evaluate the value of a property, as discussed in Section 2.1, are subjective and do not provide the level of accuracy buyers and sellers are looking for. But Artificial Intelligence is well on the way to do that.

As big data's potential keeps growing, companies need to incorporate analytics into their strategic vision, so they make better and faster decisions. Artificial Intelligence is already making some changes in the sector. At this point, it matters to introduce the concept of PropTech, which stands for Property Technology, and is a software developed for the property industry [17]. Some Commercial Real Estate firms are now applying PropTechs across their functions or even developing their own. There is also a rising number of firms who choose to invest directly in PropTech firms and start-ups. The main focus of Commercial Real Estate firms is on improving business intelligence based on enhanced analytics and on eliminating inefficiencies by strategically applying Automation, Artificial Intelligence and Machine Learning.

According to the Commercial Real Estate Innovation Report, presented by Altus Group in 2019 [18], the industry of Real Estate is already making investments to benefit from the emerging fields of Data Science, such as Artificial Intelligence and automation. Commercial Real Estate firms are now actively applying technology across a broad range of business functions, instead of keep questioning its benefits. It is clear that PropTech adoption is accelerating and that innovation is an opportunity rather than a cost. Having a clear technology and data strategy will help enable firms to more quickly respond to increasing market pressures. Back in 2019, "Scenario and Sensitivity Analysis" was already the area where Artificial Intelligence and Machine Learning were being applied the most. The purpose of this analysis is to evaluate a property's financial and operational performance to make the most reasonable real estate investment decision, which often involves dealing with a lot of data.

In 2020, Altus Group released another Commercial Real Estate Innovation Report [19], which supported the observations of 2019. It states that the industry of Commercial Real Estate has reached a tipping point on how it looks at emerging technology, and leaders are showing clear signs of taking a more forward-looking view on its impact. The technologies with high potential for significant cost savings and operational efficiencies are aimed to analytics and automation. From 2019 to 2020, the percentage of Commercial Real Estate Executives that believed Artificial Intelligence and Machine Learning have the potential for significant cost savings and operational efficiencies increased from 36% to 46%, and the percentage that believe they will create major disruptive impact augmented from 27% to 43%. This is proof of how Artificial Intelligence and Machine Learning developments are conquering the market of Real Estate and convincing the companies to adopt such technologies.

There are already a few concrete examples of companies that provide services based on Artificial Intelligence [20]. Localize [21], for example, uses AI and Machine Learning to provide accurate, useful information about a property to clients, by making advanced property analysis based on preferences. Others are improving lead generation and content marketing through customer data collection with AI-enabled consumer apps, Machine Learning interfaces and chatbots, like Convoboss [22] or Realty Chatbot [23] that automatically respond to buyer and seller leads when the real estate agent is not available at the moment. There are also companies like Skyline AI [24] and Zillow [25], which analyse patterns in

vast amounts of data, to predict the future value of a property.

In Portugal, we have Keezag [26], which, among other functionalities, predicts the fair price of a property, Reatia [27] and Casafari [28] which also do perform property analysis, and additionally present Market Analytics, using Machine Learning and Natural Language Processing.

1.3 Problem Statement

The current Real Estate industry has a high demand for an easy-operate and logical scientific price prediction model. However, the Real Estate development trend is cumbersome and cannot be forecasted accurately. Many facts such as human behaviour, mentality, decision and so on are involved in the Real Estate system. Most of the aforesaid facts are random and un-quantized, which makes it difficult to predict real estate prices [29]. Nonetheless, even if it is impossible to predict social and political factors using mathematical models, it might be feasible to introduce such predictions based on non-mathematical analysis of govern behaviour.

As we will see on Section 3, there are already some studies dedicated to create prediction tools based on Machine Learning. They are focused on experimenting and understanding which algorithms perform better in predicting Real Estate values, but their datasets are considerably small. That is why this project intends to create a model that can improve Real Estate Price prediction by using a substantial amount of data.

1.4 Goals

Most literature in this field of study performs an analysis of algorithms to predict Real Estate prices. As an improvement, this project is directed, not only to predict Real Estate prices, but also to find an adequate strategy to detect duplicates that are not so obvious in the dataset.

To achieve this purpose it was developed a tool able to predict the fair price of a property given its attributes, and a model capable to compare data entries and evaluate whether they are duplicates or not. All this using a dataset containing the characteristics of some properties in Lisbon and Setúbal, Portugal.

The outcome of this work comprises a dataset with properties from Lisbon and Setúbal, a crawling mechanism capable of continuing the data extraction to keep increasing the aforementioned dataset, and a prediction model to compute the Real Estate prices.

1.5 Document Organisation

This thesis is organised as follows. In the next section, the Background, some descriptions of the traditional ways in which the Real Estate market performs its property evaluations are presented, as well as scientific knowledge regarding the application of Artificial Intelligence to Real Estate. Next, in the State-of-the-Art section, some recent studies of this context about several algorithms and their results are analysed. In the following chapters, the Development addresses all the experiments performed as well as their results, and in the Evaluation we can find an analysis of those results. Finally, the Conclusion provides us an overview of the project and how our goals were met or what could have been done differently.

Chapter 2

Background

This section intends to provide some background not only to scientific concepts to be used in the Development section, but also some insights from how Real Estate Appraisal works in its classical ways.

2.1 Traditional Methods for Real Estate Appraisal

Real Estate appraisers are licensed professionals who estimate values of properties using standardized procedures, their knowledge of the real estate market, and appropriate amount of complete and accurate data [30]. Their work is more often supported by Automated Valuation Models and Computer Assisted Mass Appraisal, computer systems which employ various methods and models for single and mass appraisal properties [30, 31, 32]. As of now, the most common method for a Real Estate appraiser to determine the value of a property is still the Market Approach, which means determining the value of an asset based on the selling price of similar assets. The Market Approach analyses past and recent sales of similar properties, making adjustments for the differences between them, such as the property's area, the age and geographical location of the building, its amenities and for how long the property was listed for sale. It assumes that a well-informed buyer would not pay for a property more than the acquisition cost of a property of the same type that is also available on the market [33].

However, for a human it is impossible to consider all the different parameters on the property costs in detail. Therefore, it is needed a device which comprehends these patterns and impact of different parameters on the property costs [34]. More than that, when the properties are similar yet not identical in their characteristics (those being either qualitative or quantitative variables), in order to retrieve as much as possible the comparable elements of the object of appraisal, the appraiser must recur to the process of correcting or approving the properties considered similar, sometimes using logic-mathematics expressions, and often simply using the empirical descriptions (better, much better, really better, slightly worse, worse, much worse, really worse, etc.) translated by the same appraiser to the numbers (usually with percentage), especially regarding qualitative variables [35].

This approach bears a high level of subjectivity arisen from the way appraisers use their criteria for approval of similar properties found, since different correction considerations might be taken in a series variable, especially when talking about qualitative fields. Furthermore, a vast majority of appraisers does not even verify their choices on similar properties after completing the procedure. This subjectivity when conducting the Market Approach can surely affect the accuracy of calculating the value of a property [35]. Hence, a model that can eliminate subjectivity of the appraisers and foresee the future property

estimations with more noteworthy precision and reduced subjectivity is required [34, 36].

Besides the Market Approach, there are three other procedures to estimate the price of a property: the Income Approach, the Cost Approach, and the Hedonic Pricing Model.

The Income Approach focus on assessing Real Estate prices from the relation between Real Estate value and investment income. Real Estate investors purchase properties not for their own usage, but for the return on investment. In this scenario, the money paid to buy Real Estate is the investment capital and the Real Estate net income generated by that capital is the investment gains [37].

Theoretically, the principle of the Income Approach defines the purchase of a useful period of Real Estate as the period in which net income stream achieved is reduced at the appropriate interest rate (or discount rate) into the sum of the present value. According to this principle, the Real Estate price (P) is defined as

$$P = \frac{a_1}{1+r_1} + \frac{a_2}{(1+r_1)(1+r_2)} + ... + \frac{a_n}{(1+r_1)(1+r_2)...(1+r_n)}, \tag{2.1}$$

where a_1 to a_n express the income of the property in the future and r_1 to r_n represent the reduction of interest rates for the next years, concerning the Real Estate life span 'n' [37].

Lastly, the Cost Approach calculates the Real Estate price (P) by subtracting the loss of value of a property, that is, its depreciation (D), to the construction cost (C_c), without forgetting how much the land costs (L_c), as defined in

$$P = C_c - D + L_c, \tag{2.2}$$

It assumes that a potential buyer should pay for a property the equivalent price of building a similar one from scratch and with the same level of utility [38, 39]. This approach carries the limitations of assuming there is always vacant land to build a new property, which might not be the case. Also, it can be too subjective to estimate the value of depreciation of older properties, because of the many factors that need to be taken into account, and, for example, the construction materials may not be available anymore.

There is still another way to estimate the price of a product. The Hedonic Pricing Method identifies an asset's price based on the premise that it is determined both by internal characteristics of the asset and external factors affecting it.

To apply the Hedonic Pricing Model, it is required a strong degree of statistical expertise and model specification, following a period of data collection.

In the Real Estate Market, the Hedonic Pricing Method determines the price of a building or land, taking into account the characteristics of the property itself (internal factors like size, appearance, condition, etc.), and characteristics of the surrounding environment (external factors such as the crime rate of the neighbourhood, access to schools and city centre, etc.) [40].

2.2 Real Estate Market Segmentation

Researchers from [41] describe market segmentation methods as approaches for dealing with spatial heterogeneity. This can be applied to the Real Estate Market by representing the housing market into distinct submarkets. The main goal of this procedure is to segment the market in such way that allows for accurate estimates of house values.

2.3 Real Estate Appraisal using Machine Learning

The first attempt to improve market methodology came up in the early '80s, with the massive accessibility to computer statistical packages, consisting of Multiple Regression techniques [36], as an attempt to eradicate the subjectivity in the valuation process described before.

Another attempt to deal with that subjectivity is the implementation of Artificial Intelligence [42]. In the scope of Artificial Intelligence, we have Machine Learning, where the machine will predict results by learning how much importance a particular event may have on the entire system based on its pre-loaded data [43]. By replicating the functioning of the brain, Artificial Neural Networks are considered a powerful data modelling tool. Given their ability of self-learning they can approximate nonlinear functions with any precision, through training, learning and generalization [29, 36].

Moreover, Artificial Neural Networks are now regularly used in residential property valuation [44], analysing relations among economic and financial phenomena, forecasting, data filtration, generating time series and optimization [45].

Nonetheless, to train any model one needs a dataset. And in case the needed dataset does not exist already, the data must be collected before. After having a substantial amount of data, it should be processed in order to train the model as required.

2.4 Data Collection and Pre-processing

Data collection is defined as the activity of collecting information that can be used to find out about a particular subject. One possible way to collect data from websites, in order to create a database are web crawlers.

A web crawler is a bot that gathers explicit hyperlinks and HTML content from different websites. Additionally, a web scraper is responsible for extricating data from those websites. The latter is mostly centered around changing unstructured information on the web into organised information, such as a database [46].

Tchuente et al. [41] emphasise the importance of Data Pre-processing. Since we are dealing with real-world data, we have to take into account that data may be impure, incomplete, noisy, and inconsistent. They highlight that these factors might difficult the task of finding useful patterns, after all high-performance mining systems require high-quality data, and accurate data yield high-quality patterns.

In order to provide better and more accurate models, datasets need to be previously processed and organised, using some techniques that are presented in the following subsections.

2.4.1 One-hot Encoding and Ordinal Encoding

A categorical feature can take on of a limited number of values, each of which corresponds to a different category, while a binary one can take on either 0 for false or 1 for true.

Dummificating or one-hot encoding the variables , consists in transforming a categorical feature into several binary features, one for each possible category [25]. Nonetheless this procedure entails problems such as the curse-of-dimensionality, since it grows exponentially the number of features in the dataset [47].

When the categorical variables have an intrinsic order, one can apply ordinal encoding instead of one-hot encoding. In this case each categorical value will be transformed in a integer value. For example, let us consider the field "Energy Certificate", which has an intrinsic order taking N values from "A+", "A", "B",

to “F”. These categorical values can be converted to integer values by replacing the highest value (A+) to N and the lowest (F) to 1. This ordinal encoding process eliminates the curse-of-dimensionality issue and it requires less computer memory resource and computations [47], because we do not multiply the number of features.

Nonetheless, there are still features that cannot be replaced by integers. That is the case of the characteristic “City”, which can only be replaced by using one-hot encoding. For example, if this feature can take values like “Lisboa”, “Oeiras” and “Cascais”, one has to create a binary variable for each of those categories, which will be 1 if the property is located in that city and 0 otherwise. This implies that if a property is located in “Oeiras”, only the “Oeiras” variable will be set to 1, while the other will be 0.

2.4.2 Data Normalisation

Since data features have different scales, and Machine Learning models tend to perform better with data that has the same scale for each feature, Normalisation is a way of giving all features a uniform scale.

To normalise feature j, we first calculate the average μ_j and standard deviation of the feature σ_j, and replace each value of the feature, $x_{ij}|_{i=1}^{n}$ with

$$x_{ij} \leftarrow \frac{x_{ij} - \mu_j}{\sigma_j}, \tag{2.3}$$

Such normalisation is intended to make algorithms treat each feature equally rather than give more weight to certain features simply due to a difference in scale [25].

2.4.3 Dimensionality Reduction

The features in a dataset are sometimes correlated, which might result in noise in the dataset and get the model biased, since it will attribute significance to two (or more) variables, when, in fact, it could simply account one. One way to solve this is through Feature Selection. When analysing the correlation between pairs of variables, if the correlation is near to 1, we need only to keep one of those variables, in order to increase the variance between property features [25].

Principal Component Analysis is also a feature selection procedure that converts a set of features that may be linearly correlated into a set of principal components that are linearly uncorrelated, by seeking a r-dimensional basis that best captures the variance in the data. The direction with the largest projected variance is called the first principal component. The orthogonal direction that captures the second largest projected variance is called the second principal component, and so on [48].

2.5 Finding Duplicates

Duplicates are a reality in the dataset of this project, whether it is due to collection of data occurring through several months, or because a property might be associated with more than one real estate agency and end up in the website from several different sources.

For each house in the dataset, it exists the corresponding description text containing more information about that same house. It contains details on the location, places of proximity, and other key words that might help understanding whether the same house is present in the dataset more than once, for example being advertised by different Real Estate agencies. The description analysis and comparing

is in part similar to the process of Plagiarism Detection. Therefore, a research on Plagiarism Detection Techniques was made in order to explore techniques used in this context.

2.5.1 Plagiarism Detection Techniques

Plagiarism might be separated into two types: Textual Plagiarism and Source Code Plagiarism [49]. They concern, as their names indicate, the misappropriation of intellectual work in the form of text and code, respectively. For the purpose of this work, the focus will be on the techniques applied to Textual Plagiarism detection, due to its similarity to our duplicate issue.

There are String-based Methods, where documents are compared to the word or sentence level, usually using n-grams. These are capable to detect documents' overlapping relying on string matching.

Semantic-based Methods measure the similarity between words and their meanings, and then evaluate the probability of plagiarism by matching the various keywords of the text in the documents being compared. Usually, the semantic similarity between words and how documents are related is done using WordNet.

Cluster-based Methods use specific words (keywords) to find similar clusters between documents.

In Vector Space Models, documents are represented in a vector space. Instead of evaluating strings, the model extracts lexical and syntactic features and categorize them as tokens. In order to representing and comparing documents, different weighting schemes might be used, such as TF-IDF and TF-ISF, and similarity between the vectors, which are in fact documents, can be computed using Jaccard, Dice's, Overlap, Cosine, Euclidean and Manhattan distance.

2.5.2 Vector Space Models

Word embeddings represent words in a vector space, where each word has its own vector of real values and similar vectors represents words with the same meaning [47]. Word2Vec is widely used to learn word embedding and can translate a word into a numerical vector, without losing the word-level semantics [50].

One of the main learning algorithm in Word2Vec is the Bag-of-Words, that can turn a document in a vector, in which the features words, by attributing 1 if the word is present in the sentence or 0 otherwise.

A similar approach is the embedding of Term Frequency-Inverse Document Frequency

$$TF - IDF(t, d) = TF(t, d) * IDF(t), \tag{2.4}$$

TF-IDF is a weighting scheme for representing a document in a vector. This is similar to Bag-of-Words but instead of setting the word feature in the vector to 1 as the Bag-of-Words, it sets it to the TF-IDF score. This score comes from the product of Term Frequency

$$TF(t, d) = \frac{f_d(t)}{n}, \tag{2.5}$$

where $f_d(t)$ represents the frequency of term t in document d, and n refers to the number of words in document d, with Inverse Document Frequency

$$IDF(t) = log(\frac{N}{df_t}), \tag{2.6}$$

where N corresponds the total number of documents and df_t is the number of documents with term t. TF-IDF score has the purpose of measuring how relevant a certain term is to a document in a collection.

BERT is also a state-of-the-art model to learn word embeddings and it stands for Bidirectional Encoder Representation from Transformers, BERT, and is provided by Google [2]. This algorithm uses

masked language models to pre-train deep bidirectional representations from unlabeled text. In BERT, input embeddings are represented by the sum of token embeddings, segment embeddings and position embeddings, as shown in Figure 2.1.

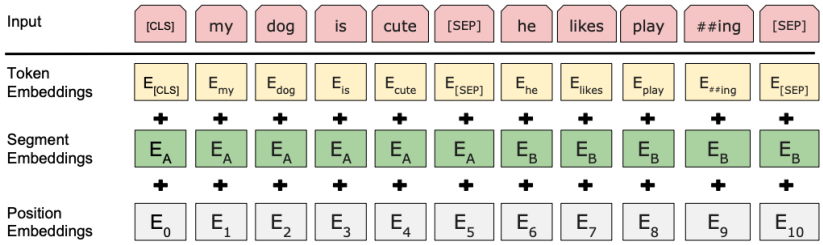


Figure 2.1: BERT input representation [2].

2.6 Machine Learning Concepts

For many years there have been developed several approaches to property valuation. In the following subsections are described some algorithms used to address problems in similar contexts.

