Noname manuscript No.
(will be inserted by the editor)

Estimation of missing prices in real-estate market agent-based simulations with machine learning and dimensionality reduction methods

Iván García-Magariño · Carlos Medrano · Jorge Delgado

Abstract The opacity of real-estate market involves some challenges in their agent-based simulation. While some real-estate websites provide the prices of a great amount of houses publicly, the prices of the rest are not available. The estimation of these prices is necessary for simulating their evolution from a complete initial set of houses. Additionally, this estimation could also be useful for other purposes such as appraising houses, letting buyers know which are the best offered prices (i.e. the lowest ones compared to the appraisals), and recommending the buyers to set an initial price. This work proposes combining dimensionality reduction methods with machine learning techniques to obtain the estimated prices. In particular, this work analyzes the use of non-negative factorization, recursive feature elimination and feature selection with a variance threshold, as dimensionality reduction methods. It compares

I. García-Magariño
Department of Computer Science and Engineering of Systems, EduQTech University of Zaragoza. Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain. Instituto de Investigación Sanitaria Aragón. University of Zaragoza. Zaragoza.
Tel.: +34-978645348
Fax: +34-978618104
E-mail: ivangmg@unizar.es

C. Medrano
Department of Electronics Engineering and Communications, EduQTech, University of Zaragoza. Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain. Instituto de Investigación Sanitaria Aragón. University of Zaragoza. Zaragoza.

J. Delgado
Department of Applied Mathematics, University of Zaragoza. Escuela Universitaria Politécnica de Teruel, c/ Atarazana 2, 44003 Teruel, Spain.

the application of linear regression, support vector regression, the k-nearest neighbors and a multi-layer perceptron neural network, as machine learning techniques. This work has applied a 10-fold cross validation for comparing the estimations and errors and assessing the improvement over a basic estimator commonly used in the beginning of simulations. The developed software and the used dataset are freely available from a data research repository for the sake of reproducibility and the support to other researchers.

Keywords agent-based simulation · machine learning · real-estate market · simulation setup

1 Introduction

Agent-based simulations (ABSs) use different autonomous entities called agents for reproducing certain individual behaviors in a realistic way, in order to simulate, analyze and predict the emergent behaviors of a certain group of individuals (Davidsson, 2002). There are many different approaches for defining ABSs and implementing them. From a software engineering point of view, the Ingenias Agent Framework was adapted to define ABSs and develop the corresponding simulators with a model-driven development approach using an adaptation of the Ingenias methodology and its language (Gómez-Sanz et al, 2010). PEABS (a Process for developing Efficient Agent-Based Simulators) allows users to develop efficient simulators from the corresponding ABS models (García-Magariño et al, 2015). From a sociology point of view, ABS models can be defined with toolkits such as NetLogo, Repast Simphony, Repast and Mason (North et al, 2013). ABSs can also use fuzzy techniques to manage uncertainty in the simulations (Hassan et al, 2010).

In the last decades, the available data are exponentially increasing in most domains (Reiser et al, 2002). The big data research field gathers the techniques and methods for analyzing and managing these large amounts of data (Provost and Fawcett, 2013). ABSs can be an appropriate approach for analyzing and predicting relevant results from large amounts of data. For addressing this goal, some works have used parallel computing (Borges et al, 2017). In particular, the HLA Actor Repast approach is aimed at distributing Repast models for high-performance simulations of scalable models (Cicirelli et al, 2011).

Nevertheless, the capability of performing realistic and complex ABS does not only require the proper high-performance execution mechanisms. In some domains, the difficult access to some data hinders the possibility of executing realistic ABSs from all the initial real data. For example, the real-estate market is opaque in some aspects such as the housing values (Chang et al, 2016).

In this context, the current work analyzes different ways of estimating non-accessible data for ABSs in the context of the real-estate market. In particular, it applies several dimensionality reduction methods and some machine learning techniques. The analyzed dimensionality reduction methods are non-negative matrix factorization (NMF), recursive feature elimination (RFE) and feature

selection (FS) with a variance threshold. The applied machine learning techniques are linear regression, support vector regression (SVR), the k-nearest neighbors (KNN) and a multi-layer perceptron (MLP) neural network.

