Real Estate Market Analysis For Investment and Buying Cycles Prediction

Maria Pereira Leitão
Instituto Superior Técnico
Lisbon, Portugal
mariapereiramq@tecnico.ulisboa.pt

ABSTRACT

The traditional methods used to estimate the 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.

Author Keywords

Real Estate, Artificial Intelligence, Machine Learning

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 [3].

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 [1]. 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.

Understanding Real Estate Market

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.

Along with economic factors there are political and social factors, environmental and scientific factors, which can also entail variations in the Real Estate market [6]. 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 [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 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.

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 unquantized, which makes it difficult to predict real estate prices [18]. 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.

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.

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.

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.

STATE OF THE ART

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.

Real Estate Market Segmentation

Tchuente et al. [15] 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.

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.

Data Collection

Tchuente et al. [15] 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.

Data Exploration

Data from [15] 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.

Data Preprocessing

The article from [15] 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.

Regarding inconsistency of the data, [15] 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.

When considering the outlier removal, [15] 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.

All the numeric variables were standardized in [15, 10], 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.

All discrete attributes in [15] were converted into Boolean dummy variables with zero or one for each of their values.

Finding Duplicates

Duplicates 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 [17]. 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.

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 [7, 5]. 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.

Experiments With Algorithms Used In This Project

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 [15, 11, 12].

Sangani et al. [12] 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 [16], the authors found Linear Regression performed poorly compared to Decision Trees and Artificial Neural Networks.

Artificial Neural Networks

Tchuente et al. [15] 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.

Random Forest

Tang et al. [14] 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 [9], they test Random Forest, among other seven machine learning tree models, and conclude Random Forest is the best performing model in their experiment.

Adaptive Boosting and Gradient Boosting

In [12], five different models are trained 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.

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, [9] 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.

Support Vector Machines

Li et al. [8] 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. Pow et al. [11], not only applied a linear Support Vector Regression but also experimented the polynomial and Gaussian kernels for regression of target prices.

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.

Data

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, so that the scraper can then collect the information in those pages.

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.

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, 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 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 1.

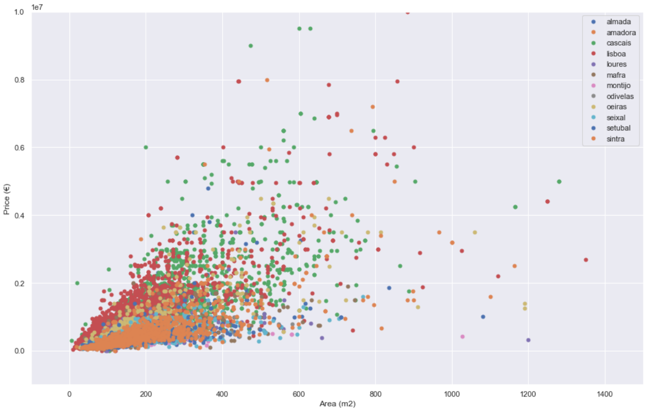


Figure 1: Price by area per municipality.

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.

Methodology

Data Preprocessing

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 stopwords, 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".

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.

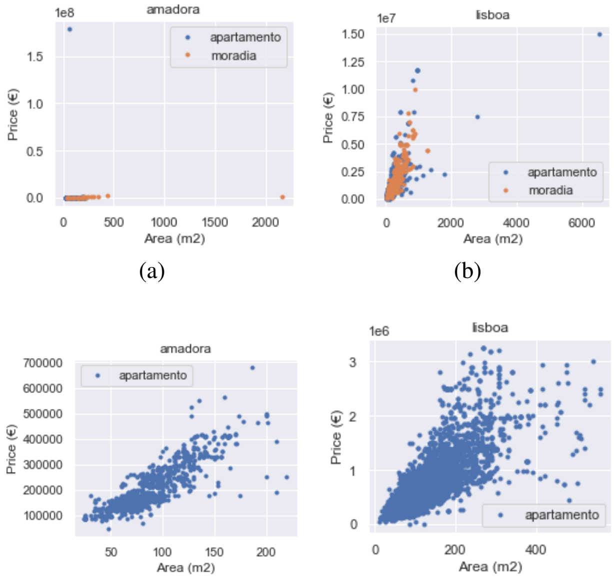
As mentioned in the Data Exploration Section, there were some entries which values did not made sense, such as Areas or Prices to 0 or very low values. These were simply eliminated, since they would compromise the performance of the model.

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 2, 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.

Another insight available in these Area-Price graphs is the variation of price with the number of rooms, depicted in Figures 3, 4, and 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.

Conserning 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 6, 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.



(a)

(b)

(c)

(d)

Figure 2: Relation between Area, Price and Property Type before (a)(b) and after (c)(d) removing outliers.

