UNIVERSIDADE
BEIRA INTERIOR

Property Appraisal Platform
Versão Final após Defesa

Mauro Mendes Gaudêncio

Relatório de estágio para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor João Carlos Raposo Neves
Co-orientador: Eng. Vasco Lopes

Dezembro de 2022

Declaração de Integridade

Eu, Mauro Mendes Gaudêncio, que abaixo assino, estudante com o número de inscrição M10784 de Engenharia Informática da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o Código de Integridades da Universidade da Beira Interior.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 17 /12 /2022

Mauro Gaudêncio

Resumo

Este documento foi criado no âmbito do estágio realizado na empresa DeepNeuronic como parte do projeto ”Plataforma de Avaliação de Propriedades”. O objetivo do mesmo foi desenvolver modelos de aprendizagem automática capazes de avaliar preços do mercado imobiliário usando modelos inteligentes e um conjunto limitado de características capazes de descrever uma propriedade. Para atingir este objetivo o projeto foi dividido em duas partes principais. Na primeira parte foi feito um estudo intensivo do estado da arte, e criada uma coleção de bancos de dados extensiva, representante do mercado imobiliário no mundo inteiro. Com esta coleção disponível, um conjunto de características foram escolhidas de acordo com a sua relevância para o problema em questão. A segunda fase consistiu nos desenvolvimentos práticos principais, envolvendo a criação de modelos e melhorias nos bancos de dados. Para isso foram escolhidas as métricas mais relevantes, e foram avaliados os modelos nos bancos de dados iniciais, criando assim um conjunto de resultados base. Seguidamente, múltiplas experiências foram feitas, abordando diferentes áreas de interesse que podiam potencialmente melhorar os resultados base. No total quatro modelos diferentes foram avaliados e as experiências realizadas todas melhoraram os resultados base obtidos. De especial relevância, na última experiência propomos a transformação do preço da propriedade para uma variável objetivo que pode ser descrita como o ”Coeficiente do preço por metro de área quadrado comparado à média do subúrbio”. Usando esta variável os resultados obtidos foram consideravelmente melhores, estas experiências foram refeitas em um novo banco de dados consideravelmente mais complexo, verificando-se também que todas estas experiências melhoram os resultados obtidos inicialmente, reforçando a ideia que estas experiências podem ser usadas mesmo em bancos de dados mais complexos.

Palavras-chave

Mercado imobiliário, Avaliação de propriedades, Aprendizagem automática, Inteligência Artificial, Redes Neuronais

Resumo alargado

Este documento foi criado no âmbito do estágio realizado na empresa DeepNeuronic como parte do projeto ”Plataforma de Avaliação de Propriedades”, com o objetivo de concluir o Mestrado em Engenharia Informática na Universidade da Beira Interior. O objetivo do mesmo foi de desenvolver modelos de aprendizagem automática capazes de avaliar preços no mercado imobiliário usando modelos inteligentes e um conjunto limitado de características capazes de descrever uma propriedade. Para atingir este objetivo o projeto foi dividido em duas partes principais. Na primeira parte foi feito um estudo intensivo do estado da arte, descobrindo assim projetos existentes na área e quais tecnologias foram classificadas como as mais bem sucedidas. Foi criada também uma coleção de bancos de dados extensiva, representante do mercado imobiliário no mundo inteiro, para ser usada durante o treino e avaliação de modelos. Com esta coleção disponível, um conjunto de características limitadas foram escolhidas de acordo com a sua relevância para o problema. Os bancos de dados que continham todas estas características foram então tratados e escolhidos para serem usados na segunda fase. Este tratamento passou por uma análise da informação contida nos mesmos, bem como a utilização de certas técnicas para eliminação de *outliers*, dados que fogem da normalidade e que poderão causar problemas durante a fase de experiências.

A segunda fase consistiu nos desenvolvimentos práticos principais, envolvendo a criação de modelos e as suas melhorias. Para isso foram escolhidas as métricas mais relevantes, e foram assim avaliados os modelos base, criando assim um conjunto de resultados base para realizar melhorias. Neste sentido, múltiplas experiências foram feitas, abordando diferentes áreas de interesse que podiam potencialmente melhorar os resultados obtidos pelos modelos nos bancos de dados base. No total quatro modelos diferentes foram avaliados, sendo destes dois modelos baseados em sistemas de árvores (*Random Forest* e *XGBoost*), e os outros dois modelos baseados em redes neuronais (*Multilayer Perceptron* e *TabNet*). Estes modelos foram escolhidos devido à análise do estado de arte feita anteriormente, pois historicamente estes foram aqueles que obtiveram melhores resultados. As experiências realizadas podem ser divididas em quatro experiências principais. A primeira tinha o objetivo de diminuir a disparidade na quantidade de dados existente nos diferentes bancos de dados utilizados. Para isso foram feitas quatro abordagens diferentes, duas em que se aumentou o tamanho dos bancos de dados mais pequenos, e outras duas em que se diminuiu o tamanho dos bancos de dados maiores. Na segunda experiência foram feitas alterações no tipo de normalização utilizada durante o treino dos modelos, sendo que no total foram feitos cinco experiências com normalizações diferentes. Na terceira experiência foi utilizado uma API chamada ”Geopy” para criar um aumento das características dos bancos de dados a partir de duas características existentes (Latitude e Longitude). Finalmente na quarta experiência proponho a alteração da variável objetivo do preço da propriedade para um valor descrito como o ”Coeficiente do preço por metro de área quadrado comparado à média do subúrbio”. Esta alteração simplifica a relação existente nas três características principais, preço, área e localização, potencialmente ajudando os modelos a obter melhores resultados. Todas estas experiências melhoraram os resultados base, com especial melhoria registada durante as últimas duas experiências. Por fim todas estas experiências foram também realizadas em um novo banco de dados mais complexo com consideravelmente mais características, verificando-se também que todas estas experiências melhoraram os resultados obtidos nos dados base, reforçando a ideia que estas experiências podem ser usadas mesmo em conjuntos de dados mais complexos.

Abstract

This document focuses on the internship in the company DeepNeuronic as part of the project ”Property Appraisal Platform”. This project’s main objective was to develop machine learning models capable of inferring real estate prices using machine learning models and a limited set of features capable of describing a property. In order to achieve the objective, the project was divided into two major phases. In the first phase the state of the art was studied and a dataset collection was put together with the aim of creating a comprehensive representation of the real estate market all across the globe. With this dataset collection available, a set of features was chosen according to their relevancy for the main problem. The second phase consisted of the major practical developments, such as the model creation and dataset improvements. With this in mind, the most relevant metrics were chosen and the models were evaluated in the chosen datasets, creating a set of baseline results to improve upon. Afterwards, multiple other experiments were done, tackling different areas of interest that could potentially improve upon the performance of the models. In total, four different models were evaluated and all the experiments improved upon the baseline results. As an highlight, in the last experiment we propose the transformation of the target label from the property price to the ”Coefficient of the price per square meter compared to the suburb average”. Using this new target label, the results obtained were considerably better. All of these experiments were redone in a new more complex dataset, with all of the experiments improving upon the baseline results obtained in this dataset, reinforcing the idea that these experiments can be used even in more complex datasets.

Keywords

Real estate, Property Appraisal, Machine Learning, Artificial Intelligence, Neural Networks

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This report was written in the context of the curricular unit ”Projeto de Dissertação ou de Estágio em Engenharia Informática”, with the purpose of concluding the Master's Degree in Computer Science at the Faculty of Engineering in University of Beira Interior (UBI). This project focuses on the development of machine learning models for real estate appraisal, which consists of evaluating how much a real property is worth. Throughout the centuries, different methods and factors have been used in this process, some used statistics and comparisons, others were rooted in mathematics, while some of them used only human expertise as a means of solid and reliable results. Eventually, with the big technology leap, a lot of the data regarding past and current transactions started being stored in computer databases to which more and more people got access to. Naturally, the methods for appraisal also started using computers to perform more precise and faster calculations in an automated way.

As we move towards a more digital world every day, all of these processes will continue improving until a point where human oriented appraisal will be obsolete. Machine learning can be especially powerful at this kind of evaluations due to its ability to use and analyse the massive amounts of existing data and inferring patterns out of it. More precisely, it can analyse the existing data about past transactions and the market as a whole to learn and create models that accurately evaluate any property in a more precise way that a human ever could. With this project our objective is to create models capable of doing these evaluations using only a limited set of features instead of a fully detailed dataset.

The objective of this chapter is to introduce the startup *DeepNeuronic* in which the internship took place, as well as establishing the objectives and the overall project structure. For better understanding of the topic, the state of the art was studied and will be thoroughly explained in chapter 2. Afterwards, an extensive dataset collection was also created, composed of many datasets from many different countries, allowing for a good representation of the real estate market all around the globe. By analysing this collection, a set made of the most relevant features for real estate appraisal was created to be used in chapter 4 for model evaluation. All the technologies used as well as the methodologies used are explained in chapter 3 with, finally, the results being presented and discussed in chapter 4.

1.1 *DeepNeuronic*

DeepNeuronic is a Portuguese tech startup created in the end of 2020 and is located in Covilhã, Portugal. The name stands for ”Deep Neural Systems for Automatic Vision” and its focus lies in developing highly efficient machine learning-based solutions to automate daily problems. Currently, the main product is a CCTV automation framework that by analysing images in real-time, provides information about several possible threatening activities, thus increasing public safety as well as decrease monetary losses. The project being developed in this dissertation lies within DeepNeuronic's strategy of developing solutions for the real estate market that extend common tools for the final user.

1.2 Objective

The objective of this internship is to create machine learning models capable of evaluating the real estate market. These models have to be able to provide accurate predictions for any property in a fast and reliable way by only analysing the input given. This input has to be limited to only a set of features capable of representing a property across any country. For this, a study of different datasets as well as the state of the art has to be done before any model creation and training. Once the set of features has been chosen, the models will be trained and experimented upon with different techniques and technologies, with the objective of improving each model performance to the best possible one. The objectives of this work can be summarized as follows:

1. The **first phase** of this project will be the analysis of the state of the art, more specifically, what techniques and machine learning models have been tested and which ones achieved the best results.

2. The **second phase** is to build a dataset collection. This collection will be comprised of multiple public datasets that will be thoroughly analysed and adapted if needed. The objective is to have a final dataset collection that contains data from multiple sources, allowing for a good representation of the real estate market all around the globe.

3. The **third phase** is to train and evaluate the models on the datasets, creating baseline results. The results of these evaluations will be extremely valuable as a means of comparison for later evaluation of other models.

4. The **fourth phase** is to improve upon the baseline results. This can be achieved by multiple ways, such as creating brand new features capable of enhancing the results, improving the quality of the datasets used or improving the models.

1.3 Report structure

This report is organized in the following way:

- The first chapter - **Introduction** - presents the internship, the company, the objectives of the internship and the report structure.

- The second chapter - **State of the art** - explains the related work that has been developed so far in the field by other experts as well as the data collection that has been made.

- The third chapter - **Technologies and Methods used** - outlines all the tools used and especially the methods and the thought process behind them during the course of the project.
- The fourth chapter - **Results and Discussion** - presents the results in a concise manner and discusses them, presenting new conclusions or questions that have risen from them.
- The fifth chapter - **Conclusion** - ends this report with a brief conclusion.

Chapter 2

State Of The Art

2.1 Introduction

In this chapter the state of the art is studied with several related works being presented. These works will be mainly focused on advanced techniques such as machine learning models and what kind of different techniques and features were used to maximize performance. After this study, our dataset collection is presented, with this collection being analysed and cleaned, using different outlier detection techniques.

2.2 Related Work

The real estate market has an extensive literature to analyse. In this context, multiple types of methods have been implemented and explored but they can be mainly grouped into two different types, the traditional valuation methods and the more modern advanced methods that use machine learning technology.

The traditional methods can be explained as methods mainly based on observation. They are grounded in direct comparisons or collections of information allowing the evaluation of the market price using regression models. The fact that these were collected by human observation means that they are also highly subjective. Examples of these types of methods would be methods such as the comparable method or the income method. The comparable method uses the sales of similar housing in the market to evaluate the market price of a certain property. Naturally not all houses are made the same and can be directly compared leading to necessary adjustments when necessary. For this reason the homogenization is very important. The income method approach ties the valuation of the property directly to the ability of said property to generate income.

The advanced methods are more quantitative and less observation focused. These methods are usually more precise and take in consideration more factors compared to the traditional methods. Examples of advanced methods are machine learning models such as neural networks or fuzzy logic methods. In this work, we focus on advanced methods for the development of a computational method to predict real estate prices from a set of house related features.

The most commonly used algorithms are Linear Regression, Random Forest, XGBoost, Artificial Neural Networks(ANNs) and Support Vector Machines(SVMs). Usually when compared between each other the best performance is achieved through Random Forest or XGBoost as seen in the works of [6],[7],[8]. It is also notable that in the work of [6] the author states that the human expert appraiser performs at an average error of 12%, value that most works using machine learning have lowered, proving the power of machine learning algorithms in real estate.

A recent work from [9] used a small dataset of 2266 datapoints to create multiple advanced machine learning models using ensembles of regression trees, Support Vector Machines, k-nearest neighbors and multi-layer perceptrons. Five fold cross-validation was used to avoid biases and after different tests, ensembles of regression trees outperformed the others. The relative best median absolute error obtained was of 5.71% and, like the authors concluded, this value is significantly smaller than the ones provided by a classical linear regression model, therefore highlighting the potential advantage of more complex machine learning algorithms. This conclusion is especially notable due to the dimension of the dataset used, since machine learning improves significantly according to the quality and dimension of the data used, it is to be expected that the performance improves even further when better datasets are used.

