HOUSE PRICE PREDICTION: A COMPARATIVE ANALYSIS OF MACHINE LEARNING APPROACHES TO STUDY MELBOURNE'S MARKET

Simona Nobile

Dissertation written under the supervision of prof. Pedro Afonso Fernandes.

Dissertation submitted in partial fulfillment of requirements for the MSc in Business Analytics, at the Universidade Católica Portuguesa, May 2024.

House Price Prediction:
A comparative analysis of machine learning approaches to study Melbourne's Market

Simona Nobile

Abstract

This thesis work investigates the application of machine learning (ML) techniques for predicting house prices, a crucial task with widespread implications. In this scope, this work presents a literature review on state-of-the-art approaches and a practical experiment using a dataset of house sales in Melbourne, Australia. The analysis focuses on identifying key features for price prediction and assessing the performance of various ML algorithms. In fact, examining feature importance over time, it is possible to understand the dynamic nature of house price prediction.

Keywords: Machine Learning; House pricing prediction; Panel data

House Price Prediction:
A comparative analysis of machine learning approaches to study Melbourne's Market

Simona Nobile

Resumo

A presente tese debruça-se sobre a aplicação de técnicas de aprendizagem automática à previsão do preço da habitação, uma tarefa crucial e com importantes implicações. Neste âmbito, o trabalho inclui uma revisão de literatura e uma aplicação ao caso de Melbourne, Austrália. A análise foca-se na identificação dos fatores determinantes do preço da habitação e na análise da performance preditiva de diversos algoritmos de aprendizagem automática. De facto, da análise da importância desses fatores ao longo do tempo, é possível compreender a natureza dinâmica do exercício de previsão do preço da habitação

Palavras chave: Aprendizagem automática; Previsão do preço da habitação; Dados de painel.

Contents

Chapter 1

Introduction

Machine Learning is a specific branch of Artificial Intelligence that focuses on learning over data. During the past few years, machine learning has been applied to approach several tasks in many domains, such as healthcare, finance, natural language processing, and image and video processing. In this project, we focus on the application of machine learning to the prediction of house prices. Indeed, predicting house prices is highly relevant in the economy as it not only plays a vital role in real-estate businesses but also impacts other things, such as the cost of life and urban development.

House price is usually highly influenced by property characteristics (e.g., size, location, number of bedrooms/bathrooms), neighborhood features (e.g., crime rate, school quality), and economic indicators (e.g., interest rates, inflation). These variables make it very complex to predict house prices accurately. Moreover, as such variables tend to vary over time, house prices are also often subject to time variations.

For these reasons, the use of machine learning is highly indicated in house price prediction. More specifically, in this thesis project, we will present the literature of current state-of-the-art methods applied to such domain, and we will apply such techniques in practice on an online dataset[1].

The dataset presents about 50 thousand samples of house sales together with several features describing the property (suburb, number of rooms, number of bathrooms, number of car spots, size,

[1] `https://www.kaggle.com/datasets/anthonypino/melbourne-housing-market`

type, sale method, postcode, region, number of properties in the area, and distance from the commercial area) and the transaction date. More specifically, all the sales were located in different areas of the city of Melbourne from March 2016 to February 2018. The practical experiment's main objective is to measure machine learning methods' ability to predict the price and individuate the most relevant features in this task. In particular, as the sales represent independent samples over a long time period, the dataset constitutes what is usually defined as a pooled cross-section, which allows us to analyze how the role of the features presented varied over time. Indeed, analyzing the most critical features is of high relevance not only because it increases the transparency and readability of the models, but also because it could allow further understanding of a complex problem such as the one of house prices prediction.

The remainder of this work is organized as follows: Chapter 2 presents the most relevant work related to ML and house price prediction; in Chapter 3, we present the case study for the practical part of the work; Chapter 4 wraps up the theoretical and practicals results of this study and analyzes possible openings for future research.

Chapter 2

Literature Review

2.1 The influence of space features in Housing Markets

Because of the significant influence of forecasting house property values, a broad array of literature exists detailing various iterations of this issue and approaches to address it. In one of the first works on this topic, Olmo [1995] conducted a study to perform a spatial estimation of housing prices in the city of Granada, Spain. More specifically, they formulate a regression problem to estimate the price from apartment age, number of bathrooms, dimensions, number of rooms, floors, and coordinates. Although for their work they use only 260 sales, they were able to conclude that the proposed approach proved to be an interesting instrument for the analysis of cross-sectional data in presence of spatial auto-correlation. Successively, other works showed that using spatial information plays an important role in obtaining better predictive performances.

In a consecutive work, Bourassa et al. [2007] demonstrated that further improvements could be obtained by adding information about housing sub-markets. In this work, the authors use case study data from Auckland, New Zealand. These results are motivated by the fact that house prices are often related to the prices of adjacent properties. For this reason, they propose to adjust the geostatistical methods from previous works with average residuals in neighborhoods.

In another work, Osland and Thorsen [2009] show how house prices are highly related to the presence of job opportunities in the urban areas. To achieve such an aim, they perform a study on data from Rogaland, Norway. By analyzing the results, they reported that local employment growth has a positive impact on prices. At the same time, they see that employment expansion paired with high urbanization in the area could negatively impact house prices.

Interestingly, in another work performed on data from the city of Lisbon, Portugal, Martínez and Viegas [2009] try to understand the relationship between transportation infrastructure and house prices. To this end, they use features describing the public transportation together with the ones from the properties. The results confirmed the expectancy by showing that the presence of metro lines tends to increase house prices. At the same time, the authors also showed that the proximity to specific metro lines could sometimes decrease the price, probably due to issues like noise or neighborhood security.

Similar to what seen so far, other works investigate the dependence between house sales and environmental factors. Other applications, for instance include proximity to cell phone towers [Filippova and Rehm, 2011], walking accessibility [Yang et al., 2018], urban centrality [Marmolejo-Duarte, 2017], and access to bike facilities [Welch et al., 2016].

Overall, all these methods confirm the high interconnection between the public services and the house prices. Indeed, such features offer proxy metrics to measure the neighborhood quality. Additional studies, for instance, confirm such direction, as in [De Nadai and Lepri, 2018], where the authors perform an analysis on 8 Italian cities and observe that the neighborhood's vitality and walkability seem to account for more than 20% of the housing value.

2.2 Including time in the equation

Using a completely different approach, Gupta and Miller [2012] studied the role of time in the prediction of house prices. More in detail, they conducted an examination of time series data across eight metropolitan areas of Southern California: Bakersfield, Los Angeles, Oxnard, Riverside, San

Diego, San Luis Obispo, Santa Ana and Santa Barbara. By analyzing the temporal relationships between these cities, they provided insights that simpler regression models, which do not consider time, fail to capture. To do this, the authors employed vector auto-regressive models, a sophisticated statistical technique designed for understanding the interconnectedness of time-dependent data. The results demonstrate that the house prices of these different cities are closely interlaced. For instance, they showed how the prices in Los Angeles have a direct influence on the prices in Bakersfield, Riverside, San Diego, and San Luis Obispo.

Building upon the ideas of Gupta and Miller [2012] and Bourassa et al. [2007] this analysis was expanded by Liu [2013]. Indeed, in this work, the authors incorporate spatial, temporal, and submarkets information into their models. The models were applied to sales in the Randstad region, in Netherlands for the period from 1997 to 2007. Through this comprehensive approach, the author concluded by demonstrating the presence of both spatial and temporal dependence in the Randstad market, with a higher relevance of the latter over the former. Again, this study reinforces the utility of advanced modeling techniques, which are helpful for studying the evolution of house markets with many practical implications.

In a related but distinct line of research, Liu et al. [2016] explored the role of time in modeling house prices using a different methodology. In their work, they proposed to model the house prices with a temporally weighted regression which takes information about travel time distance as input. Their analysis used sales data from Beijing, spanning from 1980 to 2016. By incorporating travel time distances, they were able to capture the temporal variations in house prices more accurately. This innovative approach highlights the importance of considering travel time and its impact on property values, offering a different point of view on the analysis of temporal dynamics of real estate markets.

Together, these studies underscore the critical role of time in understanding house prices and demonstrate the effectiveness of various advanced modeling techniques in capturing the complex interdependencies in housing markets. They collectively contribute to a deeper understanding of how spatial and temporal factors interact to influence real estate prices, providing valuable insights for both researchers and practitioners in the field.

2.3 Exploiting more advanced methods

While the research of relationships between house values and spatial/temporal features kept improving, statistics also moved towards the definition of more complex models. Therefore, several works started using different models from simple ordinary least squares in order to predict house sales prices, namely, machine learning methods. This trend is also justified by the improvements in the calculation capabilities of computers, which allowed modeling complex systems in less time.

Zhang et al. [2021], for instance, perform a study on the city of Toronto with spatial and temporal data from 1931 to 2011. For their study they employ a method called random forest [Rigatti, 2017]. Random forest is an ensemble method, i.e., a method which is composed of several submodels in the form of decision trees [Quinlan, 1986]. The latter, analyze the input features by applying several nested rules that form a tree. Random forests exploit this structure to build a forest of decision trees trained on different portions of the training data. Being composed of several submodels, random forest allow higher expressivity with respect to simple linear models and are therefore better suited in non-linear problems.

A similar approach is the one of Mullainathan and Spiess [2017], who perform a comparison between several machine learning methods for the prediction of house prices. For this study, they use a dataset extracted by the 2011 American Housing Survey. More specifically, they compare five different models:

- ordinary least squares, a linear model which is trained by minimizing the mean squared error between the prediction and the true value;
- LASSO, a regularized version of the ordinary least squares where an additional penalty is added to the optimization to minimize the number of features that are used. This allows the creation of general models that hopefully perform better on unseen data, thus minimizing the probability of overfitting;
- regression trees and random forest (tree-based models). In this case they also try to adjust the maximum depth of the trees to identify the best configuration;

- ensemble, a weighted combination of the predictors above. The idea is that each predictor could grasp different patterns, and utilizing them all could allow for better expressivity.

