Annals of Operations Research (2022) 308:571–608
https://doi.org/10.1007/s10479-021-03932-5

S.I. : ARTIFICIAL INTELLIGENCE IN OPERATIONS MANAGEMENT

Real estate price estimation in French cities using geocoding and machine learning

Dieudonné Tchuente[1] · Serge Nyawa[1]

Accepted: 6 January 2021 / Published online: 16 February 2021

Abstract

This paper reviews real estate price estimation in France, a market that has received little attention. We compare seven popular machine learning techniques by proposing a different approach that quantifies the relevance of location features in real estate price estimation with high and fine levels of granularity. We take advantage of a newly available open dataset provided by the French government that contains 5 years of historical data of real estate transactions. At a high level of granularity, we obtain important differences regarding the models' prediction powers between cities with medium and high standards of living (precision differences beyond 70% in some cases). At a low level of granularity, we use geocoding to add precise geographical location features to the machine learning algorithm inputs. We obtain important improvements regarding the models' forecasting powers relative to models trained without these features (improvements beyond 50% for some forecasting error measures). Our results also reveal that neural networks and random forest techniques particularly outperform other methods when geocoding features are not accounted for, while random forest, adaboost and gradient boosting perform well when geocoding features are considered. For identifying opportunities in the real estate market through real estate price prediction, our results can be of particular interest. They can also serve as a basis for price assessment in revenue management for durable and non-replenishable products such as real estate.

Keywords Real estate market · Automated valuation models · Investment · Geocoding · French cities · Machine learning · Artificial intelligence

✉ Dieudonné Tchuente
d.tchuente@tbs-education.fr

Serge Nyawa
s.nyawa@tbs-education.fr

[1] Dep. of Information, Operations and Management Sciences, Toulouse Business School, 1 Place Alphonse Jourdain, 31068 Toulouse, France

1 Introduction

Revenue management is a component of operations management that focuses on pricing to increase the profits generated from a limited amount of supply chain assets (Dana 2008). Concepts of revenue management are applied successfully across many capacity-constrained service industries for nondurable and perishable products (Berk et al. 2009), such as the airline industry for flight ticket prices (Li and Tang 2012), the car rental industry for car rental prices (Geraghty and Johnson 1997) or the hotel industry for booking prices (Harewood 2006). On the other hand, only a few studies are interested in revenue management for durable and non-replenishable (in the short term) products such as real estate (Wen et al. 2016; Padhi et al. 2015).

Considered for most countries as the largest asset class, real estate plays a major role in social and economic systems. Real estate price fluctuations have direct impacts on the financial system due to banks' central role as mortgage lenders and the frequent use of real estate as collateral (Koetter and Poghosyan 2010). However, acquiring real estate is a delicate operation that requires a precise and objective estimate of its value beforehand. Since buying a house is the largest financial transaction for most households (Pedersen et al. 2013), knowing the real value of a home is a major asset in more ways than one and allows the buyer to not only distinguish between good and bad deals but also to be able to effectively negotiate the price of the property during the transaction. On the seller's side, the precise estimate of the price of his/her home before it goes on sale allows him/her to know its exact market value. As a result, the seller can then avoid any unnecessary risk of overestimating or underestimating the sale price. It should be noted that when the sale price is overestimated, it almost surely causes a delay in the sale, while underestimation generates an unnecessary loss of profit for the seller. Additionally, an accurate estimate of house value is considered capital to an investor willing to diversify his/her portfolio because of the alternatives among housing securities and other possible investments (D'Amato et al. 2019). Therefore, it is crucial and greatly beneficial for both sellers and buyers to have tools that facilitate the estimation of real estate values.

Real estate prices are sometimes studied for rental price assessments (Gomes 2009; Gomes and Rangel 2009), but they are mostly analyzed for property price assessments with automated valuation models (Pagourtzi et al. 2003; d'Amato and Kauko 2017; Wang and Li 2019; Valier 2020). Automated valuation models (AVMs) are statistically-based models that use real estate information, such as property characteristics (e.g., age, number of rooms), comparable sales, or price trends, to provide a current estimate of the market value of a specific property. Generally, valuations are required, and they are often carried out by several different players in the marketplace, such as real estate agents, appraisers, assessors, mortgage lenders, brokers, property developers, investors, fund managers, market researchers, analysts, etc. The most commonly used approaches for automated valuation models are based on parametric and nonparametric regression techniques.

The parametric regressions used for automated valuation models are mostly based on hedonic regressions, such as multiple linear regression analysis (Narula et al. 2012). Due to the complexity and the nonlinearity of the real estate price estimation problem (Yu and Wu 2006; Kontrimas and Verikas 2011), various nonparametric regression methods, such as data envelopment analysis (Lins et al. 2005), fuzzy logic (Kuşan et al. 2010) or genetic algorithms (Morano et al. 2018), are also used. In general, machine learning methods are currently among the emerging nonparametric methods most used for automated valuation models (Viriato 2019; Valier 2020). Several studies are interested in empirically comparing the prediction accuracies of machine learning methods versus those of hedonic regression methods. Although a few studies show that some hedonic regressions can provide better results in some specific contexts (Doumpos et al. 2020), machine learning methods generally outperform hedonic regressions in many studies (Valier 2020; Mayer et al. 2018; Pérez-Rave et al. 2019). However, beyond their predictive capacities, these two approaches are often different according to their targets; hedonic regressions are explanatory, interpretable and less volatile models that can successfully address numerous economic, social, environmental and public policy issues, while machine learning models are very often less interpretable ("black box") (Yacim and Boshoff 2018) and more volatile models (Mayer et al. 2018), but they provide more powerful predictive capacity than hedonic regressions (Din et al. 2001; McCluskey et al. 2013; Mayer et al. 2018). Machine learning models are attractive to all operators who evaluate, manage or trade real estate assets. Investors can use them to evaluate the possible investments or transactions for which they are a party. Similarly, valuation service providers can use them to offer reliable estimates to their clients. In this study, we are primarily interested in the prediction accuracies of the models (e.g., from the point of view of an investor); therefore, we focus on machine learning models. Because their relevance has already been demonstrated in different contexts in the real estate price estimation literature (Isakson 1988; Kontrimas and Verikas 2011; Huang 2019; Lam et al. 2009; Mullainathan and Spiess 2017; Čeh et al. 2018; Kok et al. 2017; McCluskey et al. 2014; Baldominos et al. 2018), we consider the following seven machine learning models in this study: artificial neural networks (multilayer perceptron), ensemble learning (random forest, gradient boosting, adaboost), support vector regression, k-nearest neighbors and linear regression.

The input explanatory variables always play a major role with regard to the relevance of automated parametric or nonparametric models. Several types of explanatory variables are commonly used to estimate the prices of properties, such as the following: physical characteristic variables (e.g., living area, number of rooms), accessibility variables (e.g., proximity to amenities such as schools), neighborhood socioeconomic variables (e.g., local unemployment rates) (Johnson 2003), and environmental variables (e.g., road noise or visibility impact) (Čeh et al. 2018). Depending on the availability of all these explanatory variables, the heterogeneity of real estate features is responsible for the laboriousness of the price estimation process. However, there is a consensus in the literature on the prime importance of location/spatial variables (e.g., geographic coordinates, accessibility variables or neighborhood variables) when estimating real estate prices (Anselin 2013). Several empirical studies support this argument by handling spatial heterogeneity and spatial dependence (e.g., Basu and Thibodeau 1998; Bourassa et al. 2003; Bitter et al. 2007; Borst and McCluskey 2008; Helbich and Griffith 2016; Gröbel and Thomschke 2018; Doumpos et al. 2020). However, even if very few of these studies considered machine learning techniques, none of them consistently evaluated and quantified the relevance of spatial/location attributes among a wide range of machine learning techniques. We are tackling this latter point in this paper. Thus, our research question can be summarized as follows:

RQ What would be lost in terms of predictive power for a machine learning-based automated valuation model that fails to integrate location variables?

We study this research question by analyzing the French real estate market, which has so far received little attention. The French housing market is quite tight (Garcia and Alfandari 2018). Citizens invest in real estate to build wealth or to collect additional income. Considering the volume of rentals, the resulting tax savings and the reduced effort required to obtain savings, the rental real estate market is one of the few investment sectors in France that allows one to build up a sustainable heritage financed with credit without having exceptional income. If real estate investment is a successful these days, it is partially due to the mechanism that it offers to investors allowing for a reduction in income taxes. For example, the current *Pinel law*[1] provides income tax reductions of up to 21% over 12 years for new real estate. In this context, increasing numbers of people are interested in quickly identifying good opportunities for investing in real estate in France. However, the French real estate market is widely heterogeneous, with several different metropolitan areas. For instance, Paris is the economic and political capital, and its real estate market is particularly tight. This is also the case for other metropolitan areas, such as Nice and Bordeaux, but for different reasons (touristic and bourgeois cities, respectively). To assess their predictive capacities for such different towns, we evaluate the machine learning models for the following nine major metropolitan French areas: Paris, Marseille, Lyon, Toulouse, Lille, Bordeaux, Montpellier, Nice, and Nantes.

Overall, the main contributions of this paper compared to the literature are as follows:

- A global evaluation and a quantification of the relevance of location/spatial attributes for real estate price estimations using a wide range of machine learning methods are performed. To the best of our knowledge, this is the first study with this specific goal; at a fine-grained level, the location attributes in this study are derived from geocoding processing of the properties' addresses.
- The evaluations are performed with the same dataset, thus avoiding the bias that could appear when comparing different methods evaluated on different datasets, as in many literature reviews (e.g., Wang and Li 2019; Valier 2020).
- The study focuses on the French real estate market, which has so far received little attention. We study 5 years (2015–2019) of real sales data from notarial acts containing 480 055 house and apartment transactions.
- The machine learning models' predictive powers are evaluated and compared at a high level of granularity using data from nine different and heterogeneous metropolitan areas.

As the summary results are compared at a high level of granularity, we obtain important differences regarding the models' predictive powers (beyond 70% differences in precision in some cases) between cities with high standards of living (e.g., Paris, Bordeaux, Nice) and cities with medium standards of living (e.g., Toulouse, Lille, Montpellier). At a low level of granularity, we use geocoding to extract from and add precise geographical location features to the machine learning algorithm inputs. We obtain important improvements regarding the models' forecasting powers (improvements beyond 50% for some forecasting error measures) compared to the models trained without these features. Regarding the machine learning methods, our results reveal that neural networks and the random forest particularly outperform the other methods when geocoding features are not accounted for, while the ensemble learning methods (random forest, adaboost and gradient boosting) perform well when geocoding features are considered.

The rest of this paper is structured as follows: the next section presents some related works, followed by a description and an exploration of the dataset and a presentation of the methods used in our experiments. The subsequent section presents our experiments and the results obtained with and without geocoding processing. The succeeding section provides

[1] www.pinel-loi-gouv.fr/.

a discussion of our results, as well as implications and limitations of the study and future research directions. Finally, the last section concludes the paper.

2 Related works

The importance of location in determining housing prices is widely recognized. The key econometric issues include spatial dependence and spatial heterogeneity (Anselin 2013). Spatial dependence exists because nearby properties often have similar structural features (they were often developed at the same time) and share locational amenities (Basu and Thibodeau 1998). Spatial heterogeneity focuses on whether the marginal prices of housing characteristics are constant throughout a metropolitan area or whether they change over space (Bitter et al. 2007). To improve traditional automated valuation models, locations or spatial features are widely integrated in parametric and nonparametric methods for modeling spatial heterogeneity or spatial dependence. These methods can be classified into the following four groups: (1) market segmentation (or submarket) methods, (2) trend surface models and spatial expansion methods, (3) spatial regression methods and (4) machine learning methods with spatial attribute. Empirical studies commonly either use spatial methods in comparison with models without spatial features or compare spatial methods with one another.

2.1 Market segmentation (submarket) methods

Submarket or market segmentation methods (Bourassa et al. 1999, 2003, 2010; Goodman and Thibodeau 1998, 2003, 2007) are approaches for dealing with spatial heterogeneity by delineating the housing market into distinct submarkets. Submarkets can be defined as physical geographical areas or noncontiguous groups of dwellings with similar characteristics and/or hedonic prices. Estimates are either performed separately for each submarket or globally by adding spatial indicators, such as dummy variables for submarkets, and performing price estimates for the whole market. The aim is not necessarily to define relatively homogeneous submarkets consisting of substitutable dwellings but rather to segment the market in a way that allows for accurate estimates of house values. For example, (Bourassa et al. 2003) compared a set of spatial submarkets defined by real estate appraisers with a set of non-spatial submarkets created using factor and cluster analysis. They also considered the impacts of adjusting predictions by using the neighboring properties' residuals. Using data for Auckland, New Zealand, they found that the most accurate predictions are obtained by using a citywide equation with spatial submarket dummy variables and by adjustment with neighboring residuals. The separate submarket equations performed slightly worse or better than the citywide equation, depending on whether the predictions were or were not adjusted for the neighboring residuals, respectively. (Goodman and Thibodeau 2003) compared the predictions for three submarkets with those of a market-wide model from Dallas. The submarket models were defined based on ZIP codes, census tracts, and a hierarchical method described in (Goodman and Thibodeau 1998). They concluded that each of the submarket definitions yielded significantly better results than those of the market-wide model, but none of the submarket definitions dominated the others. (Goodman and Thibodeau 2007) compared spatial submarkets consisting of adjacent census block groups with non-spatial submarkets constructed based on dwelling sizes and prices per square foot. Both submarket methods produced significantly better predictions than the results obtained from the market-wide model, although neither clearly dominated the other.