2.6.1 K-fold Cross Validation

To understand if a predictive algorithm is working, it must be tested on a dataset. Instead of creating the predictive algorithm and then testing it on a new dataset, the procedure involves taking the initial dataset and partitioning it into a training dataset and a testing dataset.

The training set is composed of records used to build the algorithm responsible for making the predictions. Once the model is trained, the testing set is used to verify whether the predictions made by the algorithm are accurate or not. The independent variables are given to the model, and it must estimate the dependent value associated. The predicted values are then compared to the actual values present in the dependent variables in the testing set.

This process of partitioning the dataset may carry the cost reducing it. To avoid such problem, one might repeat the process k times (the number of folds), each time assigning a different group of record to the training set and to the testing set, as illustrated by Figure 2.2.

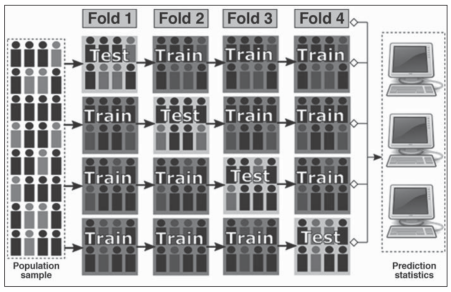


Figure 2.2: Example of 4-fold cross validation being applied to a dataset of people with or without a disease (black and grey icons, respectively) [3].

Each iteration will improve the performance of the model, by comparing between each training set's results to see what performs best and altering its overall predictive capability. Also, the model will provide more general results, since the algorithm was build on a broader group of records, reducing the risk of overfitting (being exceptionally good at predicting results on the training set, as Figure 2.3B depicts, but

less reliable in a general set of data, because the algorithm has learned the variations of the training set too well).

Figure 2.3: A model designed to separate data into two categories as in image A, might be overfitted to the data (B) or underfitted (C) [3].

2.6.2 Regression

Regression models are statistical methods for estimating the relationship between the output and the variables which have influence on the output, also referred to as influence parameters. Thus, is a way of calculating the relationship between the dependent and the independent variables.

The real output y is determined by the Regression model output $\hat{y}$, plus the error associated with the prediction,

$$y = \hat{y} + e, \tag{2.7}$$

The produced output $\hat{y}$ is influenced by the independent variables a_1 to a_n, according to the respective associated weights w_0 to w_n,

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n, \tag{2.8}$$

One way to improve the accuracy of a Regression model, is to perform the Stepwise Regression Method. Backward Stepwise Regression eliminates parameters from the model, whereas Forward Stepwise Regression considers adding new influence parameters into the model, one at a time. The overall adjusted R^2 is observed each time a new variable is introduced or removed from the model. When the adjusted R^2 is not increased by a variable, the latter is left out from the model [51]. The elimination of variables can also be executed according to the results of the F-test (significance level to stay). It is calculated the value of the F-test for each variable, from which the variable with smallest value will be eliminated. The Backward Elimination terminates when none of the F-test values is less than the critical value for elimination, meaning all remaining variables in the model meet the criterion to stay.

2.6.3 Artificial Neural Networks

An Artificial Neural Network (ANN) is a powerful tool for nonlinear problems inspired by the functioning of the brain [52]. A significant similarity is in the ability of ANN to learn and improve its performance [6].

The basic unit of the network is a simplified model of a biological neuron. Neurons in an Artificial Neural Network process the information with an activation function, as depicted on the left of Figure 2.4.

They are linked by oriented weighted connections and are organised into layers to transmit the information, like Figure 2.4 shows.

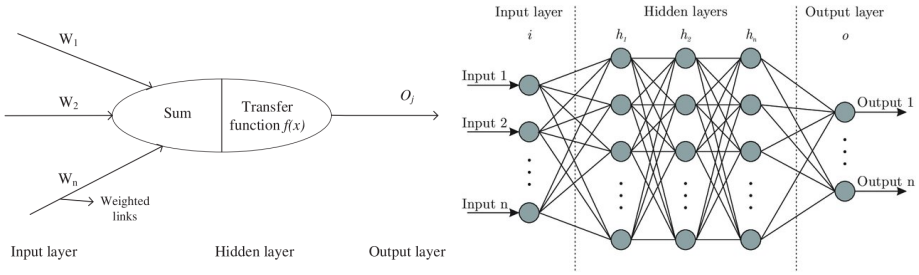


Figure 2.4: On the left, a single neuron, and, on the right, the architecture of an Artificial Neural Network with one input layer, three hidden layers, and one output layer [4, 5].

The three types of layers (input, hidden, and output) differ in the sources of their inputs and the use of their outputs.

The input layer receives and processes the data of the independent variables that are inputs to the ANN and then transmits them to the next network layer. The hidden layers process the outputs from previous layers (the input or other hidden layer) and transmits them to the next layer (other hidden layer or the output). The output layer then processes the outputs of the previous hidden layer and retrieves the value of the dependent variable as an output.

In a Feedforward Neural Network, all units in a layer pass their output to the next layer, until reaching the last layer and producing the output value.

ANNs learn and store acquired knowledge by adjusting the connection's weight values and neuron threshold values. When training Artificial Neural Networks, several learning rules can be used, as the next section describes.

2.6.4 Decision Trees and Random Forests

Decision Trees are defined as predictors that forecast the label associated with an instance x by crossing from a root node of a tree to a leaf [53]. By constructing an ensemble of trees, we can create a new classifier, where each tree is constructed by applying an algorithm A on the training set S and an additional random vector, θ, where θ is sampled i.i.d. from some distribution. This classifier is a Random Forest. The prediction of the random forest is then obtained by a majority vote over the predictions of the individual trees.

Although the Random Forest algorithm is an ensemble of many trees, the computational complexity of the ensemble is not as much as the computational complexity of all those trees together. The algorithm is actually very efficient, especially when the number of descriptors is very large [54].

2.6.5 Adaptive Boosting and Gradient Boosting

Adaptive Boosting uses a weak learner, a model that performs relatively poorly but is somewhat better than random guessing, and formulates a hypothesis with low empirical risk [53]. Usually one-level decision trees are used as weak learners, also known as decision stumps. The algorithm receives as input a set of examples and one labeling function, and performs the boosting process for a number of consecutive rounds. For each round, the booster composes a distribution on the examples given by the training set, which is received by the weak learner, together with the sample of examples. The weak

learner then produces independent and identically distributed examples according to the examples and distribution received, and returns a hypothesis considered “weak”, that is, with an accuracy barely above chance. Next, the algorithm computes the error of the hypothesis, and the inverse of such error will be the weight attributed to the aforementioned hypothesis. At the end of the round, AdaBoost updates the distribution to guarantee that examples on which the hypothesis failed will get a higher probability mass, compared to examples on which the hypothesis succeeded. By doing this procedure, on the next round the weak learner will focus on the problematic examples, and, after several rounds, the algorithm will produce a strong classifier that considers the weighted sum of all the weak hypotheses.

To sum up, AdaBoost focuses more on difficult to classify instances than on instances that are already handled well.

Gradient Boosting is very similar to Adaptive Boosting. It also builds the trees from a round based on the previous round, but instead of decision stumps, it may use trees that are deeper than a stump.

2.6.6 Support Vector Machine

Support Vector Machine is a classification method based on maximum margin linear discriminants, that is, the goal is to find the optimal hyper-plane that maximizes the gap or margin between the classes [48].

Although Support Vector Machine was created for classification purposes, its principle was extended to the task of regression and forecasting, leading to Support Vector Regression [55]. A Support Vector Regression estimates a function according to a given dataset $G = \{(x_i, y_i)\}^n$, being x_i and y_i the input vector and output value, respectively, and n denotes the total number of data. Considering the Linear Regression model, where the main goal is to minimize the sum of squared errors, the SVR also aims to find a function that covers the whole dataset, but is more flexible by allowing the model to fail within a certain margin, ϵ. Thus, the model intents to respect the constraint that the difference between the target value and the predicted value must be lower than the max error:

$$|y_i - w_i x_i| \leq \epsilon$$

, Nonetheless, we still need to account the possibility that errors are larger than ϵ, and for that purpose we introduce a slack variable. For values that fall outside the scope of ϵ, we denote its deviation from the margin as ξ. Taking into account this new margin, the goal still is to minimize the deviations as much as possible, being the new constraint:

$$|y_i - w_i x_i| \leq \epsilon + |\xi_i|$$

,

Finally, we might add another hyper-parameter, C, to decide on our tolerance for points outside of ϵ. Our tolerance will be as much as C defines, since as C increases, so does the model’s tolerance [56].

2.6.7 Grid Search

There are various parameters in a Machine Learning model that are not trained by the training set, the hyper-parameters. On the one hand, the learning rate of an Artificial Neural Network is a hyper-parameter, because it is defined before the training data is fed to the model. On the other hand, the weights are not hyper-parameters, since they are trained by the training set.

Hyper-parameters control the accuracy of the model, being particularly important in a data science project to find the hyper-parameters that yield better accuracy results.

Grid search is a technique used to compute the optimum values of hyper-parameters. It performs an exhaustive search through the specific parameter values of a model, until it finds the best values [57].

2.6.8 Evaluation Metrics

There are several evaluation measures available to assess the accuracy of Machine Learning models [4, 6]. However, the most adopted in related studies are the Coefficient of Determination (R^2),

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2}, \tag{2.9}$$

the Mean Absolute Error (MAE),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|, \tag{2.10}$$

and the Mean Absolute Percentage Error (MAPE),

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y_i}}{y_i}\right| * 100, \tag{2.11}$$

where n is the number of records, and, with y being the dependent variable of the data, for each record i, y_i is the actual value, $\hat{y_i}$ is the predicted value, and $\bar{y_i}$ is the mean value for the variable y.

A lower value of these measures demonstrates a higher accuracy of the model being evaluated, except for the Coefficient of Determination, whose value being closer to 1 means the model is well fitted to the data.

Chapter 3

State of The Art

So far, most of the work in the field of real estate valuation performs a comparative approach between Regression and Neural Networks, focusing on the contrast of error values calculated by both systems. In all cases Artificial Neural Networks take the lead with an error rate between 5% and 10%, compared to the Multiple Regression error rate between 10% and 15%. Although sometimes the error results are very similar, researchers agree that the Artificial Neural Networks are a system characterized by greater precision in contrast to the Multiple Regression [36].

In the following sections, there is an analysis performed on related literature, in order to explore what has been done before in similar works. It was explored how data was collected and pre-processed, how duplicates were treated, and which algorithms were used.

3.1 Real Estate Market Segmentation

Tchuente et al. [41] stated that Real Estate markets can be very different in each city, since political, economic, and geographic factors may vary between cities. Due to this circumstance, they aim to attain estimations of Real Estate based not only on prices per square meter, but most importantly on the Real Estate location, by analysing submarkets based on the cities in question. For example, the price per square meter is much higher in the economic and political capital, Paris, than in other regional cities. The same phenomenon happens in Portugal, where Lisbon can have a much higher price per square meter than the rest of the cities.

3.2 Data

Studies investigating the best approach to predict Real Estate prices use similar datasets, nonetheless they may differ in some characteristics and in the way the data was collected. For example, Nejad et al. [58] uses a dataset containing 3775 records of unit sales in 18 suburbs collected in a time span of seven months. There are seven features with basic information related to the property, including its address, which might be a plus, considering how much the location influences the price of a property. In another example, Pow et al. [56] uses a dataset with approximately 25000 examples and 130 features, being 60 of these features referent to socio-demographical factors.

3.2.1 Data Collection

The authors from [59], gathered their data using Call Detail Records, a procedure that consists in capturing the information on calls made by telephone systems, essentially by asking the person being called the information needed. In this case the CDR data was produced by Vodafone facilities in Budapest, Hungary, and it includes mobility entropy factors, like dweller entropy, dweller gyration, and dwellers' work distance, to be used as input variables .

On a different approach, [41] uses an open source dataset, provided by the French government, containing data from notarial acts and cadastral information on Real Estate transactions completed between 2015 and 2019. Having such data is of most value since it registers the actual price for which the houses were bought.

3.2.2 Data Exploration

Data from [41] enclosed Real Estate from French metropolitan territories and the French overseas departments and territories, with the exception of the Alsace-Moselle and Mayotte departments. However, the most significant portion of the transactions took place in the largest cities, so they chose to restrict the study to the ten largest French cities in terms of population.

When exploring the distribution of the dataset features, they chose to represent the number of transactions per city, sale type and residence type, then the price distribution per city, which allows to understand which city is the most expensive for Real Estate in France, and to conclude that the price distribution is relatively consistent according to the distributions of their populations, except for two cities that look unusually expensive seeing their sizes in terms of population, and one that, on the contrary, looks less expensive compared to its size in population. They also observed a reasonable number of outliers for all cities with very high prices, that are probably depicting luxury Real Estate, which they opted to remove to keep only the most common transactions, the ones that represent the majority of population. This kind of insights might also be particularly useful when analysing data from municipalities from Lisbon and Setúbal, since the same aspects might occur.

3.2.3 Data Preprocessing

The article from [41] selects relevant data by filtering only data from the nine cities in all the raw datasets, selecting only the valuable variables that are naturally related to the price of each transaction, and keeping only data relative to transactions of apartments and residential houses.

Missing Values

Regarding inconsistency of the data, [41] opted by simply removing all transactions with missing or bad values for postal codes, living area, and number of rooms (since they consider these are the features that most influence the target), as well as transactions with missing or bad values for prices (which is the target itself). This approach seems as the most adequate, since imputation of values would probably lead to more inconsistency.

Outliers

When considering the outlier removal, [41] removed all transactions with outliers in their prices for each city. The goal was to keep only the most common Real Estate transactions that represent the majority

of population, in order to avoid side effects. The method they used to find those outliers was through the interquartile range, where all values above the third quartile Q3 plus one half the interquartile were considered outliers.

Data Normalisation

All the numeric variables were standardized in [41, 59], meaning the data was rescaled to have a mean of zero and a standard deviation of one. This was due to many algorithms performing batter and more efficiently with standardized variables than with nonstandardized variables.

Variable Dummification

All discrete attributes in [41] were converted into Boolean dummy variables with zero or one for each of their values.

3.3 Finding Duplicates

Duplicates might be an issue when dealing with any dataset. They might compromise the performance of machine learning models either by inducing the model to believe that entry is worth more than the others, or, in case the target values are different, they might confuse the model.

Wang et al. investigated the best approach to detect duplicate questions in Stack Overflow by trying three deep learning approaches based on Word2Vec, Convolutional Neural Networks, Recurrent Neural Networks, Long Short-Term Memory [50]. Their dataset consisted of question pairs and the target value was a binary value indicating whether the pair was of duplicate or non-duplicate questions. The questions were transformed in vector representations of words, through word embeddings, and the vectors of all question pairs were fed into the deep learning models to train them. At the end, they concluded that deep learning performed better than their baseline approaches, which were based on similarity scores and overlapping of questions.

Considering that deep learning models have benefited from the target value stating whether the pairs were duplicates or not, it makes sense they perform better than a simple computation of similarities between questions. Although, when that target value does not exist yet, the similarities calculation might be of great help for targeting pairs of questions with human help.

On a similar issue with duplicates, this time dealing with defect reports at Sony Ericsson Mobile Communications, Runeson et al. experiment Natural Language Processing techniques to identify duplicate reports [60]. To evaluate if two reports are duplicates, they pre process the text applying tokenisation, stemming, and stop words removal. They find out that using a stop words list with 60 words produces better results than using a big list with 439 words or not removing stop words at all. Thereafter they represent the words of each report in a multi-dimensional vector space model, where each dimension of the space corresponds to a word, and the position along each axis in this space depends on the frequency of the word occuring in the text. After that, the similarity between two texts is measured by computing distances between vectors in the vector space. The similarity measures used were Cosine, Dice, and Jaccard, although the first performed better than the rest.

There was another relevant implementation in this experiment. From their investigation, they noticed the majory of duplicate reports were submitted within a time frame of 80 days. Hence they try on different time frames to detect if candidates are duplicates of certain record, and that reduces calculation costs.

Plagiarism Detection Techniques

Gupta et al. perform a study on plagiarism focusing on extrinsic text plagiarism detection, that is when documents are compared against a set of possible references [61, 49]. They start by pre-processing documents to keep only the relevant information, by applying sentence segmentation, tokenisation, stop word removal, and stemming. The next step is comparing the suspected document with large repositories or databases in order to retrieve near duplicate sources. For this task, the most common techniques are vector space models. After finding candidate documents, each suspicious document is intensively compared with its candidates using deep NLP techniques, as Part-of-Speech tagging as an example of syntax and semantic based techniques, Named Entity Recognition in the case of string based detection or vector space models.

3.4 Experiments With Algorithms Used In This Project

3.4.1 Linear Regression

Due to its simplicity and wide-spread use in the field of machine learning, Linear Regression models appear serving as the baseline model of some studies [41, 56, 62].

Sangani et al. [62] mentioned the relevance of one-hot encoding categorical variables, that is, turning these variables into binary ones. To test the effectiveness of normalization in Linear Regression models they train two LR models, one with normalised data and other with data that was not normalised. The latter achieved a lower value of MAE, meaning the normalization in this case was not beneficial, maybe due to ouliers, since no outlier treatment is mentioned in the preprocessing of this dataset. They also perform dimensionality reduction, by applying Principal Component Analysis to convert a set of features that may be linearly correlated into a set of principal components that are linearly uncorrelated.

In a comparison between a few machine learning models [30], the authors found Linear Regression performed poorly compared to Decision Trees and Artificial Neural Networks.

3.4.2 Artificial Neural Networks

Tchuente et al. [41] trained a variety of machine learning models, including Random Forest, Gradient Boosting and Adaptive Boosting, Linear Regression and Support Vector Regression, and Neural Networks with a Multi-layer Perceptron. From their experiments, the Neural Network model was considered the best model, for having the lowest value of MAE and RMSE among every model. Plus it has the highest value of R2, meaning it is more adequate to the data.

In another experiment, Pinter et al. [59] trained a Multi-layer Perceptron with one input layer, one hidden layer, and one output layer. Three different models were trained with this architecture, one with ten neurons in the hidden layer, other with twelve, and another with fourteen. The model that attained a better performance, with a lower MSE, was the one with only ten neurons in the hidden layers, showing that a higher number of neurons does not necessarily mean a higher accuracy.