A work from [10] proposed 2 traditional algorithms and compared them to 3 machine learning algorithms. One of the traditional algorithms was based on the analysis of the N-Latest Transactions in the region, whereas the other used the N-Nearest Similar Properties to calculate the estimated price. These algorithms were then compared to a model that used decision trees, a multilayer perceptron neural network and a linear regression. In all experiments the machine learning models performed better than the traditional algorithms, reinforcing once again the value of using machine learning in real estate appraisal. Less common algorithms can also be used successfully as demonstrated by works[11],[12]. The first one of these used a ridge regression coupled with a genetic algorithm to obtain better results when compared to multiple regression and a ANN. Meanwhile the latter work used four algorithms including C4.5, RIPPER, Naïve Bayes and AdaBoost to predict whether the price at which the houses were sold was greater or less than the listing price, transforming the problem into a more classification oriented one.

Another work [1] analysed three machine learning algorithms(Random Forest, XGBoost and LightGBM) as well as two ensemble techniques. Ensemble techniques utilize several machine learning models coupled together in order to achieve better results. The model that achieved the best results at generalising the data used the predictions of the Random Forest and LightGBM models as features for the XGBoost model. This ensemble technique is called Stacked Generalization Regression and a comparison between the actual and predicted values of this model can be seen in figure 2.1.

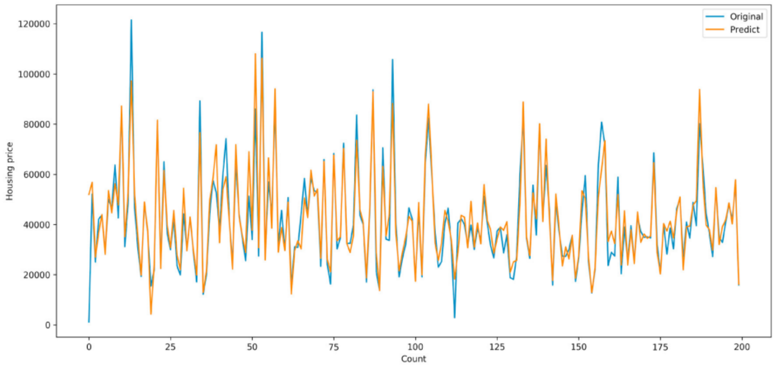


Figure 2.1: Comparison of Stacked Generalization Regression's predicted results and original test set. Image taken from [1].

Another very important factor are the features of the dataset used. Some works tried to implement image oriented datasets and traditional housing attributes datasets at the same time. A prominent approach is work[2] which relied on three datasets, one traditional dataset that included structural, neighborhood and location features, a second dataset comprised of street images and finally a third dataset comprised of aerial images. Using these datasets, the authors represented the appeal that each neighborhood had across the Great London, as seen in the figure 2.2.

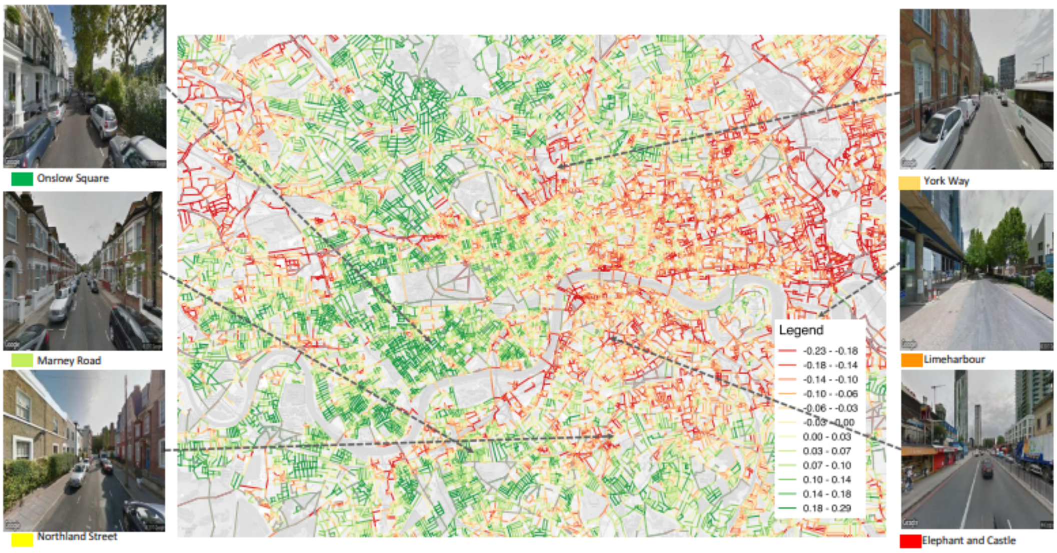


Figure 2.2: Map illustrating the visual appeal of neighborhoods across Greater London. Image taken from [2].

The work used CNNs(Convolutional neural networks, mostly used to analyze images) for image feature extraction, features that were then grouped with the traditional features and fed into a hedonic model for regression. The structure of the network used is depicted in figure 2.3 with "S" representing the street view images, "A" the aerial photos, "X" the traditional features dataset and with "F(S)" and "G(A)" representing the CNNs.

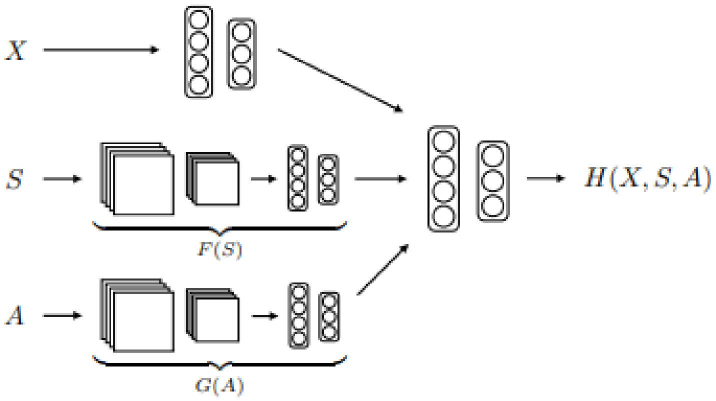


Figure 2.3: Fully nonlinear model network structure. Image taken from [2].

The result yielded by this work allowed the authors to conclude that the model augmented with features extracted from street and aerial images performs better than the model without image features.

In a more recent and very similar work [13] the authors developed a model by using a spatial neural network that uses a CNN to extract features from satellite images and then uses them together with numeric features in order for the regressor to estimate the real estate price. The experimental results show a higher performance compared to most mainstream models. The dataset used was provided by the Chinese large-scale second-hand housing trading platform Lianjia.com and was comprised of 79212 records after data cleaning. The big advantage that these deep learning models gained by using images and CNNs is a human like understanding of the property and neighbourhood factors that cannot be explained through tabular data.

Another work [3] also used images but with a different approach. The images here did not focus on the neighbourhood, instead they were frontal images of the property, as seen in figure 2.4.

Figure 2.4: Examples of house pictures used by [3].

This work employed a random walk graph by utilizing the location features to transform the problem into a sequence learning problem. This way, a novel framework was proposed by using recurrent neural networks which are particularly designed to solve sequence related problems. The prediction process involves creating multiple random sequences for each house and then averaging the predictions made in each of the sequences in order to obtain the final prediction. Another image based work [14] used heterogeneous data analysis comprised of Google satellite maps and public facilities to verify whether the use of such data could improve prediction accuracy. The authors adopted a spatial transformer network to extract the image features from the maps and proposed a joint-self attention mechanism to identify the most important features that would interest buyers. This model outperformed all other models in the experiments.

One of the most recent works with very high performance results is a work by H. Peng et al [4]. This work presents Luce, a predictive model that uses a heterogeneous information network(HIN) to model each house, as seen in figure 2.5.

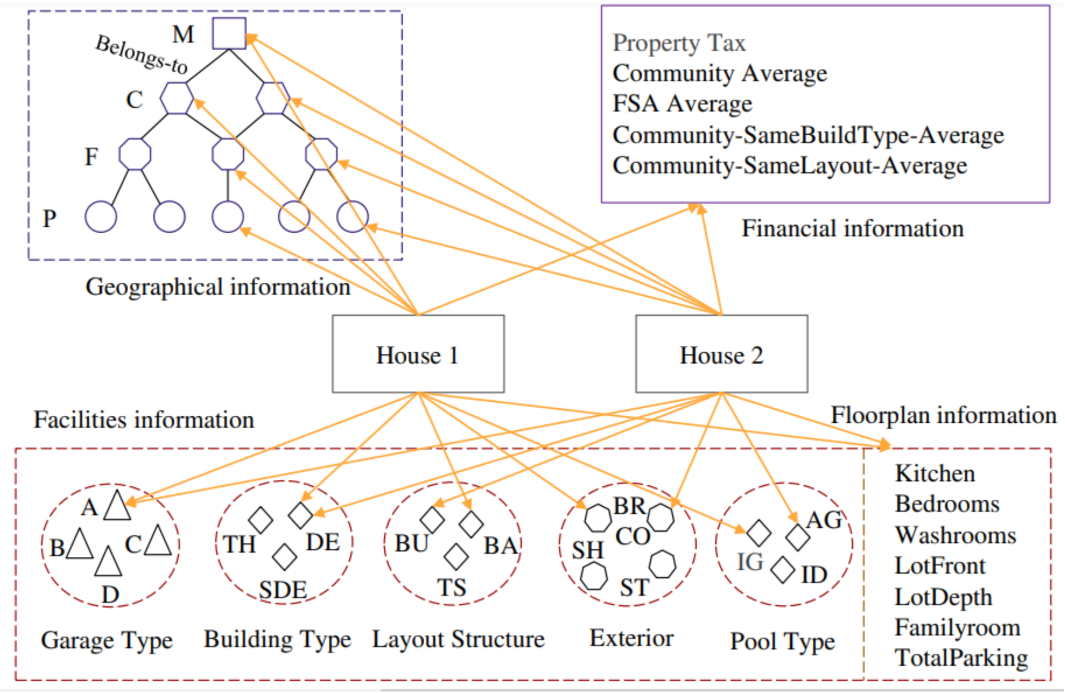


Figure 2.5: The Heterogeneous information network used by Luce. Taken from [4].

Afterwards, a Graph Convolutional Network (GCN) uses the HIN to extract the spatial information and employs a Long Short Term Memory (LSTM) network to learn the temporal price change for all the houses within the HIN. With this approach, Luce can use the prediction and transaction history to predict the current price for each house. The main advantage of Luce is that it solves the limited data and data sparsity problem with its usage of the GCN-LSTM units, outperforming state-of-the-art methods and the valuation of humans.

Fuzzy Logic can also be applied in real estate appraisal [15], [16] but it has not been explored as thoroughly as the previously mentioned machine learning models.

Cluster analysis is also an important method due to the heterogeneity and homogeneity of property data. This type of analysis groups into clusters similar properties which can be useful for appraisal, especially when paired up with other methods. A work from A. Malinowski et al [17] used six variants of traditional expert algorithms, that used sales comparison approach, with each variant being based on a different partition of the city. These partitions were created by six different clustering algorithms, out of these six, two were crisp clustering algorithms (K-Means and density-based OPTICS) and the other two were fuzzy clustering algorithms (C-Means and Fuzzy Adaptive Clustering). One of the main conclusions taken from this work was the fact that clustering may be used to delineate the boundaries of homogeneous sub markets and improve precision.

2.3 Datasets collection

As part of the research phase, a group of datasets were collected for later usage in model training and performance testing. This data collection process was done manually by exploring existing datasets on websites such as Kaggle[18] and Github[19]. A table of all collected datasets can be seen in the table 2.1.

The objective of this dataset collection phase is to later use these data to train and evaluate the performance of the models. Having a good variety of datasets from different representations can be good to ensure model flexibility and better performance. Once all the datasets were collected a further analysis was made to know how many times each feature showed up. A plot of the most occurring features can be seen in figure 2.6.

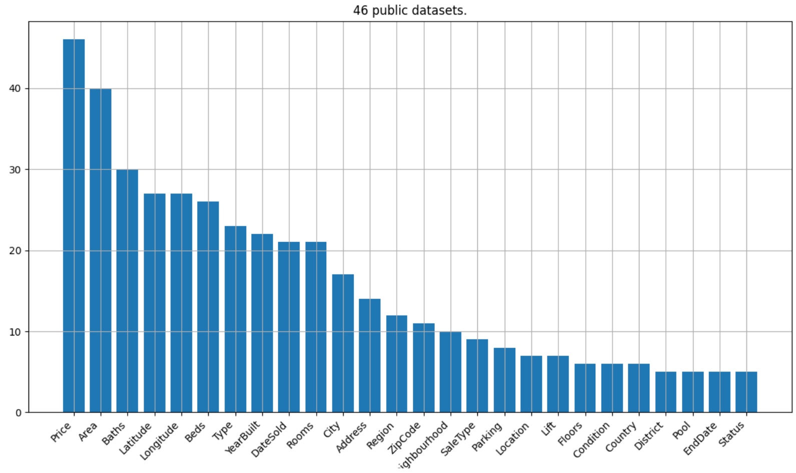


Figure 2.6: Histogram of house features. The features of 46 datasets were renamed into standard format allowing to determine the number of occurrences of each feature along the different datasets.

By analysing the plot we can easily see that the feature Price shows up in every dataset. This is obviously expected because this is our target feature, without it we could not train supervised models. After that, the most used feature is Area, existing in 40 out of the 46 datasets. After Area, it exists a big drop and the next feature is the Baths feature, followed by Latitude, Longitude and Beds.

In order to choose the best datasets a decision was made to choose datasets that had the same features. Thus, the features chosen were all of those that showed up at least 20 times. Nevertheless, even though all these features showed up more than 20 times, it does not mean that 20 datasets have them all. For example, if a dataset has all of these features except Baths it will count towards all those features existing but it will not be included in the final dataset because it is missing one of the features (Baths). This factor was especially relevant due to the YearBuilt feature, with it there were only 2 datasets that included all features with more than 20 occurrences, without it, there were 7. This big change meant that it was better to not use this feature in favor of bigger datasets, and the same could be said for the features

Rooms and Type. With this choice of features the datasets used are those that appear as bold in table 2.1.