From their analysis, Mullainathan and Spiess [2017] observe that the random forest and the ensemble model are the ones achieving the best results, thus suggesting that only using linear algorithms is not enough to model a complex problem such as house price prediction.

With the continuous development of novel machine learning techniques and the exploration of complex architectures such as neural networks, the ability of models to predict prices is constantly improving [Truong et al., 2020, Varma et al., 2018].

Having complex models allows reaching better prediction performances, but at the same time, it often opens up the problem of interpreting the models' behavior. In the next section, we explore how this problem is approached in literature and how it impacts house price prediction.

2.4 The accuracy/interpretability trade-off

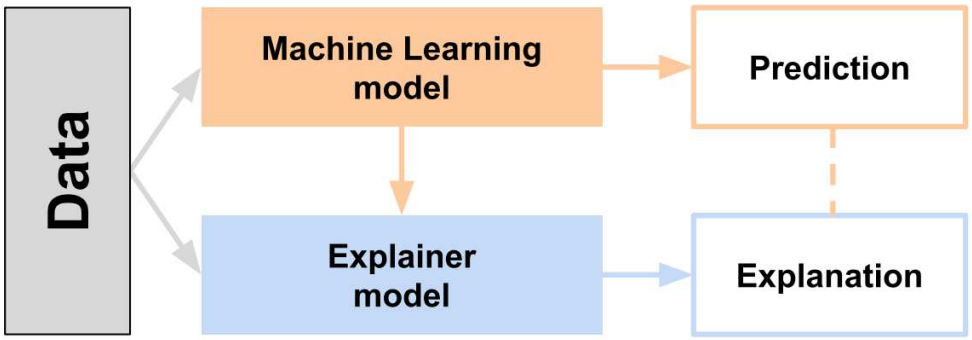


Figure 2.1: The process of explanation. With complex models, it becomes difficult to understand their behavior, for this reason, we need new techniques that allow us to obtain explanations. [Rico-Juan and de La Paz, 2021]

The introduction of intricate methods, such as random forests in machine learning, has brought to light the critical issue of understanding the reasons behind a prediction. This is a problem that is investigated in many disciplines, and it is of high impact, mostly in sensitive applications like medicine and other areas where human-related decisions are crucial. For instance, when an algorithm is asked to determine a medical diagnosis or a particular treatment, it is vital to understand the underlying rationale that led to such a decision for both trust and accountability. Additionally, government regulations such as the General Data Protection Regulation and the Artificial Intelligence Act enforce "the right of explanations". This principle essentially states that any individual impacted by an algorithm's decision has the right to an explanation for the reasons behind that decision.

In response to this, a novel branch of artificial intelligence, called explainable artificial intelligence is emerged. Explainable artificial intelligence is dedicated to the development of techniques that allow us to understand the behavior of machine learning models and make them more transparent. Such techniques are designed to investigate the decision-making processes of complex models, making it possible to interpret and trust their outputs.

Regarding the context of house prices prediction, Rico-Juan and de La Paz [2021] introduces the field of explainable artificial intelligence in house price prediction by applying several techniques that allow identifying which features the model finds most interesting and how they are used to

produce the prediction. In this work, they focus on the housing market of the city of Alicante, Spain, for the period from 1996 to 2012. By employing various explainable artificial intelligence methods, they were able to identify which features the model considered most significant and how these features were used to generate predictions. An interesting result of their study is the identification of complex dependencies between factors such as spatiality, time, and house prices, which often result in discontinuities that linear models struggle to grasp. For example, they discovered that in the northern parts of the city, the size of a house has a positive impact on its price, while the opposite trend is observed in the southern regions. This nonlinear relationship highlights the limitations of linear models, which would fail to accurately estimate prices due to their inability to account for such intricate patterns.

Through their research, Rico-Juan and de La Paz [2021] demonstrate the value of explainable artificial intelligence in providing deeper insights into the predictive models used in the real estate market. This could not only help identify new ways of improving the accuracy of predictions but also enhance our understanding of the factors influencing house prices, leading to more informed decision-making. As the field of explainable artificial intelligence continues to evolve, its techniques will become increasingly crucial in ensuring the transparency and reliability of machine learning models across various applications.

Chapter 3

Case Study: Melbourne's Housing Market

In this chapter, we delve into a practical application of the methods analyzed so far. More specifically we analyze the performance of several machine learning methods in predicting house prices in the city of Melbourne, Australia. This study also aims at identifying the factors that most influence the prices and verify their alignment with what found in literature.

We start by presenting the dataset, showing the features that are available and presenting some statistical analysis of the data. Successively we introduce the algorithms used to predict the prices, with references to their inner working and the type of interpretation we can get from them. Finally we analyze the performances of the trained models and discuss the obtained results.

3.1 Dataset

The dataset, that was obtained from Kaggle[1] contains a comprehensive list of house trades conducted done from January 2016 until April 2018 in the city of Melbourne, Australia. This dataset was meticulously collected and cleaned using data published on the website `domain.com.au`.

The dataset comprises various features that provide in-depth information about real estate proper-

[1] `https://www.kaggle.com/datasets/anthonypino/melbourne-housing-market`

Feature	Description	Data Type	Possible Values
Suburb	Suburb	Text	String
Address	Address	Text	String
Rooms	Number of rooms	Numeric	Integer
Price	Price in Australian dollars	Numeric	Float
Method	Sale method	Text	S, SP, PI, PN, SN, NB, VB, W, SA, SS
Type	Property type	Text	br, h, u, t, dev site, o res
SellerG	Real Estate Agent	Text	String
Date	Date sold	Date	YYYY-MM-DD
Distance	Distance from CBD in Kilometres	Numeric	Float
Regionname	General Region	Text	String
Propertycount	Number of properties in the suburb	Numeric	Integer
Bedroom2	Scraped number of bedrooms	Numeric	Integer
Bathroom	Number of bathrooms	Numeric	Integer
Car	Number of car spots	Numeric	Integer
Landsize	Land size in square meters	Numeric	Float
BuildingArea	Building size in square meters	Numeric	Float
YearBuilt	Year the house was built	Numeric	Integer
CouncilArea	Governing council for the area	Text	String
Lattitude	Latitude	Numeric	Float
Longitude	Longitude	Numeric	Float

Table 3.1: Dataset features. List of the features of the dataset with a small description and the corresponding data type.

ties. These features are categorized into two main types: categorical and numerical.

Categorical features in the dataset offer valuable qualitative information. For instance, the "Suburb" feature indicates the names of suburbs where the properties are located, while the "Address" feature provides the specific street addresses of these properties. The "Method" feature describes the mode of sale, which can range from properties sold to those withdrawn prior to auction. The "Type" feature indicates the property type, such as house, unit, or townhouse. Another important categorical feature is "SellerG", which identifies the real estate agent handling the sale. The "Regionname" feature specifies the general region where the suburb is situated, and the "CouncilArea" feature denotes the governing council for the area. The dataset also includes a date feature, "Date", which signifies the date when the property was sold. For this project, we transform certain categorical features such as "Regionname", "CouncilArea" and "Year" using the One-Hot notation. This method involves representing these categories using a binary vector where all elements are zero except for one, which corresponds to the category being represented.

Numerical features in the dataset provide quantitative information about the properties. These features include "Rooms", which denotes the number of rooms in a property, and "Price", which represents the property's price in Australian dollars. The "Distance" feature indicates the distance of the suburb from the Central Business District (CBD) in kilometers, while Propertycount" denotes the total number of properties within the suburb. The "Bedroom2" feature reflects the number of bedrooms, sourced from a secondary data provider, and "Bathroom" signifies the number of bathrooms in a property. The "Car" feature represents the number of available car spots. Additionally, the dataset includes Landsize" and "BuildingArea", which specify the land size and building area, respectively, in square meters. The "YearBuilt" feature denotes the year when the property was constructed. Geographic coordinates are provided by the "Latitude" and "Longitude" features, specifying the exact location of the properties.

Table 3.1 presents a detailed list of all the features, the data type, and the possible values each feature can take.

As an initial analysis, we examine the temporal distribution of the house trades recorded in the dataset. In Figure 3.1, we present a plot showing the number of house sales for each month during

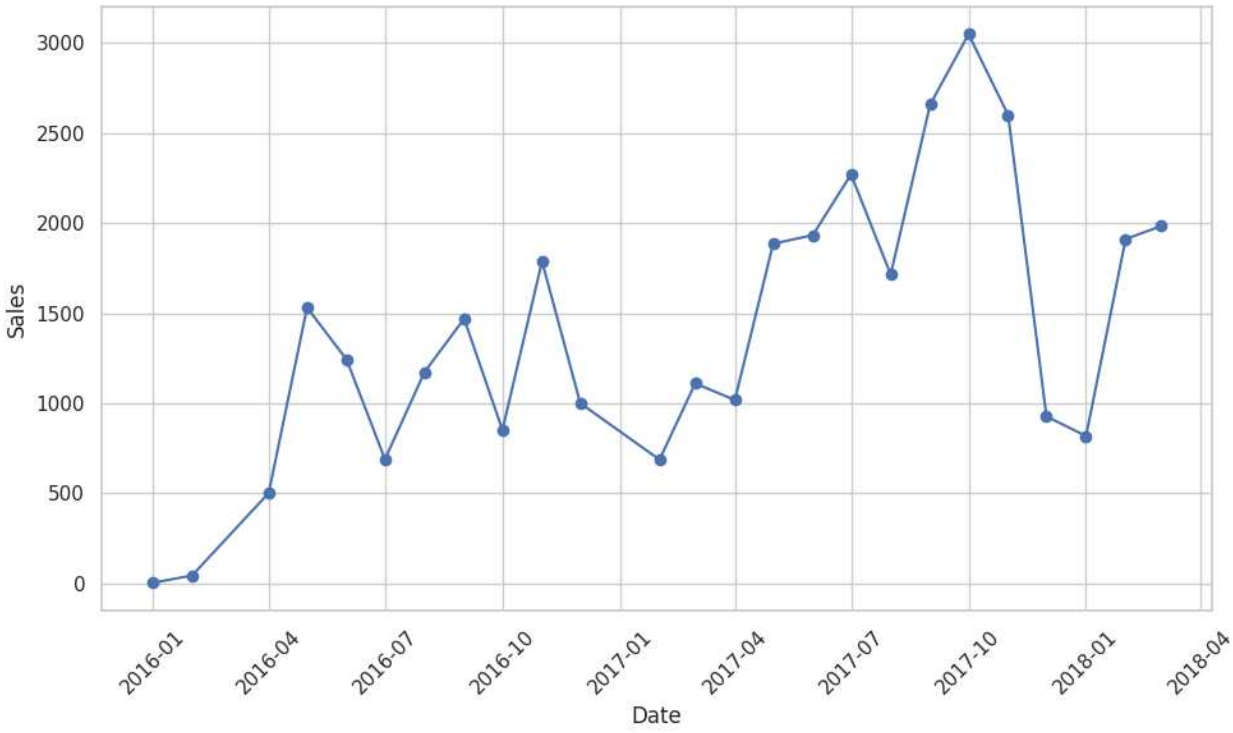


Figure 3.1: Sales Over Time

the data collection period. This figure reveals that, aside from the initial part of 2016 and the final part of 2017, the number of sales remains relatively constant throughout the entire period. This consistency suggests a stable housing market in Melbourne over the observed timeframe, with only minor fluctuations at the beginning and end of the data collection period.