Another approach [43], shows us that it is also possible to use Neural Networks as an improvement of other models. Here the authors use Linear Regression, Random Forest, and Gradient Boosting to predict housing prices. Afterwards, they apply Neural Networks to compare all the predictions made by the aforementioned algorithms, and to compute them in order to return the most accurate result. In other words, Neural Networks are used here to increase the efficiency of the prediction model.

3.4.3 Random Forest

Tang et al. [63] use a Random Forest approach with decision trees as weak learners to predict housing prices based on ensemble learning. To achieve the optimal prediction model, their experiments aim to determine the ideal depth and number of base learning Decision Trees. They also try to determine the combination strategy for predicting house prices with different integration learning algorithms.

In [58], they test Random Forest, among other seven machine learning tree models, and conclude Random Forest is the best performing model in their experiment.

3.4.4 Adaptive Boosting and Gradient Boosting

Sangani et al. [62] experimented training five different models using Gradient Boosting. One was built through XGBoost and the rest was trained by the traditional Gradient Boosting algorithm. All five models outperformed the Linear Regression ones, which makes sense since the latter merely finds a line of best fit, whereas Gradient Boosting develops an ensemble of Decision Trees. Their results also show that using the LAD loss function, which sums absolute errors, resulted in a more accurate model than using the LS loss function, which sums the squares of absolute errors and, therefore, is more affected by outliers. Considering LAD outperformed LS, they deduce their dataset contains numerous outliers.

Furthermore, they also compared the performance of Gradient Boosting and XGBoosting by training ten models of each, taking advantage of the common parameters in both algorithms, the maximum depth of each decision tree and the learning rate at which the optimal splitting point at a tree node is found. They created ten different combinations of maximum depth and learning rate, and applied it both to Gradient Boosting and XGBoosting. Considering the ten experiments, the Gradient Boosting outperformed the XGBoosting in all but one, that is, it achieved a lower MAE.

Overall, the model that achieved the best performance was generated by Gradient Boosting using Grid Search, which coheres with the purpose of the latter: to find the optimal set of parameters to train the algorithm.

In another experiment, [58] evaluates the performance of eight tree models and conclude that Gradient Boosting and XGBoosting, with MAE of 0.06748 and 0.06749 respectively, outperform all models except Random Forests, with a MAE of 0.06123.

3.4.5 Support Vector Machines

Li et al. [55] applied a Support Vector Regression to forecast Real Estate prices in China. Their input values included disposable income, consumer price index, investment in real estate development, loan interest rates, and lagged real estate price, while the real estate price is used as output variable of the SVR. In [56], they not only applied a linear Support Vector Regression but also experimented the polynomial and Gaussian kernels for regression of target prices.

3.4.6 Comparing Regression Models with Artificial Neural Networks

This paper of 2018 [4] intended to compare the predictive accuracy in property valuation between a Hedonic Pricing Model and an Artificial Neural Network model. In order to achieve a reliable comparison between the two models, the same dataset was used to train both of them.

The used dataset included transaction data of residential properties in Lagos metropolis, Nigeria, between 2010 and 2016, as well as the information of structural attributes of those properties. In to-

Model	R^2	RMSE	MAE	MAPE
HPM	0.77	61,408,856	103,370,573	38.23%
ANN	0.81	28,492,514	41,814,564	15.94%

Table 3.1: Predictive ability of the Hedonic Pricing Model and the Artificial Neural Network.

tal, there were 321 property sales transactions, with eleven independent variables and one dependent variable (the property price).

To understand how the variables could be related, several tests were performed. To remove correlated variables, it was conducted the multicollinearity test, which revealed that all the variables were uncorrelated, except for the number of bathrooms and the number of toilets (with a correlation coefficient of 0.965). Therefore, the number of toilets variable was removed from the list of independent variables and the dataset ended up with only ten independent variables. Finally, the scatter plot approach was used to verify if there was a linear relationship between property prices and the independent variables, which showed that there was, in fact, a linear relationship between them. Hence, the Hedonic Pricing Model was developed using Linear Regression.

The development of the Artificial Neural Network implies determining the number of input neurons, hidden layers and hidden neurons, and the output neurons. Usually, there is only one input layer, and its number of neurons depends on the number of independent variables to be used in developing the model. Regarding the number of hidden layers, literature previous to this paper has proved to be sufficient to have only one hidden layer for predicting property prices. Nonetheless, as to then number of neurons to be included in the hidden layer, there is no consensus.

It was then developed an Artificial Neural Network with three layers: one input layer, one hidden layer, and one output layer. The learning algorithm adopted was the Backpropagation, which was commonly used in studies previous to this paper. To determine the number of neurons to be used in the hidden layer, grid search was used so that it was found the optimal network architecture that best fitted the data, using the default parameters of learning rate, stopping criteria, and weight decay. In the end, the Artificial Neural Network that best fitted the data was a three layered with eleven neurons in the input layer, five in the hidden layer, and one in the output.

For both models, the data was randomly divided into two parts: training set (80% of the dataset) and test set (20%), using a 10-fold cross-validation to validate the models.

The accuracy of the models was then assessed using established Evaluation Metrics: the Coefficient of Determination (R^2), the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), and the Mean Absolute Percentage Error (MAPE), whose values are displayed in Table 3.1. Here we see that the ANN model produced better values of all the Evaluation Metrics. Specifically in the Mean Absolute Percentage Error, we understand that the ANN produced a MAPE value of 15.94%, while the HPM had a MAPE value of 38.23%, more than twice the value of ANN. Meaning the ANN was able to perform two times better than HPM in predicting the properties' values.

Additionally, in order to assess how suitable each of the model is for property appraisal, it was established which percentage of the predicted property values had a margin of error between 0 and 10%, which can be consulted in Table 3.2. It states that 33.33% of the predictions of ANN are within the acceptable margin of error, while only 26.67% of the HPM predictions are in that same range. Confirming once more that the ANN performed better.

The findings of this paper show that the ANN model performed significantly better than the HPM approach in estimating property values. More specifically when using a three layered Artificial Neural

	Hedonic Pricing Model		Artificial Neural Network	
Accuracy Range	*Frequency*	*Percentage*	*Frequency*	*Percentage*
$\pm 0 - 10\%$	8	26.67%	10	33.33%
$\pm 11 - 19\%$	4	13.33%	13	43.33%
$> \pm 20\%$	18	60.00%	7	23.33%

Table 3.2: Valuation accuracy of the HPM and the ANN models prediction.

LA	Neurons	R^2	RMSE	MAE	MAPE
BR	10-20	0.9749	6241.133	3561.288	3.39%
BR	15-15	0.9704	6220.818	3665.597	3.58%

Table 3.3: Evaluation metrics' results from the two best Networks.

Network, with eleven neurons in the input layer, five in the hidden layer, and one in the ouput layer.

In another study, in 2020 [6], the authors had the objective of comparing the predictive ability of the automated valuation model using Artificial Neural Networks and the hedonic pricing model using the regression method. It intended to estimate market prices of sold properties of Nitra, Slovak Republic.

By performing an analysis of ANNs trained using the Levenberg-Marquart (LM), the Bayesian Regularization (BR), and the Scaled Conjugate Gradient (SCG) learning algorithms, it investigates which one of these may provide the best prediction accuracy of the ANN pricing model.

The dataset used comprised 711 properties, from which mislabelled properties that did not meet some of the search criteria, duplicate properties, and properties that did not include all monitored parameters were excluded. After the pre-processing of the data, 256 were obtained after the selection.

For the ANNs trained with LM and the SCG learning algorithms, the data were divided into a training (70% of the dataset), validation (15%), and test (15%) sets, while, for the ANNs trained with the BR learning algorithm it was divided into a training (85%) and test (15%) sets.

In total 60 Neural Networks were trained, 20 for each learning algorithm in study. For each of the 20, four of them contained only one hidden layer, while the other 16 had two hidden layers. The number of neurons in each hidden layer varied between 5, 10, 15, and 20. Meaning each of the four Neural Networks with one hidden layer experienced one of the four numbers of neurons, and each of the 16 Neural Networks with two hidden layers experienced one of the possible 16 pairs formed by permutations with repetition of the four numbers of neurons.

The metrics used to evaluate the pricing model included the Determination Coefficient (R^2), the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE).

The results obtained in the Evaluation Metrics show that Networks trained with the BR learning algorithm with two hidden layers, ten neurons in the first hidden layer, and 20 in the second, achieve the best values of (R^2), MAE, and MAPE, while the best value of RMSE is accomplished by the Network trained with the same learning algorithm and number of hidden layers, but with 15 neurons in both layers. These results can be consulted in Table 3.3 This distinction in the results is due to the higher sensitivity of RMSE to large deviations. From all the results obtained it was concluded that the BR algorithm performed significantly better that the LM and the SCG learning algorithms.

In order to have a comparison with Neural Networks' performance, it was developed a Regression pricing model using the Stepwise Regression Method with Backward Elimination. In the nine steps that

$$Price_i = 31722.79 - 6591.80\ Loc\_1_i - 16651.07\ Loc\_3_i - 14488.27\ Loc\_4_i + 7993.72\ Num\_rooms_i + 462.55\ Area_i + 4908.63\ Floor\_2_i + 16595.35\ Num\_storeys_i + 10900.49\ Elevator_i + 3554.06\ Cellar_i + 17489.21\ Cond\_1_i + 13370.53\ Cond\_2_i + 8618.77\ Parking\_1_i$$

took to create the model, variables kept being eliminated according to the F-test values, until the last step where no variable was eliminated from the model because the F-test values of all variables were greater than the critical value for elimination. The resulting model contained twelve variables and a constant, as illustrated by the following Equation.

The values obtained in the Evaluation Metrics are shown in Table 3.4. These demonstrate that the Regression pricing model performed significantly worse than the Neural Networks above mentioned.

R^2	RMSE	MAE	MAPE
0.7898	11454.91	8516.60	8.41%

Table 3.4: Evaluation metrics' results from the Regression Pricing Model.

To have an overview of how the two best Neural Networks and the Regression Pricing Model performed at estimating property prices, seven properties were selected from the test set and had their price predicted by each of the three models. The results are displayed in Table from Figure 3.1.

Actual market price EUR	ANN pricing model BR 10_20		ANN pricing model BR 15_15		Regression pricing model	
	Estimated price EUR	Residuals	Estimated price EUR	Residuals	Estimated price EUR	Residuals
101900.00	95716.64	-6183.36	95484.04	-6415.96	99432.76	2467.24
88400.00	103351.84	14951.84	104905.06	16505.06	102164.72	-13764.72
115000.00	116713.36	1713.36	115946.06	946.06	109295.56	5704.44
90000.00	85782.71	-4217.29	89159.76	-840.24	82800.58	7199.42
92900.00	93478.66	578.66	93524.76	624.76	93748.96	-848.96
96000.00	100241.91	4241.91	102498.06	6498.06	100927.01	-4927.01
82999.00	83900.15	901.15	95122.18	12123.18	93122.31	-10123.31

Figure 3.1: Estimated property prices [6].

3.4.7 Results and Discussion

The results exhibited ANNs models achieved a better predictive ability than the pricing model based on the regression method (Table 3.3 vs. Table 3.4). Additionally, from the three learning algorithms examined, the Bayesian Regularization accomplished the best results.

Based on this research, it may be concluded that an Artificial Neural Network using the Bayesian Regularization learning algorithm and two hidden layers is suitable for estimating the price of a property. Although, the dataset used in this experiment is considerably smaller than the one I intend to use in the project. This means the number of hidden layers and neurons, and the learning algorithm used, might not perform as good as they did in the described experiment.

The last paper analysed, also proved that Artificial Neural Networks perform significantly better than Regression approaches. Furthermore, comparing the two papers, we can see that the latter achieved better accuracy. This is probably due to its higher number of neurons and for using two hidden layers instead of only one. However, when the paper of 2020 [6] performed their experiments, they varied

the number of hidden layers and neurons in each hidden layer until the topology that provided best accuracy was found. They simply recognized the best topology by displaying all their results in a table and analysing them. This is not so practicable if we want to try a considerable number of different topologies. That is why the 2018 work [4] applied Grid Search to find the best architecture for the ANN.

Chapter 4

Development

The main goals of this project are to train a model capable of predicting the fair price of a property and to find a suitable technique that manages to detect duplicates of Greater Lisbon and Setúbal. Since there is no dataset available concerning the properties of these regions and their prices, it was raised the need to collect that data first.

The ideal dataset to train the model should have, for each property, the characteristics that influence its price, as well as its price fluctuations through time. This means that the collection of data should take place in a substantial time span. Reason for which a web crawler and a web scraper were developed during the month of October, so that when the time to train the model comes, there is a dataset containing data from November to, at least, January.

4.1 Data

4.1.1 Data Collection

A web crawler was developed to navigate Imovirtual, a Portuguese real estate website that comprises more than three hundred thousand offers from several real estate agencies or individual sellers. For each property, Imovirtual displays the characteristics, as well as a short description text, sometimes with more details than the characteristics fields themselves. The crawler will gather all the web pages containing property offers in the regions of Greater Lisbon and Setúbal, due to a higher amount of properties than the rest of the country, so that the scraper can then collect the information in those pages.

4.1.2 Variables

The fields collected for each property encompasses, among other characteristics, its typology, that is the number of rooms and bathrooms, its area, its city and province, the type of the offer (if the house is for sale or for rent) and its price.

The result of the information extraction process will be a dataset containing the properties and its characteristics along with the short description texts and the timestamp from when that information was collected. That means the same property will appear more than once, but always with different timestamps. It matters to keep the records of the same property so that a variation in the price can be detected. The fields collected for each property can be consulted in Table 4.1.

Variable	Meaning	Type	Value
Net Area	The actual occupied area not including unoccupied accessory areas such as corridors, stairways, toilet rooms, and closets.	Quantitative	Float number representing square meters.
Area range	Interval where the net area of the property is inserted.	Categorical	Categories representing intervals aparted 50 square meters from each other.
Gross area	The floor area within the inside perimeter of the exterior walls of the building under consideration.	Quantitative	Float number representing square meters.
Terrain area	Land area of the property.	Quantitative	Float number representing square meters.
Number of rooms	Number of rooms of the property.	Quantitative	Integer number.
Number of bathrooms	Number of bathrooms of the property.	Quantitative	Integer number.
Province	District where the property is located.	Categorical	String.
Subregion	County of the district where the property is located.	Categorical	String.
Parish	Parish of the county where the property is located.	Categorical	String.
Condition	Represents the condition of the property,	Categorical	Category with the values ”new”, ”used”, ”renovated”, etc.
Country	Country where the property is located.	Categorical	String.
Energy certificate	Energetic efficiency of a property.	Categorical	Category from ”A+” to ”F”.
Offer type	Whether the property is for sale or for rent.	Categorical	Category with the values ”buy” or ”rent”.
Property type	Type of house in the property.	Categorical	Category with the value ”apartment”, ”house”, ”farm”, etc.
Floor	In case it is an apartment, in which floor of the building it is located.	Quantitative	Integer number.
Year of Construction	The year in which the property was built.	Quantitative	Integer number.
Lift	Whether the building is equipped with a lift.	Categorical	Binary value.
Garage	Whether the property has a garage.	Categorical	Binary value.
Swimming Pool	Whether the property is equipped with a swimming pool.	Categorical	Binary value.
Price	Asking price of the property.	Quantitative	Float number representing Euros.

Table 4.1: Description and type of values of each variable in the data.

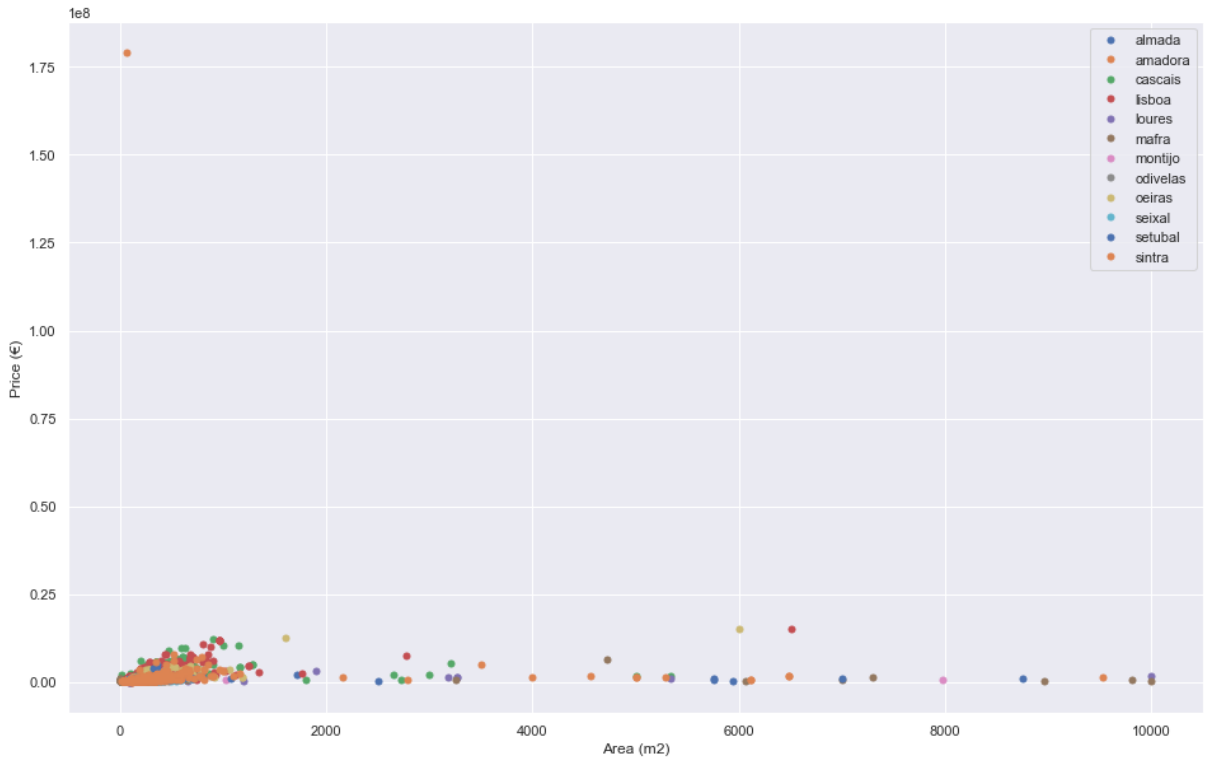


Figure 4.1: Price by area per municipality.