2.2 Trend surface models and spatial expansion methods

Trend surface models and spatial expansion methods integrate spatial attributes into traditional hedonic regression methods. The principle of a trend surface model is to use a regression function that estimates the property value at any location based on the two coordinates (latitude and longitude) of the location (Clapp 2003; Xu 2008; Orford 2017; Doumpos et al. 2020). The spatial expansion method allows house characteristics to vary over space in a traditional hedonic regression framework by the interaction of house characteristics with locational information (Thériault et al. 2003; Fik et al. 2003; Bitter et al. 2007). For example, (Thériault et al. 2003) used an expansion model that allows housing attributes to vary based on both accessibility and neighborhood attributes. In a study by Tucson, (Fik et al. 2003) specified a fully interactive expansion model employing a second-order polynomial expansion of housing attributes (properties' geographical coordinates) and dummy variables representing submarkets. The interactions between the absolute location variables and structural attributes allowed the coefficients to vary over space. This model outperformed the stationary model, and its explanatory power was far superior. Several spatial interactive terms were significant, indicating the presence of spatial heterogeneity in the prices of these attributes.

2.3 Spatial regression methods

Spatial regression models have been developed to make estimations and predictions about space by explicitly modeling the spatial correlations among observations in different locations. For automated valuation models, the most commonly used spatial regressions include methods such as geographically weighted regressions (GWRs) (Bitter et al. 2007; Borst and McCluskey 2008; Lockwood and Rossini 2011; McCluskey et al. 2013; Bidanset et al. 2017) and simultaneous autoregressive (SAR) models or conditional autoregressive (CAR) models (Bourassa et al. 2007). The GWR method is a local modeling approach that explicitly allows parameter estimates to vary over space. Rather than specifying a single model to characterize the entire housing market, GWR estimates a separate model for each sale point and weights the observations by their distance to this point, thus allowing for unique marginal-price estimates at each location. This method is appealing because it mimics, to some extent, the "sales comparison" approach to valuation used by appraisers in that only sales within proximity to the subject property are considered, and price adjustments are made based on the differences in the characteristics within this subset of properties. (Bitter et al. 2007; Helbich and Griffith 2016) found that GWR outperforms many standard hedonic regressions and spatial expansion methods. (Borst and McCluskey 2008; McCluskey and Borst 2011) applied GWR successfully to identify the existence of housing submarkets. Their findings demonstrated an increase in predictive accuracy when using the GWR approach across three large urban areas in the USA. These findings seemingly indicated that the local variation explicitly addresses spatial dependency as a continuous function, which led to the analysis of the relationships between properties, depending on the distance from one to another. In the case of lattice models, such as the SAR and CAR models, locations are restricted to the discrete set of points represented by the data used to estimate the model. Using data for Auckland, New Zealand, (Bourassa et al. 2007) compared a simple hedonic regression model that included submarket dummy variables with geostatistical (similar to GWR) and lattice (CAR and SAR) models. They showed that the lattice methods performed poorly in comparison with the geostatistical approaches or even in comparison with a simple hedonic regression model that ignores spatial dependence; however, they did not use the neighboring properties' residuals or the spatial weight matrix to improve the prediction accuracy. Their best results were obtained by incorporating submarket variables into a geostatistical framework.

2.4 Machine learning methods with spatial attributes

The last group includes a few studies that integrated machine learning methods with spatial attributes and compared them with some of the previous methods or with non-spatial methods (McCluskey et al. 2013; Mayer et al. 2018; Čeh et al. 2018; Doumpos et al. 2020). (McCluskey et al. 2013) assessed and analyzed a number of geostatistical approaches relative to an artificial neural network (ANN) model and the traditional linear hedonic model. The findings demonstrated that ANNs can perform very well in terms of predictive power and, therefore, valuation accuracy, outperforming traditional multiple regression analysis and approaching the performances of spatially weighted regression approaches. The results of (Doumpos et al. 2020) demonstrated that linear regression models developed with a weighted spatial (local) scheme provide the best results, outperforming the machine learning approaches and models that do not consider spatial effects. However, the two machine learning approaches in their study (random forest and gaussian process regression) provided the best results in a global setting but did not benefit much from implementation in a local context; this could be justified by the fact that, in a local context with only few transactions, there are not enough data for machine learning techniques to train optimal models. This study also evaluated only two machine learning techniques. Other studies, such as (Mayer et al. 2018; Čeh et al. 2018), clearly demonstrated the relevance of machine learning techniques compared to those of some other methods in a spatial context. (Mayer et al. 2018) compared three variants of hedonic linear regressions with three machine learning techniques (random forest, gradient boost and artificial neural networks). Their results showed that machine learning techniques (gradient boost in particular) are more accurate than linear models in terms of prediction accuracy, even if linear models (robust regression, in particular) are less volatile. (Čeh et al. 2018) studied the predictive performance of the random forest machine learning technique in comparison with commonly used hedonic models based on multiple regressions for the prediction of apartment prices. Their outputs revealed that the random forest method obtained significantly better prediction results than those of the hedonic models.

We can clearly observe that all these studies that considered spatial heterogeneity or spatial dependence mostly integrated spatial features in traditional hedonic linear models (the first three groups), and only a few of them also evaluated machine learning techniques (the last group). When machine learning techniques are also evaluated, they universally tend to provide better results in terms of predictive power than hedonic models. This is the reason why we specifically focus on these techniques in this paper. However, this study differs from the literature in many aspects, as follows: (1) in a spatial/location context with geocoded location attributes, we evaluate a wider range of machine learning techniques that have already shown their relevance for automated valuation models in different contexts; (2) this evaluation is performed with the same dataset for each model, thus avoiding the bias that could appear when comparing different methods evaluated on different datasets; (3) we analyze the French real estate market, which has received little attention thus far; and (4) we compare the results at high and low location granularity levels by comparing, for instance, the models' predictive powers on nine different and heterogeneous metropolitan areas in France.

3 Data and methods

3.1 Data

The raw dataset for this study is an open source dataset provided by the French government since April 2019 with an open license. This dataset, titled "Demands of land values", is published and produced by the French general directorate of public finances.[2] It provides data on real estate transactions completed during the last five years in metropolitan territories and the DOM-TOM (French overseas departments and territories), except the Alsace-Moselle and Mayotte departments. The data are from notarial acts and cadastral information. The data files are updated every six months in April and October. Each update removes and then replaces all previously published files. Datafiles (under the.csv extension) are provided on a yearly basis and are approximately 4 GB in size. In this paper, we study real estate transactions from the following 5 years: 2015, 2016, 2017, 2018 and the first three quarters of 2019. These transactions represent approximately 18 GB of data and contain almost all the real estate transactions for all French cities. However, given that the most important portion of the transactions takes place in the largest cities, we choose to restrict the study to the 10 largest French cities in terms of population, which are as follows: Paris, Marseille, Lyon, Toulouse, Nice, Nantes, Montpellier, Strasbourg, Bordeaux and Lille (Fig. 1). Due to political, economic and geographic factors, the real estate markets are very different in each of these cities. For example, the price per square meter is much higher in Paris (the French economic and political capital) than in other cities (regional cities). Our goal here is to go beyond global real estate estimation based on prices per square meter and provide precise and automatic estimations of real estate in each of these cities with the use of machine learning methods. As the city of Strasbourg is in the *Alsace-Moselle* department, transactions for this city are not provided in the dataset; therefore, our study focuses on the 9 other largest French cities.

3.1.1 Variables

For each transaction in the dataset, 43 variables are available. However, a significant number of these variables refer to technical data about notarial acts and are not relevant for our study. The variables that could be related to real estate price estimation are listed in the following in Tables 1 and 2.

The descriptive statistics of these variables for each city are provided in the following table.

[2] The link to the dataset "Demands of land values" is:https://www.data.gouv.fr/fr/datasets/5c4ae55a634f4117716d5656/.

Table 1 List of variables used

Variable	Possible Values	Comment
Date of mutation	The date of the transaction (day, month and year)	Date of signature by the notary
Nature of mutation	Sale, sale before completion, land to build, exchange, expropriation, adjudication	
Land Value	Price of the transaction	This price includes taxes, but notarial fees are not included
Address	Street number, repetition index, street type, postal code, city	
Residence type	House, apartment, industrial location, outbuilding	
Land Area	Land Area	In square meters
Living Area	Living space area	In square meters
Number of rooms	Number of rooms in the living space area	
Number of Lots	Number of lots in cases with joint properties	

3.1.2 Repartition

Number of transactions per city, year and quarter Figure 2 shows how the transactions are distributed per city, year and quarter. Figure 2A shows that the distribution of the numbers of transactions per city is generally consistent with the distribution of the populations in these cities (Fig. 1). However, we can notice that Toulouse and Bordeaux recorded many more transactions relative to their population, and this can be perceived as a good indicator for real estate development in these two cities. Figure 2B shows that there was a growth in the number of real estate transactions from 2015 to 2017, but this trend seems to have reversed since 2018. Figure 2C and D clearly show that more overall transactions are made in the last quarter of the year than in each of the other quarters.

Figure 3 shows how the transactions are distributed per year for each city. As in the previous figure, the trends are almost the same for all cities. Only the city of Lille registers continued and noninterrupted growth from 2015 to 2018. We cannot draw any conclusions for 2019, as the last quarter is not included in the data for this year.

Number of transactions per city, sale type and residence type The distributions of the transactions per sale type (nature of mutation) and residence type are provided in Fig. 4 below. Figure 4A shows that almost all the transactions were of the sale or sale before completion types. The adjudication, exchanges, land to build and expropriation types are marginals. This same behavior is also observed even when examining the repartition per city (Fig. 4C and D). Figure 4B shows that most transactions concerned apartments, followed by outbuildings, industrial locations and houses, which are also significant. Because we are only interested in real estate for residential properties, we only use the transactions for houses and apartments as residence types in our analysis. To remove any side effects due to the skewness of the distribution per sale type, we also only keep the transactions of the sales and sales before completion types in our analysis.

Price distribution per city Since our target variable is the price of each piece of real estate, Fig. 5 below shows the price distributions per city. Figure 5A clearly shows that Paris is by far the most expensive city for real estate in France. Overall, the price distributions per city are relatively consistent with respect to the distributions of their populations (Fig. 1). However, we observe a gap with regard to Bordeaux and Nice, which appear to

Table 2 Variable descriptive statistics per city

Variable	Statistic	PARIS	MARSEILLE	LYON	TOULOUSE	NICE	NANTES	MONTPELLIER	BORDEAUX	LILLE
Price (euros)	min	0	0	0	0	0	1	1	1	0
	mean	2,820,867	985,029	796,134	662,643	350,611	642,925	663,989	607,551	308,408
	median	398,480v	168,000	227,100	170,720	195,000	175,000	160,000	252,475	175,000
	max	1,249,132,030	378,000,000	77,600,000	33,427,218	23,680,000	55,244,924	174,814,352	18,429,000	20,000,000
	std	38,170,024	4,988,817	3,203,984	2,755,043	666,302	3,475,487	4,938,465	1,840,670	586,130
	nbNan	0	0	0	0	0	0	0	0	0
Living,Area,(m^2)	min	1	1	5	1	4	1	1	2	1
	mean	54	63	66	61	59	65	58	67	58
	median	42	60	63	57	54	60	54	58	50
	max	1500	780	874	720	735	769	844	540	888
	std	44	33	36	33	34	38	34	44	39
	nbNan	7	12	2	6	0	1	1	2	3
Land Area (m^2)	min	8	1	15	1	15	12	5	6	7
	mean	931	1507	2169	1025	774	965	1598	435	311
	median	396	333	421	351	500	303	324	172	114
	max	19,427	65,680	24,687	51,859	22,974	24,831	29,029	17,314	94,941
	std	1460	5202	4094	3398	1081	2320	5462	1484	3144
	nbNan	156,375	54,653	33,008	42,095	34,523	24,804	21,926	20,296	15,162
Number of Rooms	min	0	0	0	0	0	0	0	0	0
	mean	2	3	3	3	2	3	3	3	3
	median	2	3	3	3	2	3	2	3	2
	max	45	60	17	41	31	24	43	30	28
	std	1	1	1	1	1	2	1	1	2
	nbNan	7	12	2	6	0	1	1	2	3
Number of Lots	min	0	0	0	0	0	0	0	0	0

Table 2 (continued)