Dataset	**Number of datapoints**	**Number of features**	**Region covered**
Bhubaneswar region [21]	546	13	Bhubaneswar, India
Bengaluru city [22]	13320	9	Bengaluru, India
REIT [23]	1883	26	New York, US
King County [24]	**21613**	**21**	**King County, US**
Ames, Iowa [25]	2918	80	Ames, US
Analyze Boston [26]	214967	13	Boston, US
Taiwan [27]	414	8	Taiwan
Russia [28]	540000	13	Russia
Riga [29]	4689	13	Riga, Latvia
Madrid [30]	21742	58	Madrid, Spain
Japan [31]	-	38	Japan
NYC 2016/17 [32]	84548	14	New York, US
Melbourne [33]	**34857**	**18**	**Melbourne, Australia**
São Paulo [34]	13640	15	São Paulo, Brazil
Victoria [35]	100000	12	Victoria, Australia
Philadelphia [36]	615	24	Philadelphia, US
USA [37]	4600	18	USA
Saudi Arabia villas [38]	1417+930	10	Saudi Arabia
Saudi Arabia [39]	3799	23	Riyadh/ Jeddah/ Dammam/Alkhobar, Saudi Arabia
’07-’17 South Korea [40]	5891	28	South Korea
Tunisia [41]	12748	8	Tunisia
Immo Scout 24 [42]	10552	18	Germany
Australian 2021 [43]	1500+	7	Australia
Argentina [44]	**1048576+**	**19**	**Argentina**
Colombia [44]	**1048576+**	**19**	**Colombia**
Ecuador [44]	**819175**	**19**	**Ecuador**
Peru [44]	**288069**	**19**	**Peru**
Uruguay [44]	**385558**	**19**	**Uruguay**
Seattle Airbnb [45]	7576	13	Seattle, US
Seoul [46]	4021	10	Seoul, South Korea
Arizona [47]	563	8	Arizona, US
Sindian Taiwan [48]	414	7	New Taipei City, Taiwan
Raifhack [49]	282766	25	Russia
California 1990 [50]	20640	10	California, US
India metropolitan [51]	32963	38	India
Belo Horizonte [52]	6000	10	Belo Horizonte, Brazil
D.C. [53]	**158957**	**30**	**D.C., US**
Amsterdam [54]	924	7	Amsterdam, Netherlands
Kuala Lumpur [55]	53883	8	Kuala Lumpur, Malaysia
Nashville [56]	56000+	22	Nashville, US
Zameen.com Pakistan [57]	**190904**	**10**	**Pakistan**
Beijing [58]	318852	16	Beijing, China
Craigslist [59]	384977	17	US
Portland [60]	25724	16	Portland, US
Paris [61]	10000	14	Paris, France
Perth [62]	**33656**	**13**	**Perth, Australia**

Table 2.1: List of datasets collected. The datasets in bold are the one that were chosen for usage after analysis.

2.3.1 Data Cleaning

After the data collection phase was finished all of the chosen features in each dataset were converted into a standard format, for instance, all prices were converted into € and all areas into m^2. Furthermore, the features that were not chosen were removed from the datasets. All the rows that had at least one empty cell and all nonsensical data(such as negative prices or areas) were also removed. At this point all that was left to do was outlier removal.

2.3.1.1 Outlier Removal

An outlier is a data point that is significantly different from most other data points. These outliers can mislead and make machine learning methods perform worse. Due to this, it is important to detect them and remove them. For outlier detection, several methods were implemented such as Grubbs test, zscore and IQR test.

2.3.1.2 Grubb's test

This test was named after the statistician Frank E. Grubbs and it is based on the assumption that the data can be reasonably approximated by a normal distribution. It detects one outlier at a time, removes it and repeats until there are no outliers remaining. The test is defined as:

$$G = \frac{\max_{i=1,\ldots,N} \left|Y_i - \bar{Y}\right|}{s}, \quad (2.1)$$

with $\bar{Y}$ denoting the sample mean and s the standard deviation. Using G it is possible to know if there exists outliers at significance level α if:

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/2N,N-2}}{N-2+t^2_{\alpha/2N,N-2}}}, \quad (2.2)$$

with $t^2_{\alpha/2N,N-2}$ denoting the upper critical value of the t-distribution with N−2 degrees of freedom and a significance level of $\alpha/(2N)$.

2.3.1.3 Interquartile Range(IQR)

This method uses the interquartile range to detect outliers, this range is defined as the difference between the 3rd and 1st quartile of the data. The usual procedure when using this method is to multiply the IQR value by 1.5 and using it as a border to differentiate outliers from non-outliers. A representation of the method can be seen in figure 2.7.

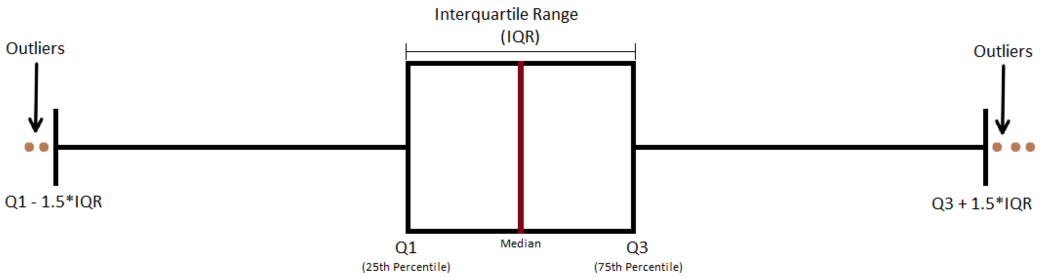


Figure 2.7: IQR method. This method uses Interquartile range to differentiate outliers from non-outliers.

2.3.1.4 Z-score

The standard score or Z-score is useful to know how many standard deviations away a data point is from the mean. Assuming a normal distribution, any data point that lies +/- 3 standard deviations can be considered an outlier. Using this knowledge we can use z-score to detect the outliers.
The Z-score of a certain data point can be calculated as:

$$z = \frac{x - \mu}{\sigma}, \quad (2.3)$$

with x being the data point, μ being the mean and σ being the standard deviation.

2.3.1.5 Comparison between methods

After all the methods were implemented, they were compared in order to understand which one was the most suitable for the data. In the figure 2.8 it is possible to see a comparison between the different final distributions of each dataset for the feature "Price" after each method was used.

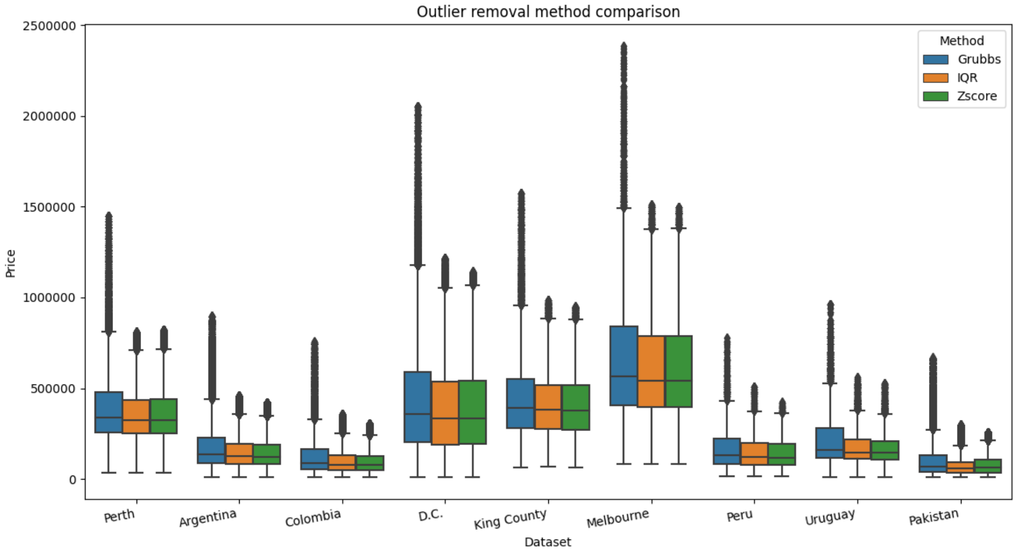


Figure 2.8: Comparison between methods. The methods were used in all datasets and then compared in order to understand which one was the best to use.

By analysing the figure it is easy to see that the grubb's test removed the least amount of outliers. This is likely due to the fact that the distributions of the datasets are not normal distributions, where the test performs best. Comparing the IQR and Z-score methods, the figure seems to point out that both methods are equally good, with z-score being very slightly better, as such, this was the chosen method out of the 3 analyzed.
After these processes the datasets are ready for usage in the model building and evaluation phase.

Chapter 3

Technologies and Methods used

3.1 Introduction

In this chapter all of the aspects regarding the multiple technologies and methodologies used in the project are thoroughly explained. The main technology used was the Python programming language which was used for code development. The methodologies are especially important as they allowed for continuous analysis and for a good exploration of the datasets and the models.

3.2 Technologies used

3.2.1 Python

Python is a high-level programming language that contains a wide range of libraries. These libraries make this language especially good for machine learning due to its simplicity, flexibility and access to great tools and frameworks. Some of the most important libraries used were Numpy, Pandas, Sklearn, MatPlotLib and PyTorch. All code developed during this project used this language.

3.2.2 PyCharm

PyCharm is a integrated development environment (IDE) specifically created for Python. It is very simple to use and allows for many helpful features such as executing code and debugging.

3.2.3 GeoPy

Geopy is a Python library with several popular geocoding web services. It was very important in this project as it allowed for some very important data augmentation on the datasets.
By providing coordinates to geopy, this library makes API requests to several geocoding softwares such as Nominatim, Google Maps, Bing Maps and others. These requests then return information about the location provided, such as, the country where its located in, the city, road, address and other useful informations.

3.3 Tree-based Models

Tree-based models rely on decision trees as the base for their output. A decision tree is a flowchart-like structure that represents a set of decisions and their consequences. Each node represents a condition and each branch coming of a node represents the outcome.

These are great models for tabular data because they are simple, easy to implement and easy to interpret. They also provide decent results on simple datasets that can be easily represented by a set of rules. A simple decision tree can be seen in figure 3.1.

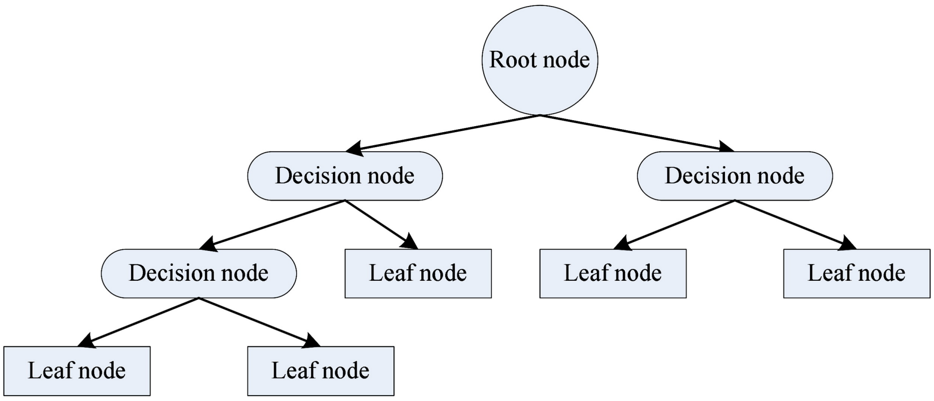


Figure 3.1: Example of a decision tree.

3.3.1 Random Forest

The random forest model is one of these tree-based models and its main focus is the usage of technique called bootstrap aggregation, or more simply, bagging. This technique consists in sampling the main dataset into multiple random subsets and then building a multitude of decision trees, in parallel, each using the different subsets. The model will then make its final classification/prediction based on the output of each decision tree. Depending on the type of problem, the model might pick the mean of all predicted values, or simply pick the value that was predicted the most. With this approach, the Random Forest model solves two of the main problems that exist within decision trees: Overfitting and high variance. It also solves the issue of high dimensional data because each tree only has a subset of the full data.

3.3.2 XGBoost

XGBoost stands for eXtreme Gradient Boosting, and the main difference when compared to the random forest model is that instead of using bagging, it uses a technique called gradient boosting. This technique is part of the boosting techniques, in which the models are trained sequentially instead of in parallel. At each iteration, each model takes on the weak points of the previous model and improves on it, learning and getting better predictions each time. Gradient boosting is a special type of boosting in which the error is minimized by using the gradient descent algorithm.

The XGBoost model is incredibly optimized, fast and widely accepted as the best tree-based model. In figure 3.2 it is possible to visualize how good this model is when compared to other popular algorithms.