4.1.3 Data Exploration

Once data was gathered and before it was pre-processed, it was explored in order to understand what was relevant and what had to be changed or deleted, since there could be some entries with unreasonable values. At this point of the project, the data was displayed in some graphs, so that a relationship between the features could be noticed and absurd entries could be spotted.

There were, indeed, some entries that did not make sense in the context, indicating houses with an Area of 0, or some other illogical low value. This was probably due to the fact that each house in the dataset was at some point inserted in the Real Estate Portal Website by a person, who may or may not had been careless about whether the details he/she was entering were right or wrong. Or it could simply be a mistake, because that is normal since we are relying on humans to insert the data from each house. These outliers had to be spotted and discarded.

Furthermore, there were also a few entries with the price set to 0, or other values equally absurdly low, that were causing errors for example when trying to make a regression model out of that data, causing the Regression to predict some house prices as negative. These might be due to an error, as mentioned in the case of Areas, but on a more specific way, might be due to the fact that people do not really want to disclose the price of the house being announced or they are simply waiting for an offer. So they simply put some other value in the field of the house's price.

Comparatively to the entries above mentioned, with critically low Areas, that are probably errors, there were a few entries with an Area above 2000 square meters, as displayed by Figure 4.1, that are not errors, and are probably relative to houses with a higher terrain area. These entries also have an excessively lower price considering the extensive area, which may be due to their location being out of

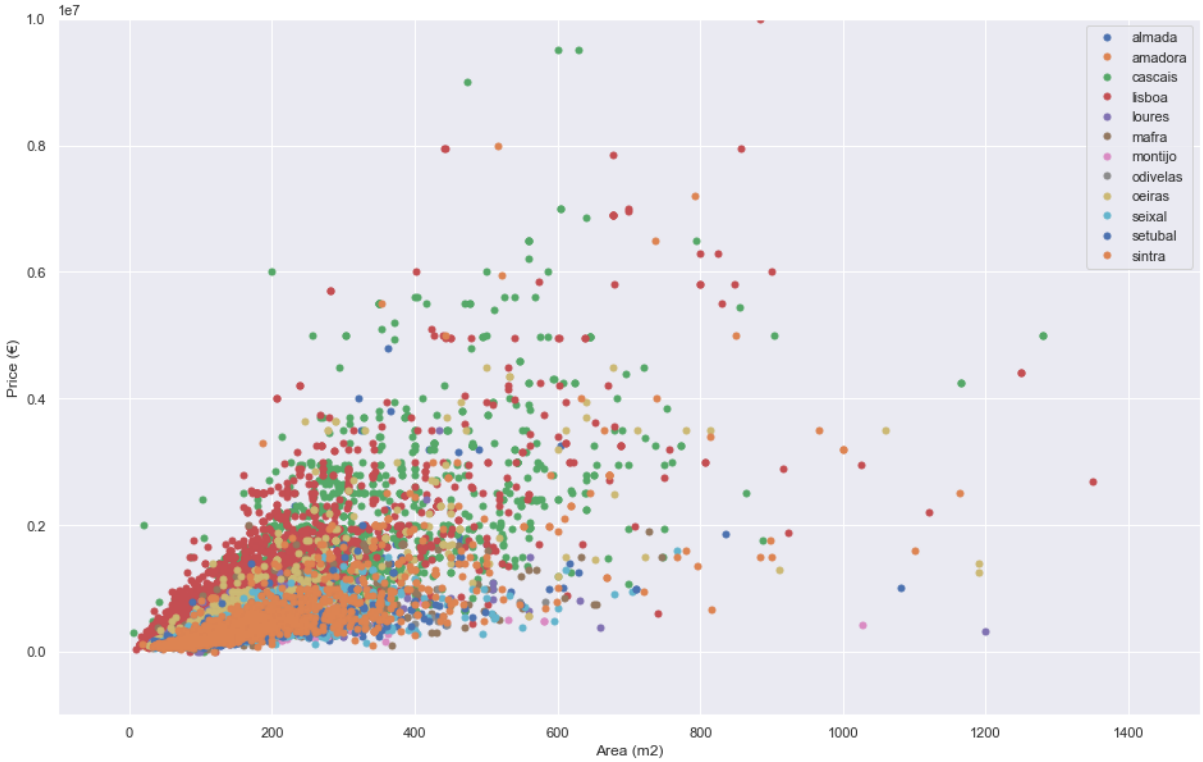


Figure 4.2: Price by area per municipality.

the city centre. Since their amount is not as significant as the rest of the houses with an area below 2000 square meters, and therefore are not part of the majority of houses present in the dataset, these entries will be ignored when training the models, so that the dataset is more consistent as shown in Figure 4.2.

Being the location one of the most important detail of a house, and probably the one that has the most influence on the price, data was separated by Municipality. Dealing with data from only two districts (Lisbon and Setúbal), I ended up with 12 graphs from the respective Municipalities (Almada, Amadora, Cascais, Lisboa, Loures, Mafra, Montijo, Odivelas, Oeiras, Seixal, Setúbal, Sintra). For each location, it could be observed the influence of a different characteristic one at a time. In the X-axis it could be found the Area, while the Y-axis represented the price. Then, each dot in the graph represented a house in the dataset, and colours were used to represent other characteristic, such as the property type, the condition of the house or the energy certificate.

4.2 Methodology

4.2.1 Data Preprocessing

Missing Values

Before deleting any data, it is intended to locate first the properties that are missing some entries, and to try and fill those entries by locating the concerning information in the short description text referent to the property.

The first point of this process was to transform each description on a list of tokens, removing stop-

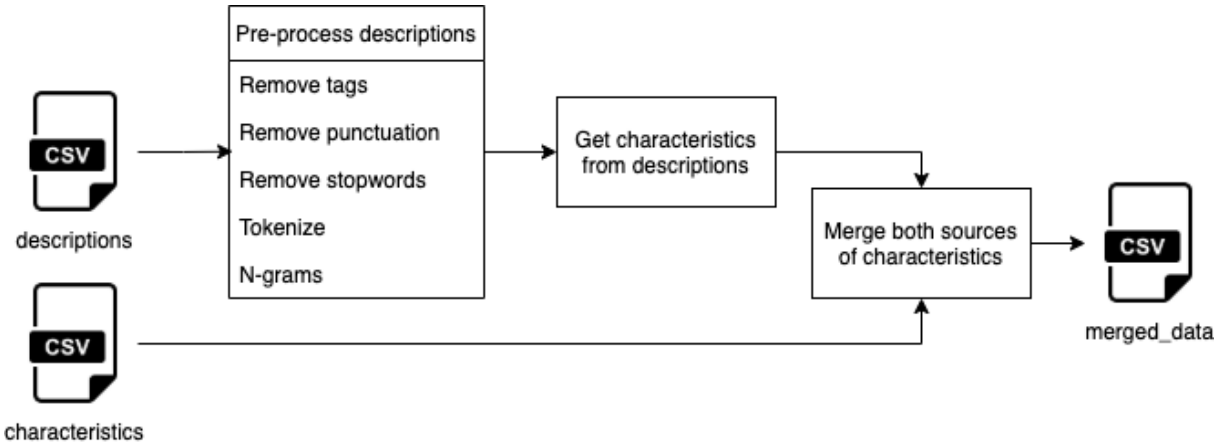


Figure 4.3: Merging data workflow.

words, special characters, punctuation, HTML tags that had been unintentionally extracted, and some adverbs that did not contribute to the house description. After removing what could be considered as noise in the text, we are left with a list of tokens, from which we can obtain also a list of bigrams, which can be useful to spot characteristics that have more than one word.

A new blank dataset is then created with the IDs of every house in the dataset, and with all the characteristics that can be extracted from descriptions set to 0. By going through every list of tokens and bigrams, we can check whether or not some details are present in those descriptions and fill those columns in the new dataset.

It is assumed that this strategy works, since it is supposed that when a house does not have a characteristic, that is not mentioned. For example, we look for the word ”pool” to find out whether or not the house has a pool, because no Real Estate advertise would write that the house does not have a pool. What might happen that might mislead this process, is the case where the advertise is actually mentioning a shared pool. In that case, we can only assure that the bigram ”shared pool” occurs and we do not consider the characteristic ”pool” but instead we consider the characteristic ”shared pool”.

This process is represented by the diagram in Figure 4.3.

If, after that procedure, a characteristic is still with a majority of missing values, it will probably be better to dismiss it, since it is not feasible to perform imputation of values because it might prejudice the accuracy of the model. For this purpose, if a feature has more than 90% of missing values, it was removed from the dataset.

Outliers

As mentioned in the Data Exploration Section, there were some entries which values did not made sense, such as Areas or Prices to 0 or very low values. These were simply eliminated, since they would compromise the performance of the model.

Most of the variables in the dataset are numeric and continuous, so an adequate way to find outliers is by performing z-score calculations. A z-score, or a standard score, measures how far from the mean a data point is. It represents how many standard deviations from the mean a data point is. These calculations are made by grouping the data by location, so that we do not risk considering an entry that looks

like an outlier for the whole dataset, but it makes sense in the region where it is placed.

A complete overview of the relation between a few variables and the effect of outliers' removal can be consulted in the Appendix A, but there are some cases worth mentioning in particular. Such as the situation illustrated by Figure 4.4, where one can observe that all entries that represented houses (moradia) were not significant compared to the apartments (apartamento) in municipalities of Amadora and Lisboa, and were considered ouliers by z-score calculations, which is coherent given that, in reality, both municipalities have a much higher offer of apartments rather than houses.

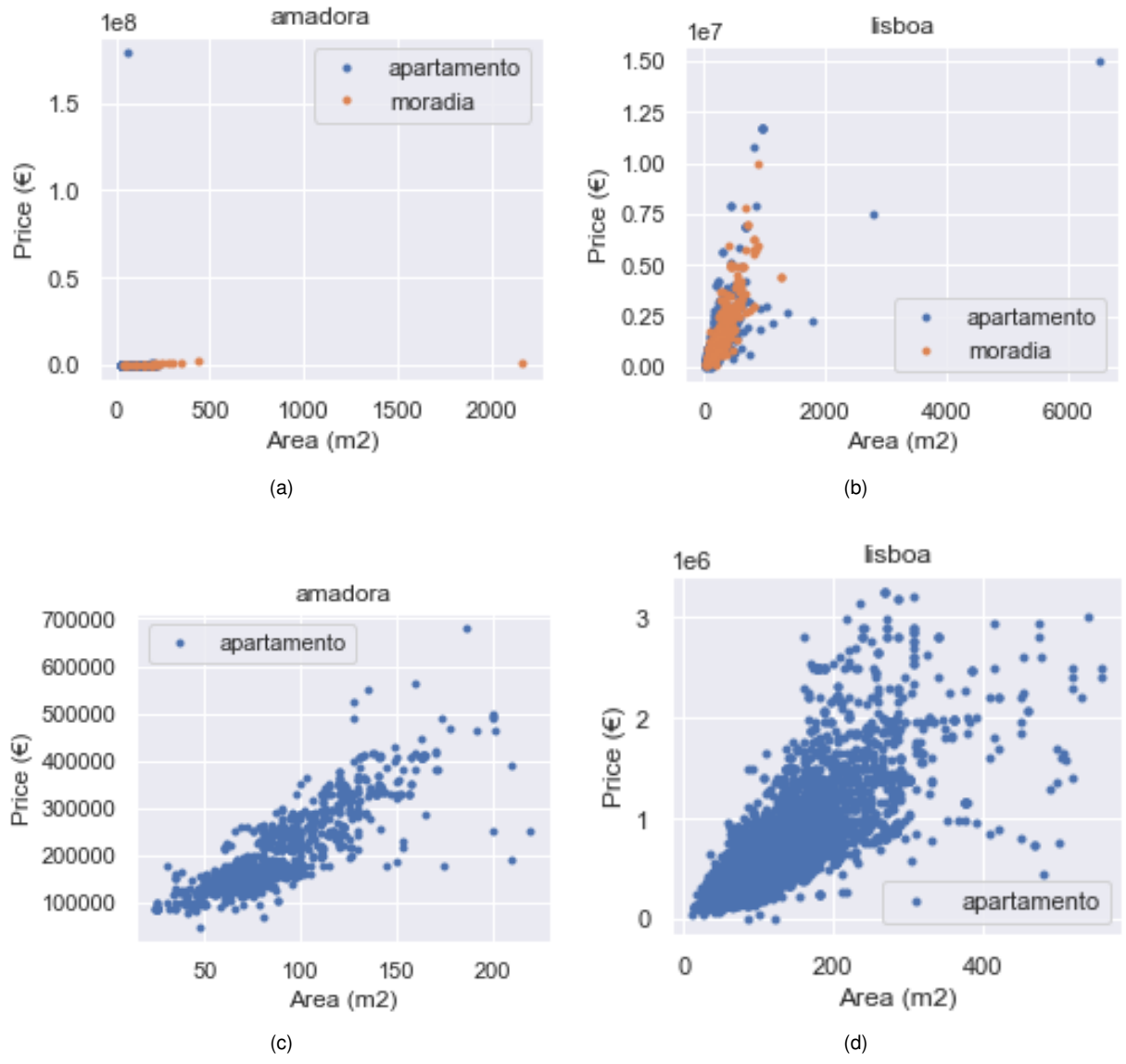
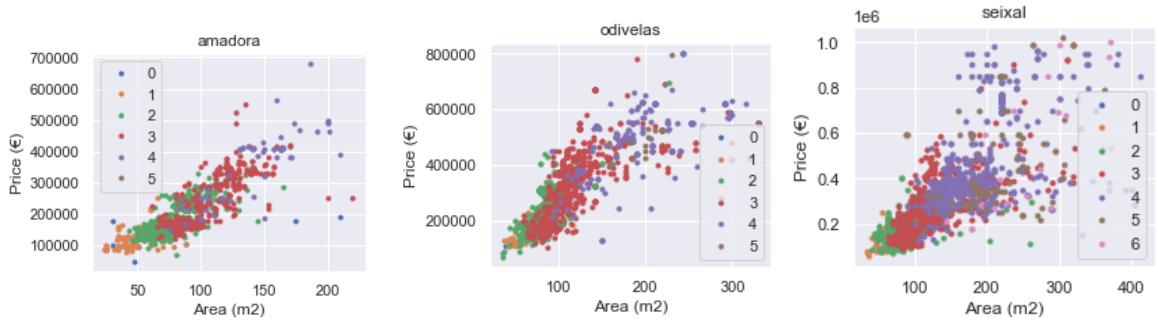


Figure 4.4: Relation between Area, Price and Property Type before (a)(b) and after (c)(d) removing outliers.

Another insight available in these Area-Price graphs is the variation of price with the number of rooms, depicted in Figure 4.5. Here we can see a clear variation of the number of rooms, distinguished by color. The higher the number of rooms, the higher the price and the area of the property, which would be obvious, since a property with more rooms is worth more, and having more rooms implies a higher area. This relation also occurs with the number of bathrooms, but not, for example, with the Energy Certificate, where no clear relationship is found by the respective graphs, which can be consulted in the

Appendix A.



(a) (b) (c)

Figure 4.5: Relation between Area, Price and Number of Rooms (depicted by color) after removing outliers.

Dimensionality Reduction

Since the dataset was composed by several features that could be in part correlated, the correlation matrix was computed, so that any correlation and causality could be spotted more easily through an image. As can be seen in Figure 4.6 and 4.7, the highest correlation detected between features is 0.66, between area and the number of rooms, which is not significant enough to remove one of the columns involved. Plus, it is interesting to evaluate the influence each feature has on the target feature, Price.

Another approach that was embraced was Principal Component Analysis. Since the baseline was a Linear Regression Model to calculate Real Estate Prices, there were two experiments of Linear Regression Models. One of them involved choosing principal features through PCA while the other did not. Since the latter performed better, that is, it provided lower values of Mean Average Precision Error and Mean Average Error, it was concluded that in this context it was not convenient to use PCA.

Duplicates

Considering there will be data referent to every day for more than two months, it is expectable that there is a huge number of repeated values, which means that the first thing to do is look out for rows whose different entries are only the timestamps from when they were collected and keep only one of them in the dataset.

Although, additionally, sometimes the same property might be available in more than one real estate agency, meaning it can look like different properties and appear more than once in the real estate portal where data will be collected from. Consequently, before using the data to train a model, it is necessary to look out for those duplicates, and treat them accordingly.

Since we will be dealing with a considerable amount of data, it is needed an automatic approach to detect the duplicates, otherwise it would be unfeasible to locate them.