Variable	Statistic	PARIS	MARSEILLE	LYON	TOULOUSE	NICE	NANTES	MONTPELLIER	BORDEAUX	LILLE
	mean	2	1	1	1	1	1	1	1	1
	median	2	1	1	1	1	1	1	1	1
	max	27	16	16	13	10	18	10	9	19
	std	1	1	1	1	1	1	1	1	1
	nbNan	0	0	0	0	0	0	0	0	0
Postal Code	nbUnique	20	16	9	6	4	4	4	5	5
	nbNan	1	2	0	2	1	1	0	0	9
Year	nbUnique	5	5	5	5	5	5	5	5	5
	nbNan	0	0	0	0	0	0	0	0	0
Quarter	NbUnique	4	4	4	4	4	4	4	4	4
	nbNan	0	0	0	0	0	0	0	0	0
Residence Type: APARTMENT	count	167,763	60,378	36,414	45,773	37,776	26,156	22,985	25,027	20,243
Residence Type: **HOUSE**	count	829	7852	921	6210	1773	6026	2384	6173	5372
Sale Type: **SALE**	count	167,352	63,114	34,454	44,707	36,331	29,638	21,900	29,667	24,174
Sale Type: BEFORE COMPLETION	count	1240	5116	2881	7276	1218	2544	3469	1533	1441

City	Population
Paris	2 206 488
Marseille	861 635
Lyon	513 275
Toulouse	471 941
Nice	342 522
Nantes	303 382
Montpellier	277 639
Strasbourg	277 270
Bordeaux	249 712
Lille	232 741

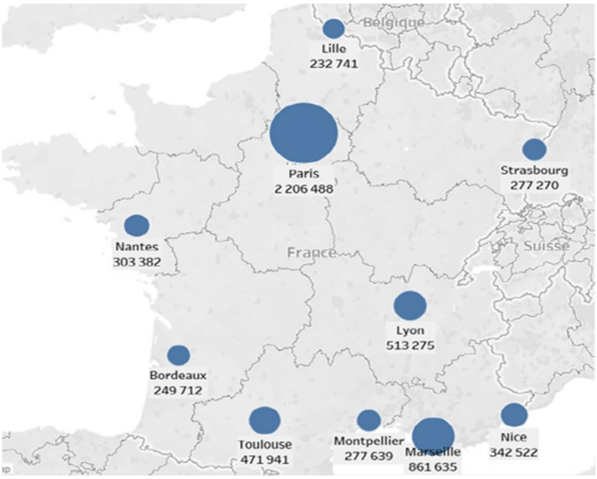


Fig. 1 Studied cities (except Strasbourg)

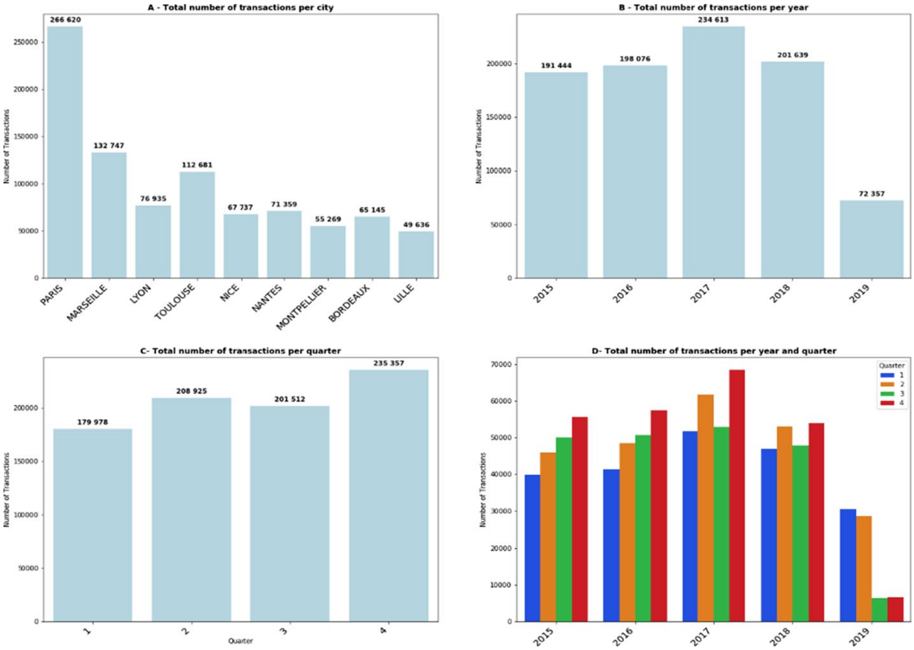


Fig. 2 Transactions per city, year and quarter

be particularly expensive compared to their sizes in terms of population. Conversely, Marseille is less expensive relative to its size in terms of population. We also observe that there are many outliers for all cities that have very high prices; these certainly represent luxury real estate. To avoid side effects, we remove these outliers in our analysis to keep only the most common real estate transactions, which represent the majority of the population. Figure 5B shows the price distributions per residence type (houses and apartments). The

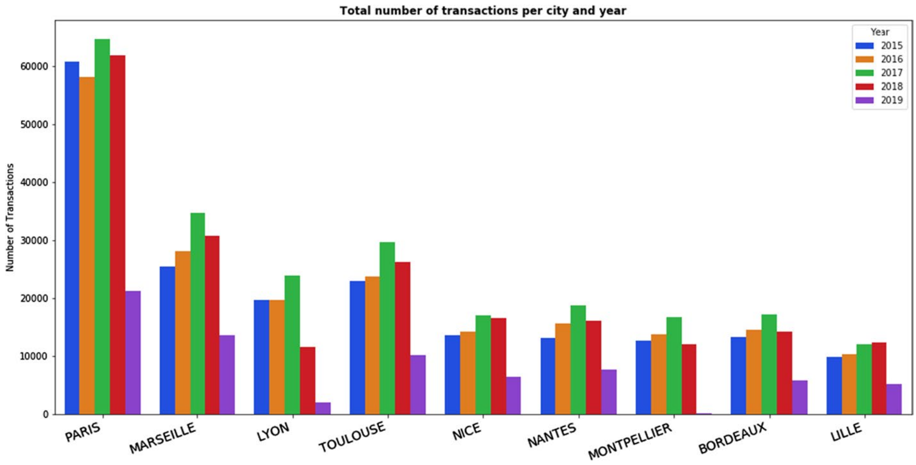


Fig. 3 Transactions per city and year

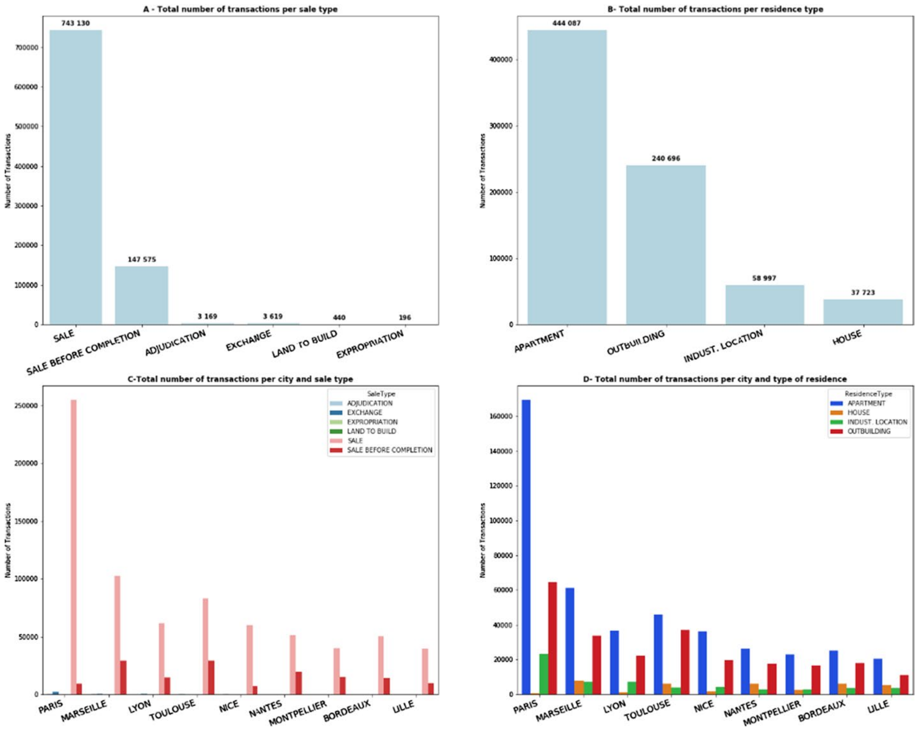


Fig. 4 Transactions per sale type, residence type and city

price distribution trends per city remain the same for apartments and houses. However, houses are obviously more expensive than apartments, except in Lille. The price difference between houses and apartments is also much more pronounced in Paris than in other cities.

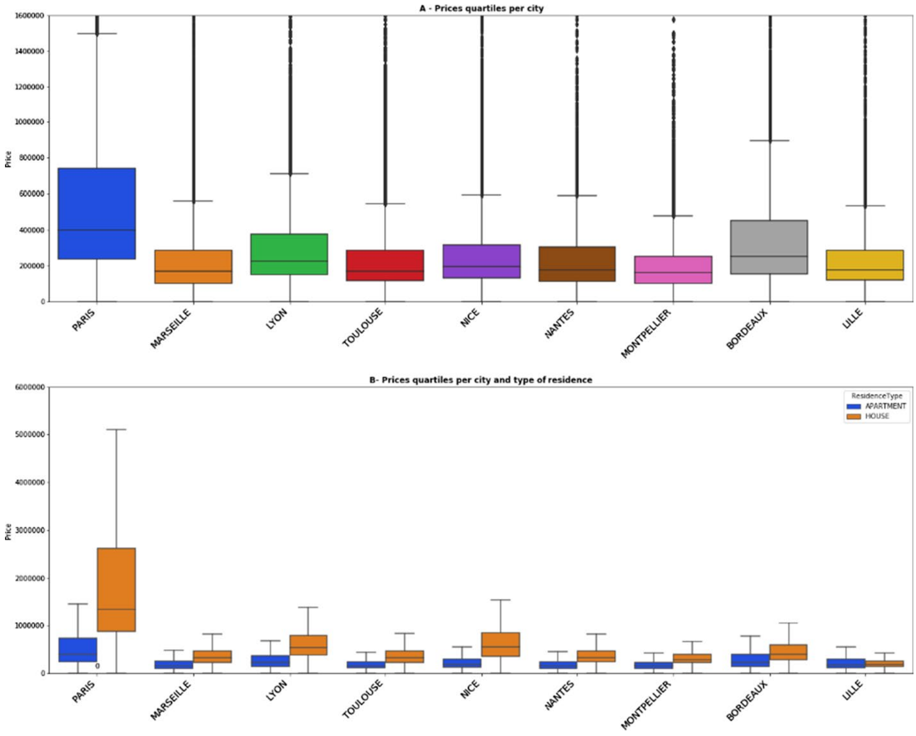


Fig. 5 Price distribution per city and residence type

3.2 Methods

Consisting of a set of well-established methods, machine learning provides algorithms for computers to discover knowledge and make decisions by first learning from given data. Machine learning techniques are becoming increasingly popular, even within the field of production, operations management or manufacturing (Choi et al. 2018; Shin and Park 2000). In those domains, machine learning algorithms are routinely used to search for new patterns in data or to generate predictive models. Subsequently, such patterns are used to improve future operational decisions (Cohen 2018; Chen et al. 2020; Kusiak 2020). This success can be explained by different factors: the improvement of computational processing that makes it cheaper and more powerful than before; affordable data storage solutions; the availability of massive and diverse sources of information; an ever-increasing demand for data-driven decision making; and a need for automatization of the decision processes. Machine learning algorithms have good reputations in terms of predictive power (Wu 1997). Using very few assumptions regarding the input and output variables and applying complex mathematical calculations, they automatically produce models that are not only able to analyze large and complex datasets but also to produce fast and accurate results (Akyildirim et al. 2020). Machine learning methods are also increasingly used for automated valuation models. When comparing machine learning methods for automated valuation models, most existing studies show that several different methods can perform well depending on each context or dataset used (Valier 2020). Most of these methods include artificial neural networks (McCluskey et al. 2013; Yacim and Boshoff 2018; Abidoye et al. 2019); ensemble learning methods, such as random forest, gradient boosting and adaptive boosting (e.g., McCluskey et al. 2014; Čeh et al. 2018; Mullainathan and Spiess 2017; Kok et al. 2017; Mayer et al. 2018; Baldominos et al. 2018); k-nearest neighbors (e.g., Isakson 1988; Borde et al. 2017); and support vector regression (e.gLam et al. 2009; Kontrimas and Verikas 2011; Huang 2019). Thus, for our specific study, we compare all these methods but in the same context and with the same dataset. We also use the linear regression model, which can serve as the baseline model. In the next subsections, we present an overview of these selected techniques.