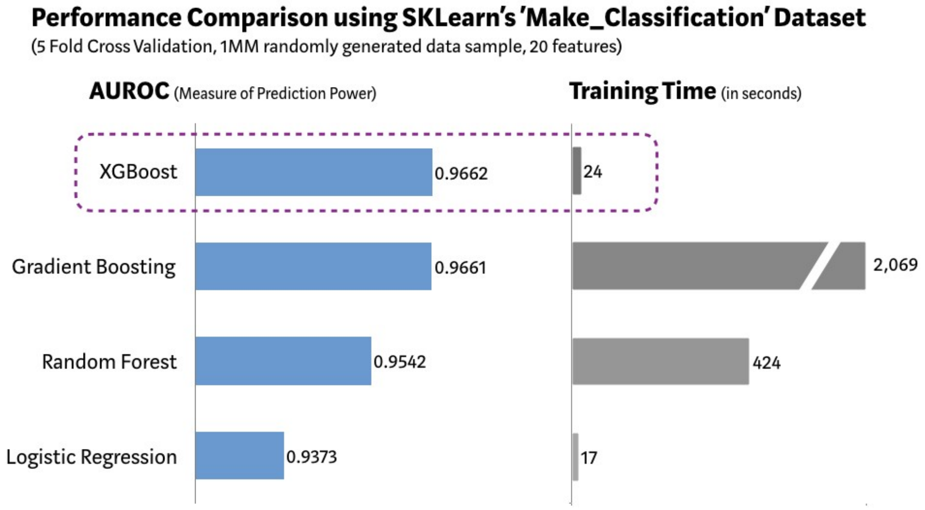


Figure 3.2: XGBoost vs. Other ML Algorithms using SKLearn's Make_Classification Dataset. (source: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d)

3.4 Neural Network Models

Neural Networks are inspired by the way a brain works on any animal. They try to replicate the behavior of biological neurons by receiving an input, and outputing a number to the next neuron. The calculation is made by using weights that are adjusted during the learning process in order to achieve better results. The neurons are organized into layers and many different architectures exist. In this project two architectures were used, the Multilayer perceptron and a model known as TabNet.

3.4.1 Multilayer perceptron

A multilayer perceptron is a simple neural network with an input layer, an output layer and one or more hidden layers. Every neuron from one layer connects with every neuron from the next layer, making this kind of neural network a fully-connected neural network. It is also a feedforward neural network because during training, the inputs of each neuron are multiplied by the weights of that neuron, and then the activation function is applied, with the result of this computation being fed into the next layer. Once the last layer (output layer) has been reached, the backpropagation algorithm is used to start the learning process of the MLP.

Backpropagation is the process on which the gradient of the loss function is computed across the entirety of the neural network and the necessary adjustments to the weights of each neuron are made. Across several iterations, this process allows for the neural network to learn and get better results each time.

In the figure 3.3 it is shown an example of an MLP with one hidden layer.

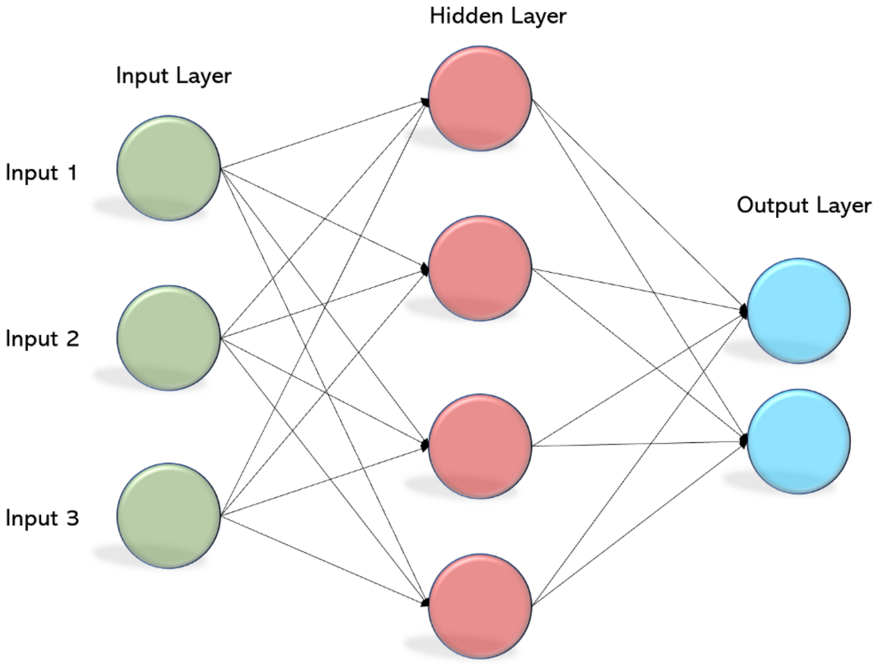


Figure 3.3: Simple MLP architecture.(source: https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron)

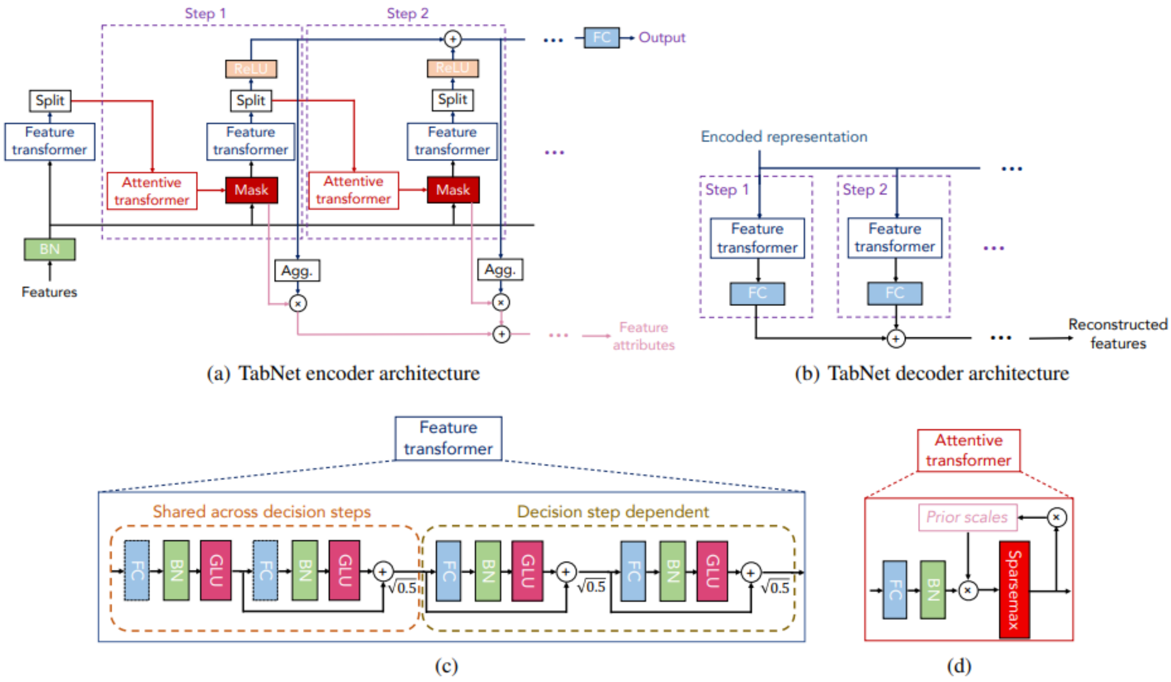
3.4.2 TabNet

The TabNet model [5] is a Deep Neural Network developed in 2019 that outperformed the best decision tree models(XGBoost, LightGBM, ...) on tabular data.

TabNet took inspiration from decision trees in its mapping functionality and it keeps a lot of its benefits while improving on its design. Its feature selection employs sparse instance-wise feature selection and builds a sequential multi-step based architecture. Each step contributes to the selected features and it allows for the model to mimick ensembling by simply increasing the number of steps and dimension. This model uses a learnable mask for soft selection of the salient features obtained by a transformer.

This model usage of learnable masks makes it possible to know how much contribution each feature had to the decision. This insight of the model behavior is a very big advantage, as it allows for better interpretability, understanding of how the model works and how it can be improved. Finnaly the TabNet model also contains a decoder to reconstruct the features from the encoded representations the model used.

The TabNet architecture is depicted in figure 3.4.



(a) TabNet encoder architecture

(b) TabNet decoder architecture

(c)

(d)

Figure 3.4: TabNet architecture [5].

3.5 Methodology

In this section the methodology used during the experiment phase is presented with all the different experiments and the thought process behind them being explained. Using the collection of datasets that was compiled previously in chapter 2.3, an initial experiment was done in order to have baseline results for the following experiments, all of which had the same objective: improve upon the quality of the baseline results. With this objective in mind, each time the results of an experiment were analysed, a new experiment would be done in order to improve upon the previous experiment. All of the thought process regarding the experiments done during the course of this project are explained ahead and the results are presented in chapter 4.

3.5.1 Research Strategy

First experiment - Getting baseline results: In this experiment our goal was to gather the baseline results so that it was possible to know if later experiments were successful or not. The datasets used in this experiment were the ones described in chapter 2.3 with z-score outlier removal applied. The main focus during this experiment was on fine tuning each model's parameters. For this, the most used strategy was to use the library known as GridSearchCV, a tool that allows for an exhaustive search over combinations of specified parameters for an estimator. By using GridSearchCV, one can experiment with a wide array of different parameters and know what combination performed better. This can be very useful because it allows for a good idea on how each parameter impacts the results. After several variations of the initial parameter grid, it became very difficult to notice any significant improvement and with this, the best results obtained for each model were classified as the baseline results. The best parameters obtained for each model in this phase were also the ones used in all other experiments because fine tuning in every new experiment would become very slow and time wasting for minimal gain.

Second experiment - Dataset augmentation: The second experiment was focused on improving the datasets. With this in mind the first adjustment made to the datasets was regarding the price. In the baseline datasets, the average price on multiple datasets was very low, with plenty of prices below 1000€ and even 100€. This happened due to several reasons, such as the conversion to € from other less valuable coins. For example, a house in Colombia that is worth 1000000 COP(Colombian Pesos) is converted to only 231,32€. Its possible to analyse the price distribution of the baseline datasets in the figure 3.5.

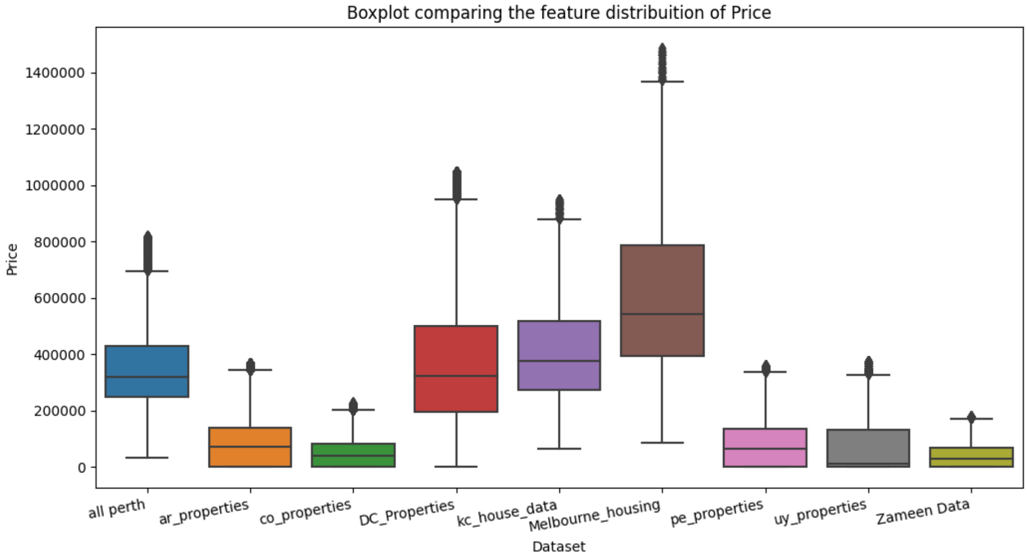


Figure 3.5: Price Distribution on the Baseline Datasets.

Ideally, the outlier removal analysis made would have gotten rid of these cases but, since there are so many, the average price in these datasets was very low, and thus, they were not considered outliers. In order to solve this problem, in this experiment, all datasets were changed such as that the minimum price accepted was 10000€. The new distribution can be seen in the figure 3.6.

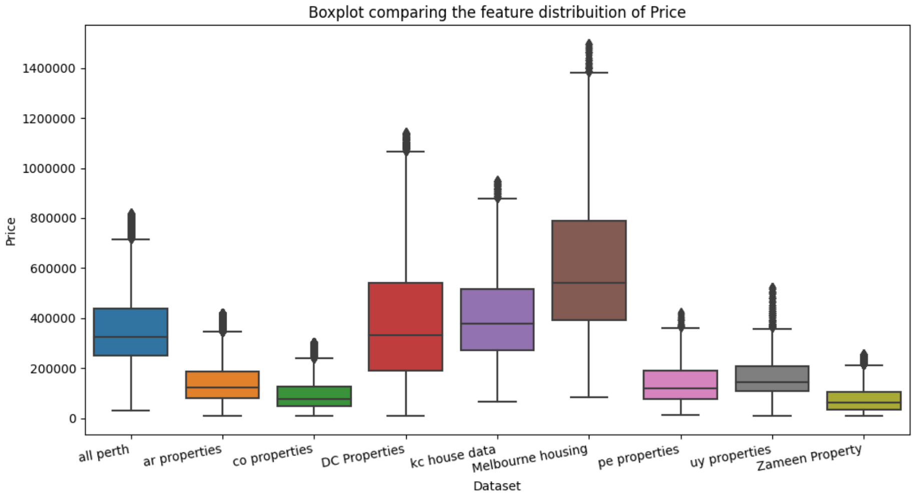


Figure 3.6: Price Distribution on the Datasets after the imposed minimum Price of 10000€.

After this adjustment was made, another possible problem was identified regarding the datasets: their size. As shown in figure 3.7, the datasets have very different sizes with the smallest dataset having 2275 data points and the biggest one having 117758 data points(51.7 times bigger).