The selected methods offer a wide range of different approaches that can be tested in this problem. Linear regression is simple, fast and its solution can be found in closed form. It cannot deal with nonlinear dependences but it can be seen as a baseline algorithm. KNN is a pure data driven method that does not assume any particular form of the regression function. For classification, it is known to have a bounded error in the limit of infinite number of samples. It is assumed that two near samples should have similar output, which seems reasonable in general. MLP is known to be able to approximate any function under not very restrictive conditions. Thus it can model nonlinear behavior in a natural way. SVR applies the ideas of sparse kernel machines to regression problems. It can work with high dimensional spaces. The prediction is usually efficient since it is based on a subset of the training points.

When working with a very big training dataset, it can be important to use dimensionality reduction techniques. These techniques can reduce the processing time when training different machine learning algorithms, which is particularly important in big data. In this work, we have considered NMF, RFE and FS with a variance threshold as dimensionality reduction techniques, and we have applied them to linear regression, SVR, KNN and MLP.

The contribution of this work is the use of machine learning techniques to predict initial missing values in simulations of real-estate markets. Besides, the dimensionality reduction NMF has been applied for the first time for house price prediction. The different combinations of dimensionality reduction and machine learning algorithms have been tested with data taken from Idealista[1], a popular Spanish real estate portal.

2 Related work

2.1 Real-estate simulations

Several works have simulated real-estate markets for different purposes. Some simulation models focus on estimating the risks of the real-estate market, as these are normally considered relevant by investors. For example, Pyhrr (1973) presented a computer simulation model that calculates the risks of investing in real-estate providing some probabilities. They considered both business risk (related to real-estate transactions and their prices) and financial risks (related to debt financing). In this line of work, Jiang et al (2007) introduced a simulation model for assessing the economic risk of a real-estate project. Their approach was based on a weighted combination of risk factors. They obtained the corresponding influencing weights by an analytical hierarchy process.

The proposal of Zhuge et al (2016) is more similar to the current one from a methodological point of view, as both works are in the context of

[1] https://www.idealista.com (last accessed October 22, 2018)

ABS models. In particular, their approach analyzed the residential location choices and the real estate prices by means of simulations. They simulated negotiations between household buyer agents and owner seller agents. They mainly considered the house prices and their accessibility. A case study about a medium-sized city of China corroborated their proposal.

Other works apply simulation models for analyzing the investment behaviors of firms. Wang et al (2017) used ten years of real-estate information of China for detecting the driving forces behind the macroeconomic variables. They analyzed several influences between private and state-owned firms due to the discriminatory financial constraints in China.

However, none of the aforementioned works has used real-estate simulations for assessing the capability of machine learning and dimensionality reduction methods for estimating the missing data at the beginning of simulations.

2.2 Machine learning methods and ABSs

Machine learning methods have been applied several times to reinforce ABSs in the literature. For instance, the Q-learning algorithm has reinforced an ABS model in the context of radiotherapy (Jalalimanesh et al, 2017). Their approach was simulated with R-NetLogo package. They mainly modeled the repercussions of different treatments considering the radio sensitivity of cells. The learning algorithm improved the treatment plans for the cure of the tumor with minimum side effects.

Machine learning has proven to be useful for incorporating some learning capabilities in the agents. For example, Wojtusiak et al (2012) included learning capabilities in agents in stochastic ABSs in the context of transportation logistics. More concretely, their autonomous logistics were enhanced with learning predictive models for environment conditions and learning for the evolutionary plan optimization. In addition, Khalil et al (2015) proposed a framework for machine learning in interactive multi-agent systems (MIL-MAS). The main goal of their framework is to enhance interactive agents to learn from their interactions. Their proposed learning method outperformed the Q-learning algorithm according to their experiments in the taxi domain.