3.2.1 Artificial neural networks

Inspired by biological neural networks, artificial neural networks mimic the human neural network and are composed of artificial neurons that are also called nodes. The neurons are connected to each other through edges. The latter are responsible for the transmission of signals from one node to another. A signal that propagates through the network can be associated with a real number, and each node is associated with a threshold, above which the signal is assumed to be significant. Additionally, a weight is assigned to each edge that measures the importance of the considered connection. The node values and edge weights are combined to define the strength of the signal. The intuition behind neural networks is that many neurons can be joined together to carry out complex computations. The structure of a neural network can be described as a graph whose nodes are neurons and whose edges are links between the output of some neuron to the input of another neuron (Shalev-Shwartz and Ben-David 2014; Anthony and Bartlett 2009). The network is organized through the following three different types of layers: the input layer, which receives the external data; the hidden layer, which is also called the black box; and the output layer, which produces the result. To be more precise, each node receives signals from other nodes (approximated by numbers); to compute the output of a specific node, the incoming signals are combined with the weights of all the input's edges and the node bias is adjusted using a transfer function. This process is applied to all nodes until the final estimated output is obtained. The final output is compared to the true value, and an observed error is computed. Then, the edge weights and node biases are adjusted through the network, and the output values are recomputed until a minimal error is obtained. Since an artificial neural network is a mathematical model with approximation functions, it has the advantage of being able to work with any data that can be made numeric. Artificial neural networks perform well with nonlinear data and large numbers of inputs. This type of model can be trained with any numbers of inputs and layers, and the predictions are fast. It is among the most powerful modeling devices in machine learning and is currently the preferred approach for addressing complex machine learning problems. Its flexibility draws from its ability to entwine many telescoping layers of nonlinear predictor interactions. However, this method is often said to be a black box with a computationally expensive and time-consuming training step. Additionally, despite its effective learning capability, a major drawback of an artificial neural network is the unreadability of the learned knowledge, i.e., the lack of an explanatory capability (Shigaki and Narazaki 1999).

3.2.2 Random forest

The random forest algorithm is based on decision trees and can be applied for classification or regression exercises (Breiman 2001; Shalev-Shwartz and Ben-David 2014). Let us assume that we want to use a training set $S = \{(x_1, y_1), \dots, (x_N, y_N)\}$ to construct a predictor for the output variable y using the inputs in x. The first step of the random forest algorithm involves selecting, with replacement, a random sample $S_1 = \{(x_{11}, y_{11}), \dots, (x_{n1}, y_{n1})\}$ of n observations from S. As a second step, from the sample S_1, we construct a decision tree T_1 with one additional randomness attribute, as follows: during the construction of each node, from the set of P attributes (or inputs), only p attributes are randomly selected and used to split the node based on the information gain or the variance reduction (in the case of regression trees). At the end of the process, we obtain a decision tree. The process is then repeated m times, leading to m decision trees $T_1, \dots, T_m$. Given an unseen observation of inputs x, the prediction of the output y is obtained by averaging the predictions from all individual regression trees $T_1, \dots, T_m$. The random forest has the advantage of reducing the overfitting problem and the variance in the decision trees. Thus, there is an improvement in the accuracy of the algorithm. Unlike curve-based algorithms, the advantages of random forest are that it is invariant to monotonic transformations of the predictors; it naturally accommodates categorical and numerical data in the same model; it can approximate severe nonlinearities; and a tree of depth L can capture (L – 1)-way interactions (Gu et al. 2020). The flexibility of random forests is also their limitation; this method is less interpretable than an individual decision tree, has a high computational cost and uses a great deal of memory. Consequently, its predictions can be slow.

3.2.3 Adaptive bBoosting and gradient boosting

Adaptive boosting (henceforth, adaboost) is a learning technique that aims to increase the efficiency of a given learning system (Freund and Schapire 1995). The theory behind boosting suggests that many weak learners may, as an ensemble, comprise a single strong learner with greater stability than that of a single complex tree. A decision tree is most often considered as the base estimator. It uses the notion of recursive partitioning: at each step, by searching for the best split across all predictors and all their values, the sample is partitioned into subsamples to create the most homogeneous subsamples in terms of the outcome. To generate the full-grown tree, the concept of node impurity is used (Shmueli and Yahav 2018). In adaboost, the decision tree is trained in several successive stages on random samples formed by assigning significant weights to individuals who are difficult to classify. At each step, a classifier is produced. The final classifier is a linear combination of step classifiers weighted by coefficients related to their performances. Additionally, adaboost can be interpreted as an optimization algorithm on an exponential cost function. Gradient boosting is a generalized boosting technique since it allows for optimization with other differentiable loss functions. During a prediction exercise, once the models have been trained, adaboost and gradient boosting can achieve very good accuracy levels with modest memory and runtime requirements. They are designed to deal with complex and high-dimensional data (Cui et al. 2018). Nevertheless, these methods suffer from difficulties in terms of their interpretability. Additionally, they perform poorly when the feature space has thousands of features with sparse values.

3.2.4 K-nearest neighbors

Based on local approximation, the k-nearest neighbors algorithm (henceforth, KNN) is a nonparametric machine learning algorithm that can be used for classification and regression (Cover and Hart 1967; Devroye et al. 1996). The intuition behind this technique is the following: let $S = \left\{ \left(x_1, y_1\right), \ldots, \left(x_N, y_N\right) \right\}$ be a sample of N observations, where x_i is the set of attributes for individual i and y_i is the outcome variable. Let us consider a new individual with coordinates (x, y), whose attributes are known and stored in a vector x. We are interested in predicting the value of the outcome variable y. From the set of points $\left(x_1, \ldots, x_N, x\right)$, using a distance metric, this algorithm observes the k nearest neighbors of x. Let us call these neighbors $\left(x_{(1)}, \ldots, x_{(K)}\right)$. Depending on the nature of the output variable (categorical or numeric), y is approximated either by the mode or the average of $\left(y_{(1)}, \ldots, y_{(K)}\right)$. In the regression case, the use of a weighted average can provide optimal results. The weight allocated to the output $y_{(k)}$ can be the inverse of the distance between $x_{(k)}$ and x. This procedure is described under the assumption that the number of neighbors k to consider is known. However, this is often not the case. Nevertheless, this number can be approximated using the root mean square error (RMSE); the optimal value for k is the one that minimizes the RMSE. Since it does not derive any discriminative function from the training data, the KNN has the advantage of being much faster than other algorithms that require training. Because of the absence of a training step, new data can be added seamlessly without impacting the accuracy of the algorithm. This method is very easy to implement since only two parameters are required for its implementation, i.e., the value of k and the distance function. However, the KNN performs poorly in high-dimensional setups (with a large number of individuals or an important number of variables or dimensions). In that case, the performance of the algorithm can be degraded by the cost of computing the distance between a new point and the massive number of existing points.

3.2.5 Linear regression

A linear regression model is used when we want to explain a dependent variable y, which is also called the output or target, or outcome variable, by a set of n-dimensional attributes stored in the variables $\left(x_1, \ldots, x_p\right)$, which are also called explanatory, input or independent variables (Stigler 1981). The following equation summarizes the link between the outcome and input variables:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

where ε is an error term, and we assume that ε follows a standard normal distribution, as follows:

$$\varepsilon \sim N(0;1)$$

The coefficients of this model are estimated using the minimization of the sum of the squared errors and are given by the following formula:

$$\hat{\beta} = (X'X)^{-1} X' y$$

where $\hat{\beta}$=$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)'$ and $X = (1, x_1, \ldots, x_p)$.

Given a set of input attributes $\left(x_{(1)}, \dots, x_{(p)}\right)$, the predicted output $\hat{y}$ is given by the following:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_{(1)} + \cdots + \hat{\beta}_p x_{(p)}$$

Linear regression models have the advantage of being easy to implement and interpret, and they are also efficient to train. They tend to demand low computation costs. Hence, they are often used in large-scale prediction tasks (Cui et al. 2018). Their major limitation is the linearity assumption between the outcome variable and the explanatory variables. In real applications, the data are rarely linearly separable. This method is very sensitive to outliers.

3.2.6 Support vector machine (SVM)

The SVM is a linear supervised classifier. Using a hyperplane to separate the data, it is trained on in-sample items to learn to classify out-of-sample items solely based on the values they show for their features (Lolli et al. 2019). To find the frontier between the categories to be separated, an SVM uses a training sample made of points whose categories are known. The frontier is obtained by searching for the hyperplane that separates the training sample while maximizing the distance between the training points and this hyperplane (this is called maximizing the margin). The training points closest to the border are called support vectors. However, the training points may not be linearly separable, in which case there is no hyperplane capable of separating the data. In this situation, we search for a transformation of the initial data that allows separation. In general, the training values are projected into a large dimensional space, where it becomes possible to find a linear separator (Shalev-Shwartz and Ben-David 2014; Cortes and Vapnik 1995; Boser et al. 1992). When the output variable being predicted is continuous-valued, the classification concept of the SVM can be generalized to the regression case. This is called support vector regression (SVR). The goal of SVR is to find a function that presents a margin of tolerance ε from the target values while being as flat as possible, that is, to find the narrowest tube centered around the surface while minimizing the distance between the predicted and true outputs. Mathematically, the problem resolved by SVR during the training process is as follows:

$$\begin{cases} Min\frac{1}{2}\|w\|^2 \\ s.t\left|y_i - \langle w, x_i \rangle - b\right| \leq \varepsilon, \forall i \end{cases}$$

where y_i, x_i, for $i = 1, \dots, n$, are the output and input variables from the training set, respectively, $\langle w, x_i \rangle + b$ is the predicted value to be compared to the target value y_i, and ε is a threshold such that all predictions must be within a range ε of the true values. In a case with a nonlinear SVM, the scalar product $\langle w, x_i \rangle$ is replaced by a kernel function $K(w, x_i)$. Because of the kernel function, the SVM method is highly flexible. Assumptions about the functional form of the transformation are avoided, and there is good out-of-sample generalization when the kernel tuning parameters are appropriately chosen. Like other nonparametric techniques, the SVM method suffers from a lack of transparency in its results. Graphical visualizations can be used to facilitate the interpretation of the results.

3.3 Performance evaluation

To assess the predictive performances of machine learning estimators for real estate price forecasting in major French cities, some evaluation metrics are needed. As is common in the literature (Botchkarev 2019), we rely on the following measures:

- Q1: defines the first quartile of the prediction error distribution (the error values larger than 25% of all the prediction errors).
- MedAE: represents the median error (the error values larger than 50% of all the prediction errors).
- Q3: defines the third quartile of the prediction error distribution (the error values larger than 75% of all the prediction errors).
- MAE measures the mean absolute error; for a set of $\boldsymbol{n}$ error terms $\left\{\boldsymbol{e}_i, \boldsymbol{i} = 1, \ldots, \boldsymbol{n}\right\}$, the MAE is defined by the following:

$$\boldsymbol{MAE} = \frac{\sum_{i=1}^{n} \left|\boldsymbol{e}_i\right|}{\boldsymbol{n}}$$

- RMSE: quantifies the root mean square error; for a set of $\boldsymbol{n}$ error terms $\left\{\boldsymbol{e}_i, \boldsymbol{i} = 1, \ldots, \boldsymbol{n}\right\}$, the RMSE is defined by the following:

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left|\boldsymbol{e}_i\right|^2}{\boldsymbol{n}}}$$

- MSLE: defines the mean squared logarithmic error; for a set of $\boldsymbol{n}$ prices $\left\{\boldsymbol{y}_i, \boldsymbol{i} = 1, \ldots, \boldsymbol{n}\right\}$ and a set of $\boldsymbol{n}$ predicted price values $\left\{\widehat{\boldsymbol{y}}_i, \boldsymbol{i} = 1, \ldots, \boldsymbol{n}\right\}$, the MSLE is defined by the following:

$$\boldsymbol{MSLE} = \frac{1}{\boldsymbol{n}} \sum_{i=1}^{n} \left(\mathbf{log}\left(\boldsymbol{y}_i + 1\right) - \mathbf{log}\left(\widehat{\boldsymbol{y}}_i + 1\right)\right)^2$$

- R2: computed for the regression model; it represents the proportion of the variance of the dependent variable (output) that is explained by the independent variables (inputs).

For each evaluation metric, we are first interested in its values for the best performing city (considered as nonreference) and the worst performing city (considered as reference); second, we are interested in its values regarding the real estate price prediction information with geocoding (considered as nonreference) and without geocoding (considered as reference). For each case, these values are used to compute an improvement ratio, which is defined as follows:

$$\mathit{Improvement\ ratio} = \frac{\mathit{Metric\ value\ for\ the\ reference} - \mathit{metric\ value\ for\ the\ nonreference}}{\mathit{Metric\ value\ for\ the\ reference}}$$

4 Experiments

Our overall experimental process is described in the figure below.

The main steps are data preparation (with and without geocoding), model training with machine learning techniques and cross validation, and finally, selection of the best model, which will be used for the evaluations and interpretations. All these steps are described in the next sections.

4.1 Data preparation

It is now well known that the most important step in machine learning or predictive modeling is the data preparation step. In practice, it has been generally found that data cleaning and preparation account for approximately 80% of the total data engineering effort (Zhang et al. 2003). Data preparation comprises those techniques concerned with analyzing raw data to yield high-quality data and mainly includes the following processes: data collection, data integration, data transformation, data cleaning, data reduction, and data discretization. Data preparation is a fundamental step for many reasons. First, although real-world data are impure, high-performance mining systems require high-quality data, and accurate data yield high-quality patterns. Second, real-world data may be incomplete (e.g., missing attribute values, missing certain attributes of interest, or only aggregate data are available), noisy (e.g., containing errors or outliers), and inconsistent (containing discrepancies in codes or names), and these types of data can disguise useful patterns.

Data preparation involves generating a dataset smaller than the original dataset that can significantly improve the efficiency of data mining and includes the following tasks:

- Selecting relevant data: selecting attributes (filtering and wrapper methods), removing anomalies, or eliminating duplicate records.
- Reducing data: sampling or instance selection.