In Figure 3.2, we display the distributions of the key numerical features present in the dataset. The plots provide insights into the nature of these features and highlight some important patterns. For example, certain features follow classic statistical distributions: the distributions of Latitude and Longitude exhibit a Gaussian (normal) shape, indicating that most properties are clustered around central geographic coordinates with fewer properties located at the extremes.

However, other features require preprocessing before they can be effectively used in building predictive models. For instance, the target variable, Price, shows a significant imbalance. The majority of property prices are concentrated between 0 and 4 million Australian dollars, while prices in the range of 4 to 10 million are sparsely represented. This skewed distribution could impact the performance of predictive models, as they might become biased towards predicting values within the

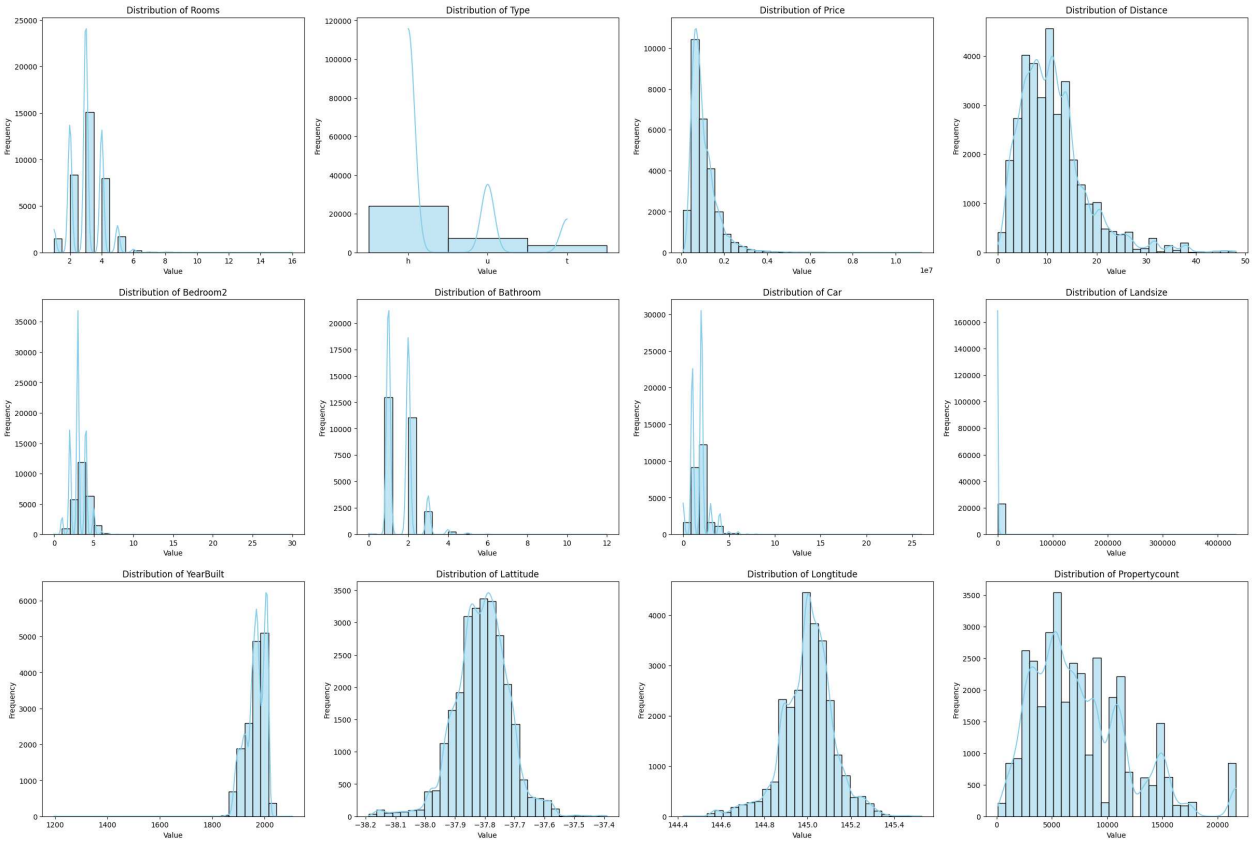


Figure 3.2: Distribution of numerical features

more densely populated price range, thereby underestimating higher-priced properties.

Additionally, anomalies are evident in some features, such as the number of bedrooms and bathrooms. These features exhibit a wide range of values, from 0 to 30 bedrooms and 0 to 12 bathrooms, respectively. Such high values are uncommon for typical residential properties and are likely indicative of luxury estates or hosting facilities. These outliers should be addressed separately, as they fall outside the scope of this study, which focuses on more standard housing market conditions.

To better understand the relationships between the features and the house prices, we also analyzed the correlation matrix, which is reported in Figure 3.3. This matrix provides insights into how different features correlate with each other and with the target variable, price.

Our analysis reveals several interesting correlations. We observe a positive correlation between house price and features such as the number of rooms, bedrooms, and car spots. This result indicates that properties with more rooms, bedrooms, and available car spaces tend to have higher

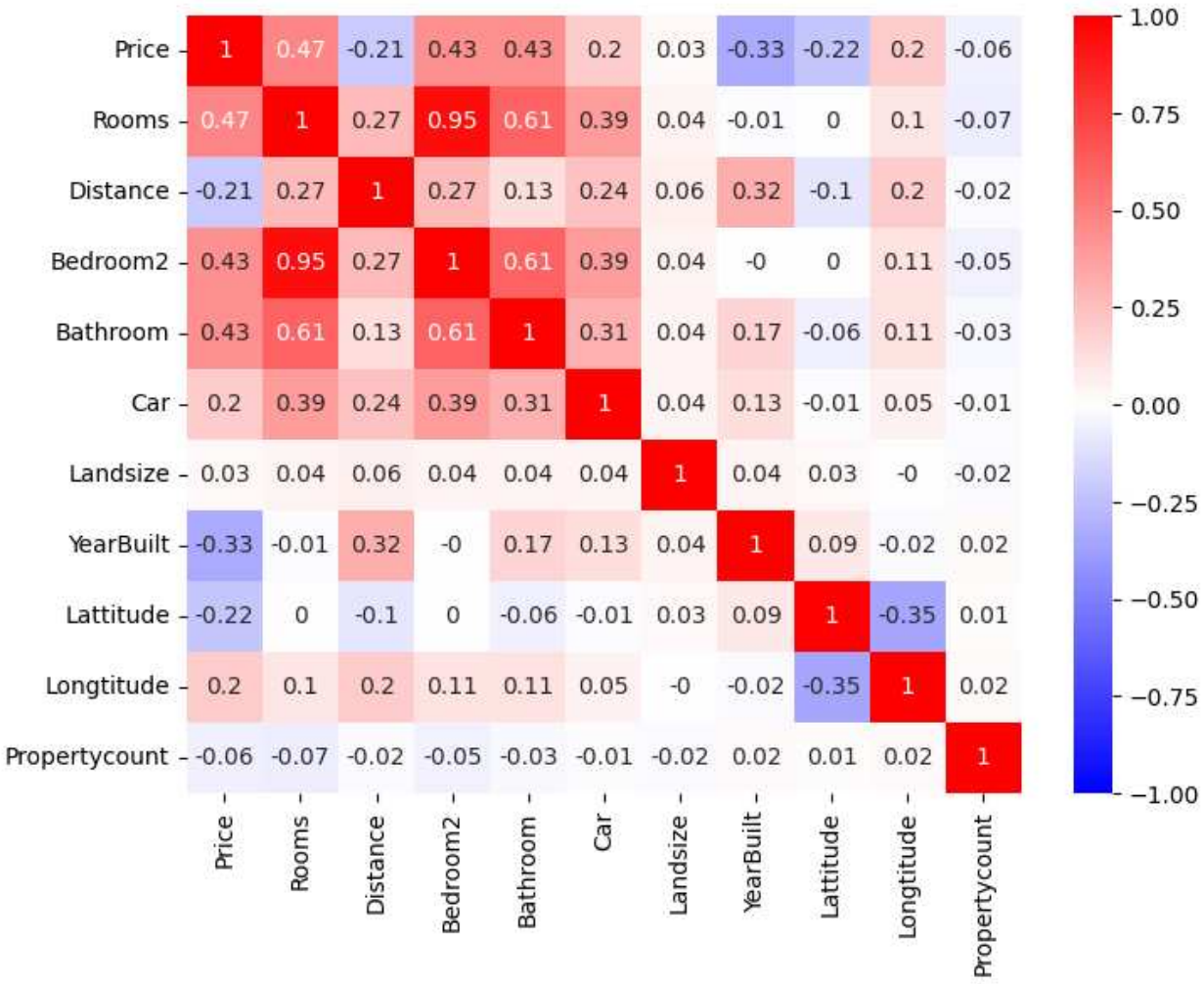
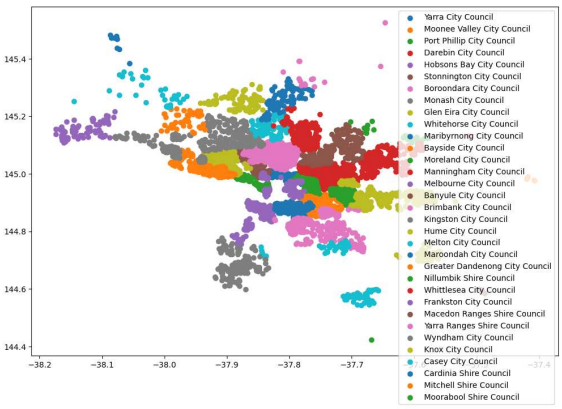