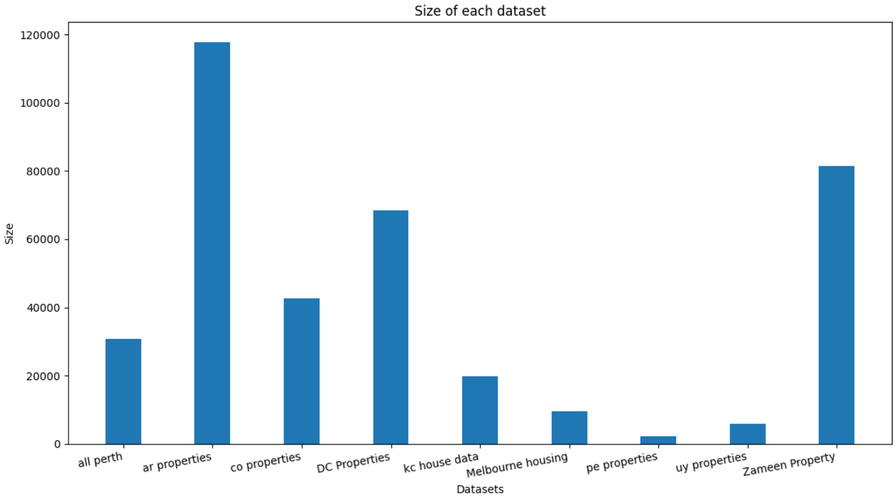


Figure 3.7: Size of each Dataset.

While this is not necessarily a problem when training on each dataset individually, when grouping them all up there will be a big disparity in the representation of each dataset, leading to possible problems, such as bigger datasets dwarfing smaller datasets to irrelevance. With this in mind, in order to balance each dataset size, the following strategies were tested:

- Size augmentation: The size of the smaller datasets was increased by sampling random rows up to the size of the largest dataset;
- Size reduction: The size of the bigger datasets was decreased by removing random rows.

Third experiment - Normalization:

Normalization is the process in which the data is transformed from its original state into a certain range. This can help plenty of models to achieve better results as it balances out each feature so that no feature biases the model performance, making all features equally important. It is considered a must do process especially in neural network methods. This experiment focused on how different types of normalization would impact the final results. To compare the performance of different normalization techniques the following experiments were carried out:

- No Normalization;
- Standardization - removes the mean and scales each feature to unit variance.;
- Min-Max Scaling - Transforms the features by scaling each feature to a given range, the range used was 0-1;
- Log transformation - Transforms the features by replacing each value with the logarithm of that value;
- Standardization and Log transformation;

  In figure 3.8 its possible to visualize the difference between each of the methods.

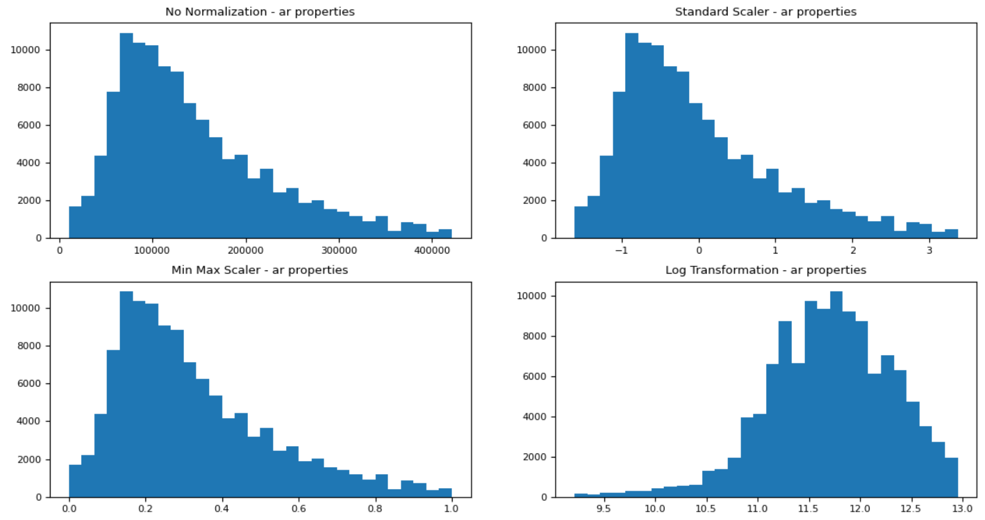


Figure 3.8: Difference between the normalization methods when applied to the perth dataset on the target variable (Price).

Fourth experiment - Feature augmentation(Geopy):

Feature augmentation is the process of exploiting the existing features to create new ones increasing the dimensionality of the dataset. This process is usually one of the best strategies to improve results in machine learning since it relieves the model from the burden of infering false correlations between original features. In this experiment, the feature augmentation used two of the main features, Latitude and Longitude. By using a python geolocation service called Geopy [63], its possible to obtain several informations regarding a location by using only its latitude and longitude. These informations allowed for the following new features:

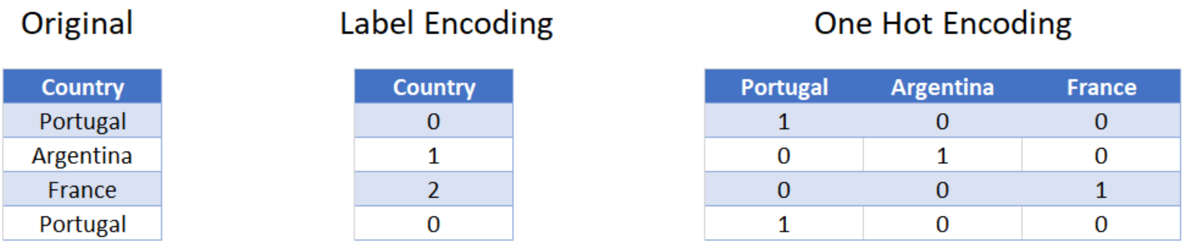
- Country;
- State;
- Municipality;
- Road;
- Suburb;
- Postcode.

The feature "Road" ended up not being used due to lots of missing data, making it very unreliable. One of the datasets - Zameen Data - also did not work well when using geopy, with most of the features being always missing. Due to this, for this experiment and any involving geopy, this dataset was discarded.

Since these features were the first categorical features used, in order to allow the models to train, they were encoded using the OneHot Encoding strategy, with the exception of the Random Forest model, which used the Label Encoding strategy instead because the increase in complexity while using One Hot Encoding was too much and made training too long. These methods consist of the following:

- One Hot Encoding - The categorical data is transformed into numeric data by splitting the original column into multiple columns, one for each unique categorical value. These columns are then filled with 0 and 1, with 1 corresponding to a "True" value and 0 a "False" value.
- Label Encoding - Each unique value in the target column is transformed into a unique numeric value.

While Label Encoding is considerably simpler, it may induce the model into miss understanding the data to have an order(3>2>1>0). This makes One Hot Encoding the best method to use when encoding categorical data that has no order, like those created with Geopy. An example of both encoders can be seen in image 3.9.



Original

Country
Portugal
Argentina
France
Portugal

Label Encoding

Country
0
1
2
0

One Hot Encoding

Portugal	Argentina	France
1	0	0
0	1	0
0	0	1
1	0	0

Figure 3.9: Difference between Label Encoding and One Hot Encoding.

Theoretically speaking, since all these new features are related with two already used features - Latitude and Longitude - it would be expected that the models would be able to retrieve patterns capable of achieving similar results without using these new features. Thus, the main questions of this experiment is the following: Does adding these geolocation features help the models simplify the complexity of the Latitude and Longitude features? Or is it that the new features will help to better define the neighborhood in a way that the Latitude and Longitude features could not?

Fifth experiment - Coefficient approach:
With the fifth experiment the objective was to emulate the approach that some entities use in order to calculate the price of certain terrains/properties. For example, the government of Portugal uses a localization coefficient(simulator available here [64]) in order to calculate a tax regarding real estate.
With this in mind, the coefficient created for this experiment uses the geopy feature ”suburb” as the location indicator and can be described as the **Coefficient of the Price per square meter compared to the suburb average**. Using this new feature as the target label, the models may be able to understand the relation between price, area and location better and achieve better results This coefficient is calculated by analysing each suburb and calculating the average Price per Area of all the houses within it. The coefficient of each house will then be the Price/Meter value divided by the average of the suburb.

$$Cps = PM/APMS, \quad (3.1)$$

with Cps being the Coefficient, PM the Price per Meter and $APMS$ the average Price per Meter in the suburb.

Chapter 4

Results and Discussion

4.1 Introduction

In this chapter all of the metrics used across the experiments done are presented, discussed and explained. Afterwards, all five experiments are presented, with a brief explanation beforehand for each one of what they are about and what is the objective of said experiment. Once the results are shown a brief discussion is done, with the conclusions being taken and explained. Lastly, the experiments are redone on a new different dataset with more features to test certain conclusions drawn from the original datasets. All these experiments and code developed can be found in Github on the following link: `https://github.com/Mielikki26/PropertyAppraisalPlatform` [65].

4.2 Metrics

In order to analyse the performance of each model correctly the following metrics were measured for each experiment:

- MAPE - The main metric used was the mean absolute percentage error (MAPE) as this metric is easy to interpret and can be compared easily between other works. This value is the measurement of the precision the model obtained with its predictions. It is defined by the following formula:

  $$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (4.1)$$

  with n being the number of data points, y_i the true value and $\hat{y}_i$ the predicted value by the model.

- R^2 - This metric is the coefficient of determination, mostly called R-squared(R^2). Its main purpose is to measure how well the model predictions approximate the real data points. In the best case scenario, the R^2 value is 1, which means the predicted values exactly match the true values. A model that always predicts the mean value($\bar{y}_i$) will have R^2 value of 0, and one that predicts worse than the mean value will have negative R^2. It is defined by the following formula:

  $$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}, \quad (4.2)$$

with y_i being the true value, $\hat{y}_i$ the predicted value and $\bar{y}_i$ the mean of the true data values.

- MAE - This is the most simple of the three metrics, the mean absolute error (MAE) is the average difference between the true values and the predicted ones by the model. In the context of this project it represents how many € on average were the model's predictions incorrect when compared to the true prices. It is defined by the following formula:

$$MAE = \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{n} \right|, \quad (4.3)$$

with n being the number of data points, y_i the actual value and $\hat{y}_i$ the prediction value.

4.3 Experiments

Each experiment was repeated by using explicitly 5 different seeds in the pseudo random generator, and the mean and standard deviation obtained in the following experiments result from this sampling strategy. Also, this allows to decrease the variance that a single experiment might have, providing more accurate results.

4.3.1 Baseline results

As described in 3.5.1 the first experiment had the only objective of defining the baseline results to which the following experiences would attempt to improve upon. For this, the datasets used were the baseline ones obtained from the dataset collection after the outlier detection was made. These are considered the baseline results because no significant alterations were made to the datasets and only fine tuning was applied to the models. These results are shown in two tables, one for the tree-based models (table 4.1) and another for the neural network models (table 4.2).

	XGBoost			RF		
	MAPE	R^2	MAE	MAPE	R^2	MAE
Perth	16.17 ± 0.29	0.75	50767 ± 480	15.65 ± 0.23	0.77	48473 ± 171
Argentina	23.06 ± 0.09	0.80	24284 ± 43	25.17 ± 0.30	0.79	25910 ± 140
Colombia	20.81 ± 0.24	0.81	17238 ± 227	20.58 ± 0.21	0.82	16961 ± 168
D.C.	24.04 ± 0.48	0.84	62337 ± 512	23.66 ± 0.22	0.86	59848 ± 447
King County	14.69 ± 0.19	0.80	55230 ± 580	13.97 ± 0.15	0.82	52092 ± 552
Melbourne	17.74 ± 0.39	0.75	102703 ± 1401	16.93 ± 0.05	0.77	97791 ± 1093
Peru	28.01 ± 2.19	0.68	32462 ± 1058	27.49 ± 2.18	0.71	31000 ± 738
Uruguay	28.45 ± 1.77	0.69	32507 ± 722	29.18 ± 2.94	0.71	31334 ± 469
Zameen	21.36 ± 0.18	0.85	13573 ± 76	21.76 ± 0.23	0.87	13393 ± 48
all data	22.22 ± 0.08	0.91	34198 ± 102	23.91 ± 0.10	0.91	35555 ± 88

Table 4.1: Baseline Results obtained with XGBoost and Random Forest.

	TabNet			MLPR		
	MAPE	R^2	MAE	MAPE	R^2	MAE
Perth	17.64 ± 0.39	0.73	54097 ± 1442	24.09 ± 0.99	0.58	69391 ± 1125
Argentina	33.52 ± 0.96	0.66	34496 ± 853	39.17 ± 0.70	0.57	39781 ± 574
Colombia	26.49 ± 0.49	0.75	21453 ± 93	30.51 ± 0.59	0.70	23842 ± 366
D.C.	27.56 ± 0.86	0.83	67873 ± 666	42.48 ± 1.24	0.70	96885 ± 1516
King County	15.44 ± 0.73	0.79	57813 ± 1771	17.37 ± 0.44	0.74	65149 ± 2026
Melbourne	18.13 ± 0.53	0.74	105780 ± 1687	21.69 ± 0.36	0.68	120628 ± 2762
Peru	33.59 ± 3.71	0.56	39274 ± 2786	40.20 ± 2.42	0.57	42576 ± 1100
Uruguay	35.18 ± 3.64	0.62	37440 ± 2755	32.69 ± 1.42	0.62	38885 ± 1664
Zameen	34.33 ± 1.46	0.73	20713 ± 742	37.36 ± 1.10	0.68	23017 ± 175
all data	43.65 ± 0.28	0.72	65639 ± 321	58.57 ± 5.23	0.62	79766 ± 4223

Table 4.2: Baseline Results obtained with TabNet and MLPR.

By analysing both tables it is possible to see that both tree-based models have considerable better performance than the neural network models. It is also very clear that the datasets ”Perth”, ”King County” and ”Melbourne” have the best performance no matter what model is used.

The biggest difference between the tree-based models and the neural network models exists when all data is combined. In this scenario, the tree based models have a performance that is very alike the performance on each individual dataset, meanwhile the neural network models have a significantly worse performance when all data is combined.