In the website where data was collected in the beginning, different Real Estate agencies might have inserted the same house with slightly different characteristics, which might compromise the performance of the model. For example, if the same house is represented with slightly different characteristics but with the same price, or with exactly the same characteristics but a different price, that might lead to a

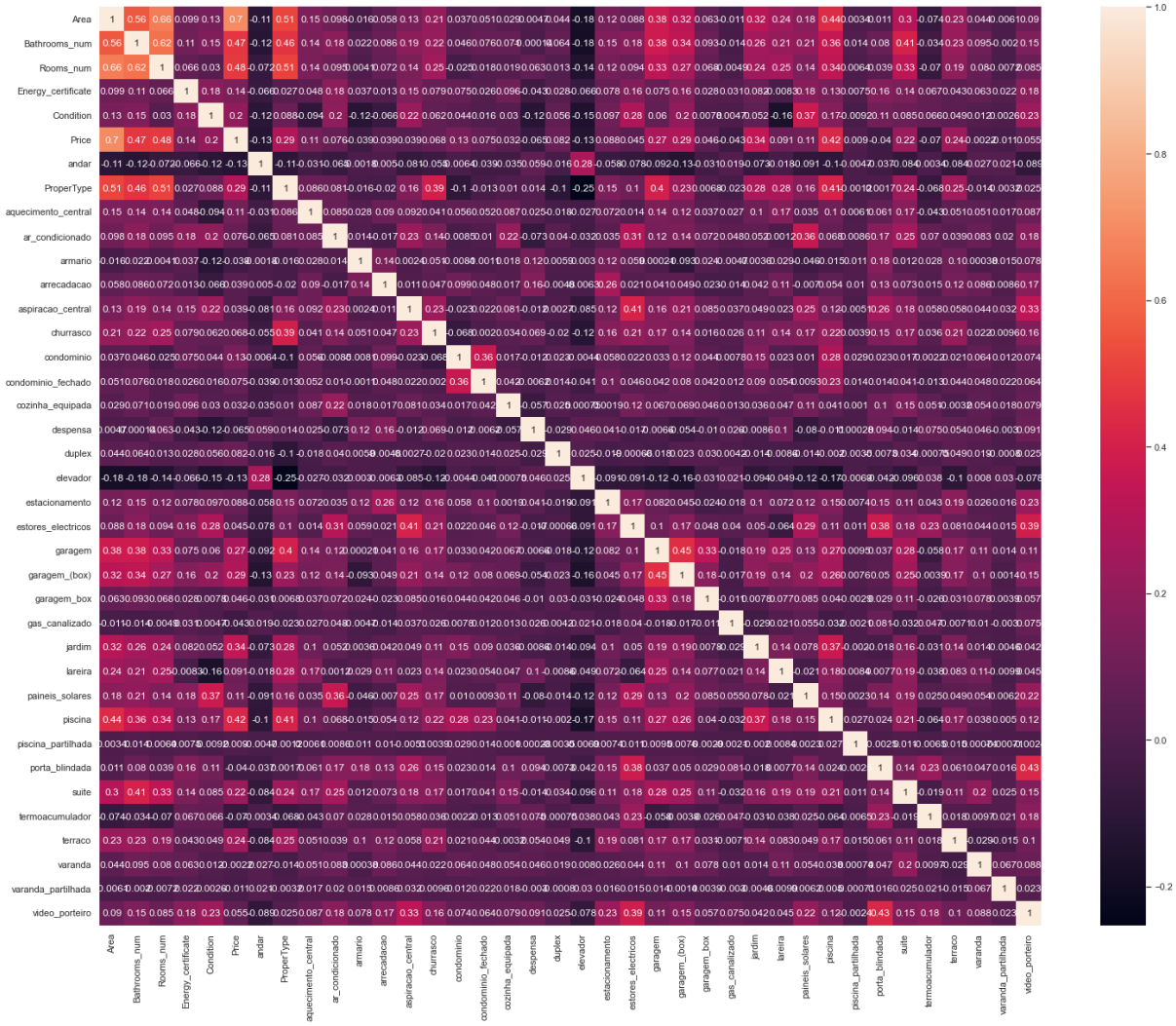


Figure 4.6: Correlation Matrix.

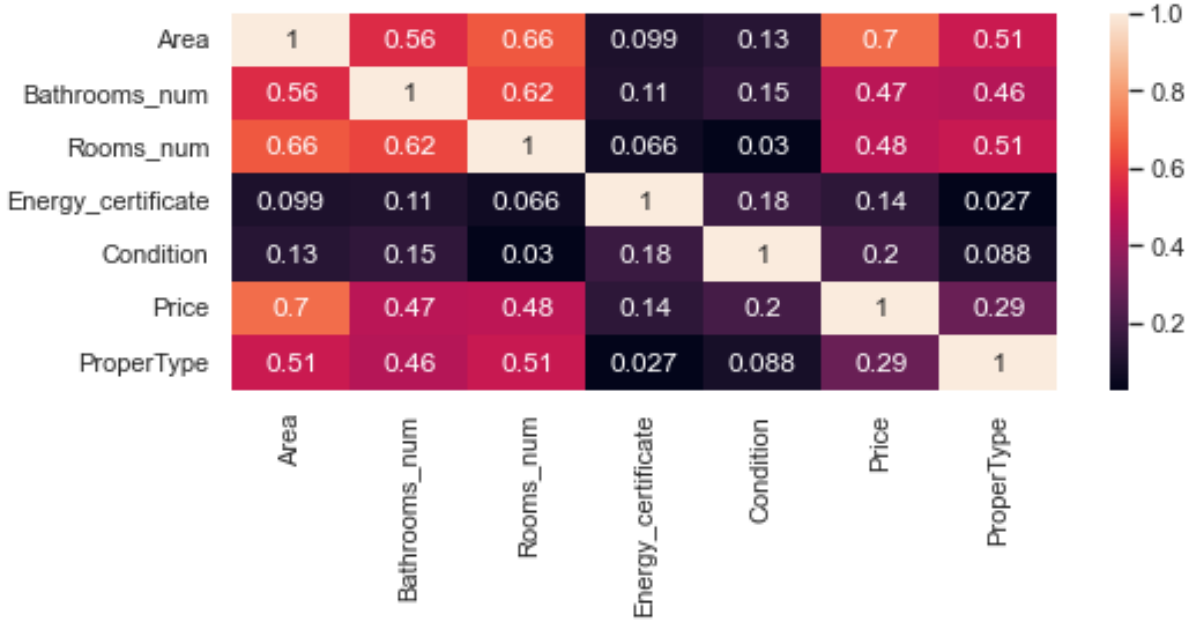


Figure 4.7: Correlation Matrix of fundamental features.

Bayes Error. So these duplicates must be found and properly treated.

There was not a control set stating whether or not a pair of houses represented duplicates, since it would be unfeasible to evaluate the complete dataset manually to find the true duplicates. Although, there was a shorter set of data that was manually evaluated in order to locate duplicates before applying any algorithm to detect duplicates.

On the complete dataset, it was performed an analysis on the set of descriptions, applying different algorithms and similarity measures, in order to evaluate how adequate each approach would be to detect duplicates, based on text descriptions.

To avoid comparisons between houses that are obviously not duplicates, such as houses in different locations, or with considerably different characteristics, an aggregation by location, number of bathrooms and area range is made, so that possible duplicates are already retrieved at this point. Afterwards, the NLP algorithms will only compute similarities between the groups aggregated before.

The purpose of each NLP algorithm is to yield an embedding for every text entry, so that the set of descriptions can be represented as vectors in a vector space, and, subsequently, vector distances can be computed in order to find the closest ones, which might represent the duplicates we are looking for.

The pre-processing of the text was the same for every algorithm: tokenising the text, removing punctuation, stopwords, special characters, and a few irrelevant adverbs, as well as HTML tags that had been extracted involuntarily. On a first approach, a Bag of Words and a TF-IDF models were implemented. The Bag of Words will represent each text as the times each word occured, while the TF-IDF will represent them as a score that mirrors the relevance of each word in the whole set. On the same conditions, that is, the same set of descriptions and considering the same similarity measure, the TF-IDF model seems to be more reliable. The BoW is recongnizing a lot more duplicates than the TF-IDF, and that might be because it does not take into account the words in the collection as the latter does.

As mentioned before, the control set that we had was considerably low compared to the rest of the dataset, so the evaluation of the duplicate detection was done mainly by hand. Thus, based on the aforementioned procedure, using the complete dataset, a set of pairs of possible duplicates was exported and covered manually, to tag which pairs were in fact duplicates or misclassifications. This sample of classifications will allow us to compute true positives and false positives, but will be inadequate to compute true negatives and false negatives. In order to address this issue, the smaller sample of entries was covered to explore the existence of duplicates, and that same sample was then processed in the operation described before with BoW, TF-IDF, and BERT algorithms.

Ordinal and One-hot Encoding

In Table 4.1, we can observe that that are a considerable number of categorical variables. Those variables must be transformed in numeric variables before being used to train the model.

There are features that involve an intrinsic order, such as the Condition of the property, where a new house will obviously carry a higher value than an old one, or the Energy Certificate, which holds letters with a specific order and a house classified with A will be of more value than one classified with B. These features must be Ordinal Encoded, transforming each category to an integer that respects its order.

Furthermore, there is also a fundamental categorical feature that has only two values, Property Type. This variable indicates whether the entry is an apartment or a house, not having a specific order, so it is easily one hot encoded. For that reason, the category of apartment is translated to the number 1 and a house will be represented by 0.

Data Normalisation

In order to have all the features on the same scale, the dataset will be normalised.

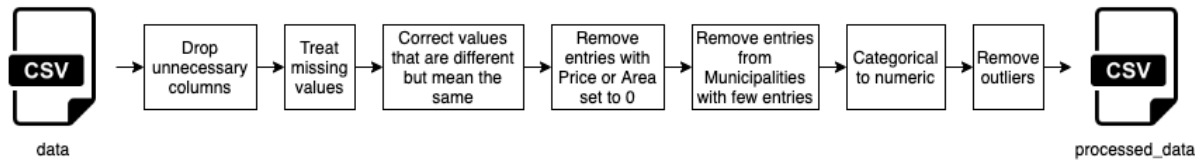


Figure 4.8: Data pre-processing workflow.

4.2.2 Training

After all pre-processing techniques, the dataset used to train the models comprised of almost 26000 instances, and it was only being considered data from the twelve municipalities with the higher amount of instances, which can be consulted in Table 4.2.

Location of subset	Number of instances
All data	25731
Almada	1327
Amadora	1069
Cascais	3535
Lisboa	7528
Loures	864
Mafra	1441
Montijo	873
Odivelas	1641
Oeiras	1820
Seixal	1867
Setúbal	1183
Sintra	2583

Table 4.2: Number of instances by municipality.

Since each dataset performs differently on different approaches, it has to be assessed which algorithm performs better, i.e., provides a better accuracy. For that reason, before deciding on an algorithm to train the model, some experiments were be conducted.

As above mentioned, the solutions that produced better results on similar problems used Artificial Neural Networks and Regression. In reality, ANN even performed better than Regression. Within the scope of Neural Networks different decisions might be taken, such as which activation function to use, the number of layers and neurons in each layer.

The model to predict the selling price of a house is a machine learning model that, receiving a property's characteristics as inputs, will calculate its fair price and return it as output. The dataset available will be separated in three sets, a training set (70% of the dataset), a testing set (15%), and a

validation set (15%), using K-fold Cross Validation. The model will be trained using the training set and its accuracy will then be assessed using the testing set.

On a first approach, the ANN topology will be based on the similar work of 2020 [6] that achieved such good results as MAPE values of 3.39% and 3.58%. However, it needs to be taken into account that in the experiment they were dealing with a really short dataset compared to the one we will be using. Meaning we might start with a small number of hidden layers and neurons, but the model must be tested with an extensive multitude of topologies with higher numbers of hidden layers and neurons. Additionally, the similar work of 2018 [4] was more efficient in finding a suitable architecture for their Neural Network, using Grid Search, so that technique will be adopted.

The methodology of work can be consulted in the chart of Figure 4.9.

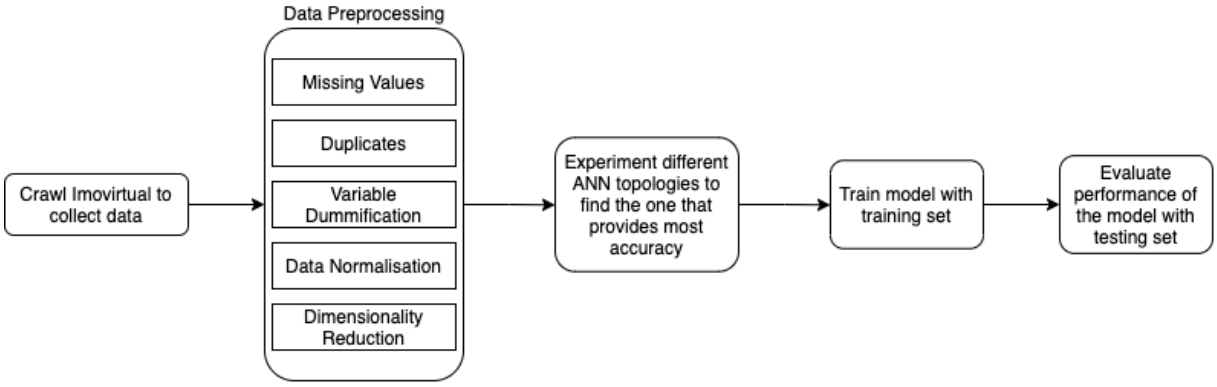


Figure 4.9: Work flow model.

All the following algorithms were implemented in two different ways. First by training one model for the whole dataset, and then by separating the dataset and training one model per municipality.

To tune the hyper-parameters of each algorithm, except Linear Regression, it was performed Grid Search with cross-validation so that the optimal hyper-parameters were found and then used to train the model. This entails that, in some cases, the same algorithm might have different ideal hyper-parameters considering the municipality we are treating.

Linear Regression

As a baseline model it was implemented a Linear Regression. The main goal of a baseline model is to quickly fit a dataset without much effort and computation. Linear Regression fills this requirement, since it is relatively easy to set up and has a considerable chance of providing reasonable results.

Artificial Neural Networks

Considering that different Neural Network topologies might lead to different training results, and that different types of data might respond better to different hyper-parameters, through Grid Search were run several experiments for each municipality data and the whole dataset. After all it was possible to distinguish the most adequate topology and combination of hyper-parameters for each type of data.

Random Forest

Random Forest for regression was used in similar experiments achieving proper results. Therefore an experiment on this algorithm was performed in this project as well. Due to having less hyper-parameters

that are not overly sensitive, Random Forest is an algorithm relatively easy to tune. By finding the most adequate hyper-parameters, we aim to increase the generalization performance of the algorithm.

The most relevant hyper-parameters that are considered worth to tune are the number of trees (n_estimators), the criteria with which to split on each node (criteria), the maximum number of features to consider at each split (max_features), and the maximum depth of each tree (max_depth).

Boosting Algorithms

Boosting was experimented in three ways: Adaptive Boosting, Gradient Boosting and Extreme Gradient Boosting, with decision stumps as weak learners. As in the algorithms aforementioned, different combinations of hyper-parameters were tested for each Boosting algorithm and for each municipality. The weak learners used were decision stumps.

Support Vector Machines

A regression model based on support vector machines, that is, a support vector regression was developed and its hyper-parameters were tested to find the most adequate for the dataset in question.

Chapter 5

Results

To evaluate the performance of the model, the metrics used will be the ones presented in the Background: R^2, MAE and MAPE.

5.1 Duplicates

The three procedures to find duplicates were applied to the complete dataset, and from this resulted a list of pairs of possible duplicates. The duplicates identified by the experiments were evaluated manually in order to understand which were True Positives and False Positives. The TF-IDF approach detected 134 pairs of duplicates, but only 113 were in fact duplicates, resulting in a Precision of 0.84. Bag-of-Words found 136 pairs of duplicates, but only 115 were in fact duplicates, resulting in a Precision of 0.84. Lastly, BERT retrieved 319 pairs of duplicates, but only 202 were in fact duplicates, resulting in a Precision of 0.63.

Using the smaller set of data, concerning one municipality, that had true duplicates, none of the three algorithms found a possible pair of duplicates. This was probably due to this sample comprising only about 100 instances while the complete dataset contained almost 26000 entries.

5.2 Linear Regression

The Linear Regression model, implemented as a baseline, as explained before, achieved the results registered in Table 5.1.

	R2	MAE	MAPE (%)
All Data	0.70	167418.29	42.82
Almada	0.64	62703.50	21.06
Amadora	0.71	27757.08	13.24
Cascais	0.69	325744.40	39.36
Lisboa	0.67	164185.29	30.32
Loures	0.74	72132.94	22.63
Mafra	0.52	95114.95	27.77
Montijo	0.60	55506.62	20.46
Odivelas	0.81	43257.96	15.09
Oeiras	0.70	124946.38	23.40
Seixal	0.75	62792.97	21.26
Setúbal	0.57	62272.69	22.02
Sintra	0.68	79809.15	24.27

Table 5.1: Linear Regression Results.

5.3 Artificial Neural Networks

To make sure we could find the most adequate Artificial Neural Network for each type of data, different arrangements of hyper-parameters were tested, and their results can be consulted in Table 5.2. It is worth mentioning that the number of epochs is the same for every case, 1000, as well as the activation function, ReLu. The number of epochs was chosen as a balance between what is computationally feasible in terms of processing time and what is necessary to achieve convergence of the model, meaning a higher number would not lead to more convergence but would take too many resources. In the case of the activation function, for a regression output it could only vary between Linear and ReLu, but since there are not negative prices, it only made sense to use ReLu.

The number of samples fed to the model at each iteration, the batch size, was varied between 32, 64 and 256. As we can see, for every experiment, the batch size that translates in better results is the lowest, 32. The learning rate (lr) was in its case, varied between two different values, 0.1 and 0.01. In respect to the number of hidden layers, n_layers, it was experimented with 2, 4 and 6, and their respective number of neurons, n_units, varied between 60, 80, and 120. The lowest number of neurons is as high as the number of input variables.

We can observe that, from the different subsets of data, the most complex network belongs to Lisboa, with four hidden layers and 120 units per layer. This is probably due to Lisboa being the municipality with the highest amount of data and, therefore, more diverse data that needs a compound network to cover it.

	R2	MAE	MAPE (%)	Hyper-parameters
All Data	0.72	126538.47	24.01	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 60
Almada	0.12	102907.27	31.73	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 60
Amadora	0.80	25832.77	13.46	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 4, 'n_units': 80
Cascais	0.50	338438.32	40.09	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 60
Lisboa	0.81	124098.30	22.67	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.01, 'n_layers': 4, 'n_units': 120
Loures	0.78	65621.50	26.73	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 4, 'n_units': 60
Mafra	0.24	114977.16	31.50	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 80
Montijo	0.81	42588.82	17.49	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 80
Odivelas	0.92	21780.31	7.58	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.01, 'n_layers': 4, 'n_units': 80
Oeiras	0.70	140208.78	23.52	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 60
Seixal	0.74	64401.85	22.19	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 60
Setúbal	0.46	72027.55	26.79	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.01, 'n_layers': 2, 'n_units': 120
Sintra	0.69	82464.00	21.56	'batch_size': 32, 'epochs': 1000, 'hidden_act': 'relu', 'lr': 0.1, 'n_layers': 2, 'n_units': 60

Table 5.2: Artificial Neural Networks Grid Search Results.

5.4 Random Forest

In Random Forest experiments, the hyper-parameters were tuned to find the best ones for each model, and their results can be found in Table 5.3.