Data preparation generates high-quality data, which lead to high-quality patterns. For example, we can:

- Recover incomplete data: fill in the values missed or reducing ambiguity.
- Purify the data: correct errors or remove outliers (unusual or exceptional values).
- Resolve data conflicts: use domain knowledge or expert decisions to settle discrepancies.
- Add additional valuable data by data linkage.

In our case, we use almost all of these data preparation techniques for each experiment (without geocoding and with geocoding).

4.1.1 Data Preparation Without Geocoding

In the experiments without geocoding, the data preparation step is summarized by the figure below.

The successive steps are as follows: attribute selection, inconsistency removal, outlier removal, filling in missing values, standardization and one-hot encoding.

The attribute selection step consists of selecting only data from the 9 cities in all the raw datasets. As stated in the Data section, the raw dataset contains 43 variables for each transaction. In this step, we also select only the valuable variables (the 10 variables shown in the figure) that are naturally related to the price of each transaction. Because we are only interested in real estate transactions for residential properties, we also only keep the transactions with the sales and sales before completion sale types for apartments and residential houses.

In the inconsistency removal step, we particularly remove all transactions with missing or bad values for the following key attributes: postal code (because we have a strong belief regarding the importance of house locations in this study), price (since it is our target dependent variable), living area and number of rooms (since they are naturally strong predictors for the price).

In the outlier removal step, for each city, we remove all transactions with outliers in their prices (Fig. 5A). To avoid side effects, we remove outliers in our analysis to keep only the most common real estate transactions that represent the majority of the population. The outlier price values are identified with a common method, which consists of using the interquartile range, i.e., all values above the third quartile Q3 plus one half the interquartile range.

The step of filling in missing values consists of replacing the missing values of the land area variable with zero. This is because this variable is usually missing for apartment transactions.

Because many algorithms (e.g., neural networks, support vector regressors) are perform better and more efficiently with standardized variables than with nonstandardized variables, we perform a transformation for all the continuous variables, all of which are almost normally distributed (land area, living area, number of rooms, number of lots). Standardization typically means rescaling the data to have a mean of zero and a standard deviation of 1 (unit variance).

Finally, for all other discrete attributes (postal code, sale type, residence type), we perform the one-hot encoding transformation to convert them into continuous and Boolean dummy variables with 0 or 1 for each of their values. For instance, we have 6 different postal codes in Toulouse, so the postal code variable for this city is replaced by 6 different dummy variables, with each of them taking the value 0 or 1 for each transaction. Many machine learning algorithms (e.g., neural networks, support vector regression or linear regressions) require this transformation for the effective handling of discrete attributes.

At the end of this step, for a city such as Toulouse, we end up with, for instance, 17 independent variables (along with the dependent variable “price”) in the prepared dataset to be used as the input for the machine learning algorithms. (Fig. 7)

4.1.2 Data Preparation with Geocoding

In our framework with geocoding, the data preparation step is detailed below.

This set of steps differs from the previous set by one additional step, i.e., the geocoding of each transaction to obtain the precise latitude and longitude for a piece of real estate. By adding the latitude and longitude of each transaction, we should be able to evaluate the relevance of the spatial/location features for improving the real estate estimations of the models. Since we have variables that address the details of each transaction in the raw data (e.g., street number, repetition index, street type, postal code and city), we should be able, by using a geocoding service to provide the geographical coordinates (latitude and longitude) of each property of a transaction in the data. There are many existing geocoding services worldwide (e.g., Google, ArcGis, HERE), that are available for free, paid or, most often, paid at a daily usage rate (Singh 2017; Di Pietro and Rinnone 2017). However, in

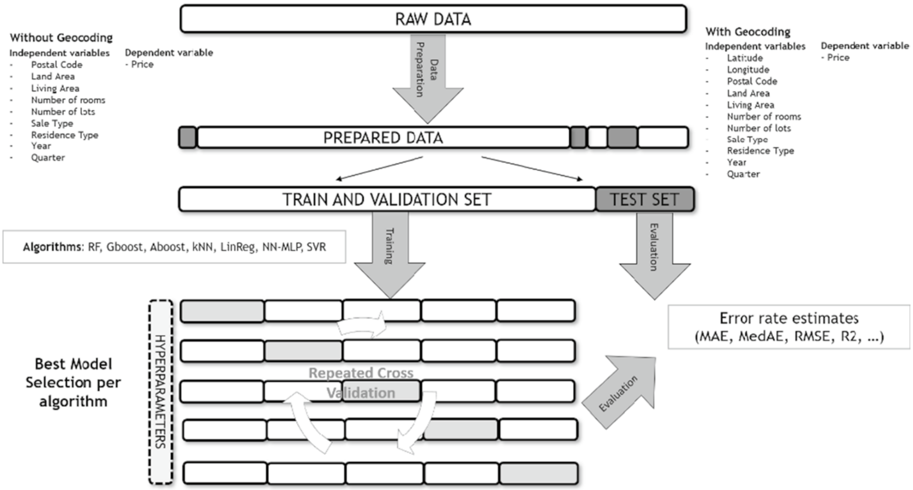


Fig. 6 Overall experimental process

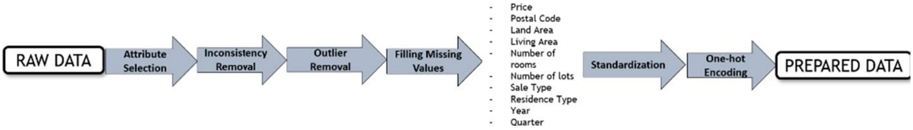


Fig. 7 Main steps of data preparation without geocoding

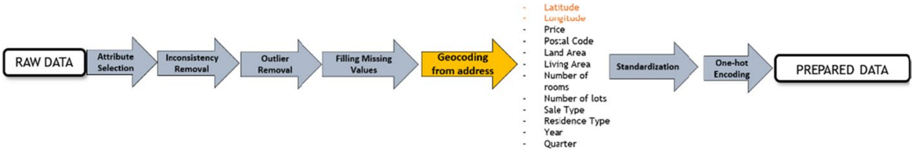


Fig. 8 Main steps of data preparation with geocoding

our case, with only French addresses, we can use the available free geocoding service[3] from the French government that provides many APIs for geocoding in France. We use this service for retrieving the geographical coordinates of each transaction address, as shown in the figure below.

For each address provided, the geocoding service returns additional information, such as the latitude of the address, the longitude of the address, the resulting address variables (house number, street, postal code and city), and a probability score that gives us an idea of the accuracy of the result. Geocoding, in general, is a complex task that can sometimes provide inaccurate, erroneous or no results. When there is no match for the input address, all the result fields are empty. When a match is obtained, the additional information provided

[3] https://geo.api.gouv.fr/adresse.

Table 3 Hyperparameters used for each machine learning algorithm

ML Algorithm	Hyperparameters chosen	Different values used
Neural Networks (MLP)	Network architecture (hidden_layer_sizes)	150, (150,50), (50, 20)
	Activation function (activation)	relu, logistic
	Learning rate (learning_rate_init)	0.001, 0.005, 0.1
	Optimizer (solver)	adam, lbfgs
Random Forest	Max depth of decision tree (max_depth)	8, 32
	Number of decision trees (n_estimators)	1000, 2000, 2500
Adaboost	Type of estimator (base_estimator)	DecisionTreeRegressor
	Decision tree max depth (max_depth)	8, 32
	Number of estimators (n_estimators)	1000, 2000, 2500
	Learning rate (learning_rate)	0.001, 0.05, 0.1
Gradient Boosting	Decision tree max depth (max_depth)	8, 32
	Number of estimators (n_estimators)	1000, 2000, 2500
	Learning rate (learning_rate)	0.001, 0.05, 0.1
	Loss function (loss)	ls, huber
K-Nearest Neighbors	Number of neighbors (n_neighbors)	5, 30, 100
	Neighbors' weight functions (weights)	Uniform, distance
	Neighbors' algorithm (algorithm)	ball_tree, kd_tree, brute, auto
Support Vector Regression	Intercept fitting (fit_intercept)	True, False 1.0, 2.0, 3.0
	Regularization parameter (C)	1000, 2000
	Max number of iterations (max_iter)	Epsilon_insensitive,
	Loss function (loss)	Epsilon_squared_insensitive
Linear Regression	Intercept fitting (fit_intercept)	True, False
	Normalization (normalize)	True, False

(in addition to the latitude and longitude) helps to eliminate potential errors. For example, in our case, we retain only transactions where the geocoding result provides the same street name and postal code as the input, as well as a result score probability greater than 60%. Overall, approximately 90% of the transactions are successfully geolocated by the service, and this can be considered a good ratio. Because we have a very high number of transactions to be geocoded, we use the batch service of the API, and we perform whole-file geocoding for each city. For each successful result, we only retain the latitude and longitude as additional variables to be used in the machine learning algorithms (Fig. 8). For example, for the city of Toulouse, we have 17 independent variables in the prepared dataset without geocoding (after one-hot encoding); thus, we have 19 independent variables in our prepared dataset with geocoding (after one-hot encoding), but we lose approximately 10% of the transaction data due to geocoding errors or non-matches during the geocoding exercise.

4.2 Training

The training process is performed in the same way with and without geocoding, as presented in Fig. 6. For each city, the prepared dataset is first divided: 75% for the training set (i.e., 33 475 transactions for the city of Toulouse) and 25% for the test set (i.e., 11 159 transactions for the city of Toulouse). The training process is performed with fivefold cross

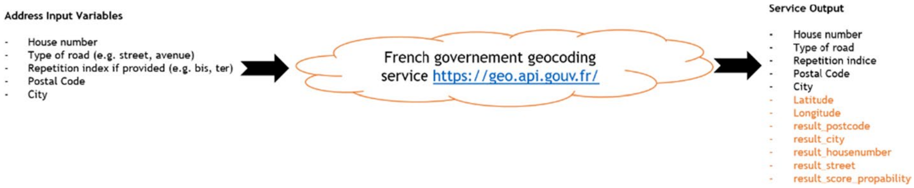


Fig. 9 Preview of the inputs and outputs of the French geocoding service

validation and a set of hyperparameters for each machine learning algorithm, as presented in the following in Table 3.

For each city and for all fivefold cross validation steps, this process gives us 180 different trained neural network models, 30 different trained random forest models, 90 different trained adaboost models, 180 different trained gradient boosting models, 120 different trained k-nearest neighbors models, 180 different trained support vector regression models and 20 different trained linear regression models. This training process provides a total of 800 trained models per city for each experiment, and this corresponds to a total of 1600 models for training both experiments (with and without geocoding) for each city. Overall, we have a total of 14 400 trained models for all 9 cities. For each city and each algorithm, we only select the best model for the experiment without geocoding and the best model for the experiment with geocoding for a comparative analysis.(Fig. 9)

4.3 Evaluation of results

Here, we present an evaluation of results for the experiment without geocoding and for the experiment with geocoding, as well as a comparison between these two results.

4.3.1 Results of the experiment without geocoding

Model performances without geocoding

Figure 10 shows the resulting metrics for each machine learning model used in this experiment. If we examine, for instance, the three best predictors for each metric, we always obtain the random forest, neural network and k-nearest neighbors models as the best predictors, except for gradient boosting in the case of the Q1 metric. However, in general, the neural network technique appears to be the best predictor among all the algorithms used. The hyperparameters of the neural networks used to achieve these results are as follows:

- 2 layers with 150 neurons in the first layer and 50 neurons in the second layer.
- ReLU activation function.
- Adam solver.
- 1000 max iterations.
- Learning rate of 0.1.

For the best predictors in general:

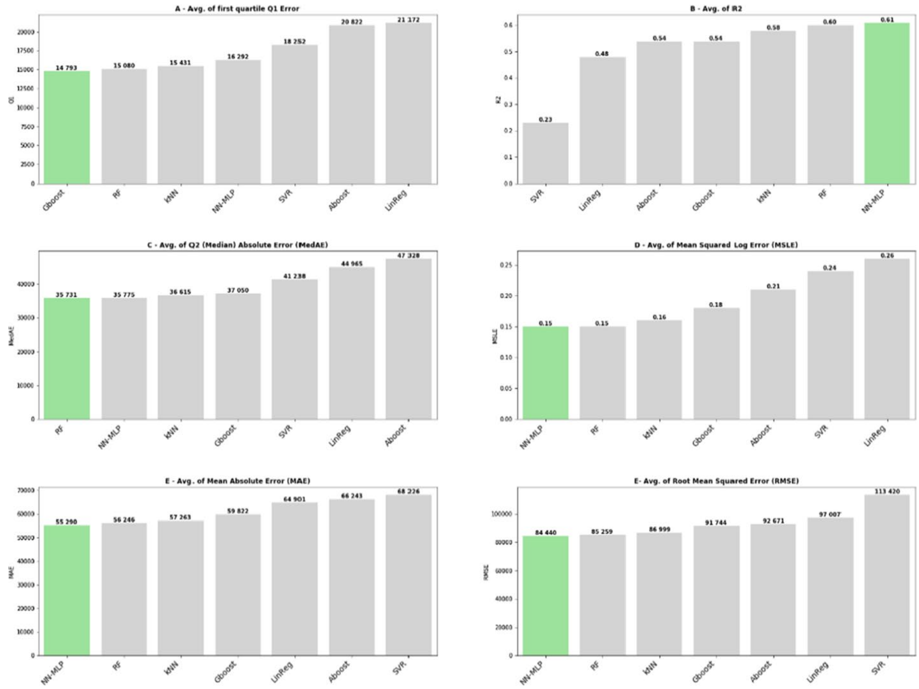


Fig. 10 Model evaluations in terms of metrics for the experiment without geocoding

- The Q1 of absolute errors is approximately 15,000 euros.
- The median error is approximately 35,000 euros.
- The mean absolute error is approximately 55,000 euros.