4.3.2 Datasets augmentation

As described in chapter 3.5.1, initially in this experiment, all datasets were changed such as that the minimum price accepted was 10000€ for all datasets, this was called the 10k experiment. After this alteration, the size of the datasets was the biggest concern. Thus, four experiments were done to tackle this issue:

- Min50 - For each dataset with size longer than the double of the smallest one(2275), the data points were cut in 50% at random.
- Min25 - For each dataset with size longer than the double of the smallest one(2275), the data points were cut in 75% at random.
- Max - For each dataset with size smaller than the average of all datasets, the size of the dataset was increased to match the average size. This process was done by sampling random original data points and subsequently changing their value (a variance of at most 1% was introduced) to increase the variability of the data.
- Gauss - For each dataset with size smaller than the average of all datasets, the size of the dataset was increased to match the average size. This process was done by sampling random original data points and introducing a variance obtained from a Gaussian distribution.

These four experiments allowed the datasets sizes to be more even while maintaining the original distributions. The results of each experiment are shown in four tables, one for each model(Tables 4.3, 4.4, 4.5 and 4.6), the MAE metric was hidden in favor of easier table interpretation.

	MLPR									
	10k		Min50		Min25		Max		Gauss	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	23.79	0.59	**23.14**	0.60	23.39	0.58	27.78	0.45	28.74	0.40
Argentina	39.67	0.59	40.04	0.57	39.16	0.57	39.51	0.58	**38.77**	0.59
Colombia	30.40	0.70	30.87	0.69	30.79	0.70	**30.23**	0.70	30.73	0.70
D.C.	42.39	0.70	43.19	0.70	43.81	0.68	42.32	0.70	**42.30**	0.70
King County	17.24	0.74	17.14	0.75	**16.77**	0.75	19.71	0.67	26.91	0.49
Melbourne	21.51	0.69	**21.16**	0.68	21.56	0.68	29.39	0.46	34.61	0.29
Peru	38.91	0.56	39.05	0.56	38.36	0.56	31.81	0.68	**29.42**	0.71
Uruguay	33.53	0.62	**32.71**	0.62	33.86	0.60	36.30	0.60	37.55	0.53
Zameen	37.73	0.67	38.66	0.67	36.53	0.68	37.70	0.68	**36.26**	0.68
all data	**55.34**	0.66	58.37	0.63	58.23	0.66	56.51	0.66	77.03	0.56

Table 4.3: Results obtained with MLPR when using different types of dataset augmentation.

	TabNet									
	10k		Min50		Min25		Max		Gauss	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	**17.92**	0.72	18.48	0.70	17.95	0.73	19.50	0.68	26.51	0.46
Argentina	32.97	0.67	34.37	0.66	33.86	0.66	32.84	0.67	**32.74**	0.66
Colombia	26.28	0.76	26.17	0.75	28.21	0.74	**25.94**	0.76	26.19	0.76
D.C.	28.37	0.83	29.80	0.82	28.81	0.83	**28.24**	0.83	28.59	0.83
King County	**14.96**	0.80	15.60	0.79	15.16	0.79	16.89	0.75	28.16	0.47
Melbourne	**17.66**	0.75	18.70	0.72	18.79	0.73	23.44	0.63	33.08	0.33
Peru	36.70	0.58	35.19	0.56	40.04	0.56	**22.35**	0.82	23.30	0.81
Uruguay	33.36	0.63	33.54	0.61	34.16	0.62	**30.39**	0.69	38.93	0.56
Zameen	**34.47**	0.73	34.58	0.72	35.29	0.71	34.60	0.71	36.19	0.73
all data	42.75	0.72	45.67	0.71	44.17	0.72	**40.96**	0.73	70.23	0.59

Table 4.4: Results obtained with TabNet when using different types of dataset augmentation.

	RF									
	10k		Min50		Min25		Max		Gauss	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	**15.51**	0.77	16.56	0.74	15.82	0.75	15.56	0.77	20.28	0.63
Argentina	**25.03**	0.79	26.33	0.77	25.88	0.78	25.09	0.79	25.16	0.79
Colombia	20.77	0.82	22.17	0.80	21.46	0.81	20.78	0.82	**20.73**	0.82
D.C.	**23.47**	0.85	25.16	0.84	24.21	0.85	23.81	0.86	23.96	0.85
King County	13.97	0.82	14.69	0.80	13.99	0.82	**12.05**	0.85	22.95	0.58
Melbourne	17.19	0.77	18.17	0.74	17.35	0.76	**11.65**	0.86	27.27	0.49
Peru	25.98	0.73	26.73	0.73	26.42	0.73	8.53	0.95	**5.14**	0.97
Uruguay	28.02	0.72	30.88	0.68	27.91	0.70	**15.83**	0.88	28.03	0.72
Zameen	21.91	0.87	23.15	0.85	22.42	0.86	**21.80**	0.87	21.93	0.87
all data	23.76	0.91	24.84	0.90	24.15	0.91	**22.49**	0.92	31.33	0.79

Table 4.5: Results obtained with Random Forest when using different types of dataset augmentation.

	XGBoost									
	10k		Min50		Min25		Max		Gauss	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	16.38	0.75	17.16	0.71	16.48	0.74	**14.86**	0.76	20.55	0.60
Argentina	**23.14**	0.80	26.12	0.75	24.37	0.78	23.20	0.79	23.48	0.79
Colombia	21.09	0.80	22.87	0.78	21.65	0.79	**20.95**	0.80	20.91	0.80
D.C.	24.12	0.84	24.77	0.83	24.61	0.84	**24.06**	0.84	24.07	0.84
King County	14.84	0.79	15.74	0.78	15.19	0.78	**10.93**	0.85	23.75	0.55
Melbourne	17.86	0.74	18.86	0.72	18.26	0.73	**8.44**	0.88	27.86	0.43
Peru	26.83	0.69	28.78	0.67	27.49	0.67	**3.63**	0.97	5.06	0.97
Uruguay	27.82	0.69	29.55	0.66	29.15	0.67	**9.45**	0.90	26.51	0.71
Zameen	21.17	0.85	23.49	0.83	22.04	0.84	21.22	0.85	**21.17**	0.86
all data	22.23	0.91	23.57	0.90	22.91	0.91	**19.68**	0.93	28.74	0.80

Table 4.6: Results obtained with XGBoost when using different types of dataset augmentation.

By analyzing each table and comparing them with the baseline results, the following conclusions can be drawn:

- The 10k experiment had the most consistent results, always improving or maintaining on the baseline results;
- The dataset ”Peru” improved drastically(10% for MLPR and TabNet, 20+% for RF and XGBoost), on the experiences that increased its size(Max and Gauss);
- Decreasing the size of the datasets was almost always worse than the 10k experiment;
- In all experiments the tree-based models were considerably better than the neural network models.

Even though the Max and Gauss experiences improved the performance of certain datasets considerably, it also decreased the performance of other datasets when using tree-based models. This, and the increase in training time that results from the increased data size makes it a not worthwhile change when compared to the simple 10k experiment that mostly improved every result slightly. Due to this, the only change that was kept for the next experiments was the 10k experiment change.

4.3.3 Normalization

As explained in chapter 3.5.1, Normalization is the process of transforming the data from its original state into a certain range, usually smaller. In the previous experiments the Normalization used was always the Standard normalization, in this experiment, four other types of normalization will be tested. Similar to the previous experiments, the results of each experiment are shown in four tables, one for each model (Table 4.7, 4.8, 4.9 and 4.10), the MAE metric was hidden in favor of easier table interpretation.

	MLPR									
	No Normalization		Standard		MinMax		Log Transformation		Log & Standard	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	98.58	-5.55	23.79	0.59	24.62	0.52	37.51	-0.04	**21.10**	0.58
Argentina	84.02	-2.39	39.67	0.59	44.43	0.47	58.66	-0.07	**34.29**	0.56
Colombia	89.71	-2.00	30.40	0.70	31.46	0.69	61.86	-0.08	**26.74**	0.70
D.C.	95.92	-2.42	42.39	0.70	42.19	0.70	82.61	-0.11	**35.95**	0.69
King County	99.18	-5.40	17.24	0.74	18.89	0.69	38.77	-0.05	**16.76**	0.72
Melbourne	99.73	-4.57	21.51	0.69	24.96	0.58	40.27	-0.04	**18.78**	0.69
Peru	99.73	-2.84	38.91	0.56	41.74	0.53	56.49	-0.08	**35.32**	0.55
Uruguay	99.36	-3.44	33.53	0.62	39.14	0.51	48.20	-0.07	**30.36**	0.62
Zameen	78.38	-1.44	37.73	0.67	38.93	0.66	74.62	-0.10	**32.66**	0.66
all data	65.92	-0.59	55.34	0.66	54.14	0.67	80.57	0.10	**44.64**	0.59

Table 4.7: Results obtained with MLPR when using different types of normalization.

	TabNet									
	No Normalization		Standard		MinMax		Log Transformation		Log & Standard	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	19.69	0.69	17.92	0.72	18.80	0.70	19.10	0.66	**16.91**	0.72
Argentina	32.49	0.69	32.97	0.67	34.36	0.65	31.69	0.63	**30.88**	0.64
Colombia	25.83	0.77	26.28	0.76	26.81	0.75	25.49	0.73	**24.14**	0.75
D.C.	27.76	0.83	28.37	0.83	29.77	0.82	**25.64**	0.81	26.34	0.82
King County	49.26	-1.40	14.96	0.80	15.66	0.79	16.13	0.75	**14.57**	0.80
Melbourne	85.49	-3.65	17.66	0.75	47.84	0.04	52.39	-0.49	**17.21**	0.74
Peru	64.14	-0.50	36.70	0.58	46.45	0.39	183.83	-3128.12	**31.91**	0.58
Uruguay	63.83	-1.30	33.36	0.63	41.20	0.49	64.92	-0.26	**33.19**	0.59
Zameen	32.17	0.75	34.47	0.73	36.70	0.70	31.12	0.69	**29.56**	0.72
all data	43.10	0.72	42.75	0.72	50.01	0.70	41.68	0.64	**38.08**	0.67

Table 4.8: Results obtained with TabNet when using different types of normalization.

	RF									
	No Normalization		Standard		MinMax		Log Transformation		Log & Standard	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	15.55	0.77	15.51	0.77	15.80	0.77	**14.86**	0.76	15.12	0.76
Argentina	25.11	0.79	25.03	0.79	24.86	0.79	**22.95**	0.78	23.04	0.78
Colombia	21.01	0.82	20.77	0.82	20.82	0.82	**19.34**	0.82	19.41	0.82
D.C.	23.75	0.85	23.47	0.85	23.65	0.86	22.95	0.84	**22.81**	0.84
King County	13.95	0.82	13.97	0.82	13.95	0.81	13.68	0.82	**13.61**	0.82
Melbourne	16.97	0.77	17.19	0.77	16.91	0.77	**16.31**	0.77	16.37	0.77
Peru	26.62	0.75	25.98	0.73	27.56	0.74	25.06	0.72	**25.04**	0.73
Uruguay	27.27	0.73	28.02	0.72	28.23	0.72	**25.78**	0.72	25.93	0.70
Zameen	21.91	0.87	21.91	0.87	22.00	0.87	**20.28**	0.86	20.34	0.86
all data	23.87	0.91	23.76	0.91	23.90	0.91	**21.79**	0.90	21.80	0.90

Table 4.9: Results obtained with Random Forest when using different types of normalization.

	XGBoost									
	No Normalization		Standard		MinMax		Log Transformation		Log & Standard	
	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2	MAPE	R^2
Perth	16.32	0.75	16.38	0.75	16.33	0.75	**15.55**	0.75	15.75	0.74
Argentina	23.22	0.79	23.14	0.80	23.27	0.80	**21.44**	0.80	21.60	0.79
Colombia	20.63	0.81	21.09	0.80	21.12	0.80	**19.35**	0.81	19.75	0.80
D.C.	23.60	0.84	24.12	0.84	24.25	0.84	**22.89**	0.84	23.52	0.83
King County	14.64	0.80	14.84	0.79	14.89	0.80	**14.30**	0.80	14.73	0.79
Melbourne	17.75	0.74	17.86	0.74	17.87	0.74	**16.50**	0.75	17.38	0.74
Peru	27.01	0.69	26.83	0.69	29.48	0.66	**24.39**	0.69	25.93	0.67
Uruguay	26.74	0.70	27.82	0.69	27.43	0.68	**25.70**	0.69	26.66	0.68
Zameen	21.18	0.86	21.17	0.85	21.26	0.85	**19.57**	0.86	20.29	0.86
all data	22.14	0.91	22.23	0.91	22.46	0.91	**20.44**	0.91	20.65	0.90

Table 4.10: Results obtained with XGBoost when using different types of normalization.

By analyzing each table and comparing them with the baseline results, the following conclusions can be drawn:

- As expected, not using normalization is significantly worse when using the neural network models. This is especially notable on the MLPR model. The TabNet model is less affected, this is likely due to the several attributes unique to tabnet such as:
  1. TabNet's mapping functionality inspired by decision trees;
  2. The feature transformer that, after each block, performs normalization.
- The Log transformation also performs badly on neural network models. This makes sense due to the fact that this is not a normalization of the data and more of a transformation. On the tree-based models this is the best performing experiment.
- Considering all the models, the most successful experiment was the ”Log & Standard” normalization, with special improvement on the neural network models with all datasets combined.

With this experiment the most logical conclusion to drawn is that ”Log & Standard” normalization was the best one and that normalization has special influence on the neural network models while still helping the tree-based models. After this experiment, the normalization used was always the ”Log & Standard” normalization.

4.3.4 Feature augmentation - Geopy

As explained in chapter 3.5.1, in this experiment the datasets were augmented using geopy, a python geolocation service. With this change six new features were added, but only five were used:

- Country;
- State;
- Municipality;

- Road - Not used;
- Suburb;
- Postcode.

The results of this experiment are shown in two tables, one for the tree-based models(table 4.11) and another for the neural network models(table 4.12).