The number of trees in each model, n_estimators, was tested between 500, 1000, and 1500. For each tree in the model, its maximum depth, max_depth, was also experimented between 15, 20, 50, and 100, and its maximum number of features to consider at every split, max_features, varied between 0.3, 0.5, and 0.8, which represents taking 30%, 50%, or 80%, respectively, of variables. For this hyper-parameter, one notices that the highest value, 0.8, is never chosen in any case, probably because a higher number of features is only better when the dataset is very noisy, which is not the case, due to the pre-processing done beforehand. Thus, using 30% or 50% of features to train each tree works well enough. Finally, the criteria to split each node, criterion, was also tuned between mean absolute error,

mae, and mean squared error, mse. For every experiment, mean squared error led to better results.

	R2	MAE	MAPE (%)	Hyper-parameters
All Data	0.86	88554.53	17.35	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.5, 'n_estimators': 1500
Almada	0.73	60181.97	17.56	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.3, 'n_estimators': 500
Amadora	0.85	20641.23	10.07	'criterion': 'mse', 'max_depth': 20, 'max_features': 0.3, 'n_estimators': 1000
Cascais	0.80	209190.56	23.32	'criterion': 'mse', 'max_depth': 100, 'max_features': 0.5, 'n_estimators': 1500
Lisboa	0.85	108513.93	19.24	'criterion': 'mse', 'max_depth': 100, 'max_features': 0.3, 'n_estimators': 500
Loures	0.89	43577.87	17.13	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.3, 'n_estimators': 1500
Mafra	0.68	70021.59	19.92	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.3, 'n_estimators': 1000
Montijo	0.88	31096.69	12.09	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.3, 'n_estimators': 500
Odivelas	0.95	16724.87	6.12	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.3, 'n_estimators': 500
Oeiras	0.85	91119.85	13.60	'criterion': 'mse', 'max_depth': 15, 'max_features': 0.5, 'n_estimators': 500
Seixal	0.85	42059.96	13.47	'criterion': 'mse', 'max_depth': 50, 'max_features': 0.3, 'n_estimators': 1000
Setúbal	0.83	41380.87	16.60	'criterion': 'mse', 'max_depth': 100, 'max_features': 0.3, 'n_estimators': 500
Sintra	0.83	57986.42	14.61	'criterion': 'mse', 'max_depth': 100, 'max_features': 0.5, 'n_estimators': 1000

Table 5.3: Random Forest Grid Search Results.

5.5 Adaptive Boosting

The most adequate AdaBoost models for each type of data were found by tuning the number of trees, $n\_estimators$, the weight applied to each estimator at each boosting iteration, $learning\_rate$, and the loss function to use when updating the weights after each boosting iteration, $loss$. The outcome of such experiments is indicated in Table 5.4.

The number of estimators varied between 50, 100, and 500, but such high number of trees only worked well for Amadora data. Regarding the learning rate, it was tested with 0.1 and 0.01, and the majority of experiments performed better with a lower learning rate. Finally, the possibilities for loss function were linear, square or exponential.

	R2	MAE	MAPE (%)	Hyper-parameters
All Data	0.67	181808.81	46.05	'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50
Almada	0.61	76719.68	23.40	'learning_rate': 0.1, 'loss': 'exponential', 'n_estimators': 50
Amadora	0.72	31203.44	16.37	'learning_rate': 0.1, 'loss': 'linear', 'n_estimators': 500
Cascais	0.65	307457.88	35.86	'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 100
Lisboa	0.71	180309.97	34.55	'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 50
Loures	0.80	64871.74	23.08	'learning_rate': 0.01, 'loss': 'square', 'n_estimators': 100
Mafra	0.50	96762.00	28.74	'learning_rate': 0.01, 'loss': 'exponential', 'n_estimators': 100
Montijo	0.77	49915.01	19.41	'learning_rate': 0.1, 'loss': 'exponential', 'n_estimators': 100
Odivelas	0.85	43840.04	14.96	'learning_rate': 0.1, 'loss': 'linear', 'n_estimators': 100
Oeiras	0.77	127096.82	20.67	'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 100
Seixal	0.75	61281.48	21.03	'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50
Setúbal	0.76	50296.43	20.87	'learning_rate': 0.01, 'loss': 'linear', 'n_estimators': 50
Sintra	0.77	78673.84	22.76	'learning_rate': 0.01, 'loss': 'square', 'n_estimators': 100

Table 5.4: Adaptive Boosting Grid Search Results.

5.6 Gradient Boosting

Gradient boosting has hyper-parameters that are very similar to the above-mentioned Adaptive Boosting. To achieve good results in these models, it was tuned, as before, the number of estimators, the learning rate, the loss function, but also the loss function, the subsample, and the criteria to measure the quality of a split. The results of such experiments can be seen at Table 5.5.

The number of estimators, i.e. the number of trees, took the values of 50, 100, and 500, but in every case it performed better with the highest number of trees. The learning rate was also the same in every experiment. It was experimented between 0.01 and 0.1, but it ended up providing better results with the latter. The loss function to be optimized could vary between least square loss, ls, least absolute deviation, lad, and a combination of LS and LAD, $huber$. Regarding the fraction of samples to be used for fitting the individual base learners, $subsample$, it was tested for 0.5 and 1. Lastly, the function to measure the quality of a split, $criterion$, may be mean squared error with improvement score by Friedman, $friedman\_mse$, mean squared error, $squared\_error$, and mean absolute error, mae.

	R2	MAE	MAPE (%)	Hyper-parameters
All Data	0.87	90624.52	32.74	'criterion': 'mse', 'learning_rate': 0.1, 'loss': 'lad', 'n_estimators': 500, 'subsample': 0.5
Almada	0.66	62970.73	16.69	'criterion': 'mae', 'learning_rate': 0.1, 'loss': 'lad', 'n_estimators': 500, 'subsample': 1
Amadora	0.81	22646.22	10.78	'criterion': 'mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 0.5
Cascais	0.80	214521.60	24.02	'criterion': 'friedman_mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 1
Lisboa	0.81	128581.14	22.20	'criterion': 'friedman_mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 0.5
Loures	0.87	46843.93	16.61	'criterion': 'mae', 'learning_rate': 0.1, 'loss': 'lad', 'n_estimators': 500, 'subsample': 1
Mafra	0.61	75792.04	20.33	'criterion': 'mse', 'learning_rate': 0.1, 'loss': 'lad', 'n_estimators': 500, 'subsample': 0.5
Montijo	0.85	35548.73	13.30	'criterion': 'mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 0.5
Odivelas	0.95	19510.60	6.83	'criterion': 'mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 0.5
Oeiras	0.84	91945.83	13.77	'criterion': 'friedman_mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 1
Seixal	0.85	42365.96	14.04	'criterion': 'mae', 'learning_rate': 0.1, 'loss': 'ls', 'n_estimators': 500, 'subsample': 1
Setúbal	0.79	45953.96	17.42	'criterion': 'mae', 'learning_rate': 0.1, 'loss': 'lad', 'n_estimators': 500, 'subsample': 1
Sintra	0.80	62343.98	15.86	'criterion': 'friedman_mse', 'learning_rate': 0.1, 'loss': 'huber', 'n_estimators': 500, 'subsample': 0.5

Table 5.5: Gradient Boosting Grid Search Results.

5.7 Extreme Gradient Boosting

Being a specific implementation of the previous algorithm, XGBoost provides more accurate approximations because it uses second order derivative, regularization and parallel computing. The Grid Search on this model was performed on the number of trees, the maximum number of levels in each tree, the learning rate, the booster, and the L1 regularization term on weights. The outcome of such experiments is stated in Table 5.6.

The number of trees, $n\_estimators$, was explored between 50, 100, and 500. For each tree, there is a maximum number of levels, which is represented by max_depth, and was tested for 10, 15, and 20. The learning rate consists of the weight applied to each estimator at each boosting iteration, and varies between 0.1 and 1.

	R2	MAE	MAPE (%)	Hyper-parameters
All Data	0.89	129672.38	29.26	'alpha': 0.01, 'booster': 'dart', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 50
Almada	0.68	63758.13	17.76	'alpha': 0.01, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 50
Amadora	0.83	21088.97	10.11	'alpha': 0.1, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 15, 'n_estimators': 50
Cascais	0.77	209479.94	23.10	'alpha': 0.1, 'booster': 'dart', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 500
Lisboa	0.85	107208.47	19.05	'alpha': 0.01, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 500
Loures	0.87	45481.82	17.23	'alpha': 0.01, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 500
Mafra	0.63	74966.02	21.37	'alpha': 0.01, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 50
Montijo	0.86	32087.09	12.24	'alpha': 0.1, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 15, 'n_estimators': 500
Odivelas	0.95	17282.00	6.01	'alpha': 0.01, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 500
Oeiras	0.81	100957.44	14.90	'alpha': 0.01, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 15, 'n_estimators': 50
Seixal	0.81	44273.38	13.98	'alpha': 0.1, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 15, 'n_estimators': 100
Setúbal	0.73	47671.98	17.51	'alpha': 0.1, 'booster': 'gbtree', 'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 50
Sintra	0.80	61123.41	15.35	'alpha': 0.1, 'booster': 'dart', 'learning_rate': 0.1, 'max_depth': 20, 'n_estimators': 100

Table 5.6: Extreme Gradient Boosting Grid Search Results.

5.8 Support Vector Regression

Support Vector Regression also needs its hyper-parameters tuned. For that purpose, the kernel type to be used in the algorithm was explored between linear and polynomial, $poly$. For the polynomial function, the intention was to experiment values 3, 5, and 10, and for the regularization parameter, 1, 50, and 100. Although, it was computationally unbearable to test all those combinations, and the experiment had to be limited between a linear kernel and a polynomial with degree 2. For the regularization parameter it was set a considerably high value that was computationally feasible, 100. The results of these experiments can be found at Table 5.7.

	R2	MAE	MAPE (%)	Hyper-parameters
All Data	0.55	161080.92	28.22	'C': 100, 'gamma': 'auto', 'kernel': 'linear'
Almada	0.67	68572.40	19.53	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Amadora	0.79	25043.74	12.32	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Cascais	0.66	277649.39	28.17	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Lisboa	0.69	162806.05	26.37	'C': 100, 'gamma': 'auto', 'kernel': 'linear'
Loures	0.75	66274.37	21.23	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Mafra	0.53	84632.11	22.64	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Montijo	0.61	51275.79	17.80	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Odivelas	0.80	44258.04	14.01	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Oeiras	0.74	129199.35	18.26	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Seixal	0.82	52539.59	17.02	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Setúbal	0.76	49037.80	18.14	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'
Sintra	0.73	72570.80	17.55	'C': 100, 'degree': 2, 'gamma': 'auto', 'kernel': 'poly'

Table 5.7: Support Vector Regression Grid Search Results.

5.9 Evaluation

As an overview of all the previous results, we can consult Table 5.8 to observe MAPE values of the algorithms tested for each municipality and for the dataset as a whole.

It is clear that training models without applying Real Estate market segmentation, that is, treating each municipality as an individual dataset, will lead to poorer performances, in general. That was expectable, since, by separating heterogeneous data, it will be easier for each sub-model to generalize and achieve better accuracies. The majority of subsets achieved better results with Random Forest and Gradient Boosting, as represented by the highlighted values. However, for Artificial Neural Networks and Random Forest, the model concerning all data provides better results than certain submodels.

For some locations it was harder to find a precisive model, such as Cascais and Lisboa. This may be due to the diversification of housing in both municipalities. In Cascais, as well as in Lisbon, we can found some luxury Real Estate and, at the same time, some social neighbourhoods, sometimes being geographically close.

	LR	ANN	RF	AdaBoost	Gradient Boost	XGBoost	SVM
All Data	42.82	24.01	**17.35**	46.05	32.74	29.26	28.22
Almada	21.06	31.73	17.56	23.40	**16.69**	17.76	19.53
Amadora	13.24	13.46	**10.07**	16.37	10.78	10.11	12.32
Cascais	39.36	40.09	23.32	35.86	24.02	**23.10**	28.17
Lisboa	30.32	22.67	19.24	34.55	22.20	**19.05**	26.37
Loures	22.63	26.73	17.13	23.08	**16.61**	17.23	21.23
Mafra	27.77	31.50	**19.92**	28.74	20.33	21.37	22.64
Montijo	20.46	17.49	**12.09**	19.41	13.30	12.24	17.80
Odivelas	15.09	7.58	6.12	14.96	6.83	**6.01**	14.01
Oeiras	23.40	23.51	**13.60**	20.67	13.77	14.90	18.26
Seixal	21.34	22.19	**13.47**	21.03	14.04	13.98	17.02
Setúbal	22.02	26.79	**16.60**	20.87	17.42	17.51	18.14
Sintra	24.27	21.56	**14.61**	22.76	15.86	15.35	17.55

Table 5.8: Overview of MAPE (%) values for every algorithm tested.

Chapter 6

Conclusion

Reviewing the main goals of this project, it is worth mentioning the necessity of an objective prediction model that could compute property prices and avoid the subjectivity of traditional methods for property evaluation. The dataset uses Real Estate from Lisbon and Setúbal due to the higher amount of Real Estate in these locations. Furthermore, the project also addressed the need of a proper duplicate detection mechanism. Thus, this project intended to explore a dataset containing data from the two districts, take some insights from it, find an adequate technique to detect duplicates in the dataset, and develop a reliable model that could take into account several features, and, from that, calculate the value of a property.

To achieve such purpose, the first task had to be to collect the needed data. That is why a web crawler was developed and data was collected from November 2020 until October 2021. Such task demanded a constant maintenance, since the website where data was being collected suffered some updates during the time this project was elapsing. Next, data was pre-processed and duplicate detection was experimented through different NLP techniques. The dataset containing data from Lisbon and Setúbal was separated by location and each subset was used to trained several models in order to understand which were more adequate to each type of data. Overall, Random Forest and Gradient Boosting performed better than the rest of the algorithms experimented, and Real Estate market segmentation has proven to be a valuable procedure, since separating the dataset and creating several models provided better results than training a single model using the whole dataset.

6.1 Contributions

The main contribution of this work consists on comparison of algorithms, providing a series of models, some more accurate than others, that are capable of computing property prices from a considerable amount of municipalities in Lisbon and Setúbal. Each municipality had data distributed in a certain way, and that is the main reason why some algorithms work better in data from one place but may perform poorly in another group of data.

There were also experiments on duplicate detection using Natural Language Processing. They were majorly based on text description that were associated with each property ad.

As an object of study, this project leaves a dataset with property records collected from November 2020 to October 2021, as well as a scraper capable to keep increasing such dataset.

6.2 Future Work

As previously mentioned, the crawler used might keep collecting useful data for similar works in the future. Nonetheless, it must be updated considering changes in the website. The data already collected may be used in further experiments, with algorithms not tested in this work, or with more complex tests that were not covered before. It might also be interesting to represent data as time series, considering there are data collected in different moments of time.

Furthermore, it would also be interesting to explore data from other sources and with different features. The data used in this work does not include the actual price for which a property was sold nor socio-demographic factors, and it would provide valuable insights if the experiments performed here were also applied to such data.

In the scope of duplicate detection, description text could be more thoroughly pre-processed in order to increase accuracy. Some descriptions contained information relative to Real Estate agencies that could be deleted, but such pre-processing task would be too complex to complete in the time span of this project.

Bibliography

[1] "How to determine where we are in the real estate market cycle - marshallcf." https://marshallcf.com/how-to-determine-where-we-are-in-the-real-estate-market-cycle/. (Accessed on 01/05/2021).

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, S. Huang, M. Brooks, M. J. Lee, and H. Asadi, "Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods," *American Journal of Roentgenology*, vol. 212, no. 1, pp. 38–43, 2019.

[4] R. B. Abidoye and A. P. Chan, "Improving property valuation accuracy: A comparison of hedonic pricing model and artificial neural network," *Pacific Rim Property Research Journal*, vol. 24, no. 1, pp. 71–83, 2018.

[5] F. Bre, J. M. Gimenez, and V. D. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks," *Energy and Buildings*, vol. 158, pp. 1429–1441, 2018.

[6] M. Štubňová, M. Urbaníková, J. Hudáková, and V. Papcunová, "Estimation of residential property market price: Comparison of artificial neural networks and hedonic pricing model," *Emerging Science Journal*, vol. 4, no. 6, pp. 530–538, 2020.

[7] "Real estate: Definition, types, how the industry works." https://www.thebalance.com/real-estate-what-it-is-and-how-it-works-3305882. (Accessed on 01/02/2021).

[8] "Six fundamental human needs we need to meet to live our best lives." https://www.forbes.com/sites/quora/2018/02/05/six-fundamental-human-needs-we-need-to-meet-to-live-our-best-lives/?sh=702c16e5344a. (Accessed on 01/05/2021).

[9] "Real estate market: Your complete guide — millionacres." https://www.fool.com/millionacres/real-estate-market/. (Accessed on 01/02/2021).

[10] "Understanding the four phases of the real estate cycle — fortunebuilders." https://www.fortunebuilders.com/real-estate-cycle/. (Accessed on 01/02/2021).

[11] "4 phases of a real estate cycle." https://www.biggerpockets.com/member-blogs/10401/69798-4-phases-of-a-real-estate-cycle. (Accessed on 01/02/2021).

[12] "The four phases of the real estate cycle — crowdstreet." https://www.crowdstreet.com/resources/investing/real-estate-cycle/. (Accessed on 01/02/2021).

[13] P. Syms, "Real estate economics," 2013.

[14] "Novo recorde: mercado imobiliário já vale 12% do pib - tsf." https://www.tsf.pt/economia/mercado-imobiliario-aumentou-24-no-ano-passado-10721075.html. (Accessed on 01/05/2021).

[15] I. Geipele, L. Kauskale, N. Lepkova, and R. Liias, "Interaction of socio-economic factors and real estate market in the context of sustainable urban development," in *Environmental Engineering. Proceedings of the International Conference on Environmental Engineering. ICEE*, vol. 9, p. 1, Vilnius Gediminas Technical University, Department of Construction Economics . . . , 2014.

[16] "Factors affecting the real estate market," Jul 2020.

[17] "What is proptech in real estate, and what does it mean? — hqo." https://www.hqo.co/what-is-proptech-and-what-does-it-mean/. (Accessed on 01/02/2021).

[18] "Cre innovation report 2019 - our commercial real estate services — altus group." https://www.altusgroup.com/services/en-eu/reports/cre-innovation-report-2019/. (Accessed on 01/02/2021).