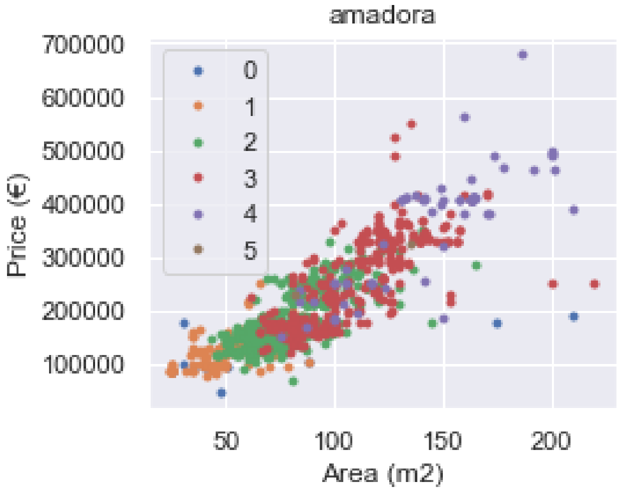


Figure 3: Relation between Area, Price and Number of Rooms (depicted by color) after removing outliers in Amadora.

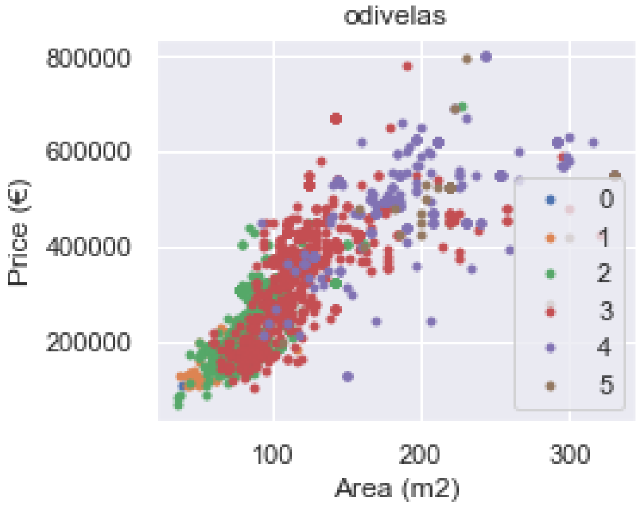


Figure 4: Relation between Area, Price and Number of Rooms (depicted by color) after removing outliers in Odivelas.

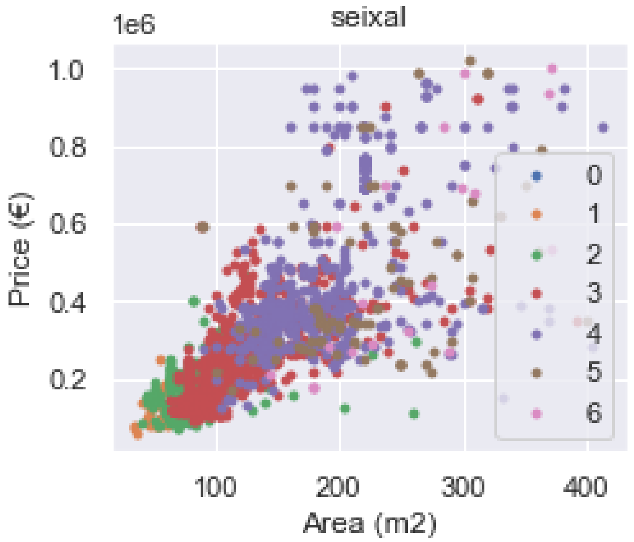
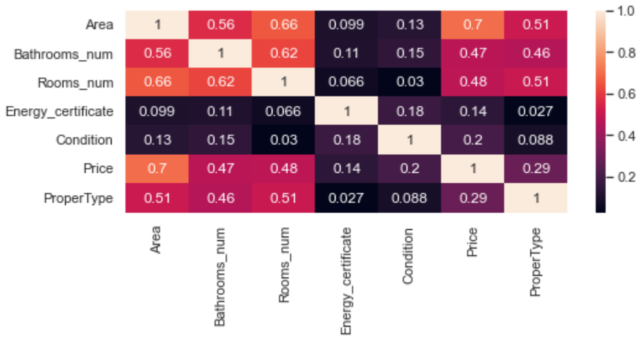


Figure 5: Relation between Area, Price and Number of Rooms (depicted by color) after removing outliers in Seixal.



	Area	Bathrooms_num	Rooms_num	Energy_certificate	Condition	Price	ProperType
Area	1	0.56	0.66	0.099	0.13	0.7	0.51
Bathrooms_num	0.56	1	0.62	0.11	0.15	0.47	0.46
Rooms_num	0.66	0.62	1	0.066	0.03	0.48	0.51
Energy_certificate	0.099	0.11	0.066	1	0.18	0.14	0.027
Condition	0.13	0.15	0.03	0.18	1	0.2	0.088
Price	0.7	0.47	0.48	0.14	0.2	1	0.29
ProperType	0.51	0.46	0.51	0.027	0.088	0.29	1

Figure 6: Correlation Matrix.