From another point of view, using the R2 metric leads to 61% of the price variance being explained by the input variables of our models.

Performances of the best models per city without geocoding

If we only consider our best model (the neural network model), the metrics per city are presented in Fig. 11. The performances of the models are different for each city. Except for the R2 metric, real estate price predictions are more accurate for cities with medium costs of living in terms of real estate prices (e.g., Toulouse, Montpellier, and Nantes) and are less accurate for cities with high costs of living(e.g., Paris, Bordeaux, and Nice). The R2 metric shows that even if the price variation is better explained for an expensive city such as Paris than for an inexpensive city, the price forecasting precision remains low.

The metric improvement ratios between the best-performing city and the worst-performing city are mostly over 60%, as shown in the Table 4 below.

If we compare the results of the best-performing city with the average results for all cities, we can notice the following:

- The first quartile of the prediction error distribution is approximately 10,000 euros (compared to the average of 15,000 euros for all cities).

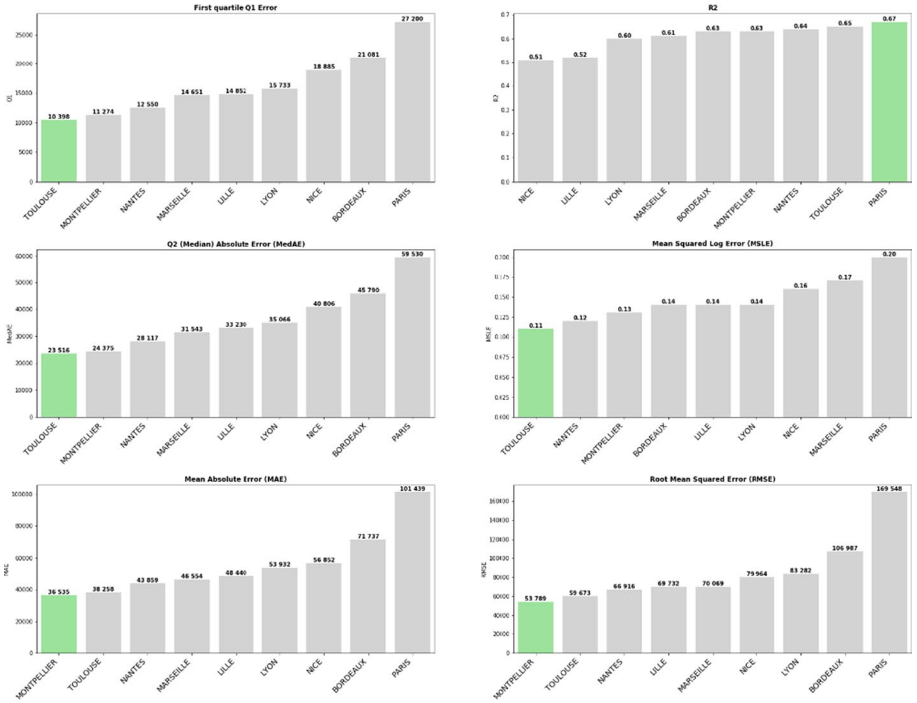


Fig. 11 Metrics of the best model (the neural network model) per city for the experiment without geocoding

- The median prediction error is approximately 23,000 euros (compared to the average of 35,000 euros for all cities).
- The mean absolute error is approximately 36,000 euros (compared to the average of 55,000 euros for all cities).
- The R2 variance is approximately 67% (compared to the 61% average for all cities).

In the next section, we present the results obtained with geocoding.

4.3.2 Results for the experiment with geocoding and improvement

Model performances with geocoding

Figure 12 shows the resulting metrics for each machine learning model used along with the geocoded variables. Relative to the experiment without geocoding, we can observe two major findings, as follows:

- The ensemble learning algorithms (random forest, gradient boosting and adaboost) outperform all other algorithms for all metrics.
- Real estate price predictions with geocoding are far better than predictions without geocoding in terms of all metrics.

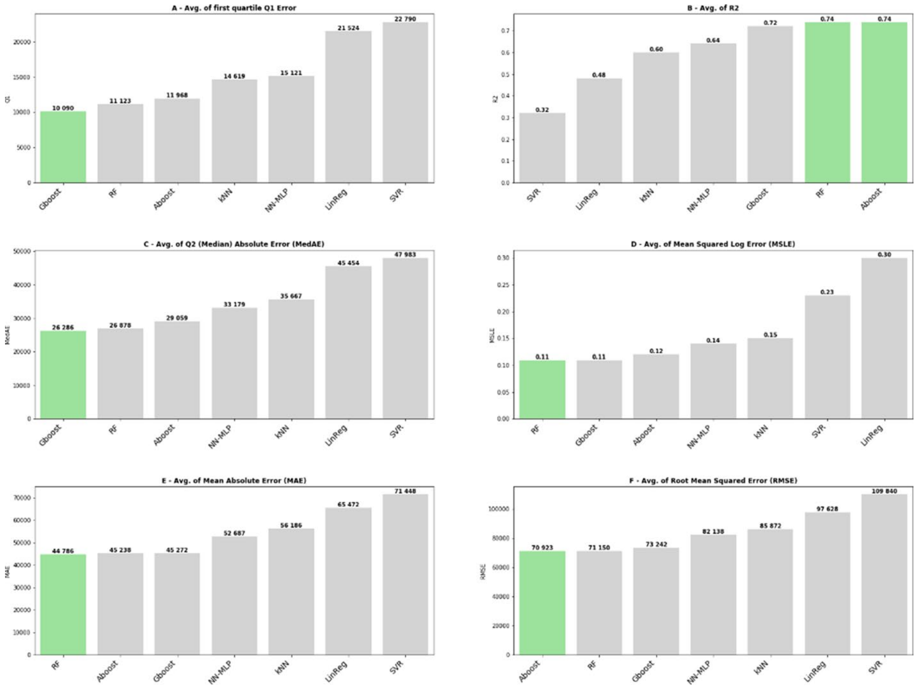


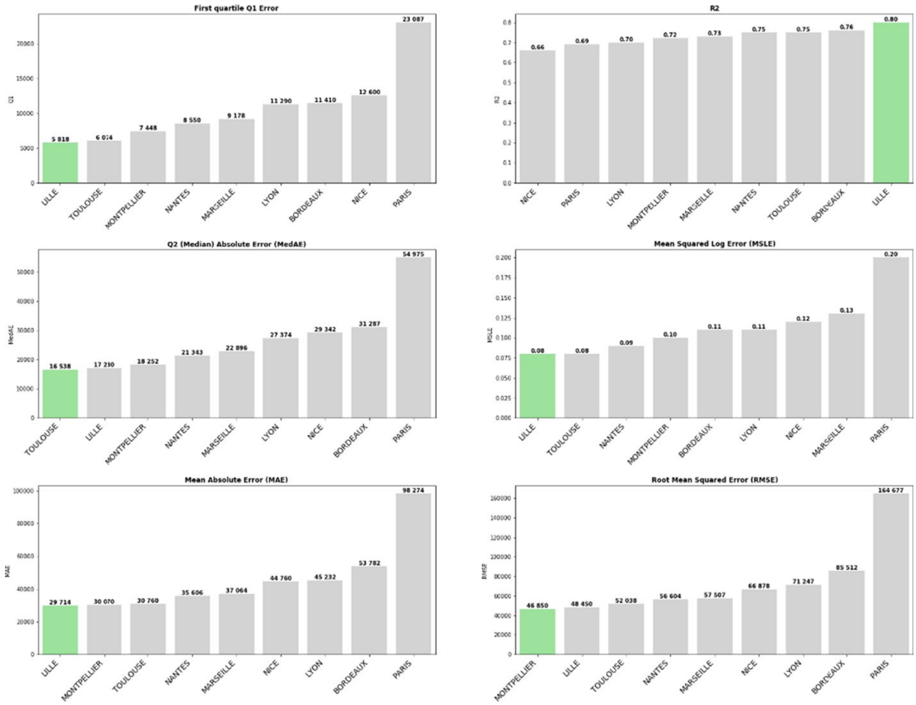
Fig. 12 Model evaluations in terms of metrics for the experiment with geocoding

Table 4 Improvement ratios between the best-performing and worst-performing cities for the experiment without geocoding

Metric	Best-performing city (value)	Worst-performing city (value)	Improvement ratio (%)
Q1	Toulouse (10,398)	Paris (27,200)	61.7
MedAE	Toulouse (23,516)	Paris (59,530)	60.4
MAE	Montpellier (36,535)	Paris (101,439)	63.9
RMSE	Montpellier (53,789)	Paris (169,548)	68.2
MSLE	Toulouse (0.11)	Paris (0.20)	45
R2	Paris (0.67)	Nice (0.51)	31

The following Table 5 presents the hyperparameters used during the evaluation of the ensemble learning algorithms. For all these algorithms, the best max depth parameter for the decision trees was 32, and the optimal number of estimators (decision trees) was 2500. The use of small values for the learning rates leads to better results (0.05 for adaboost and 0.1 for gradient boosting) than the use of large values.

Compared to the experiment without geocoding, when we examine the accuracy of the models, the following Tables 6, 7 and 8 show the average improvements for the different metrics for all cities and for each ensemble learning algorithm. Overall, for all metrics, we observe a mean improvement of 36.11% for adaboost, 31.13% for gradient boosting and 24.66% for random forest, thereby clearly showing the relevance of integrating the


First quartile Q1 Error

R2

Q2 (Median) Absolute Error (MedAE)

Mean Squared Log Error (MSLE)

Mean Absolute Error (MAE)

Root Mean Squared Error (RMSE)

Fig. 13 Metrics of the best model (random forest) per city for the experiment with geocoding

Table 5 Best hyperparameters for the ensemble learning algorithms

Algorithm	Hyperparameter	Best value
Random Forest	Bootstrap	True
	Max depth	32
	Number of estimators	2500
Adaboost	Base estimator	Decision Tree Regressor
	Max depth	32
	Number of estimators	2500
	Learning rate	0.05
Gradient Boosting	Number of estimators	2500
	Max depth	32
	Learning rate	0.1

geocoding step for real estate estimation for all the cities used in this experiment. If we consider, for instance, the adaboost algorithm, we obtain the following observations:

- The first quartile of the forecasting errors is approximately 10,000 euros (compared to the average of 18,000 euros without geocoding).

Table 6 Best adaboost predictor for all cities

Metric	With Geocoding	Without Geocoding	Geocoding Improvement (%)
MedAE	26,441	42,119	37.22
R2	0.74	0.54	37.04
MAE	39,396	57,444	31.42
RMSE	59,829	79,328	24.58
MSLE	0.11	0.2	45.0
Q1	10,814	18,412	41.27
Q3	50,884	79,777	36.22

Table 7 Best gradient boosting predictor for all cities

Metric	With Geocoding	Without Geocoding	Geocoding Improvement (%)
MedAE	22,642	32,887	31.15
R2	0.73	0.54	35.19
MAE	38,447	51,907	25.93
RMSE	61,437	78,471	21.71
MSLE	0.1	0.17	41.18
Q1	8482	12,955	34.53
Q3	49,683	69,236	28.24

Table 8 Best random forest predictor for all cities

Metric	With Geocoding	Without Geocoding	Geocoding Improvement (%)
MedAE	23,423	32,049	26.92
R2	0.74	0.6	23.33
MAE	38,300	49,157	22.09
RMSE	59,835	73,124	18.17
MSLE	0.1	0.14	28.57
Q1	9 610	13 488	28.75
Q3	49,144	65,312	24.76

- The median error is approximately 26,000 euros (compared to the average of 42,000 euros without geocoding).
- The mean absolute error is approximately 39,000 euros (compared to the average of 57,000 euros without geocoding).
- The R2 variance is approximately 0.74 (compared to the average of 0.54 without geocoding).

Performances of the best models per city with geocoding

We consider one of our best ensemble learning models (random forest, for instance); the metrics obtained per city are presented in Fig. 13. The performances of the model are slightly different for each city. The results are more accurate for cities with high costs of living in terms of real estate prices (e.g., Lille, Toulouse, Montpellier, and Nantes) and are less accurate for cities with high costs of living (e.g., Paris, Bordeaux, and Nice).

The metric improvement ratios between the best-performing city and the worst-performing city are mostly over 70%, as shown in the Table 9 below, except for that of the R2 metric (improvement ratio of 21%).

If we compare the results of the best-performing city with the average results for all cities, we can notice the following:

- The first quartile of the forecasting errors is approximately 5000 euros (compared to the 10,000 euros average for all cities).
- The median error is close to 16,000 euros (compared to the 26,000 euros average for all cities).
- The mean absolute error is close to 29,000 euros (compared to the 44,000 euros average for all cities).
- The R2 variance is approximately 80% (compared to the 74% average for all cities).