[19] "Cre innovation report 2020 - argus — software solutions for commercial real estate." https://www.altusgroup.com/argus/resources/insights/cre-innovation-report-2020. (Accessed on 01/02/2021).

[20] "5 ways ai is changing the real estate sector — by axeleo — axeleo — medium." https://medium.com/axeleo/5-ways-ai-is-changing-the-real-estate-sector-a726bf600a83. (Accessed on 01/02/2021).

[21] "Localize." https://www.localize.city/. (Accessed on 01/02/2021).

[22] "Convoboss." https://convoboss.com/real-estate-chatbot. (Accessed on 01/02/2021).

[23] "Realty chatbot." https://realtychatbot.com/. (Accessed on 01/02/2021).

[24] "Skyline ai." https://www.skyline.ai/. (Accessed on 01/02/2021).

[25] "Zillow: Real estate, apartments, mortgages & home values." https://www.zillow.com/. (Accessed on 01/02/2021).

[26] "keezag: Compare comissões, serviços e faça o melhor negócio." https://www.keezag.com/. (Accessed on 01/02/2021).

[27] "Reatia — página inicial." https://reatia.com/pt/. (Accessed on 01/02/2021).

[28] "A base de dados mais limpa e completa do mercado imobiliário." https://pt.casafari.com/. (Accessed on 01/02/2021).

[29] H. Xiaolong and Z. Ming, "Applied research on real estate price prediction by the neural network," in *2010 The 2nd Conference on Environmental Science and Information Application Technology*, vol. 2, pp. 384–386, IEEE, 2010.

[30] B. Trawiński, Z. Telec, J. Krasnoborski, M. Piwowarczyk, M. Talaga, T. Lasota, and E. Sawiłow, "Comparison of expert algorithms with machine learning models for real estate appraisal," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 51–54, IEEE, 2017.

[31] W. J. McCluskey, M. McCord, P. Davis, M. Haran, and D. McIlhatton, "Prediction accuracy in mass appraisal: a comparison of modern approaches," *Journal of Property Research*, vol. 30, no. 4, pp. 239–265, 2013.

[32] M. D'Amato and T. Kauko, "Advances in automated valuation modeling," *Springer International Publishing AG, doi*, vol. 10, pp. 978–3, 2017.

[33] J. Alico, *Appraising machinery and equipment*. McGraw-Hill, 1989.

[34] N. N. Ghosalkar and S. N. Dhage, "Real estate value prediction using linear regression," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–5, IEEE, 2018.

[35] K.Agnieszka, "Aplicación de redes neuronales artificiales en la valuación inmobiliaria," in *MBA Thesis*, 2008.

[36] Y. E. Hamzaoui and J. A. H. Perez, "Application of artificial neural networks to predict the selling price in the real estate valuation process," in *Proceedings of the 2011 10th Mexican International Conference on Artificial Intelligence*, pp. 175–181, 2011.

[37] W. Biao, "On the analysis of income approach for real estate," in *MSIE 2011*, pp. 988–991, IEEE, 2011.

[38] V. Kontrimas and A. Verikas, "The mass appraisal of the real estate by computational intelligence," *Applied Soft Computing*, vol. 11, no. 1, pp. 443–448, 2011.

[39] "Cost approach (real estate) - overview, how to calculate, limitations." `https://corporatefinanceinstitute.com/resources/knowledge/valuation/cost-approach-real-estate/`. (Accessed on 01/05/2021).

[40] "Hedonic pricing definition." `https://www.investopedia.com/terms/h/hedonicpricing.asp`. (Accessed on 01/06/2021).

[41] D. Tchuente and S. Nyawa, "Real estate price estimation in french cities using geocoding and machine learning," *Annals of Operations Research*, pp. 1–38, 2021.

[42] Y. El Hamzaoui, J. Hernandez, M. Cruz-Chavez, and A. Bassam, "Search for optimal design of multiproduct batch plants under uncertain demand using gaussian process modeling solved by heuristics methods," *Chemical Product and Process Modeling*, vol. 5, no. 1, 2010.

[43] A. Varma, A. Sarma, S. Doshi, and R. Nair, "House price prediction using machine learning and neural networks," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 1936–1939, IEEE, 2018.

[44] I. D. Wilson, S. D. Paris, J. A. Ware, and D. H. Jenkins, "Residential property price time series forecasting with neural networks," in *Applications and Innovations in Intelligent Systems IX*, pp. 17–28, Springer, 2002.

[45] J. J. D.D. Hawley and D. Raina, "Artificial neural systems: A new tool for financial decision-making," *Financial Analysis Journal*, pp. 63–72, 1990.

[46] S. Goel, M. Bansal, A. K. Srisvastava, and N. Arora, "Web crawling-based search engine using python," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 436–438, IEEE, 2019.

[47] P. Ng, "dna2vec: Consistent vector representations of variable-length k-mers," *arXiv preprint arXiv:1701.06279*, 2017.

[48] M. J. Zaki, W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[49] H. A. Chowdhury and D. K. Bhattacharyya, "Plagiarism: Taxonomy, tools and detection techniques," *arXiv preprint arXiv:1801.06323*, 2018.

[50] L. Wang, L. Zhang, and J. Jiang, "Duplicate question detection with deep learning in stack overflow," *IEEE Access*, vol. 8, pp. 25964–25975, 2020.

[51] B. Yildiz, J. I. Bilbao, and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," *Renewable and Sustainable Energy Reviews*, vol. 73, pp. 1104–1122, 2017.

[52] J. Bin, S. Tang, Y. Liu, G. Wang, B. Gardiner, Z. Liu, and E. Li, "Regression model for appraisal of real estate using recurrent neural network and boosting tree," in *2017 2nd IEEE international conference on computational intelligence and applications (ICCIA)*, pp. 209–213, IEEE, 2017.

[53] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[54] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[55] D.-Y. Li, W. Xu, H. Zhao, and R.-Q. Chen, "A svr based forecasting approach for real estate price prediction," in *2009 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 970–974, IEEE, 2009.

[56] N. Pow, E. Janulewicz, and L. Liu, "Applied machine learning project 4 prediction of real estate property prices in montréal," *Course project, COMP-598, Fall/2014, McGill University*, 2014.

[57] "What is grid search?. explaining how to obtain optimal. . . — by farhad malik — fintechexplained — medium." `https://medium.com/fintechexplained/what-is-grid-search-c01fe886ef0a`. (Accessed on 01/06/2021).

[58] M. Z. Nejad, J. Lu, and V. Behbood, "Applying dynamic bayesian tree in property sales price estimation," in *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 1–6, IEEE, 2017.

[59] G. Pinter, A. Mosavi, and I. Felde, "Artificial intelligence for modeling real estate price using call detail records and hybrid machine learning approach," *Entropy*, vol. 22, no. 12, p. 1421, 2020.

[60] P. Runeson, M. Alexandersson, and O. Nyholm, "Detection of duplicate defect reports using natural language processing," in *29th International Conference on Software Engineering (ICSE'07)*, pp. 499–510, IEEE, 2007.

[61] D. Gupta *et al.*, "Study on extrinsic text plagiarism detection techniques and tools.," *Journal of Engineering Science & Technology Review*, vol. 9, no. 5, 2016.

[62] D. Sangani, K. Erickson, and M. Al Hasan, "Predicting zillow estimation error using linear regression and gradient boosting," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 530–534, IEEE, 2017.

[63] Y. Tang, S. Qiu, and P. Gui, “Predicting housing price based on ensemble learning algorithm,” in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pp. 1–5, IEEE, 2018.

Appendix A

Data Exploration

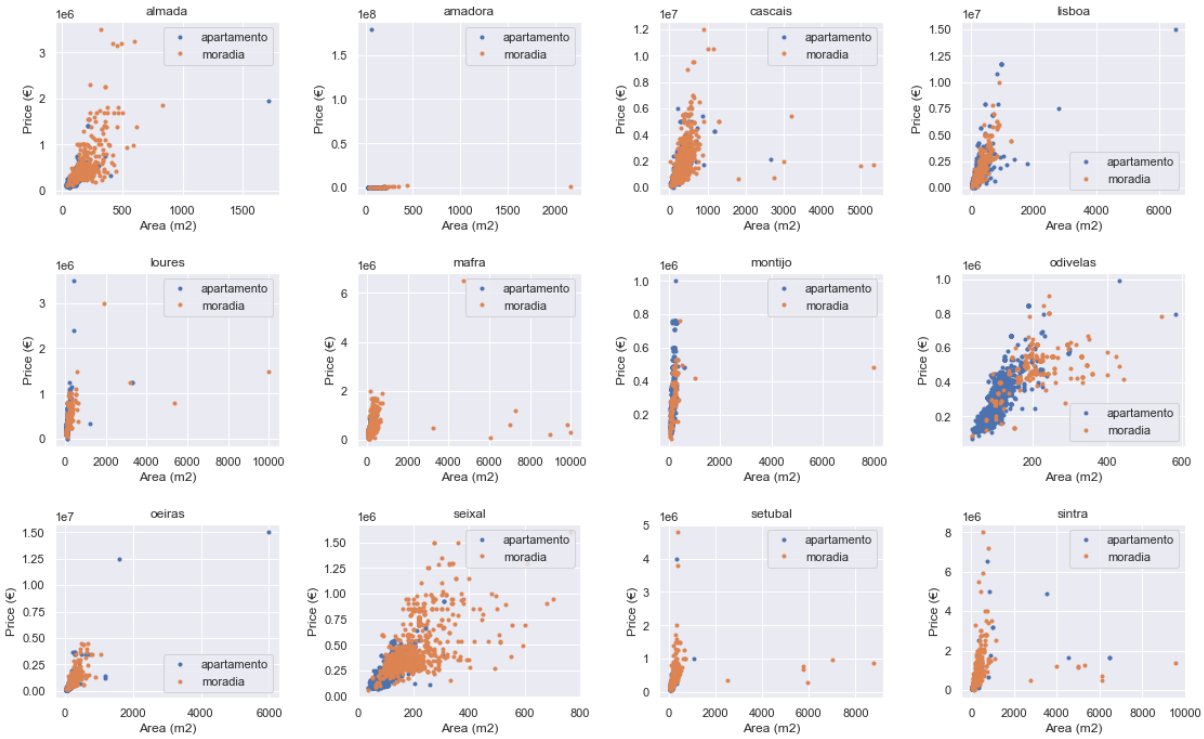


Figure A.1: Area-Price property type before removing outliers.

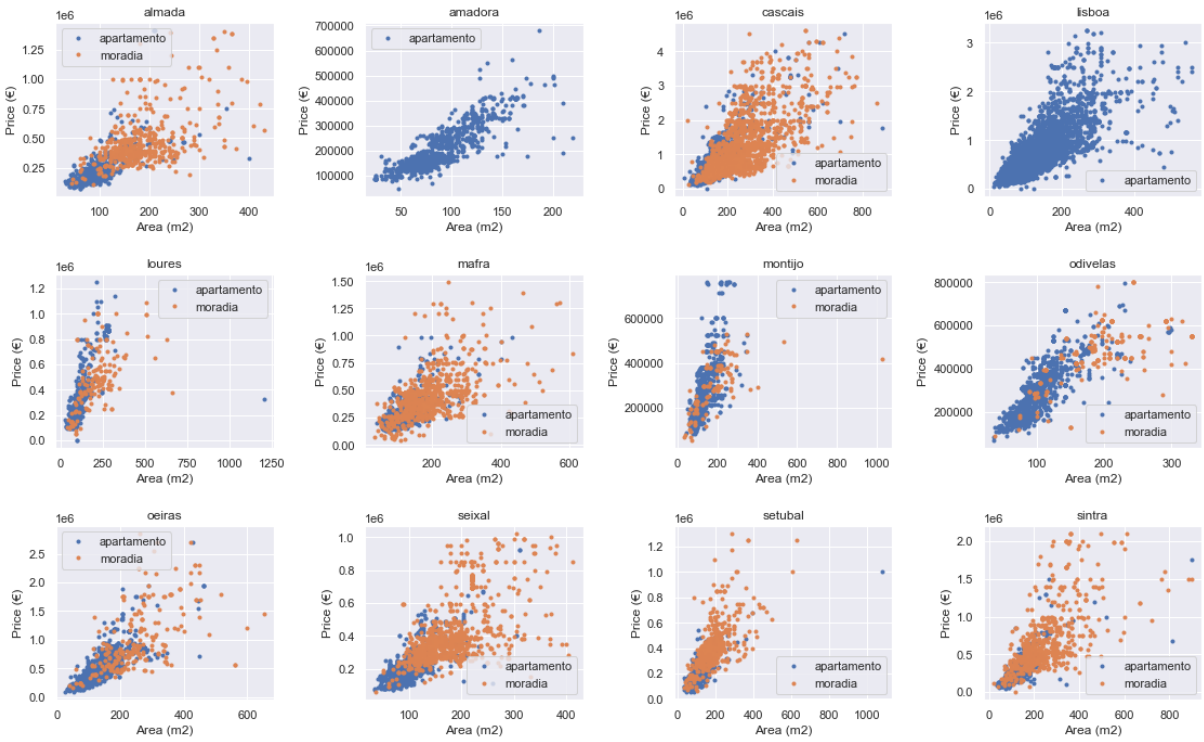


Figure A.2: Area-Price property type after removing outliers.

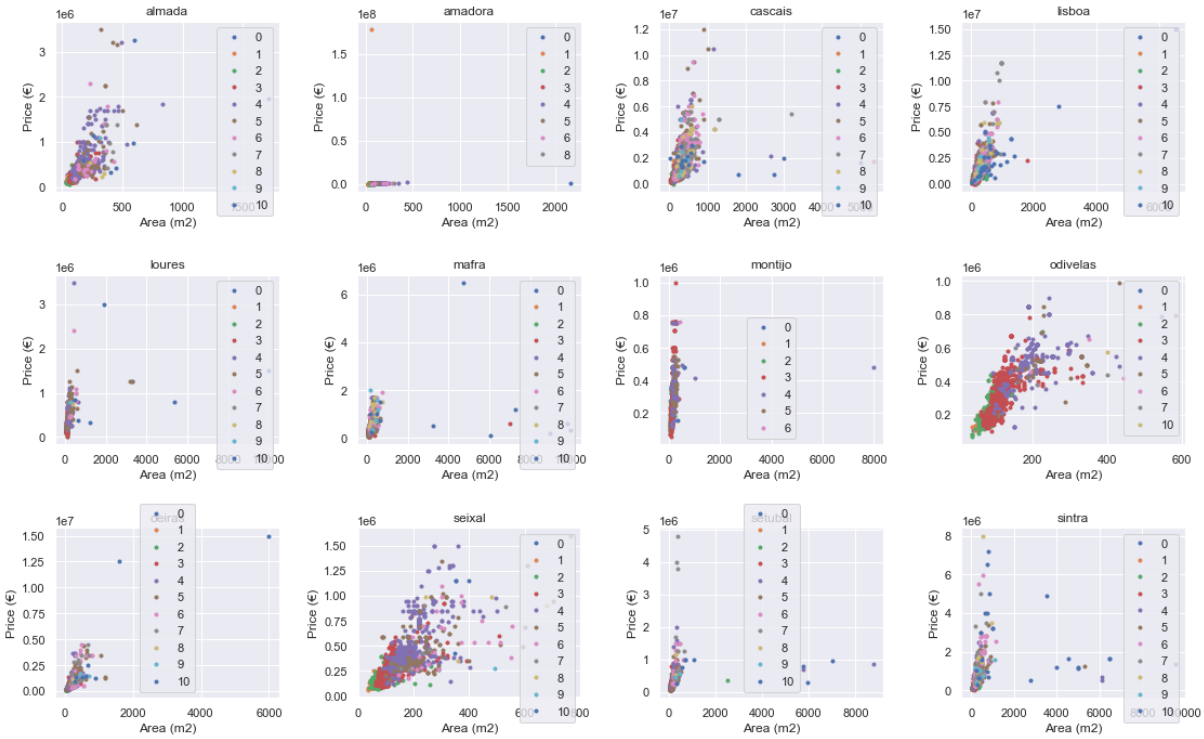


Figure A.3: Area-Price number of rooms before removing outliers.

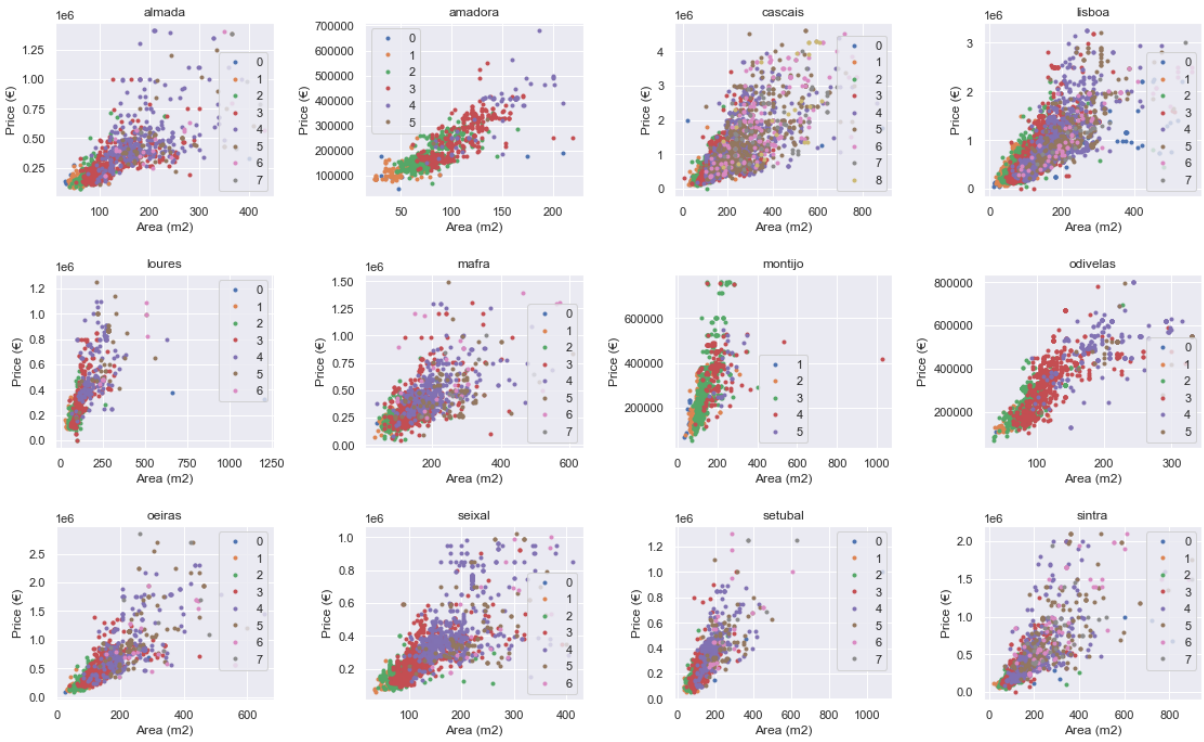


Figure A.4: Area-Price number of rooms after removing outliers.