	XGBoost			RF		
	MAPE	R^2	MAE	MAPE	R^2	MAE
Perth	15.47 ± 0.19	0.76	49280 ± 290	15.06 ± 0.23	0.76	48616 ± 504
Argentina	22.28 ± 0.14	0.79	25343 ± 212	22.60 ± 0.19	0.78	25680 ± 144
Colombia	18.96 ± 0.21	0.83	17309 ± 184	18.19 ± 0.20	0.84	16881 ± 306
D.C.	23.45 ± 0.24	0.83	63968 ± 513	22.66 ± 0.37	0.85	61730 ± 438
King County	14.58 ± 0.10	0.79	55254 ± 627	13.78 ± 0.13	0.81	52454 ± 484
Melbourne	16.80 ± 0.16	0.75	100447 ± 578	15.99 ± 0.17	0.77	95822 ± 1361
Peru	25.75 ± 1.51	0.69	31111 ± 817	23.51 ± 1.05	0.71	29429 ± 759
Uruguay	25.91 ± 1.11	0.67	32880 ± 741	23.95 ± 1.58	0.72	31011 ± 403
all_data	23.03 ± 0.12	0.86	47189 ± 146	23.38 ± 0.10	0.86	47545 ± 266

Table 4.11: Results obtained with XGBoost and Random Forest when geopy features were added to the datasets.

	TabNet			MLPR		
	MAPE	R^2	MAE	MAPE	R^2	MAE
Perth	15.75 ± 0.64	0.75	49691 ± 827	16.54 ± 0.36	0.74	51948 ± 899
Argentina	27.16 ± 0.80	0.73	30360 ± 445	26.22 ± 0.40	0.72	30603 ± 201
Colombia	21.37 ± 0.21	0.80	19886 ± 222	21.55 ± 0.20	0.79	20416 ± 115
D.C.	25.65 ± 0.40	0.83	66873 ± 1075	30.99 ± 0.66	0.76	82869 ± 2428
King County	14.49 ± 0.31	0.79	55661 ± 806	14.78 ± 0.23	0.79	57007 ± 497
Melbourne	17.36 ± 0.43	0.74	102970 ± 1232	17.14 ± 0.54	0.75	101804 ± 2272
Peru	35.42 ± 5.36	0.59	38665 ± 3487	28.54 ± 1.21	0.66	33420 ± 571
Uruguay	29.50 ± 2.07	0.65	35657 ± 1262	27.18 ± 1.13	0.67	34704 ± 278
all_data	26.63 ± 0.44	0.83	53612 ± 1640	27.98 ± 0.45	0.80	58428 ± 859

Table 4.12: Results obtained with TabNet and MLPR when geopy features were added to the datasets.

With this experiment the following conclusions can be taken:

- Geopy augmentation greatly helps neural network models when all the datasets are used together(16.66% improvement for MLPR and 11.45% improvement for TabNet when compared to previous experiment). It also helps on each individual dataset to a lesser extent(4.5% average improvement for MLPR and 1.05 average improvement for TabNet);
- The XGBoost model got worse results in this experiment when compared to the previous one. Meanwhile the Random forest only slighly improved on every individual dataset, but got a worse performance of 1.58% on all data combined. This is likely because the geopy features made the geolocation features more biased and thus decreased the importance of the other features, resulting in worse performance. This result means that the geopy augmentation is not worthwhile using on Random Forest and XGBoost, because the results do not justify the increased time needed to train the models in the bigger datasets.

- The increase in complexity made all models considerably slower to train, making it debatable if it is even worthwhile using geopy features on models such as the XGBoost and Random Forest models that did not obtain a improvement with the new features.

With these conclusions it is finally possible to get the following answer to the initial question: The Geopy features greatly helped the neural network models in simplifying the complexity of the Latitude and Longitude features. On the contrary, the tree-based models were unable to use these new features effectively and got worse results. This is likely due to the fact that these geopy features are basically redundant data that the tree-based models could already use well. It is also very likely that the XGBoost model got affected in a worse way when compared to the Random Forest model due to the high increase in the data dimensionality occured during the usage of the "One Hot Encoding" technique explained in chapter 3.5.1. Since the Random Forest model used the "Label Encoding" technique instead, the data dimensionality did not increase by a significant amount, thus, the results were not as bad as the ones obtained by XGBoost even though the technique is worse for performance.

4.3.5 Coefficient approach

The final experiment studied the effect of changing the target label from Price to another label capable of representing the value of an house. For this, we propose the usage of the "Coefficient of the Price per square meter compared to the suburb average", as explained in chapter 3.5.1.

The results of this experiment are shown in two tables, one for the tree-based models(table 4.13) and another for the neural network models(table 4.14).

	XGBoost			RF		
	MAPE	R^2	MAE	MAPE	R^2	MAE
Perth	13.73 ± 0.08	0.78	46548 ± 381	14.03 ± 0.11	0.77	47730 ± 354
Argentina	16.92 ± 0.18	0.82	23257 ± 235	17.85 ± 0.09	0.81	24465 ± 143
Colombia	16.64 ± 0.21	0.82	18034 ± 107	17.01 ± 0.22	0.83	18315 ± 214
D.C.	18.51 ± 0.31	0.79	73958 ± 484	17.81 ± 0.29	0.80	70572 ± 1070
King County	15.07 ± 0.31	0.72	63844 ± 1723	13.96 ± 0.20	0.77	58134 ± 942
Melbourne	16.53 ± 0.31	0.76	99821 ± 1561	16.37 ± 0.14	0.76	98251 ± 1290
Peru	19.17 ± 1.47	0.75	27966 ± 2032	19.69 ± 1.11	0.77	27712 ± 606
Uruguay	17.40 ± 0.18	0.75	27706 ± 420	17.14 ± 0.66	0.78	27352 ± 704
all_data	17.72 ± 0.03	0.89	40469 ± 206	18.33 ± 0.06	0.89	41426 ± 185

Table 4.13: Results obtained with XGBoost and Random Forest when using the Coefficient approach.

	TabNet			MLPR		
	MAPE	R^2	MAE	MAPE	R^2	MAE
Perth	14.36 ± 0.35	0.75	49030 ± 915	14.79 ± 0.24	0.75	49271 ± 497
Argentina	19.97 ± 0.18	0.76	27786 ± 135	20.01 ± 0.26	0.76	27783 ± 249
Colombia	19.72 ± 0.20	0.78	21161 ± 313	19.93 ± 0.90	0.78	21474 ± 428
D.C.	19.21 ± 1.11	0.80	74023 ± 2943	22.90 ± 1.02	0.73	87584 ± 3331
King County	15.04 ± 0.45	0.73	63663 ± 693	14.54 ± 0.42	0.74	61163 ± 1917
Melbourne	17.93 ± 1.58	0.72	107222 ± 9087	17.00 ± 0.26	0.74	102953 ± 1265
Peru	22.80 ± 1.76	0.66	32961 ± 2000	21.31 ± 1.12	0.72	31030 ± 656
Uruguay	19.46 ± 0.56	0.70	31266 ± 585	19.17 ± 0.38	0.73	30285 ± 433
all_data	19.55 ± 0.24	0.87	44282 ± 779	20.49 ± 0.47	0.85	47132 ± 859

Table 4.14: Results obtained with TabNet and MLPR when using the Coefficient approach.

As seen in the tables, the results of this experiment were incredibly successful with the average MAPE decreasing in every single model by a considerable amount when compared to the Geopy experiment:

- XGB: Average -3.65% on individual datasets, -5.31% on all datasets combined;

- RF: Average -2.74% on individual datasets, -5.05% on all datasets combined;

- TabNet: Average -4.78% on individual datasets, -7.08% on all datasets combined;

- MLPR: Average -4.16% on individual datasets, -7.49% on all datasets combined.

4.3.6 Main Conclusions

These experiments were very successful in improving the performance of each model on the datasets. The variety of datasets used allow us to infer that these experiments would be useful to use on any dataset that might be lacking in features, size or complexity. The dataset augmentation experiment proves that more data does not automatically mean better results, meanwhile, the normalization experiment proves that using techniques not commonly used can prove to be the best approach. Finally, the Coefficient experiment also proves that just because we are adding something that may seem redundant, it may help the model significantly in simplifying the patterns existent in the features. Obviously, this experiment was only possible thanks to the Geopy experiment done before which proved how valuable the feature engineering technique can be in improving a dataset complexity by only using already existent features. Overall these experiments prove that even lackluster datasets can be improved a decent amount by not even changing the model.

A representation of the MAPE evolution across the experiments can be visualized in the figures 4.1 and 4.2.

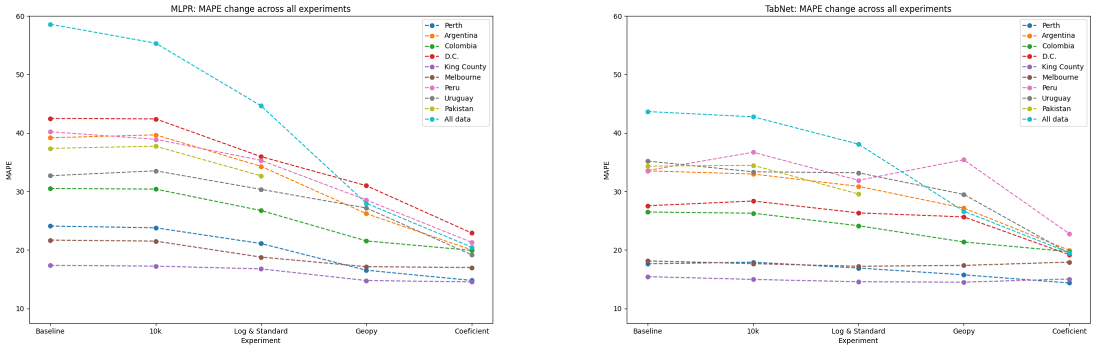


Figure 4.1: MAPE change across all experiments and all datasets when using MLPR and TabNet.

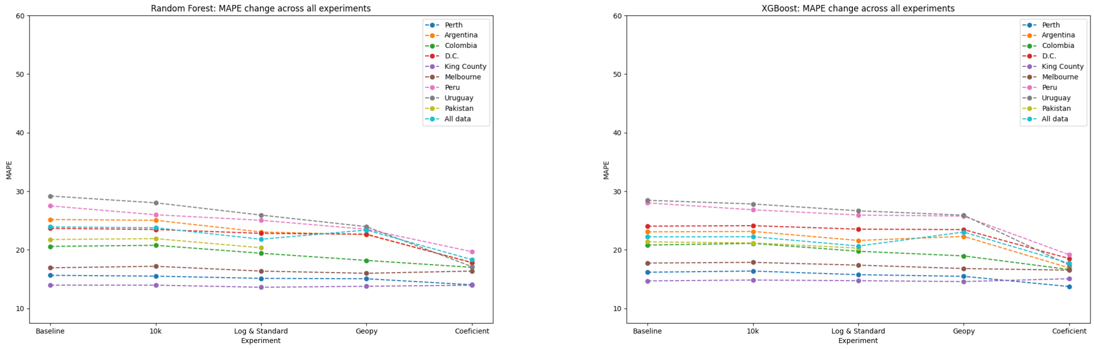


Figure 4.2: MAPE change across all experiments and all datasets when using Random Forest and XGBoost.

These figures perfectly describe the improvements that each experiment provided. It is especially evident on the Neural Network models how much these experiments helped the performance of the models, notably when all the data was combined into a singular dataset.

4.4 ERA dataset

In order to test the impact of these experiments on a different dataset with different features, the ERA dataset was made available thanks to DeepNeuronic. This dataset is focused on the city of Lisbon and Setúbal, with the most common county being Almada. This dataset is comprised of 9363 data points and 44 features including the set of features used on the other datasets. These new features provide information very different to the set of features originaly used, such as the Typology of the house, the Energy certification, whether or not the house has been renewed and more.

The main objective of using this dataset was to test whether or not even more complex datasets can utilize the approaches used in the original experiments to improve performance. As such, all experiments were redone on only this dataset, with the exception of the dataset augmentation experiment, as this dataset is already complex enough. For the coefficient experiment, the already existent feature of ”Parish” was used instead of the ”suburb” feature used in the other datasets for the calculation of the coefficient. The main reasons for this change is due to the ”Parish” feature being more accurate and the fact that the Geopy API returned a lot of missing values. The six most common parishes can be seen in the figure 4.3.

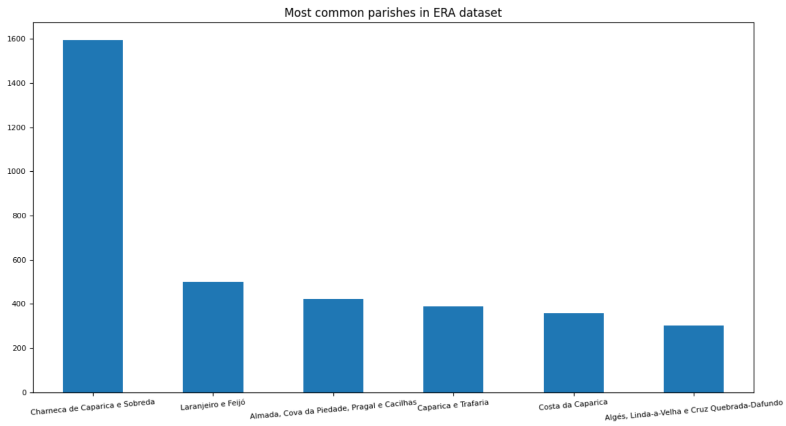


Figure 4.3: Most common parishes existent in ERA dataset.

The results obtained in the experiments using the new dataset are shown in table 4.15.