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.

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 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. Thus, 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, due to the lack of an existing control set, the accuracy of the duplicate detection had to be done manually. Thus, based on the aforementioned procedure, 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, a sample of entries from a specific municipality 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.

Due to a considerable number of categorical variables, such variables must be transformed in numeric ones 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.

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

Training

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 [13] 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.

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 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.

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.

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 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 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.

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.

RESULTS

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

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.

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. 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 to 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 iterations, 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.

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 more compound network to cover it.

Random Forest

In Random Forest experiments, the hyper-parameters were tuned to find the best ones for each model. 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.

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 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.

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 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*.

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 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.

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.

Evaluation

As an overview of all the previous results, we can consult Table 1 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	Grad. 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 1: Overview of MAPE (%) values for every algorithm tested.

CONCLUSION

Reviewing the main goals of this project, it is worth mentioning the necessity of an objective prediction model that could compute property prices in Lisbon and Setúbal and 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.

Contributions

The main contribution of this work consists on 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 its data distributed in its 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.

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.

REFERENCES

[1] 2018. 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`. (February 2018). (Accessed on 01/05/2021).

[2] 2020. Factors Affecting the Real Estate Market. (Jul 2020). `https://www.getsmarter.com/blog/market-trends/factors-affecting-the-real-estate-market/`

[3] 2021. Real Estate: Definition, Types, How the Industry Works. `https://www.thebalance.com/real-estate-what-it-is-and-how-it-works-3305882`. (May 2021). (Accessed on 02/06/2021).

[4] Rotimi Boluwatife Abidoye and Albert PC Chan. 2018. Improving property valuation accuracy: A comparison of hedonic pricing model and artificial neural network. *Pacific Rim Property Research Journal* 24, 1 (2018), 71–83.

[5] Hussain A Chowdhury and Dhruba K Bhattacharyya. 2018. Plagiarism: Taxonomy, tools and detection techniques. *arXiv preprint arXiv:1801.06323* (2018).

[6] Ineta Geipele, Linda Kauskale, Natalija Lepkova, and Roode Liias. 2014. 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. Vilnius Gediminas Technical University, Department of Construction Economics . . . , 1.

[7] Deepa Gupta and others. 2016. Study on Extrinsic Text Plagiarism Detection Techniques and Tools. *Journal of Engineering Science & Technology Review* 9, 5 (2016).

[8] Da-Ying Li, Wei Xu, Hong Zhao, and Rong-Qiu Chen. 2009. A SVR based forecasting approach for real estate price prediction. In *2009 International Conference on Machine Learning and Cybernetics*, Vol. 2. IEEE, 970–974.

[9] Mehrdad Ziaee Nejad, Jie Lu, and Vahid Behbood. 2017. Applying dynamic Bayesian tree in property sales price estimation. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE, 1–6.

[10] Gergo Pinter, Amir Mosavi, and Imre Felde. 2020. Artificial intelligence for modeling real estate price using call detail records and hybrid machine learning approach. *Entropy* 22, 12 (2020), 1421.

[11] Nissan Pow, Emil Janulewicz, and L Liu. 2014. Applied Machine Learning Project 4 Prediction of real estate property prices in Montréal. *Course project, COMP-598, Fall/2014, McGill University* (2014).

[12] Darshan Sangani, Kelby Erickson, and Mohammad Al Hasan. 2017. Predicting zillow estimation error using linear regression and gradient boosting. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 530–534.

[13] Michaela Štubňová, Marta Urbaníková, Jarmila Hudáková, and Viera Papcunová. 2020. Estimation of Residential Property Market Price: Comparison of Artificial Neural Networks and Hedonic Pricing Model. *Emerging Science Journal* 4, 6 (2020), 530–538.

[14] Yajuan Tang, Shuang Qiu, and Pengcheng Gui. 2018. Predicting housing price based on ensemble learning algorithm. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. IEEE, 1–5.

[15] Dieudonné Tchuente and Serge Nyawa. 2021. Real estate price estimation in French cities using geocoding and machine learning. *Annals of Operations Research* (2021), 1–38.

[16] Bogdan Trawiński, Zbigniew Telec, Jacek Krasnoborski, Mateusz Piwowarczyk, Michał Talaga, Tedeusz Lasota, and Edward Sawiłow. 2017. 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)*. IEEE, 51–54.

[17] Liting Wang, Li Zhang, and Jing Jiang. 2020. Duplicate question detection with deep learning in stack overflow. *IEEE Access* 8 (2020), 25964–25975.

[18] Hu Xiaolong and Zhong Ming. 2010. 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. IEEE, 384–386.