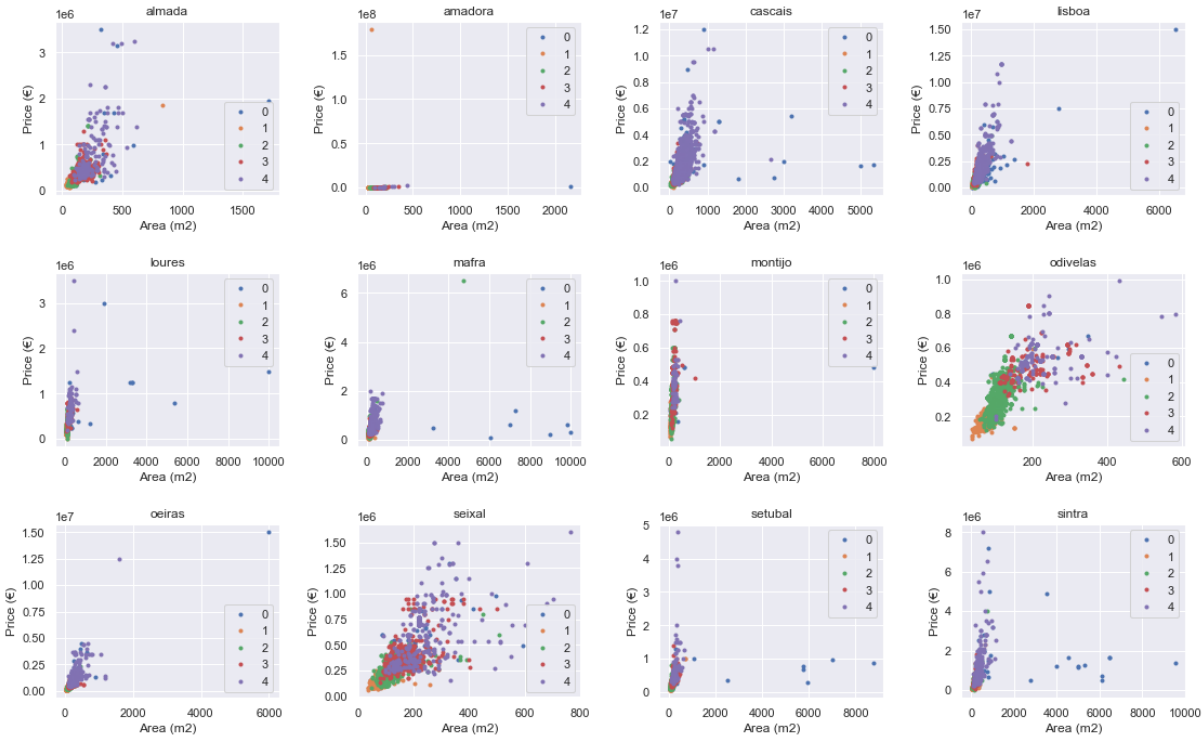


Figure A.5: Area-Price number of bathrooms before removing outliers.

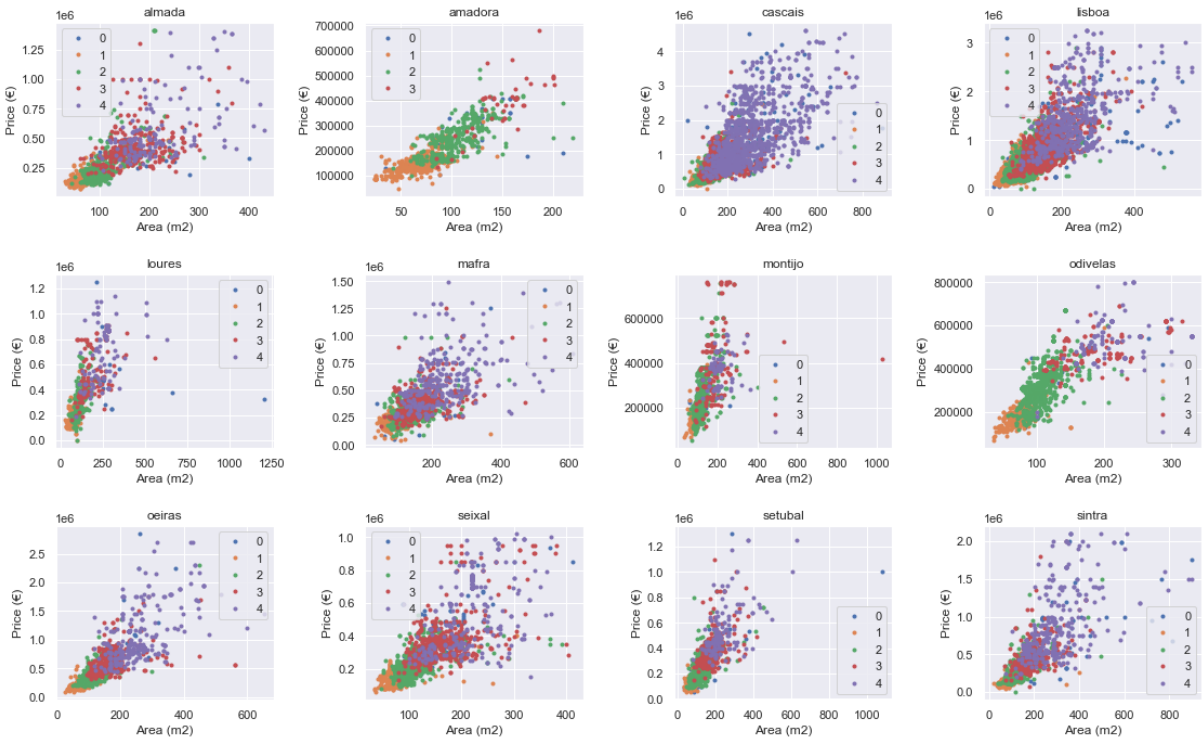


Figure A.6: Area-Price number of bathrooms after removing outliers.

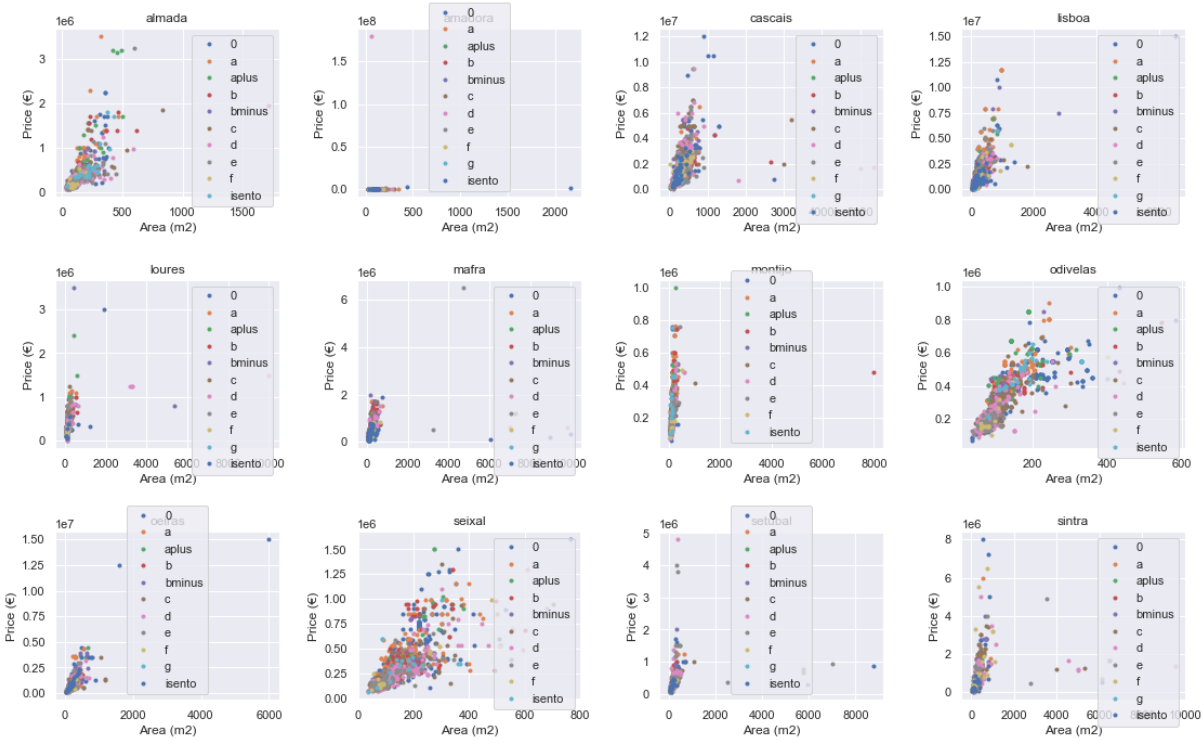


Figure A.7: Area-Price energy certificate before removing outliers.

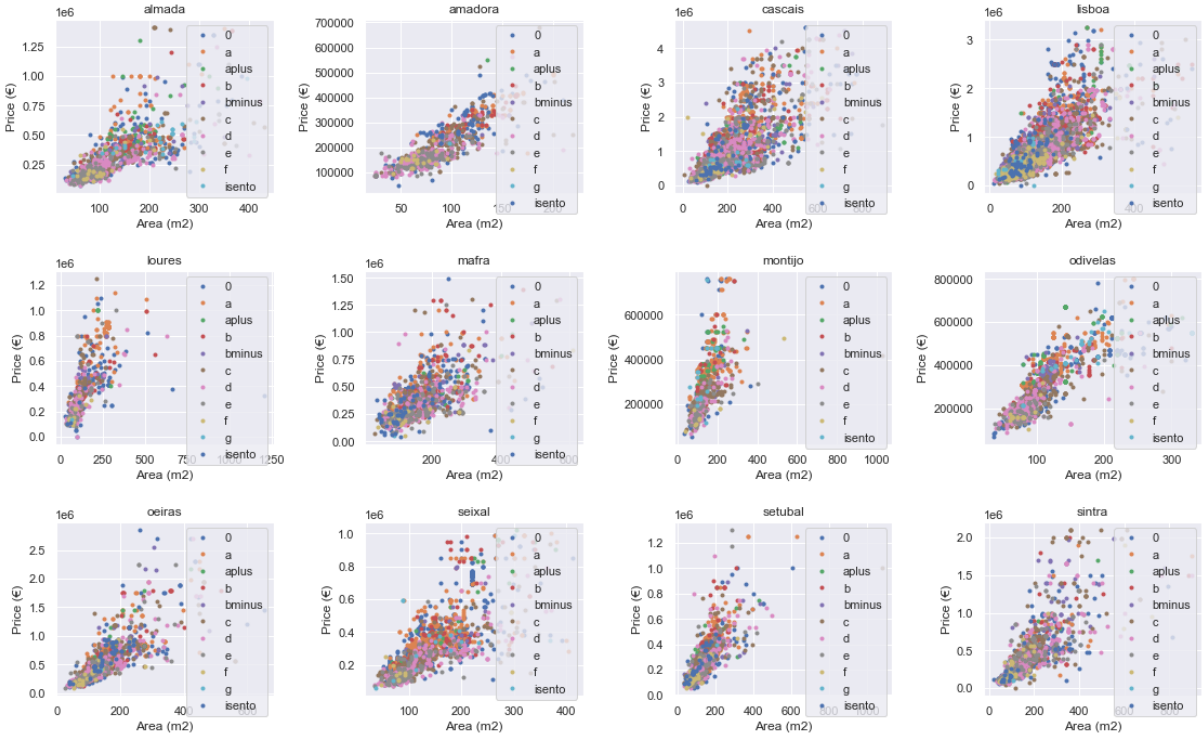


Figure A.8: Area-Price energy certificate after removing outliers.

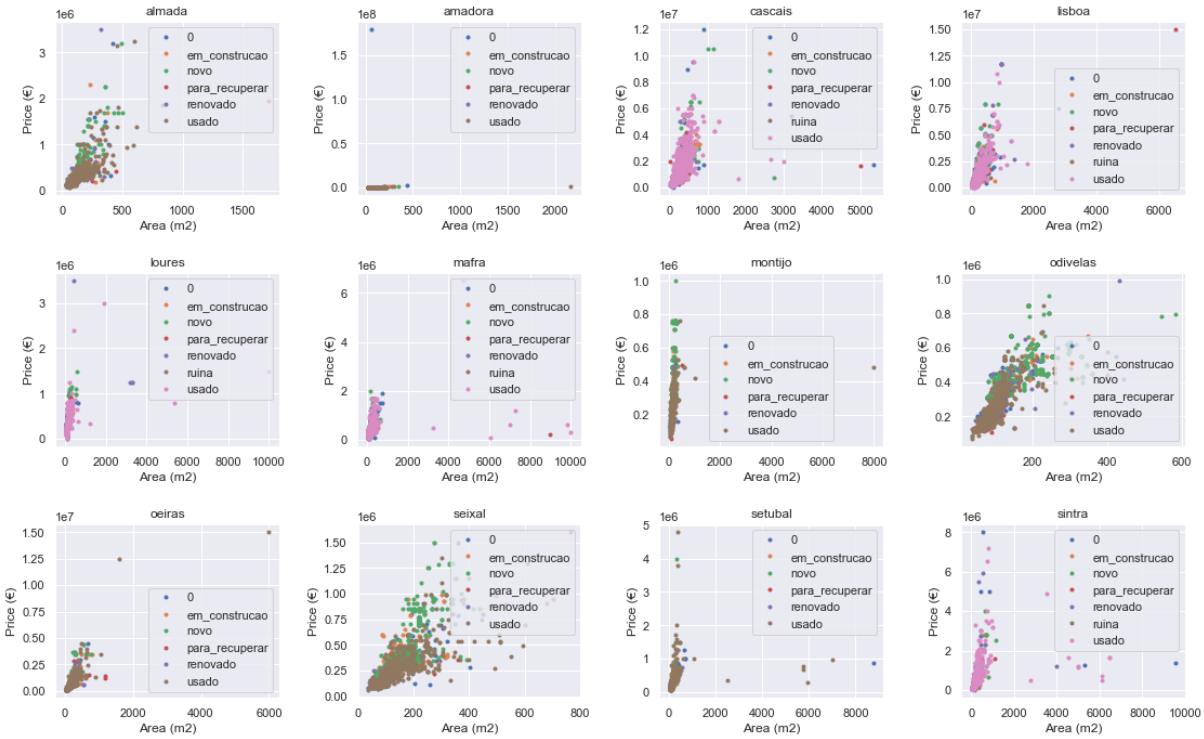


Figure A.9: Area-Price condition before removing outliers.

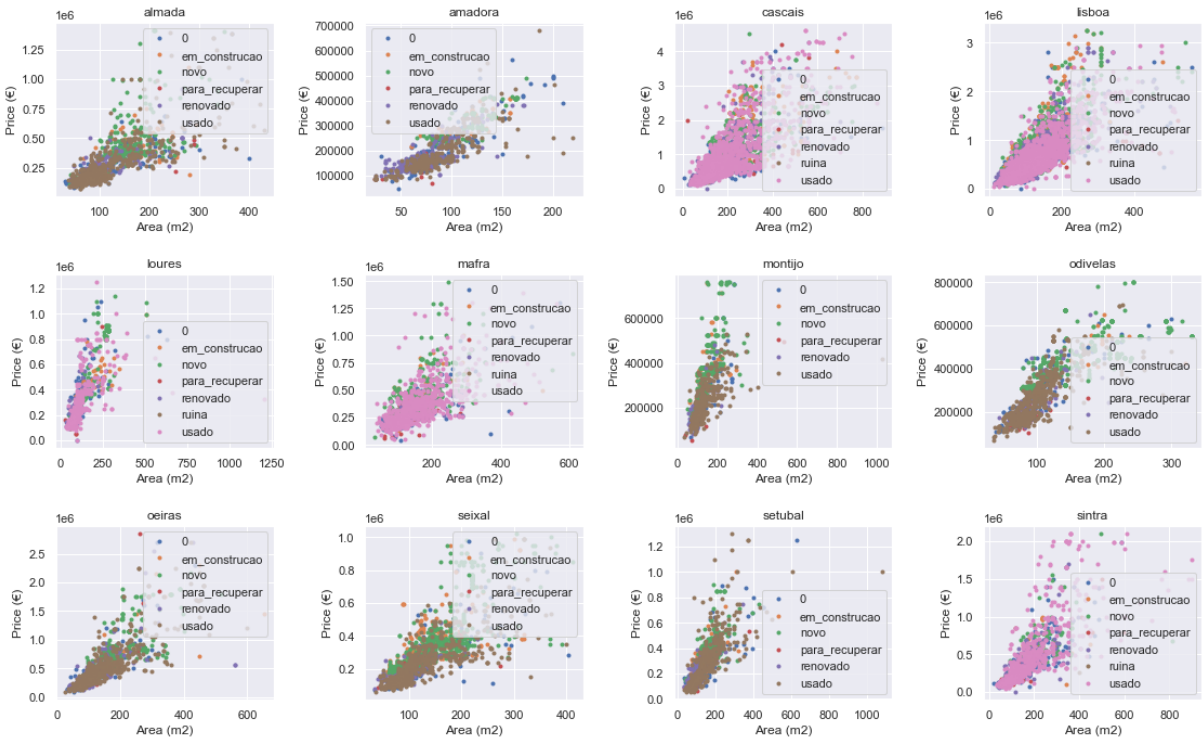


Figure A.10: Area-Price condition after removing outliers.