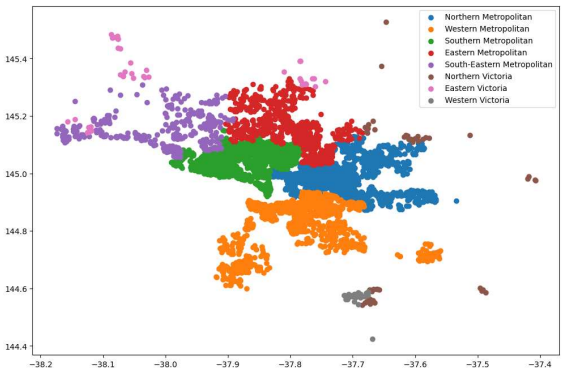
Figure 3.3: Distribution of numerical features

prices, which aligns with the literature and common expectations in the real estate market where larger and more accommodating properties typically lead to higher values.

Another intriguing finding is the small negative correlation between price and latitude. Given that Melbourne is in the Southern Hemisphere, latitudes are negative; hence, a smaller latitude value indicates a location located more southern. The negative correlation suggests that properties situated in more southerly locations tend to have higher prices. In addition to this, there is also a small positive correlation between price and longitude, implying that properties in the eastern suburbs also tend to be more expensive. Together, these correlations suggest that southeastern suburbs of Melbourne may report higher house prices, which could be due to various factors such as desirability of location, amenities, or other socioeconomic factors.



(a) Scatter plot of the properties colored using the council area



(b) Scatter plot of the properties colored using the region name

Figure 3.4: Three simple graphs

The correlation matrix also confirms some expected trends. For instance, there is a negative correlation between the distance to the urban center and the house price, indicating that properties closer to the city center tend to be more expensive. This is in line with the literature, where it was shown that proximity to central business districts and services generally increases property values due to better access to services, employment, and amenities.

Additionally, we observe a negative correlation between the construction year and the price. This suggests that newer properties tend to be more expensive, which could be attributed to modern construction standards, newer amenities, and lower maintenance costs. Conversely, older properties might be less expensive due to the potential need for renovations.

Using this explorative analysis, we are able to understand how data should be preprocessed before performing the modeling. Based on our findings, we decide to focus on a specific subset of the dataset to ensure the accuracy and relevance of our analysis. Specifically, we only consider sales of type “h”, which includes houses, cottages, villas, semi-detached houses, and terraces. This selection helps in creating a more homogeneous dataset for our study.

We also narrowed down our focus to a specific set of features that have been identified as significant through our analysis. These features are: Rooms, Council Area, Region Name, Bathroom, Landsize, Car, Year, Month, Latitude, and Longitude. By concentrating on these features, we can streamline our preprocessing steps and improve the performance of our predictive models.

For the Bathroom feature, we preprocess it by scaling it with respect to the number of rooms. This transformation allows us to represent the ratio between bathrooms and rooms, providing a normalized measure that can be more informative than the raw count, also given its original correlation with the number of Rooms. For the Car feature, we convert it into a binary variable depending on the presence of a car spot. This binary transformation simplifies the feature and makes it more suitable for modeling purposes.

Council Area and Region Name are categorical values that need to be converted into a numerical format for machine learning algorithms. We achieve this by transforming these features into the "one hot" form, which means creating a binary feature for each possible value. This method ensures that the categorical data is appropriately represented without introducing any ordinal relationships.

For visualization purposes, we include maps of the different council areas and region names in Figure 3.4a and Figure 3.4b, respectively. These visualizations provide a geographical context for the categorical features, aiding in the understanding of spatial patterns in the data.

To further refine our dataset, we perform data cleaning steps. We discard rows with missing values to ensure that our analysis is based on complete information. Additionally, we exclude properties with land sizes outside the 10-1000 square meters range, as these extreme values could impact our results and are not representative of typical properties. Furthermore, we limit our analysis to houses with a bathroom-to-rooms ratio smaller than 1.5. Properties outside this range are underrepresented in the dataset and could introduce bias into our models.

3.2 Methodology

The primary goal of this project is to determine the most effective model for forecasting house prices and utilize it to examine the significant variables influencing these prices both in a general sense and as they evolve over time. In this section, we will outline the methodology adopted to carry out the experimental phase of the project. This methodology is structured into three parts, each specifically tailored to address one of the three main objectives of this thesis. Through this

approach, we aim to provide a comprehensive understanding of the dynamics shaping housing market trends and enable informed decision-making in this domain.

What is the best model?

In our regression task aimed at predicting house prices, we train four distinct algorithms: ordinary least squares (linear regression), LASSO regression, decision trees, and random forests. This section introduces the four algorithms and presents the implemented procedure for the training on house prices.

Linear Models: OLS and Lasso

The first two algorithms we use belong to the family of linear models. Linear models are a class of models that assume a linear relationship between the input features and the target variable. The predicted value is a weighted sum of the input features. Mathematically, a linear model can be expressed as:

$$\hat{y} = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b \tag{3.1}$$

, where $\hat{y}$ is the predicted value, x_i are the input features, w_i are the linear weights (coefficients), and b is the intercept term, also referred to as bias. Linear models are widely used due to their simplicity, interpretability, and ease of implementation.

Ordinary least squares consist of a fundamental linear regression method that aims to find the best-fitting line by minimizing the sum of the squared differences between the observed and predicted values. Mathematically, the optimization problem is defined as the minimization of the quadratic error between the true value and the predicted one:

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} (y_i - (\mathbf{x}_i^T \mathbf{w} + b))^2. \tag{3.2}$$

Here, $\mathbf{x}_i$ represents the vector of the feature for the i-th record, and $\mathbf{w}$ is instead the vector of

coefficients. Ordinary least squares provide a straightforward approach to modeling the relationship between input features and the target variable. The coefficients **w** directly indicate the influence of each feature on the predicted outcome, making the model easy to interpret. For small to moderately sized datasets, with a relatively small number of features, ordinary least squares can be computed efficiently. It requires solving a system of linear equations, which can be done quickly using matrix operations. However, ordinary least squares come with some limitations that often make them not suited for solving some regression problems. Ordinary least squares is highly sensitive to outliers because it minimizes the sum of squared errors. Large deviations can disproportionately influence the fit of the model, leading to biased estimates. Ordinary least squares assume a linear relationship between the input features and the target variable. This may not hold true for complex, non-linear relationships, limiting the model's predictive performance in such cases. When the input features are highly correlated, the estimates of the coefficients become unstable and exhibit high variance. This multicollinearity can make the interpretation of the coefficients difficult and the model less reliable. With a large number of features, ordinary least squares can overfit the training data, capturing noise rather than the underlying trend. This results in poor generalization to new, unseen data.

To overcome these limitations, particularly the ones related to overfitting and multicollinearity, a technique called LASSO (Least Absolute Shrinkage and Selection Operator) was developed. LASSO regression is a regularized version of linear regression that adds an L1 penalty to the loss function. This penalty term is designed to feature selection by shrinking some coefficients to zero, effectively removing them from the model. To add the regularization term, the LASSO optimization problem is defined as:

$$\min_{\mathbf{w},b} \left(\sum_{i=1}^{N} (y_i - (\mathbf{x}_i^T \mathbf{w} + b))^2 + \lambda \sum_{j=1}^{p} |w_j| \right) \tag{3.3}$$

,

where λ is a regularization parameter that controls the strength of the penalty, and p is the number of features. During training, the Lasso algorithm iteratively updates the model coefficients by balancing the fit of the model (measured by the residual sum of squares) with the complexity of

the model (measured by the sum of the absolute values of the coefficients). The regularization parameter λ determines how much weight is given to the penalty term. As λ increases, more coefficients are driven to zero, resulting in a simpler model. Moreover, LASSO tends to produce sparse models with fewer non-zero coefficients. This property makes the model easier to interpret and reduces the risk of overfitting by effectively performing feature selection. By driving some coefficients to zero, in fact, LASSO effectively selects a subset of relevant features. This can be particularly useful when dealing with datasets with many features, some of which may be irrelevant or redundant. Additionally, the regularization parameter λ introduces bias into the model but can significantly reduce variance, leading to better generalization to new data. LASSO can even handle multicollinear data better than ordinary least squares by selecting one feature from a group of highly correlated features, thus improving model stability. When features are highly correlated, in fact, LASSO tends to select one and ignore the others, reducing the problem of multicollinearity.

While LASSO, like ordinary least squares, can be sensitive to outliers, the regularization can somewhat mitigate their impact by shrinking large coefficients, thus preventing any single feature from dominating the model.

Decision trees and Random forests

Decision trees [Quinlan, 1986] and random forests [Rigatti, 2017] are two powerful algorithms often used for regression tasks, particularly when dealing with non-linear relationships and complex interactions among features.