	MLPR		TabNet		RF		XGBoost	
	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE
Baseline	34.72	42087	28.84	33189	18.42	22407	15.12	15532
Log & Standard	**24.91**	40208	23.31	35477	**15.24**	23659	**9.80**	14319
Geopy	26.50	41175	**21.61**	34936	15.37	22723	10.01	14830
Coefficient	28.27	45287	23.02	38800	17.31	28752	13.03	17012

Table 4.15: Results obtained with the Era dataset across all models.

By analyzing the table, the most obvious result is the fact that all experiments do improve upon the baseline results that the models achieve. This reinforces the idea that all these experiments can be utilized on more complex datasets for better performance. The other big information that can be taken from the results, is that the geopy and coefficient experiments did not improve the performance of the models when compared to the Log & Standard experiment (with the only exception being using Geopy with TabNet). This is certainly a surprising result but not necessarily unexpected. Feature augmentation always has best performance when used on more basic datasets with few features, since this dataset has 44 original features it is understandable why more features would not necessarily improve performance.

Chapter 5

Conclusion and Future Work

5.1 Main Conclusions

The main objective of this work was to build machine learning models capable of exploring multiple real estate markets and evaluating different houses with different properties. Due to the nature of the real estate market, as the economy of each country changes, the prices will be also changing, and logically, a machine learning model can never be perfect, being the best when paired with human expertise. By initially studying the vast state of the art, the best techniques and correct procedures were found. Then, a big dataset collection was created, composed with datasets from multiple countries and multiple real estate markets. Thanks to this, it was possible to study them and analyse them in order to find the best common features to use so that it was possible to represent any real estate market to a good level. In order to maximize the performance across all the datasets, multiple experiments were done, with each one tackling different aspects of data analysis and model building. All of these experiments improved on the previous results proving that even in simple datasets with only 7 features, lots of improvements can be done and the performance can be improved a decent amount without necessarily changing anything about the models. The biggest achievement from this work is the ”Coefficient approach” used that allowed us to improve the performance of all models by changing the target label to a value that represents the relation between the features that are most commonly seen as the defining traits of a property (Price, Area and Location). With this work, we also showed that these ideas can be adapted into more complex datasets to improve performance, as shown by our usage of the ERA dataset comprised of 44 features, significantly more complex than our 7 feature set used in the other datasets. Regarding the models used, the best performing models were the tree-based models, more specifically the XGBoost model. This is not surprising due to all of the data used being tabular data, where tree-based algorithms perform the best.

Thanks to this work it was possible to determine the best strategies and methodologies to be used when evaluating the real estate market across several countries. Using this, DeepNeuronic can create an interface/platform that allows users to use the methods developed and evaluate any house they desire to input in real time.

5.2 Future Work

Even though all the objectives regarding this work were completed successfully, some ideas are left to be explored, such as:

- More data augmentation using other basic features;

- More fine tuning;
- Experiments with different model architectures such as ensembles;
- Experiments with bigger datasets.

These are the main ideas that were not fully explored, besides these there are many more experiments that can be done, due to the massive size of techniques and ideas that exist in machine learning.

Finally there was also an idea to create a web platform capable of using the developed models to evaluate any input given by a user. This idea is very interesting but was not done due to being outside of the scope of this project.

Bibliography

[1] Q. Truong, M. Nguyen, H. Dang, and B. Mei, "Housing price prediction via improved machine learning techniques," *Procedia Computer Science*, vol. 174, pp. 433–442, 2020, 2019 International Conference on Identification, Information and Knowledge in the Internet of Things. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920316318 xi, 6, 7

[2] S. Law, B. Paige, and C. Russell, "Take a look around," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 5, p. 1–19, Nov 2019. [Online]. Available: http://dx.doi.org/10.1145/3342240 xi, 7, 8

[3] Q. You, R. Pang, and J. Luo, "Image based appraisal of real estate properties," *CoRR*, vol. abs/1611.09180, 2016. [Online]. Available: http://arxiv.org/abs/1611.09180 xi, 8

[4] H. Peng, J. Li, Z. Wang, R. Yang, M. Liu, M. Zhang, P. Yu, and L. He, "Lifelong property price prediction: A case study for the toronto real estate market," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021. xi, 9

[5] S. O. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," 2019. [Online]. Available: https://arxiv.org/abs/1908.07442 xi, 18, 19

[6] M. Dellstad, "Comparing three machine learning algorithms in the task of appraising commercial real estate," 2018. 5

[7] S. B. Jha, V. Pandey, R. K. Jha, and R. F. Babiceanu, "Machine learning approaches to real estate market prediction problem: A case study," *CoRR*, vol. abs/2008.09922, 2020. [Online]. Available: https://arxiv.org/abs/2008.09922 5

[8] W. Ho, B.-S. Tang, and S. Wong, "Predicting property prices with machine learning algorithms," *Journal of Property Research*, vol. 38, pp. 1–23, 10 2020. 5

[9] A. Baldominos, I. Blanco, A. Moreno, R. Iturrarte, □. Bernárdez, and C. Afonso, "Identifying real estate opportunities using machine learning," *Applied Sciences*, vol. 8, p. 2321, 11 2018. 6

[10] B. Trawinski, Z. Telec, J. Krasnoborski, M. Piwowarczyk, M. Talaga, T. Lasota, and E. Sawilow, "Comparison of expert algorithms with machine learning models for real estate appraisal," 07 2017, pp. 51–54. 6

[11] J. Ahn, H. Byun, K. J. Oh, and T. Kim, "Using ridge regression with genetic algorithm to enhance real estate appraisal forecasting," *Expert Systems with Applications*, vol. 39, pp. 8369–8379, 07 2012. 6

[12] B. Park and J. K. Bae, "Using machine learning algorithms for housing price prediction: The case of fairfax county, virginia housing data," *Expert Systems with Applications*, vol. 42, no. 6, pp. 2928–2934, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414007325 6

[13] R. F.-Y. Lin, C. Ou, K.-K. Tseng, D. Bowen, K. Yung, and W. Ip, “The Spatial neural network model with disruptive technology for property appraisal in real estate industry,” *Technological Forecasting and Social Change*, vol. 173, no. C, 2021. [Online]. Available: https://ideas.repec.org/a/eee/tefoso/v173y2021ics0040162521004996.html 8

[14] P.-Y. Wang, C.-T. Chen, J.-W. Su, T.-Y. Wang, and S.-H. Huang, “Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism,” *IEEE Access*, vol. 9, pp. 55 244–55 259, 2021. 8

[15] A. G. B. Sarip and M. B. Hafez, “Fuzzy logic application for house price prediction,” 2015. 9

[16] H. Kuşan, O. Aytekin, and İlker Özdemir, “The use of fuzzy logic in predicting house selling price,” *Expert Systems with Applications*, vol. 37, no. 3, pp. 1808–1813, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417409006885 9

[17] A. Malinowski, M. Piwowarczyk, Z. Telec, B. Trawinski, O. Kempa, and T. Lasota, *An Approach to Property Valuation Based on Market Segmentation with Crisp and Fuzzy Clustering: 10th International Conference, ICCCI 2018, Bristol, UK, September 5-7, 2018, Proceedings, Part I*, 01 2018, pp. 534–548. 9

[18] “Kaggle,” https://www.kaggle.com, last access: 18-12-2021. 10

[19] “Github,” https://github.com, last access: 18-12-2021. 10

[20] W. Lim, L. Wang, Y. Wang, and Q. Chang, “Housing price prediction using neural networks,” 08 2016, pp. 518–522.

[21] “gyanadata/real-estate-housing-price-prediction-project-,” https://github.com/gyanadata/Real-Estate-Housing-Price-Prediction-Project-, last access: 22-11-2021. 11

[22] “satishgunjal/house-price-prediction-project,” https://github.com/satishgunjal/House-Price-Prediction-Project/blob/master/Bengaluru_House_Data.csv, last access: 22-11-2021. 11

[23] “eliekawerk/real_estate_tycoon_project,” https://github.com/eliekawerk/Real_Estate_Tycoon_project/blob/master/project_files/real_estate_data.csv, last access: 22-11-2021. 11

[24] “kc_house_data,” https://www.kaggle.com/shivachandel/kc-house-data, last access: 25-11-2021. 11

[25] “digipodium/real-estate-analysis-and-prediction,” https://github.com/digipodium/Real-Estate-Analysis-and-Prediction/tree/master/data, last access: 22-11-2021. 11

[26] “Property assessment,” https://data.boston.gov/dataset/property-assessment, last access: 22-11-2021. 11

[27] “Real estate price prediction,” https://www.kaggle.com/quantbruce/real-estate-price-prediction. 11

[28] “Russia real estate 2018-2021,” https://www.kaggle.com/mrdaniilak/russia-real-estate-20182021, last access: 22-11-2021. 11

[29] “Riga real estate dataset,” https://www.kaggle.com/trolukovich/riga-real-estate-dataset, last access: 22-11-2021. 11

[30] “Madrid real estate market,” https://www.kaggle.com/mirbektoktogaraev/madrid-real-estate-market, last access: 22-11-2021. 11

[31] “Japan real estate prices,” https://www.kaggle.com/nishiodens/japan-real-estate-transaction-prices, last access: 22-11-2021. 11

[32] “Nyc property sales,” https://www.kaggle.com/new-york-city/nyc-property-sales, last access: 22-11-2021. 11

[33] “Melbourne housing market,” https://www.kaggle.com/anthonypino/melbourne-housing-market?select=Melbourne_housing_FULL.csv, last access: 25-11-2021. 11

[34] “Sao paulo real estate - sale / rent - april 2019,” https://www.kaggle.com/argonalyst/sao-paulo-real-estate-sale-rent-april-2019, last access: 22-11-2021. 11

[35] “Victoria real estate,” https://www.kaggle.com/ruizjme/realestate-vic-sold, last access: 22-11-2021. 11

[36] “Philadelphia real estate,” https://www.kaggle.com/harry007/philly-real-estate-data-set-sample, last access: 22-11-2021. 11

[37] “House price prediction,” https://www.kaggle.com/shree1992/housedata?select=data.dat, last access: 22-11-2021. 11

[38] “Villas real estate price,” https://www.kaggle.com/maha48/villas-price-dataset?select=train_data.csv, last access: 22-11-2021. 11

[39] “Saudi arabia real estate (aqar),” https://www.kaggle.com/lama122/saudi-arabia-real-estate-aqar, last access: 22-11-2021. 11

[40] “Apartment data,” https://www.kaggle.com/gunhee/koreahousedata, last access: 22-11-2021. 11

[41] “Property prices in tunisia,” https://www.kaggle.com/ghassen1302/property-prices-in-tunisia, last access: 22-11-2021. 11

[42] “German house prices,” https://www.kaggle.com/scriptsultan/german-house-prices, last access: 22-11-2021. 11

[43] “Commercial real estate for sale,” https://www.kaggle.com/aramacus/commercial-real-estate-for-sale, last access: 22-11-2021. 11

[44] "Property listings for 5 south american countries," https://www.kaggle.com/rmjacobsen/property-listings-for-5-south-american-countries, last access: 22-11-2021. 11

[45] "Seattle airbnb listings," https://www.kaggle.com/shanelev/seattle-airbnb-listings, last access: 22-11-2021. 11

[46] "Seoul real estate datasets," https://www.kaggle.com/jcy1996/seoul-real-estate-datasets, last access: 22-11-2021. 11

[47] "Arizona houses 2021," https://www.kaggle.com/antoniong203/arizona-houses-2021, last access: 22-11-2021. 11

[48] "Real estate dataset," https://www.kaggle.com/smitisinghal/real-estate-dataset, last access: 22-11-2021. 11

[49] "Raifhack-ds-2021-fall," https://www.kaggle.com/lildatascientist/raifhackds2021fall, last access: 22-11-2021. 11

[50] "California housing prices," https://www.kaggle.com/camnugent/california-housing-prices, last access: 22-11-2021. 11

[51] "Housing prices in metropolitan areas of india," https://www.kaggle.com/ruchi798/housing-prices-in-metropolitan-areas-of-india, last access: 22-11-2021. 11

[52] "House pricing in belo horizonte," https://www.kaggle.com/guilherme26/house-pricing-in-belo-horizonte, last access: 22-11-2021. 11

[53] "D.c. residential properties," https://www.kaggle.com/christophercorrea/dc-residential-properties, last access: 22-11-2021. 11

[54] "Amsterdam house price prediction," https://www.kaggle.com/thomasnibb/amsterdam-house-price-prediction, last access: 22-11-2021. 11

[55] "Property listings in kuala lumpur," https://www.kaggle.com/dragonduck/property-listings-in-kuala-lumpur, last access: 22-11-2021. 11

[56] "Nashville housing data," https://www.kaggle.com/tmthyjames/nashville-housing-data, last access: 22-11-2021. 11

[57] "Zameen.com property data pakistan," https://www.kaggle.com/huzzefakhan/zameencom-property-data-pakistan, last access: 22-11-2021. 11

[58] "Housing price in beijing," https://www.kaggle.com/ruiqurm/lianjia, last access: 22-11-2021. 11

[59] "Usa housing listings," https://www.kaggle.com/austinreese/usa-housing-listings, last access: 22-11-2021. 11

[60] "Portland housing prices/sales jul 2020 - jul 2021," https://www.kaggle.com/threnjen/portland-housing-prices-sales-jul-2020-jul-2021, last access: 22-11-2021. 11

[61] "Paris housing price prediction," https://www.kaggle.com/mssmartypants/paris-housing-price-prediction, last access: 22-11-2021. 11

[62] "Perth house prices," https://www.kaggle.com/syuzai/perth-house-prices, last access: 25-11-2021. 11

[63] "Geopy," https://geopy.readthedocs.io/en/stable/, last access: 18-06-2022. 23

[64] "Simuladorat," https://zonamentopf.portaldasfinancas.gov.pt/simulador/default.jsp, last access: 02-08-2022. 24

[65] "Property appraisal platform," https://github.com/Mielikki26/PropertyAppraisalPlatform, last access: 09-10-2022. 25