Multi-agent systems (MASs) are the precursor of ABS and represent a wider concept, concerning interactive autonomous entities, but not necessarily aimed at simulations. Machine learning based on neural networks has been combined with them in the literature. For instance, Cui et al (2016) applied a MAS in combination with a neural network to achieve a distributed adaptive consensus for reducing the burden of network communication. In addition, Becker et al (2016) combined MAS and neural network methods for taking routing decisions in a logistic facility. Their experiments showed the improvement of their method over other well-known routing mechanism in a simulated scenario based on real data from a car terminal.

Nevertheless, to the best of authors' knowledge, machine learning techniques have not been applied to obtain the initial missing values for ABSs.

2.3 Big data

The exponential growth of data over time implies new challenges that are studied by the big data field (Yaqoob et al, 2016). This steep increase of data not only requires from new solutions for the upcoming challenges, but also provides appropriate scenarios for testing and validating certain research theories. In this context, the free open big data repositories are continuously becoming more popular and useful.

The official and national repositories of some countries (e.g. USA and UK) are especially relevant for their variety and amount of information and their free open access. Some research studies have used these repositories. For example, the official UK repository[2] was applied to estimate the demands of certain kinds of activities based on the natural environment information (Tratalos et al, 2016). This UK repository has also allowed Brown et al (2016) to train a tool for estimating the impact of mega-nourishment interventions on coast areas affected by the construction growth. Several European big data repositories have supported other research studies. For instance, Jayaram et al (2015) analyzed the effective technological marketing strategies in several countries of Easter Europe according to features such as privacy laws, demographics, competitive conditions, attitude towards technology and corruption, thanks to one of these repositories. They considered different marketing technologies like digital profiling, websites, content management, social media, digital collaborations and analytics.

Several private companies own large and increasing datasets generated from their business model. For example, 100 h of video is uploaded to YouTube every minute, and 100 terabytes of data are uploaded daily in Facebook (Yaqoob et al, 2016). More concretely in the real-estate context, Idealista is a popular Spanish website that includes about 9,000 new house offers per day. The exportation and importation of the information from these private companies are not as promoted and assisted as in the aforementioned official repositories, but still some works like the current one can benefit from the information of these private companies.

Certain big data scenarios have supported works about ABSs and more generally MASs. For instance, the Care High Performance Simulation (HPS) tool (Borges et al, 2017) defined and simulated ABS models with parallel computing. In this way, developers were able to run ABSs using big data with the corresponding parallel calculations. Anya et al (2015) proposed the application of agent-based models for simulating enterprise operations based on big data, applying the already existing Gaia methodology. Wang et al (2016) presented a design of a smart industry factory in which the coordination of smart objects (e.g. machines, conveyers and products) was performed by means of a MAS. This MAS received feedback based on big data.

The management of big data has been assisted by dimensionality reduction in several works. Dimensionality reduction is useful because normally the gi-

[2] http://data.gov.uk/ (last accessed September 16, 2017)

gantic datasets have significant amounts of redundancies (Houari et al, 2016). Different domains have benefited from dimensionality reduction for managing big data. For example, Sun and Wen (2017) proposed a new dimensionality reduction method for recognizing facial expressions. Sabarina and Priya (2015) applied a tensor based feature reduction model for ignoring unwanted data dimensions for precision agriculture in the management of real-time data analysis. NMF is a dimensionality reduction technique that has supported the management of big data. For instance, Žibert et al (2016) used NMF to analyze the relations between air pollution and meteorological situations. Their work was based in two well-known European large datasets, and corroborated the existence of some particulate matter patterns and discovered new ones. Maruyama et al (2014) applied NMF for detecting cells on calcium imaging data. In the case of noisy data, NMF outperformed the alternative component analysis algorithm. In the work of Chen et al (2014), NMF supported the sense induction of each topic senses from the large variety of social tags about it. Notice that the social tags about web resources are written by different users and can be ambiguous.

Nonetheless, these works did not apply dimensionality reduction techniques for estimating the missing data at the initial stage of real-estate ABSs.