When examining the model precision for one city, such as Lille, compared to the experiment without geocoding, the following Tables 10, 11 and 12 show the mean improvement for all metrics with the ensemble learning algorithms. Overall, we observe for all metrics a mean improvement of 40.85% for Ada Boost, 39.77% for Gradient Boost and 31.7% for Random Forest, which also clearly shows the relevance of integrating the geocoding step for real estate estimation at the city level. If we consider, for instance, the Ada Boost algorithm for that city, we have the following:

- The first quartile of the forecasting errors (25% of the predictions) is approximately 4000 euros (compared to the average of 9000 euros without geocoding, an improvement of approximately 52.36%).
- The median error is approximately 16,000 euros (compared to the average of 26,000 euros without geocoding, an improvement of approximately 39.22%).
- The mean absolute error is approximately 29,000 euros (compared to the average of 43,000 euros without geocoding, an improvement of approximately 33.14%).
- The R2 variance is approximately 0.79 (compared to the average of 0.54 without geocoding, an improvement of approximately 46.3%).

5 Discussion and implications

5.1 Experimental discussion

With respect to our research question, the aim of this paper is to evaluate what would be lost in terms of predictive power for an automated valuation model that fails to integrate location variables. We designed an experiment that particularly focuses on machine learning models evaluated on a complete dataset containing the 5-year historical real estate transactions in nine major French cities. We used geocoding to add precise geographic location coordinates to the features to be used as inputs for each machine learning model. We built specific models for each city of the experiment with and without adding geographic coordinate features as model inputs to compare the predictive powers of the models in both cases.

Table 9 Improvement ratios between the best-performing and worst-performing cities for the experiment with geocoding

Metric	Best-performing city (value)	Worst-performing city (value)	Improvement ratio (%)
Q1	Lille (5818)	Paris (23,087)	74.8
MedAE	Toulouse (16,538)	Paris (54,975)	70
MAE	Lille (29,714)	Paris (98,274)	69.7
RMSE	Montpellier (46,850)	Paris (164,677)	71.5
MSLE	Lille (0.08)	Paris (0.20)	60
R2	Lille (0.80)	Nice (0.66)	21

Table 10 Best adaboost predictor for Lille

Metric	With Geocoding	Without Geocoding	Geocoding Improvement (%)
MedAE	21,500	33,000	34.85
R2	0.8	0.54	48.15
MAE	31,173	47,453	34.31
RMSE	47,905	68,121	29.68
MSLE	0.08	0.18	55.56
Q1	7000	13,027	46.27
Q3	41,500	66,000	37.12

Table 11 Best gradient boosting predictor for Lille

Metric	With Geocoding	Without Geocoding	Geocoding Improvement (%)
MedAE	16,179	26,620	39.22
R2	0.79	0.54	46.3
MAE	29,203	43,680	33.14
RMSE	48,907	68,375	28.47
MSLE	0.08	0.15	46.67
Q1	4413	9263	52.36
Q3	38,774	57,190	32.2

The results clearly show that adding geographic coordinates to the list of input features leads to a significant increase in precision for the most popular model evaluation metrics (MedAE, Q1, Q3, R2, MAE, RMSE, and MSLE). More precisely, for all cities, the mean precision improvement can reach 36% on average for all metrics and up to 45% on average for some specific metrics with the best predictor models. In terms of the models built for each city, this precision improvement can reach 40% on average for all metrics (e.g., Lille city) and even 52% for specific metrics. At a high level of granularity, we also compare the differences in terms of each model's precision for the nine cities used in the experiment. The results show that each model's precision for almost all the metrics was approximately 60% more precise for cities with medium costs of living (e.g., Toulouse, Lille, and Montpellier) than for cities with high costs of living (e.g., Paris, Bordeaux, and Nice). Moreover,

Table 12 Best random forest predictor for Lille

Metric	With geocoding	Without Geocoding	Geocoding improvement (%)
MedAE	18,282	27,013	32.32
R2	0.8	0.6	33.33
MAE	30,225	41,953	27.96
RMSE	47,992	63,377	24.28
MSLE	0.08	0.13	38.46
Q1	7223	11,195	35.48
Q3	38,554	55,154	30.1

this precision difference reaches 70% when considering models using geographical coordinates as input features. Finally, regarding the machine learning techniques used, our results reveal that neural networks and random forest particularly outperform the other methods when geographical coordinates are not accounted for, while the ensemble learning methods (random forest, adaboost and gradient boosting) perform well when geographical coordinates are considered.

Our results are in line with studies in the literature that shows that including location attributes in automated valuation models results in improved prediction accuracies for techniques such as submarket methods, trend surface and spatial expansion methods, spatial regression methods, and machine learning methods with spatial attributes (Bourassa et al. 2003; Bitter et al. 2007; McCluskey et al. 2013; Čeh et al. 2018; Doumpos et al. 2020). However, from our research question and our experiments, this study additionally provides an estimation of what would be lost in terms of predictive power for a model (specifically a machine learning model) that fails to integrate location attributes. The losses increase up to 52% for the best model predictors for a metropolitan city in our experiment. This metric provides a better perception than other metrics of the high importance of location attributes for automated valuation models. At a high level of granularity, our results also provide a quantification of the relevance of using submarket methods (e.g., Bourassa et al. 2010; Goodman and Thibodeau 2007). In our case, we built different models for each city, and we observed that we can obtain model precision differences of up to 70% between medium-cost cities and high-cost cities. This result can be viewed as a difference in the spatial dependence and spatial heterogeneity between these medium-cost cities and high-cost cities (Anselin 2013; Basu and Thibodeau 1998; Bitter et al. 2007). Finally, regarding the best machine learning methods, many studies in the literature have already demonstrated similar results with ensemble learning algorithms as their best predictors (e.g., McCluskey et al. 2014; Čeh et al. 2018; Mullainathan and Spiess 2017; Kok et al. 2017; Mayer et al. 2018; Baldominos et al. 2018) or with artificial neural networks outperforming the other methods (McCluskey et al. 2013; Yacim and Boshoff 2018; Abidoye et al. 2019). However, some other studies in the literature contrast the results with those of k-nearest neighbors (e.g., Isakson 1988; Borde et al. 2017) or support vector regression as the best predictors (e.g., Lam et al. 2009; Kontrimas and Verikas 2011; Huang 2019). However, all these related experiments are realized in different contexts and with different datasets, and they do not always consider all these algorithms in the same experiment. Our experiment overcomes these biases and could be viewed as a more reliable comparison between all these algorithms considering the use of the same context and the same dataset throughout the experiment.

5.2 Implications

Our studies may have many research and practical implications.

Research implications

To the best of our knowledge, this is the first study focusing on evaluating and quantifying the impact of geographic locations on real estate price estimations. Many existing studies in the literature (described in Sect. 2) have already demonstrated the relevance of location features in real estate price estimations, but none of them provide metrics that precisely quantify the relevance of location features. Our research question in this study is thus quite new and can lead to many other similar empirical studies with machine learning methods, as well as with other automated valuation methods, such as submarket methods, trend surface methods, spatial expansion methods, and spatial regression methods.

In the operations management field, only a few studies are interested in revenue management for durable and non-replenishable products such as real estate (Wen et al. 2016; Padhi et al. 2015). This study could serve as a basis for assessing real estate prices for strategic revenue management under the uncertainty of real estate projects. For instance, this study could help to set the number of each type of property and price for which it is difficult to handle revenue management under uncertain customer demands, customer preferences, and volatile commodity prices (Padhi et al. 2015; Bogataj et al. 2016).

Practical implications

This study could have direct implications in terms of real estate price estimations, particularly for the French market, which has so far received little attention from automated valuation models or in operations management. Our study is based on a reliable data source containing 5 years of historical real estate transactions from notarial acts. We can express the practical implications of this study in two aspects.

First, the trained machine learning models could help everyone obtain a quick estimation of the value of a real estate property from a sale or purchasing perspective, and this can also apply to real estate agencies or investors. As shown in our experiment, adding precise geographic location features considerably improves the price estimations of a given model. For instance, for many cities we have median errors of approximately 15 000 euros and first quartile errors of approximately 5 000 euros, which could be very promising as margin errors for an automated estimator while taking into account that many other important house characteristics are missing in the studied dataset (e.g., the age of the house, presence of a lift, presence of parking spaces, presence of a swimming pool, presence of terraces, presence of a garden, number of floors, community costs, etc.). This makes it possible to envisage highly relevant results with multiple characteristics.

Second, our study makes it easy to understand and compare the real estate markets of major French cities. For instance, we can clearly notice that the real estate prices in medium-cost cities, such as Lille, Toulouse, and Montpellier, can be estimated more precisely than those of more expensive cities, such as Paris, Bordeaux, and Nice. Such comparative information could provide a quality indicator when interpreting automated price estimations from different cities or when choosing only cities where price predictions are sufficiently precise to be exploited. All of this could provide valuable information for individuals, agencies or investors interested in the real estate market.

5.3 Limitations and directions for future research

The approach presented in this paper shows promising results but can be improved experimentally and conceptually in many ways.

Experimentally, the studied dataset does not contain many important house characteristics that are valuable in real estate estimations, such as the age of the asset, details of asset composition (e.g., presence of parking spaces, lifts, gardens, etc.), community costs, etc. Adding such missing characteristics would naturally improve the model accuracy rates. Linking the dataset with external data sources, such as online real estate ads or social media (Bekoulis et al. 2018), could help in extracting and adding some missing characteristics in the experiment. We also choose, in this study, to quantify the relevance of spatial attributes by adding the geographic coordinates of each transaction as a feature variable for training the machine learning models. However, other studies have also successfully included model locations with other variables, such as accessibility variables (e.g., proximity to amenities, such as schools), neighborhood socioeconomic variables (e.g., local unemployment rates), and environmental variables (e.g., road noise or visibility impact) (Čeh et al. 2018; Bourassa et al. 2010; Case et al. 2004). One other experimental improvement could be to quantify the relevance and differences (using machine learning techniques) between these other location-related variables compared to the singular use of geographical coordinates. Additionally, rather than using geographic coordinates directly, one can also use first-group transactions in small geographic tile area features (McNeill and Hale 2017) with many sizes for capturing geographical areas with different and flexible levels of granularity (e.g., low, intermediate or high). This latter approach would consider flexible, geographically-based submarkets (Bourassa et al. 1999) in the preparation steps before the process of model training with machine learning techniques. From another point of view, we mainly focus on the predictive capacities of machine learning techniques in this study because they represent the main advantage of these techniques and can be relevant for providing good estimates to many real estate actors, such as real estate agencies or investors. However, it could also be interesting to go beyond this limitation and practically quantify and compare the levels of volatility of these techniques (Mayer et al. 2018).

Conceptually, we think the approach presented in this paper could be complementary to many existing approaches for automated valuation models, particularly when integrating hedonic modeling and machine learning algorithms (Hu et al. 2019).

6 Conclusion

We presented an experiment on real estate price estimations using seven machine learning techniques with 5 years of historical data of real estate transactions in major French cities. We particularly focused on demonstrating and quantifying the relevance of location features in real estate estimations with high and fine levels of granularity, with one main objective being to provide an idea of what would be lost in terms of predictive power for an automated valuation model that fails to integrate location variables. From a practical point of view, this could also allow for the training of more accurate real estate models that could help in identifying the best opportunities for marketplace players, such as real estate agencies or investors. For instance, at a high level of granularity, we clearly observed that there were very important differences regarding the models' forecasting errors (sometimes with precision differences beyond 70%) between high-cost cities (e.g., Paris, Bordeaux, and Nice) and medium-cost cities (e.g., Toulouse, Lille, and Montpellier). Thus, this fact could imply that it would be more relevant to train specific models for some geographical submarkets (cities in this case) rather than global models including all cities. At a low level of granularity, we made use of geocoding to extract and add precise geographic location features to the machine learning algorithms' inputs. We observed important improvements in the models' forecasting powers (sometimes an improvement greater than 50%) when adding these geographic location features over models trained without these features. These results are promising and could provide data modeling alternatives using machine learning techniques in real estate price estimation procedures. However, our approach could also be complementary to many automated valuation models or revenue management methods and thus offers many perspectives for future research.

References

Abidoye, R. B., Chan, A. P., Abidoye, F. A., & Oshodi, O. S. (2019). Predicting property price index using artificial intelligence techniques. *International Journal of Housing Markets and Analysis, 12,* 1072.

Akyildirim, E., Goncu, A., & Sensoy, A. (2020). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*. https://doi.org/10.1007/s10479-020-03575-y.

Anselin, L. (2013). *Spatial Econometrics: Methods and Models*. Berlin: Springer.

Anthony, M., & Bartlett, P. L. (2009). *Neural Network Learning: Theoretical Foundations*. Cambridge: Cambridge University Press.

Basu, S., & Thibodeau, T. G. (1998). Analysis of spatial autocorrelation in house prices. *The Journal of Real Estate Finance and Economics, 17*(1), 61–85.

Bekoulis, G., Deleu, J., Demeester, T., & Develder, C. (2018). An attentive neural architecture for joint segmentation and parsing and its application to real estate ads. *Expert Systems with Applications, 102,* 100–112.

Berk, E., Gürler, Ü., & Yıldırım, G. (2009). On pricing of perishable assets with menu costs. *International Journal of Production Economics, 121*(2), 678–699.