A decision tree is a non-parametric model that predicts the target variable by learning simple decision rules inferred from the data features. It splits the data into subsets based on the value of input features, forming a tree-like structure. Each internal node represents a condition on an attribute (e.g., whether a feature value is greater than a threshold), each branch represents the outcome of the condition, and each leaf node represents a predicted value. At each node, the algorithm chooses the best feature and threshold to split the data into two subsets. The best split is determined by optimizing a certain criterion, such as the mean squared error (MSE) for regression tasks. Decision trees come with several advantages. In fact, the tree structure makes it easy to visualize and

understand the model's decisions. Additionally, they can capture non-linear relationships between features and the target variable. However, decision trees often suffer from overfitting as a fully grown decision tree can be very complex and prone to overfitting, capturing noise in the training data.

Random forests address some of the limitations of decision trees by constructing an ensemble of trees and averaging their predictions. This method improves the model's robustness and accuracy. Multiple decision trees are trained on different subsets of the training data, created by random sampling with replacement (bootstrap sampling). At each split in the tree, a random subset of features is considered. This introduces additional randomness and reduces the correlation between trees. For regression tasks, the final prediction is obtained by averaging the predictions of all individual trees. By averaging multiple trees, random forests reduce the risk of overfitting compared to individual decision trees. Random forests can handle missing values effectively by averaging predictions from different trees, some of which may use different features. However, the model is less interpretable compared to a single decision tree, as it aggregates predictions from many trees. Finally, training and predicting with random forests can be computationally intensive, especially with a large number of trees and features.

Overall, tree-based models offer a valuable alternative to linear models in the case of complex tasks where the relationship between the input feature and the output is not linear.

Training procedure

An important aspect of training consists in the selection of the hyperparameters of the models and their validation. This is usually done by enumerating different combinations of hyperparameters and identifying the one that produces the best results. To ensure a correct model evaluation, we start our analysis by splitting the dataset into two portions: a training set and a test set, represented by 70% and 30% of the data, respectively. The training set is used to train the models, while the test set is used to calculate the evaluation metrics.

To select the best hyperparameters for our models, we employ grid-search combined with 3-fold

Table 3.2: Model Parameters

Model	**Parameters**
OLS	-
Decision Tree	max_depth: 3, 5, 7, 10, 20 criterion: squared_error, absolute_error, friedman_mse, poisson
Random Forest	n_estimators: 5, 10, 20, 50, 100, 200 max_depth: 3, 5, 7, 10, 20 criterion: squared_error, absolute_error, friedman_mse, poisson
Lasso	alpha: 0.01, 0.1, 1, 10, 100

Table 3.3: Hyperparameters. List of hyperparameters tested for each of the algorithms.

cross-validation. This technique is chosen because the records in our dataset represent transactions of various properties over a specific time period. We assume these transactions represent independent data points from a pooled cross-section, defined as a combination of independent data collected over time. This assumption justifies our use of grid-search and cross-validation, ensuring that our model selection process is robust and well-suited to the nature of our dataset.

In our methodology, each model is trained using both raw and scaled data features to identify the optimal learning setting. We apply two scaling methods: Min-Max scaling, which transforms features to lie within the range $[0,1]$, and Standard scaling, which normalizes the features' distributions to have a mean (μ) of 0 and a standard deviation (σ) of 1. This dual approach allows us to determine whether scaling has a significant impact on model performance and, if so, which scaling method is most effective.

We provide a detailed list of hyperparameters that we vary for several algorithms, including linear regression, lasso, decision tree, and random forest, in Table 3.3. These hyperparameters are systematically adjusted during the grid-search process to identify the configuration that yields the best performance.

The results of our hyperparameter tuning and model evaluation are presented in Table 3.4. For each algorithm, we report the best scaling method, train and test evaluation metrics, and the optimal

training hyperparameters. Our evaluation metrics include Root Mean Square Error (RMSE) and the coefficient of determination (R^2). RMSE is the square root of the Mean Squared Error (MSE), providing a measure of the average magnitude of the prediction errors. The R^2 metric indicates the proportion of variance in the dependent variable that is predictable from the independent variables in a regression model, and is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.4}$$

Given the complexity and nonlinear nature of price forecasting, we anticipate that tree-based models, such as decision trees and random forests, will perform better compared to linear models. These models are capable of capturing complex interactions between features that are often present in price prediction tasks. In the subsequent section, we will analyze the experimental results to confirm or refute our hypothesis regarding the superiority of tree-based models in this context. By thoroughly evaluating each model's performance, we aim to identify the most accurate and reliable approach for price forecasting based on our dataset.

What are the most influential features?

What has been observed so far enables us to identify the best-performing models and the most effective initialization settings. Our next objective is to examine the influential features within these models. For this analysis, we leverage the inherent characteristics of the algorithms, utilizing their intrinsic parameters to pinpoint the features they rely on the most.

For linear models (OLS and LASSO), we can directly use the learned coefficients, as they explicitly highlight the most relevant features. Each coefficient scales the contribution of a corresponding feature towards the output, making the features with the highest absolute coefficient values the most influential. Since variables can have different ranges, we also consider the coefficients multiplied by the mean values of the variables. This adjustment ensures that we account for the differences in the feature scales, providing a more accurate assessment of each feature's impact.

In the case of tree-based algorithms (decision trees and random forests), the influential features are identified during the training process. As mentioned in the previous sections, these algorithms use specific criteria to construct decision trees, selecting variables to split on based on their importance scores. For this reason, these scores indicate how much each feature contributes to reducing the prediction error and can be used to identify the features that are considered as most influential by the models. For regression problems, various criteria can be used to determine feature importance:

- Squared error: This criterion minimizes the mean squared error (MSE) of the predictions.
- Absolute error: This criterion minimizes the mean absolute error (MAE) of the predictions.
- Friedman mean squared error: An enhancement of the MSE criterion, it is more robust to outliers and improves prediction accuracy.
- Poisson: This criterion is based on Poisson deviance, suitable for modeling count data.

At each node in the tree, the algorithm chooses the feature that results in the highest decrease in the criterion value for splitting. By analyzing these values, we can determine which features are the most influential in the model.

We anticipate that the models will identify features commonly associated with house prices, such as the number of rooms and the size of the house. More importantly, as discussed in the previous chapter, where several studies have demonstrated the significant influence of neighborhoods on house prices, we also expect to see the importance of particular suburbs or regions reflected in the models. This insight into spatial factors will further our understanding of how location-specific characteristics impact property values.

Do feature influences vary over time?

To further study if and how the features influenced prices over time, we then proceed to split the dataset into three subsequent time periods and train a separate model for each of these periods.

Specifically, our data covers a range from January 2016 until April 2018. To facilitate detailed analysis, we divide this entire span into three distinct one-year periods as follows:

- Period 1: from January 2016 until December 2016
- Period 2: from January 2017 until December 2017
- Period 3: from January 2017 until May 2018

In this approach, we maintain the same splits for the training and testing sets, but further divide each of these sets into three additional subsets corresponding to the aforementioned periods. This method allows us to rigorously analyze the influence of different features on prices within each specific time frame, thus providing a clearer understanding of temporal effects.

This particular task presents significant challenges, primarily due to the inherent complexity introduced by the temporal dependence. The evolving nature of the data over time makes it difficult to formulate certain assumptions in advance. However, based on insights from related studies, it is evident that time significantly impacts prices. Furthermore, we anticipate that the influence of various features on prices will also vary over time, reflecting the dynamic interplay between time and feature relevance.

To elaborate, temporal dynamics can introduce shifts in market conditions, consumer preferences, and other external factors that could affect the relationship between features and prices. By training separate models for each time period, we can capture these shifts more effectively. This enables us to not only assess how feature importance evolves over time but also to identify any trends or patterns that may emerge.

Moreover, by examining these periods individually, we can discover potential effects where the influence of certain features may only become influential in specific time delays. Such insights are crucial for developing robust predictive models that account for the temporal variability in the data. This granularity in analysis enhances our ability to make more accurate forecasts and provides a deeper understanding of the underlying mechanisms that drive price changes over time.

3.3 Findings

In this section, we analyze the results of the experiments described above. Again, we split our dissertation into three parts for each research aim, delving into identifying the best model, the features that most influence the price, and how their importance varies over time.

What is the best model?

Table 3.4: Models Evaluation. We report the metrics of the best models for each algorithm according to the grid search with cross-validation.

Model	Scaling	Train RMSE	Train R^2	Test RMSE	Test R^2	Hyperparameters
Linear Regression	MinMax Scaling	408k	0.64	393k	0.65	-
Lasso	MinMax Scaling	409k	0.64	394k	0.65	alpha: 100
Decision Tree	MinMax Scaling	373k	0.70	357k	0.71	criterion: 'absolute_error', max_depth: 7
Random Forest	MinMax Scaling	125k	0.97	279k	0.82	criterion: 'poisson', max_depth: 20, n_estimators: 50

In line with our assumptions, we observe that the linear models, namely linear regression and lasso, cannot produce good quality prediction, reporting high MSE and low R^2. The decision tree is slightly improved, but its performance is still imperfect. An opposite result is instead obtained with random forest. In this case, the model reports an R^2 of 0.82. Although the model performs well, we still acknowledge slight overfitting due to the difference between the performances in the train and test splits.

Additionally, we analyze the price ranges where the model performs better. Indeed, in general, we expect the models to be more accurate around the mean values of the training set. For this analysis, in Figure 3.5, we plot the mean absolute errors through various price ranges. We observe that under 2mln, the model tends to have a mean absolute error of about 250 thousand Canadian dollars, and this error tends to decrease with the decrease in price. This result highlights the high differences between the houses, which highly impact the price. Therefore, it would probably be better to have multiple models for different price ranges. Overall, we find the performances acceptable, as the majority of the houses are usually priced under 2 million Canadian dollars.