2.4 Applications based on Idealista

The main aim of Idealista website is to facilitate the free interchange of information about Spanish house units so that these can be easily bought or rented. The existence of several works support the reliability of the real-time dataset of Idealista. For instance, Chasco Yrigoyen and Le Gallo (2012) analyzed the spatial autocorrelation of market prices of housing units by means of this dataset.

In addition, some works have studied different aspects of the Spanish construction bubble by using data of the Idealista web. For example, García (2010) analyzed the factors that provoked the breakdown of the Spanish urban growth model, and compared these factors with the ones related to a similar situation in USA. Bárcena Ruiz et al (2011) studied the effect of the real-estate bubble by means of this dataset, in particular in the city of Bilbao.

Idealista has even supported the analysis of social behaviors. Bosch et al (2015) used Idealista for assessing the discrimination in rental housing. They contacted trough this website by messages showing interest in certain properties. They used both native and foreign-sounding names to measure the differences in negative responses rates.

Therefore, the current work has considered Idealista website as a reliable resource for using their data in the current research. In addition, the analysis of Idealista data for comparing machine learning and dimensionality reduction methods is novel in the context of the estimation of initial missing data in ABSs, to the best of our knowledge.

3 ABS challenge in real-estate market

The application of ABSs requires certain steps. One of these is the set-up of initial configuration, including the features of the initial state of the individual. Gilbert and Terna (2000) mentioned that the initial configuration of the system was a task of the modeler. Basically, one can distinguish the mechanism of setting up the initial population of agents between two different ways, (a) setting up every agent based on the corresponding real value, or (b) generating the agents from an initial global value. The former category is precise, but it is not always possible and convenient. In some cases, there can be some missing initial data, and in others the user may not have time to introduce all the data manually.

In ABSs with big data, normally there are some missing values, which need to be automatically generated. The generation of these data can be performed from a very simplistic way or addressed in a domain-specific way. As an example of a simplistic way, all the values are set to the average in some existing ABSs, like ABS-MindHeart (García-Magariño and Plaza, 2017). In some domains such as the simulation of cancer spreading, the initial set-up is performed by the formula of the normal distribution regarding the position of the corresponding cell (Zhang et al, 2009).

The current work focuses on the specific domain of the real-estate market. In this market, some prices are intentionally omitted by some owners and/or agency properties, while other prices are publicly available by some websites. Hence, the target problem of the current approach is formalized as follows:

Problem 1 Given some houses with certain known prices and some other houses whose prices are not available, the goal is to estimate the missing prices according to the features of the houses.

For example, we have developed an ABS that simulates different selling and buying strategies in real-estate markets (García-Magariño and Lacuesta, 2017). In this ABS, the seller agents represent owners of houses, and each one can observe the market and decide whether to offer its house for sale. If so, it also has to decide the price of the offer. In addition, the buyer agents represent potential buyers that observe the houses offered for sale and decide whether to buy a house and which one. The decisions of both kinds of agents rely on information such as the trends of average price in the last simulation iterations, the frequency of real-estate transactions in a certain number of last iterations, and the current ratio between the buyers and sellers. The ABS is easily extensible so that users can define new buying/selling strategies and simulate them.

In particular, buyer and seller agents used different kinds of input information for taking decisions about respectively whether to buy and how to change prices. Some agents calculated the price trends. For instance, buyer agents waited to obtain better prices in decreasing trends, and urged to buy in increasing trends. Seller agents updated the prices following the current trend to be near market price. Other agents used the frequency of transactions and

compared this to a threshold to detect real-estate bubbles. In this way, buyers avoided buying houses with unfair prices, and sellers took advantage by selling houses with high prices thanks to the bubble. Other agents considered the ratio between sellers and buyers. In this way, when sellers were fewer than buyers, they increased prices. Oppositely, if the number of sellers was higher than the number of buyers, then the former ones decreased the prices. A few agents followed more basic patters, denoted respectively as urgent and patient, considering whether they need to perform the real-estate transaction soon or not. For example, urgent sellers decreased the prices until selling the house, and urgent buyers bought a house soon even if the prices were high.