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

Bidanset, P.E., et al. (2017). "Further evaluating the impact of kernel and bandwidth specifications of geographically weighted regression on the equity and uniformity of mass appraisal models." In *Advances in Automated Valuation Modeling*, Springer, 191–99.

Bitter, C., Mulligan, G. F., & Dall'erba, S. . (2007). Incorporating spatial variation in housing attribute prices: a comparison of geographically weighted regression and the spatial expansion method. *Journal of Geographical Systems, 9*(1), 7–27.

Bogataj, D., McDonnell, D. R., & Bogataj, M. (2016). Management, financing and taxation of housing stock in the shrinking cities of aging societies. *International journal of production economics, 181,* 2–13.

Borde, S., Rane, A., Shende, G., & Shetty, S. (2017). Real estate investment advising using machine learning. *International Research Journal of Engineering and Technology (IRJET), 4*(3), 1821–1825.

Borst, R. A., & McCluskey, W. J. (2008). Using geographically weighted regression to detect housing submarkets: Modeling large-scale spatial variations in value. *Journal of Property Tax Assessment & Administration, 5*(1), 21–54.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Conference on Learning Theory (pp*: 144–152).

Botchkarev, A. (2019). A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge & Management, 14,* 45.

Bourassa, S. C., Cantoni, E., & Hoesli, M. (2007). Spatial dependence, housing submarkets, and house price prediction. *The Journal of Real Estate Finance and Economics, 35*(2), 143–160.

Bourassa, S. C., Hamelink, F., Hoesli, M., & MacGregor, B. D. (1999). Defining housing submarkets. *Journal of Housing Economics, 8*(2), 160–183.

Bourassa, S. C., Hoesli, M., & Vincent, S. P. (2003). Do Housing Submarkets Really Matter? *Journal of Housing Economics, 12*(1), 12–28.

Bourassa, S., Eva, C., & Hoesli, M. (2010). Predicting House Prices with Spatial Dependence: A Comparison of Alternative Methods. *Journal of Real Estate Research, 32*(2), 139–159.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32.

Case, B., John, C., Robin, D., & Rodriguez, M. (2004). Modeling spatial and temporal house price patterns: A comparison of four models. *The Journal of Real Estate Finance and Economics, 29*(2), 167–191.

Čeh, M., Kilibarda, M., Lisec, A., & Bajat, B. (2018). Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. *ISPRS International Journal of Geo-Information, 7*(5), 168.

Chen, B., Bai, R., Li, J., Liu, Y., Xue, N., & Ren, J. (2020). A multiobjective single bus corridor scheduling using machine learning-based predictive models. *International Journal of Production Research*. https://doi.org/10.1080/00207543.2020.1766716.

Choi, T. M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. *Production and Operations Management, 27,* 1868–1883.

Clapp, J. M. (2003). A semiparametric method for valuing residential locations: application to automated valuation. *The Journal of Real Estate Finance and Economics, 27*(3), 303–320.

Cohen, M. C. (2018). Big data and service operations. *Production and Operations Management, 27*(9), 1709–1723.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning, 20*(3), 273–297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, 13*(1), 21–27.

Cui, R., Gallino, S., Moreno, A., & Zhang, D. J. (2018). The operational value of social media information. *Production and Operations Management, 27*(10), 1749–1769.

D'Amato, V., Di Lorenzo, E., Haberman, S. et al. 2019. "Pension Schemes versus Real Estate." *Annals of Operations Research*: 1–13.

d'Amato, M., & Kauko, T. (2017). *Advances in Automated Valuation Modeling*. Berlin: Springer.

Dana, J. D., Jr. (2008). New directions in revenue management research. *Production and Operations Management, 17*(4), 399–401.

Devroye, L., Györfi, L., & Lugosi, G. (1996).*A Probabilistic Theory of Pattern Recognition*, Springer, Berlin

Din, A., Hoesli, M., & Bender, A. (2001). Environmental variables and real estate prices. *Urban Studies, 38*(11), 1989–2000.

Doumpos, M., Papastamos, D., Andritsos, D., & Zopounidis, C. (2020). Developing automated valuation models for estimating property values: a comparison of global and locally weighted approaches. *Annals of Operations Research*. https://doi.org/10.1007/s10479-020-03556-1.

Garcia, J. C. E., & Alfandari, L. (2018). Robust location of new housing developments using a choice model. *Annals of Operations Research, 271*(2), 527–550.

Fik, T. J., Ling, D. C., & Mulligan, G. F. (2003). Modeling spatial variation in housing prices: a variable interaction approach. *Real Estate Economics, 31*(4), 623–646.

Freund, Y., & Schapire, R. E. (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory* (pp 23–37).

Geraghty, M. K., & Johnson, E. (1997). Revenue management saves national car rental. *Interfaces, 27*(1), 107–127.

Gomes, L. F. A. M. (2009). An application of the TODIM method to the multicriteria rental evaluation of residential properties. *European Journal of Operational Research, 193*(1), 204–211.

Gomes, L. F. A. M., & Rangel, L. A. D. (2009). Determining the utility functions of criteria used in the evaluation of real estate. *International Journal of Production Economics, 117*(2), 420–426.

Goodman, A. C., & Thibodeau, T. G. (1998). Housing market segmentation. *Journal of Housing Economics, 7*(2), 121–143.

Goodman, A. C., & Thibodeau, T. G. (2003). Housing market segmentation and hedonic prediction accuracy. *Journal of Housing Economics, 12*(3), 181–201.

Goodman, A. C., & Thibodeau, T. G. (2007). The spatial proximity of metropolitan area housing submarkets. *Real Estate Economics, 35*(2), 209–232.

Gröbel, S., & Thomschke, L. (2018). Hedonic pricing and the spatial structure of housing data–an application to Berlin. *Journal of Property Research, 35*(3), 185–208.

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies, 33*(5), 2223–2273.

Harewood, S. I. (2006). Managing a Hotel's perishable inventory using bid prices. *International Journal of Operations & Production Management*. https://doi.org/10.1108/01443570610691094.

Helbich, M., & Griffith, D. A. (2016). Spatially varying coefficient models in real estate: eigenvector spatial filtering and alternative approaches. *Computers, Environment and Urban Systems, 57,* 1–11.

Hu, L., et al. (2019). Monitoring housing rental prices based on social media: An integrated approach of machine-learning algorithms and hedonic modeling to inform equitable housing policies. *Land Use Policy, 82,* 657–673.

Huang, Y. (2019). Predicting home value in California, United States via machine learning modeling. *Statistics, Optimization & Information Computing, 7*(1), 66–74.

Isakson, H. R. (1988). Valuation analysis of commercial real estate using the nearest neighbors appraisal technique. *Growth and Change, 19*(2), 11–24.

Johnson, M. P. (2003). Single-period location models for subsidized housing: Tenant-based subsidies. *Annals of Operations Research, 123,* 105–124.

Koetter, M., & Poghosyan, T. (2010). Real estate prices and bank stability. *Journal of Banking & Finance, 34*(6), 1129–1138.

Kok, N., Koponen, E. L., & Martínez-Barbosa, C. A. (2017). Big data in real estate? *The Journal of Portfolio Management, 43*(6), 202–211.

Kontrimas, V., & Verikas, A. (2011). The mass appraisal of the real estate by computational intelligence. *Applied Soft Computing, 11*(1), 443–448.

Kuşan, H., Aytekin, O., & Özdemir, İ. (2010). The use of fuzzy logic in predicting house selling price. *Expert systems with Applications, 37*(3), 1808–1813.

Kusiak, A. (2020). Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research, 58*(5), 1594–1604.

Lam, K. C., Yu, C. Y., & Lam, C. K. (2009). Support vector machine and entropy based decision support system for property valuation. *Journal of Property Research, 26*(3), 213–233.

Li, J., & Tang, O. (2012). Capacity and pricing policies with consumer overflow behavior. *International Journal of Production Economics, 140*(2), 825–832.

Lockwood, T., & Rossini, P. (2011). Efficacy in modelling location within the mass appraisal process. *Pacific Rim Property Research Journal, 17*(3), 418–442.

Lolli, F., Balugani, E., Ishizaka, A., Gamberini, R., Rimini, B., & Regattieri, A. (2019). Machine learning for multi-criteria inventory classification applied to intermittent demand. *Production Planning and Control, 30*(1), 76–89.

Mayer, M., Bourassa, S. C., Hoesli, M., & Scognamiglio, D. (2018) Estimation and updating methods for hedonic valuation. *Swiss Finance Institute Research Paper* (18–76).

McCluskey, W. J., et al. (2013). Prediction accuracy in mass appraisal: A comparison of modern approaches. *Journal of Property Research, 30*(4), 239–265.

McCluskey, W. J., & Borst, R. A. (2011). Detecting and validating residential housing submarkets. *International Journal of Housing Markets and Analysis, 4,* 290.

McCluskey, W. J., Daud, D. Z., & Kamarudin, N. (2014). Boosted regression trees: An application for the mass appraisal of residential property in Malaysia. *Journal of Financial Management of Property and Construction*. https://doi.org/10.1108/JFMPC-06-2013-0022.

McNeill, G., & Hale, S. A. (2017). *Generating tile maps* (pp. 435–445). Wiley Online Library: In Computer Graphics Forum.

Morano, P., Tajani, F., & Locurcio, M. (2018). Multicriteria analysis and genetic algorithms for mass appraisals in the Italian property market. *International Journal of Housing Markets and Analysis*. https://doi.org/10.1108/IJHMA-04-2017-0034.

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives, 31*(2), 87–106.

Shigaki, I., & Narazaki, H. (1999). A machine-learning approach for a sintering process using a neural network. *Production Planning and Control, 10*(8), 727–734.

Narula, S. C., Wellington, J. F., & Lewis, S. A. (2012). Valuating residential real estate using parametric programming. *European Journal of Operational Research, 217*(1), 120–128.

Orford, S. (2017). *Valuing the built environment: GIS and house price analysis*. London: Routledge.

Padhi, S. S., Theogrosse-Ruyken, P., & Das, D. (2015). Strategic revenue management under uncertainty: A case study on real estate projects in India. *Journal of Multi-Criteria Decision Analysis, 22*(3–4), 213–229.

Pagourtzi, E., Assimakopoulos, V., Hatzichristos, T., & French, N. (2003) Real estate appraisal: A review of valuation methods. *Journal of Property Investment & Finance*.

Pedersen, A. M. B., Weissensteiner, A., & Poulsen, R. (2013). Financial planning for young households. *Annals of Operations Research, 205,* 55–73.

Lins, M. P. E., de Lyra Novaes, L. F., & Legey, L. F. L. (2005). Real estate appraisal : A double perspective data envelopment analysis approach. *Annals of Operations Research, 138,* 79–96.

Pérez-Rave, J. I., Correa-Morales, J. C., & González-Echavarría, F. (2019). A machine learning approach to big data regression analysis of real estate prices for inferential and predictive purposes. *Journal of Property Research, 36*(1), 59–96.

Di Pietro, G., & Rinnone, F. (2017). Online geocoding services: A benchmarking analysis to some European cities. In *2017 Baltic Geodetic Congress (BGC Geomatics)*, IEEE, 273–81.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: from theory to algorithms*. Cambridge: Cambridge University Press.

Shin, C. K., & Park, S. C. (2000). A machine learning approach to yield management in semiconductor manufacturing. *International Journal of Production Research, 38*(17), 4261–4271.

Shmueli, G., & Yahav, I. (2018). The forest or the trees? Tackling Simpson's paradox with classification trees. *Production and Operations Management, 27*(4), 696–716.

Singh, S. K. (2017). Evaluating two freely available geocoding tools for geographical inconsistencies and geocoding errors. *Open Geospatial Data, Software and Standards, 2*(1), 11.

Stigler, S. M. (1981). Gauss and the invention of least squares. *Annals of Statistics, 9*(3), 465–474.

Thériault, M., Des Rosiers, F., Villeneuve, P., & Kestens, Y. (2003). Modelling interactions of location with specific value of housing attributes. *Property Management*. https://doi.org/10.1108/02637470310464472.

Valier, A. (2020). Who performs better? AVMs vs Hedonic Models". *Journal of Property Investment & Finance, 38,* 213.

Viriato, J. C. (2019). AI and machine learning in real estate investment. *The Journal of Portfolio Management, 45*(7), 43–54.

Wang, D., & Li, V. J. (2019). Mass appraisal models of real estate in the 21st century: A systematic literature review. *Sustainability, 11*(24), 7006.

Wen, X., Xu, C., & Hu, Q. (2016). Dynamic capacity management with uncertain demand and dynamic price. *International Journal of Production Economics, 175,* 121–131.

Wu, R. C. (1997). Neural network models: Foundations and applications to an audit decision problem. *Annals of Operations Research, 75,* 291–301.

Xu, T. (2008). Heterogeneity in housing attribute prices. *International Journal of Housing Markets and Analysis, 1,* 166.

Yacim, J. A., & Boshoff, D. G. B. (2018). Impact of artificial neural networks training algorithms on accurate prediction of property values. *Journal of Real Estate Research, 40*(3), 375–418.

Yu, D., & Wu, C. (2006). Incorporating remote sensing information in modeling house values. *Photogrammetric Engineering & Remote Sensing, 72*(2), 129–138.