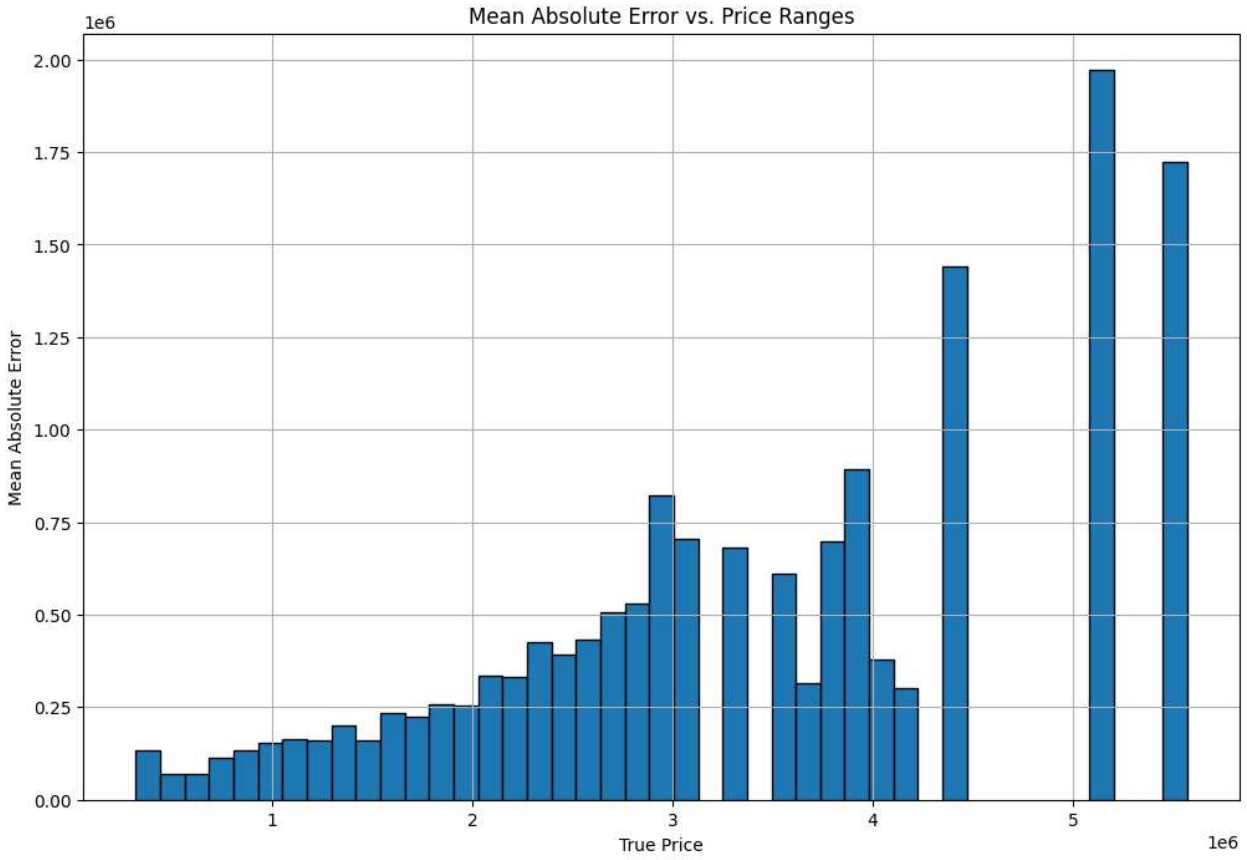


Figure 3.5: Mean absolute error distribution along different price ranges.

What are the most influential features?

In this section, we seek to identify those features that are found to be more determinant with respect to the sale prices. To study the most influential features, we use the random forest model, as it is the best-performing one, and its performance is drastically higher compared to the other models. More specifically, we report the feature importances as a bar plot, highlighting the 15 most influential features, in Figure 3.6. As mentioned above, the feature importances are derived from the model durign training for the selection of the features used to build conditions on the nodes.

We observe that common properties such as the number of rooms, the parking slot and the number of bathrooms are found as primarily used variables. More interestingly, we confirm the hypothesis that neighborhood information is highly determinant for predicting house prices. Specifically, longitude and latitude are very important, confirming the high correlation identified in the explorative

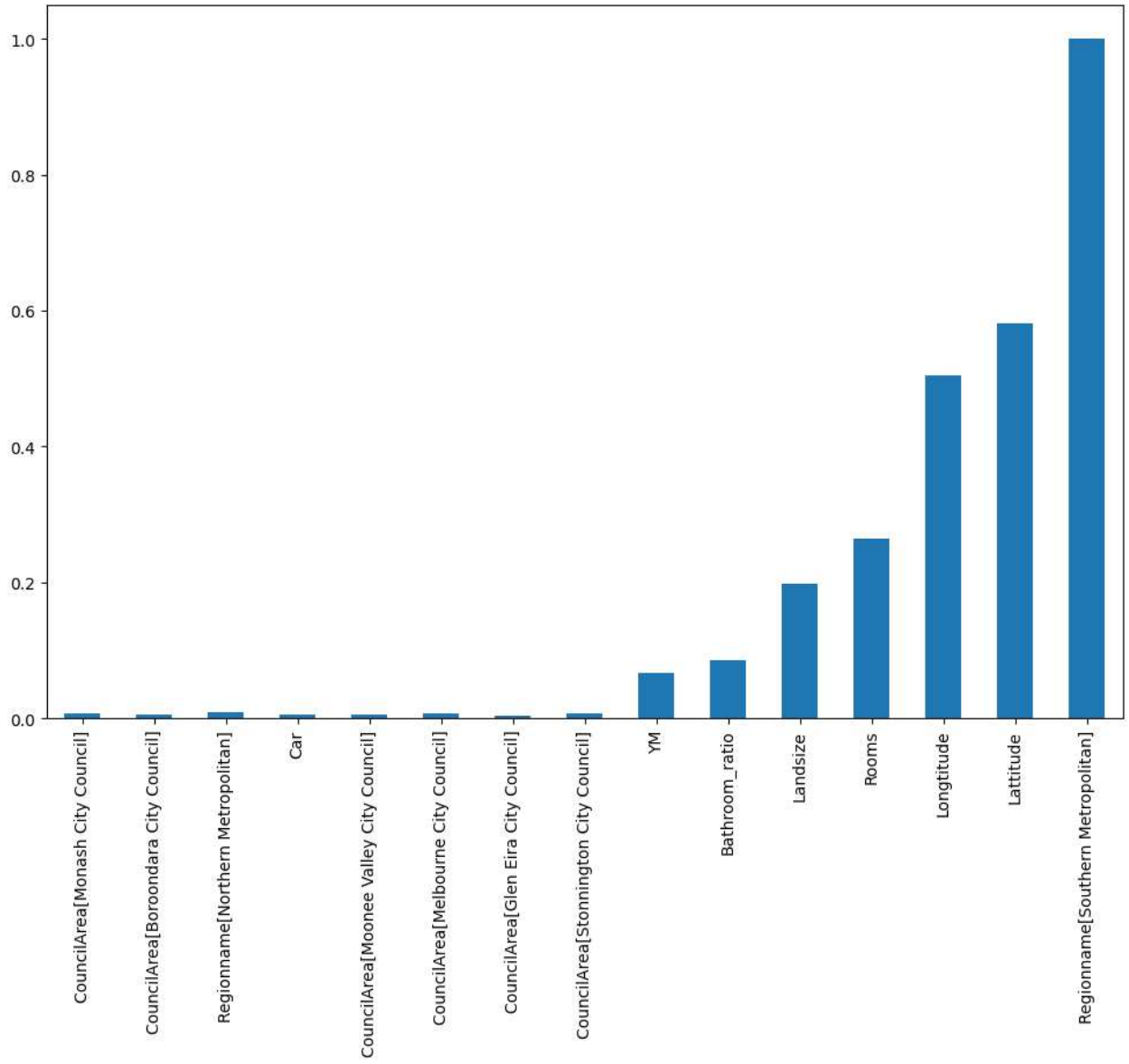


Figure 3.6: Bar plot reporting the 15 highest feature importances extracted from the random forest model.

data analysis.

Additionally, we identify the Southern Metropolitan region to have the highest influence in the prediction. Again, this result confirms what was analyzed in the previous section, where we could suppose south-east areas of Melbourne to be the most expensive due to the price correlation with longitude and latitude. It is also important to note that the model highlights several council areas are influencing, such as the areas of Monash, Borondara, Moonee Valley, Glen Eira and Stonnington.

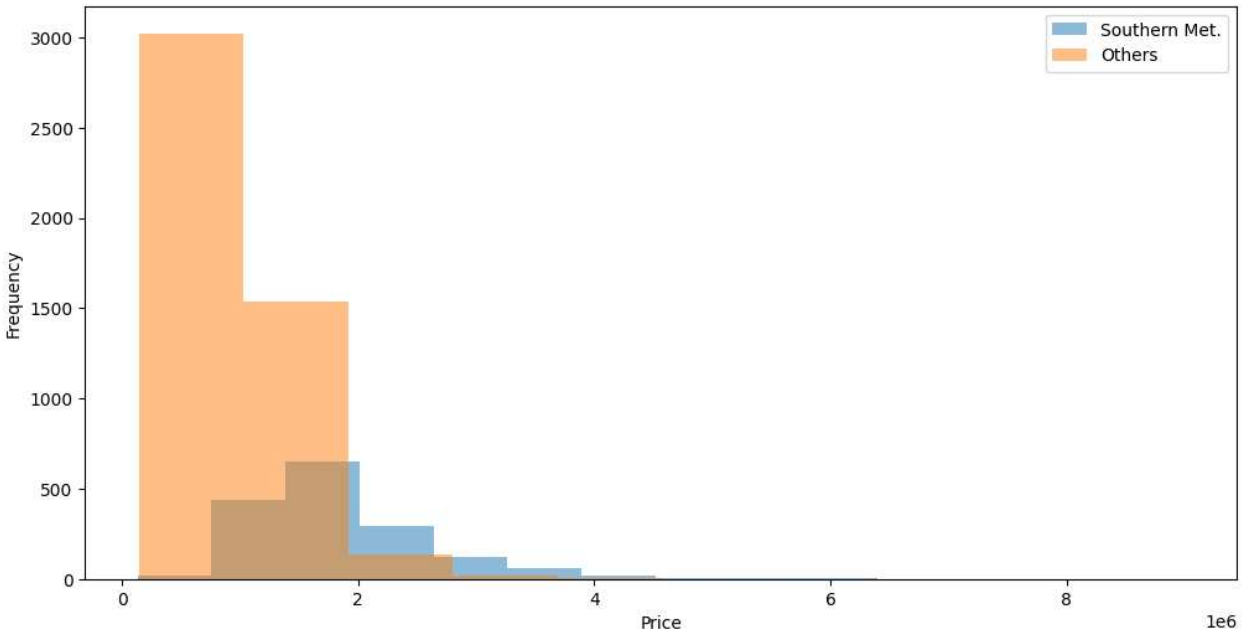


Figure 3.7: Distribution of prices inside and outside Southern Metropolitan Area.