Figure 1 presents the hierarchy of the most relevant agents. Notice that Buyer and Seller agents were abstract supertypes of agents that were used to gather all the common functionalities of respectively buyer and seller agents. All the concrete agent types extended these agents since there were normally both buying and selling strategies behind each main criterion. Thus, agent types were organized as pairs, and each of these had a buyer and a seller agent usually sharing a component that calculated the necessary information for taking decisions with the corresponding criterion. For example, one criterion was the estimation of bubble versus inactive market dynamics based on the frequency of transactions, and the agents were referred as "Frequency Transaction Buyer" and "Frequency Transaction Seller". These agents kept track of the number of accumulated transactions updating the corresponding integer variable inside the corresponding module, and based on the transactions frequency they provided an estimation of whether the market was experiencing a bubble. Similarly, the pair of the agent types "Price Trend Buyer" and "Price Trend Seller" kept track of the average prices detecting when the prices were increasing or decreasing, by keeping the first recorded price in a certain window time. Finally, the "Ratio Supply and Demand Seller" and the "Ratio Supply and Demand Buyer" agent types internally managed some superior and inferior thresholds, to determine different behaviors for different ratios of buyers divided by sellers. All these agents implemented their strategies for taking different decisions with the "Decide" and "Decide Finally" methods invoked depending whether the corresponding agent has been waiting for an excessive time amount. Besides these concrete components, all the buyer agents had an inherited property called "budget" about the maximum money that the agent had planned to spent for a given moment. All the buyers had some preferred features such as the number of rooms. All buyer agents also had access to components for initializing the price and changing it with some basic methods and parameters. All the seller agents had a reference to a house, which had the properties of the house. They also had a price for offering the house when available. Two flag indicated respectively whether their house was available for buyers and whether they had already sold the house. Notice that in some strategies, the sellers observed the market before making the house available for sale.

Although the strategies can be quite different, the interaction protocol was the same between buyer agents and seller agents. Notice that buyers can

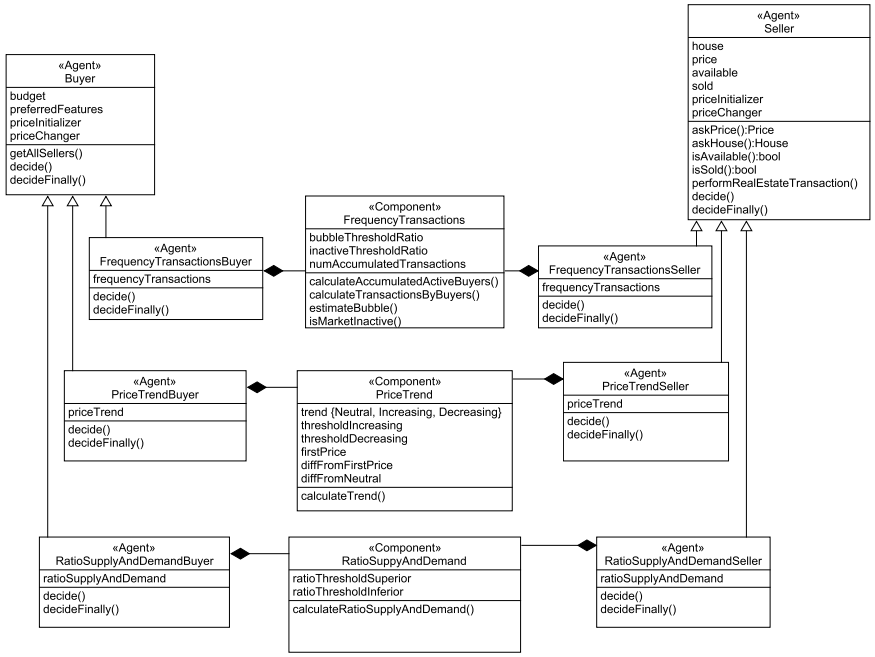

Fig. 1 Hierarchy of agent types expressed with a UML class diagram

estimate the situation by asking many sellers. Thus, the interaction was defined at the supertype level between buyers and sellers following the sequence of messages determined in the sequence diagram of Figure 2. The seller agent can ask whether a seller has their housing unit available for selling. Then, the buyer can ask for information about the house for comparing it with their preferences. The buyer can also ask for the price. After this, the buyer decides whether to buy the house, and if so they can perform the real estate transaction.