To further validate this assumption, we plot in Figure 3.7 the distribution of house prices for the Southern Metropolitan region compared to other regions. Interestingly, we note that the prices are distributed differently from other regions, explaining why the model specifically focuses on that information for predicting the price.

Finally, it is also important to mention that the model highlights that there could be a correlation between the month of the transaction and the actual price, indicating that sales done in some periods could be more expensive than others.

Do feature influences vary over time?

Table 3.5: Models Evaluation over time. We report the metrics of the random forest models trained on data from different years.

Model	Year	Test R^2	Test RMSE
Random Forest	Full Data	0.82	279k
Random Forest	2016	0.77	308k
Random Forest	2017	0.79	291k
Random Forest	2018	0.77	331k

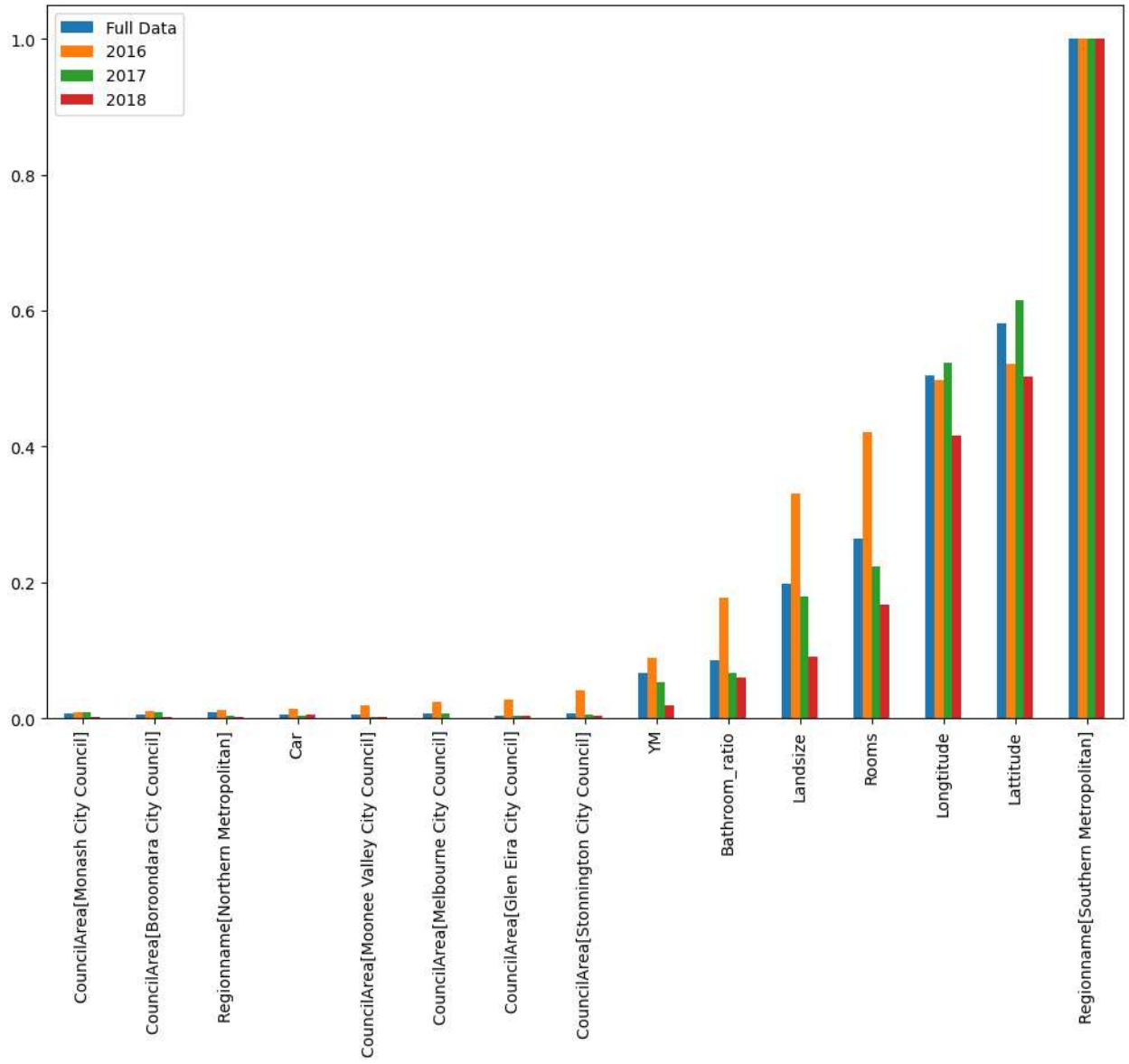


Figure 3.8: Bar plot reporting the 15 highest feature importances extracted from the random forest models trained on data from different years.

This section analyzes whether time plays a role in determining the features' importances. This analysis could, in fact, return interesting results, such as whether some properties gained or lost attention over time or if any neighborhoods incurred a positive or negative variation in house prices.

For this analysis, we train three additional random forest models using the same setting as the previous experiment. In particular, we split the training and test sets into 3 parts for each analyzed year. We then train the models in the corresponding years and evaluate them on the test sets for

the same periods. We report in Table 3.5 the results for the R^2 and RMSE. We observe that the performances are slightly worse than the ones of the model trained on the full training set. We address this decay to the smallest data available to the models. In fact, in the case of 2017, which is the most populated year, as pointed out in the preliminary analysis, the model reaches higher performances compared to the other years.

We finally report in Figure 3.8 the the 15 highest feature importances of the four models to analyze the presence of specific patterns.

We observe that, more or less, the trend is similar over the models. All the models highlight the Southern Metropolitan area as a determinant for the price. Also, latitude and longitude are attributed to be relevant. Interestingly, we note that, over the years, the difference between longitude and latitude seems to be increasing, suggesting a "migration" of the prices toward the latitude. The results indicate a negative trend over the years regarding property details such as land size and the number of rooms and bathrooms. This could indicate that the price could be determined more from the actual location of the house rather than its size.

Also, the scores attributed to the council areas and regions indicate a possible shift in the prices regarding geography. In fact, we observe different patterns for the ones that lie among the 15 most relevant features. For instance, while the Borondara City Council reports a decreasing trend, the results are the opposite for the Monash City Council. Finally, we report a decreasing trend of importance on the transaction period.

3.4 Discussion

In this section, we wrap up the experiments and compare our results with those in the literature, providing a comprehensive analysis of our findings in the context of existing research.

We observe that complex algorithms such as random forests yield better results than simpler regression models. This finding aligns with what was observed by Mullainathan and Spiess [2017], who also noted the superior performance of advanced algorithms in capturing the complexities of

the housing market. This superiority can be attributed to the high nonlinearity of the domain, where multiple factors interact to determine house values.

To further understand these factors, we calculated the importance of features through random forests, focusing on the elements that most influence the housing market. Our analysis revealed that house prices are generally influenced by three main categories:

- House characteristics;
- Spatial and geographical features;
- Temporal factors.

House characteristics that most significantly influence the price include the number of car spots, bathrooms, land size, and rooms. This result is consistent with findings in the literature. For instance, Olmo [1995] identified that apartment age, number of bathrooms, dimensions, rooms, floors, and geographical coordinates are crucial predictors of house prices. Our results reinforce these observations, highlighting the critical role of physical property attributes in price determination.

Spatial and geographical features are also crucial in determining house prices. This is especially pertinent in the Melbourne area, which covers a diverse range of neighborhoods. We identified the southern metropolitan region as particularly significant, showing a distinct price distribution compared to other areas. This finding is in line with Bourassa et al. [2007], who also noted the importance of location in real estate pricing. The spatial variability within Melbourne underscores the need for localized models to capture price dynamics accurately.

Temporal factors, as highlighted by Gupta and Miller [2012], are key to understanding price fluctuations over time. People's preferences and market trends evolve, influenced by events and changes in the territory. Our experiments demonstrated that the importance of various features can shift over time, although their overall relevance remains. This temporal aspect is crucial for developing models that can adapt to changing market conditions and provide more accurate forecasts.

In conclusion, our findings confirm the multi-faceted nature of house price determination, influenced by structural, spatial, and temporal factors. This comprehensive approach not only validates existing literature but also enhances our understanding of the housing market dynamics. Future research should continue to explore these dimensions, potentially incorporating more sophisticated models and additional data sources to further refine predictive accuracy.

Chapter 4

Conclusion

In this thesis work, we presented the problem of forecasting house prices through machine learning methods. This study aimed to explore various approaches and techniques to predict housing prices accurately, contributing to the broader field of real estate analytics.

We presented the main literature on this topic, providing a comprehensive review of existing research and methodologies. We identified three main aspects that should be taken into account when approaching such a task: structural information, spatial variables, and temporal factors. Initially, the problem was confronted only using structural information about the properties, such as the number of bedrooms, bathrooms, and overall square footage. Subsequently, spatial variables, including neighborhood characteristics and proximity to amenities, were taken into consideration, allowing us to achieve better results. Finally, time was accounted for by incorporating historical price trends and economic indicators to predict the value of the houses, further improving performance.

Parallel to this, the research focused on the application of several methods, which became more and more sophisticated over time. Traditional methods such as linear regression and decision trees were initially employed, but more advanced techniques like random forests and neural networks were later utilized. These novel methods allow reaching high performances at the cost of a decreased interpretability, which remains a critical issue in the field.

We then focused on the application of machine learning techniques to a real-world use case for

the Melbourne area. We conducted a thorough analysis to determine the models that were most suited to treat such a task, including ensemble methods and advanced regression techniques. We then shifted our attention to the features that were most responsible for price determination, such as location, property size, and market trends.

Our findings confirmed that all three aspects—structural, spatial, and temporal—play an important role in our use case, validating what was previously presented in the literature. This comprehensive approach enabled us to produce more accurate and reliable forecasts, highlighting the importance of a multifaceted analysis.

This work limits the application of machine learning techniques without any attention to deep learning models such as neural networks. While these could further improve performance, they could also further decrease the model's interpretability, making it challenging for stakeholders to understand the underlying decision-making process.

Future work could then focus on the application of deep learning in this domain and how to bridge the trade-off between performance and interpretability. This could involve developing hybrid models that combine the strengths of traditional machine learning and deep learning or creating visualization tools that make complex models more transparent. Further research could also explore the integration of additional data sources, such as social media trends or economic forecasts, to enhance the predictive power of these models.

Bibliography

Jorge Chica Olmo. Spatial estimation of housing prices and locational rents. *Urban Studies*, 32: 1331–1344, 8 1995. ISSN 0042-0980. doi: 10.1080/00420989550012492.

Steven C. Bourassa, Eva Cantoni, and Martin Hoesli. Spatial dependence, housing submarkets, and house price prediction. *The Journal of Real Estate Finance and Economics*, 35:143–160, 8 2007. ISSN 0895-5638. doi: 10.1007/s11146-007-9036-8.

Liv Osland and Inge Thorsen. Predicting housing prices at alternative locations and under alternative scenarios of the spatial job distribution. *Letters in Spatial and Resource Sciences*, 2: 133–147, 10 2009. ISSN 1864-4031. doi: 10.1007/s12076-009-0030-z.

L. Miguel Martínez and José Manuel Viegas. Effects of transportation accessibility on residential property values. *Transportation Research Record: Journal of the Transportation Research Board*, 2115:127–137, 1 2009. ISSN 0361-1981. doi: 10.3141/2115-16.

Olga Filippova and Michael Rehm. The impact of proximity to cell phone towers on residential property values. *International Journal of Housing Markets and Analysis*, 4:244–267, 8 2011. ISSN 1753-8270. doi: 10.1108/17538271111153022.

Linchuan Yang, Bo Wang, Jiangping Zhou, and Xu Wang. Walking accessibility and property prices. *Transportation Research Part D: Transport and Environment*, 62:551–562, 7 2018. ISSN 13619209. doi: 10.1016/j.trd.2018.04.001.

Carlos Marmolejo-Duarte. Does urban centrality influence residential prices? an analysis for the barcelona metropolitan area. *Revista de la construcción*, 16:57–65, 5 2017. ISSN 0718915X. doi: 10.7764/RDLC.16.1.57.

Timothy F. Welch, Steven R. Gehrke, and Fangru Wang. Long-term impact of network access to bike facilities and public transit stations on housing sales prices in portland, oregon. *Journal of Transport Geography*, 54:264–272, 6 2016. ISSN 09666923. doi: 10.1016/j.jtrangeo.2016.06.016.

Marco De Nadai and Bruno Lepri. The economic value of neighborhoods: Predicting real estate prices from the urban environment. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 323–330, 2018. doi: 10.1109/DSAA.2018.00043.

Rangan Gupta and Stephen M. Miller. The time-series properties of house prices: A case study of the southern california market. *The Journal of Real Estate Finance and Economics*, 44:339–361, 4 2012. ISSN 0895-5638. doi: 10.1007/s11146-010-9234-7.

Xiaolong Liu. Spatial and temporal dependence in house price prediction. *The Journal of Real Estate Finance and Economics*, 47:341–369, 8 2013. ISSN 0895-5638. doi: 10.1007/s11146-011-9359-3.

Jiping Liu, Yi Yang, Shenghua Xu, Yangyang Zhao, Yong Wang, and Fuhao Zhang. A geographically temporal weighted regression approach with travel distance for house price estimation. *Entropy*, 18:303, 8 2016. ISSN 1099-4300. doi: 10.3390/e18080303.

Yu Zhang, Dachuan Zhang, and Eric J. Miller. Spatial autoregressive analysis and modeling of housing prices in city of toronto. *Journal of Urban Planning and Development*, 147, 3 2021. ISSN 0733-9488. doi: 10.1061/(ASCE)UP.1943-5444.0000651.

Steven J Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1):31–39, 2017.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

Sendhil Mullainathan and Jann Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

Quang Truong, Minh Nguyen, Hy Dang, and Bo Mei. Housing price prediction via improved machine learning techniques. *Procedia Computer Science*, 174:433–442, 2020. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2020.06.111. URL `https://www.sciencedirect.`

com/science/article/pii/S1877050920316318. 2019 International Conference on Identification, Information and Knowledge in the Internet of Things.

Ayush Varma, Abhijit Sarma, Sagar Doshi, and Rohini Nair. House price prediction using machine learning and neural networks. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1936–1939, 2018. doi: 10.1109/ICICCT.2018.8473231.

Juan Ramón Rico-Juan and Paloma Taltavull de La Paz. Machine learning with explainability or spatial hedonics tools? an analysis of the asking prices in the housing market in alicante, spain. *Expert Systems with Applications*, 171:114590, 6 2021. ISSN 09574174. doi: 10.1016/j.eswa.2021.114590.

Appendix

Data cleaning

```
import pandas as pd

# Read the dataset
df = pd.read_csv("melbourne_housing_dataset/Melbourne_housing_FULL.csv")

# Calculate Bathroom ratio
df["Bathroom_ratio"] = df["Bathroom"] / df["Rooms"]

# Extract Year and Month from Date
df["Year"] = pd.to_datetime(df["Date"], format="%d/%m/%Y").dt.year
df["Month"] = pd.to_datetime(df["Date"], format="%d/%m/%Y").dt.month

# Create a feature representing Year and Month
df["YM"] = (df["Year"] - 2016) + (df["Month"] / 13)

# Filter rows where Type is 'h'
df = df[df["Type"] == "h"]

# Filter rows with Landsize less than 1000 and greater than 10
df = df[(df["Landsize"] < 1000) & (df["Landsize"] > 10)]

# Filter rows with Bathroom ratio less than 1.5
df = df[df["Bathroom_ratio"] < 1.5]

```

```
# Remove rows with missing Price values
df = df[~df["Price"].isna()]

# Drop rows with any missing values
df = df.dropna()
```

Data Preprocessing

```
# Define columns of interest for X (features) and y (target)
x_cols = ["CouncilArea", "Rooms", "Bathroom_ratio", "Landsize",
        "Regionname", "Car", "Year", "YM", "Lattitude", "Longtitude"]
# Extract features into X_df
X_df = df[x_cols].copy()
# Extract target variable into y_df
y_df = df["Price"]

# Convert Car feature into binary indicating if the property has a car
    spot or not
X_df["Car"] = (X_df["Car"] > 0).astype(int)

# Define categorical columns for one-hot encoding
categorical_cols = ["Regionname", "CouncilArea", "Year"]
# Perform one-hot encoding for categorical variables
for col in categorical_cols:
    # Get one-hot encoded columns
    ohe_col = pd.get_dummies(X_df[col])
    # Rename one-hot encoded columns to include the original column name
    ohe_col.columns = [f"{col}[{c}]" for c in ohe_col.columns]

    # Concatenate one-hot encoded columns with X_df, dropping the original
    categorical column
    X_df = pd.concat([X_df.drop(columns=[col]), ohe_col], axis=1)
```

Training and evaluation

```
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Define models to be used
models = {
    "Linear Regression": LinearRegression(),  # Ordinary Least Squares
    "Decision Tree": DecisionTreeRegressor(),
    "Random Forest": RandomForestRegressor(),
    "Lasso": Lasso()
}

# Define hyperparameters grid for grid search
params = {
    "Linear Regression": {},
    "Decision Tree": {
        "max_depth": [3, 5, 7, 10, 20],
        "criterion": ["squared_error", "absolute_error",
                      "friedman_mse", "poisson"],
        "random_state": [0]
        },
    "Random Forest": {
        "n_estimators": [5, 10, 20, 50, 100, 200],
        "max_depth": [3, 5, 7, 10, 20],
        "criterion": ["squared_error", "absolute_error",
                      "friedman_mse", "poisson"],
        "random_state": [0]
        },
    "Lasso": {
        "alpha": [0.01, 0.1, 1, 10, 100]}
```

```
}

best_models = {}

# Train models and perform grid search
for model_name, model in models.items():
    # Extract parameters for the specific model
    p = {f"model__{k}": v for k, v in params[model_name].items()}
    scaling_models = {
        "w/o Scaling": Pipeline([
                ("model", model)
                ]),
        "w/ Min Max Scaling": Pipeline([
                ("scaler", MinMaxScaler()),
                ("model", model)
                ]),
        "w/ Standard Scaling": Pipeline([
                ("scaler", StandardScaler()),
                ("model", model)
                ]),
    }
    for scaling_name, scaling_model in scaling_models.items():
        name = f"{model_name} {scaling_name}"
        grid_search = GridSearchCV(scaling_model, p,
                    cv=3, verbose=2,
                    scoring='neg_mean_squared_error')
        # Fit the model
        grid_search.fit(X_train, y_train)
        best_models[name] = grid_search
        print(f"Best parameters for {name}: {grid_search.best_params_}")

# Evaluate models
for name, model in best_models.items():
    # Evaluate on training data
    y_pred_train = model.best_estimator_.predict(X_train)
    mse_train = mean_squared_error(y_train, y_pred_train)
```

```
    r2_train = r2_score(y_train, y_pred_train)
    print(f"Train Eval for {name}: {np.sqrt(mse_train)} - {r2_train}")

    # Evaluate on test data
    y_pred_test = model.best_estimator_.predict(X_test)
    mse_test = mean_squared_error(y_test, y_pred_test)
    r2_test = r2_score(y_test, y_pred_test)
    print(f"Test Eval for {name}: {np.sqrt(mse_test)} - {r2_test}")
    print()
```