Figure 3 shows the main interface of an execution example of this ABS about real-estate market. In this case, the ABS was adapted to separately consider some of the main district areas of the Teruel city (Spain), which were San Julián, San León, and Ensanche.

In this execution example, since we used a realistic distribution of the different kinds of buyers and the different kinds of sellers, the simulation obtained realistic results as one can observe in the evolution chart provided by the ABS in Figure 4. This simulation included (a) 66 sellers and 40 buyers using the price trend, (b) 66 sellers and 40 buyers considering the frequency of transactions, (c) 63 sellers and 40 buyers based on the ratio between sellers and buyers, (d) 3 urgent sellers and 10 urgent buyers, and (e) 10 patient buyers. In addition, 23% of these agents preferred/owned houses in San Julian neighborhood, 30% preferred/owned houses in San Leon neighborhood, and 47% did it in Ensanche neighborhood. Furthermore, other data were presented in

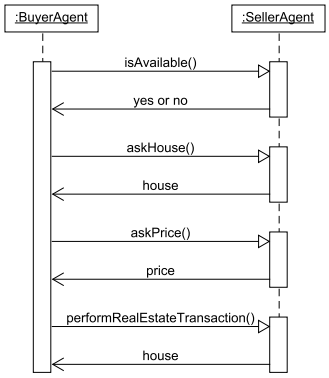

Fig. 2 Interactions between buyer and seller agents expressed with a UML sequence diagram

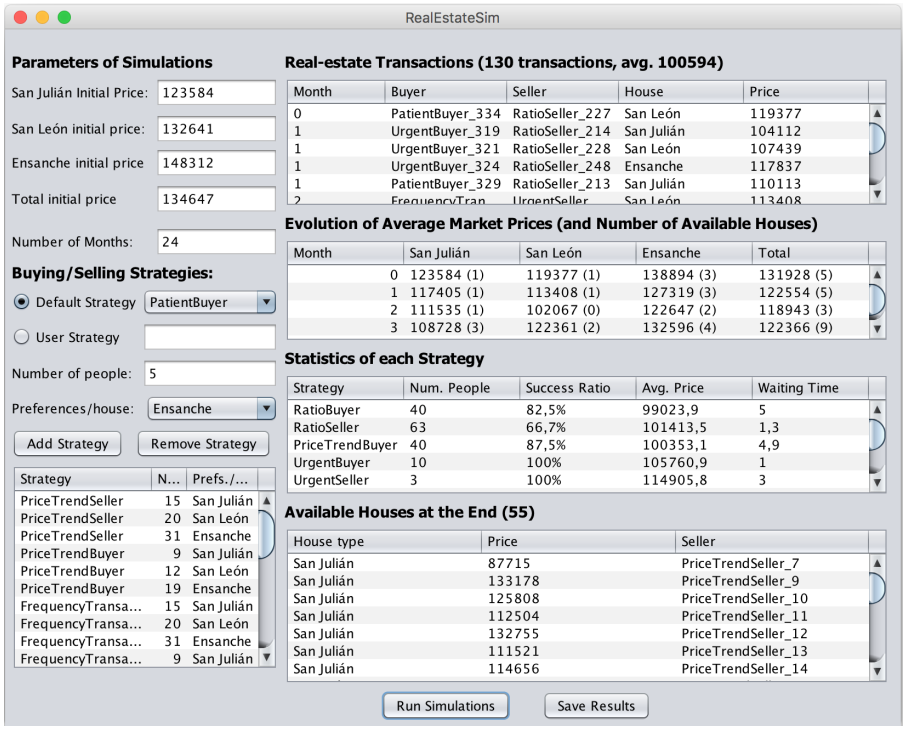

Fig. 3 Main interface of the ABS for simulating real-estate markets.

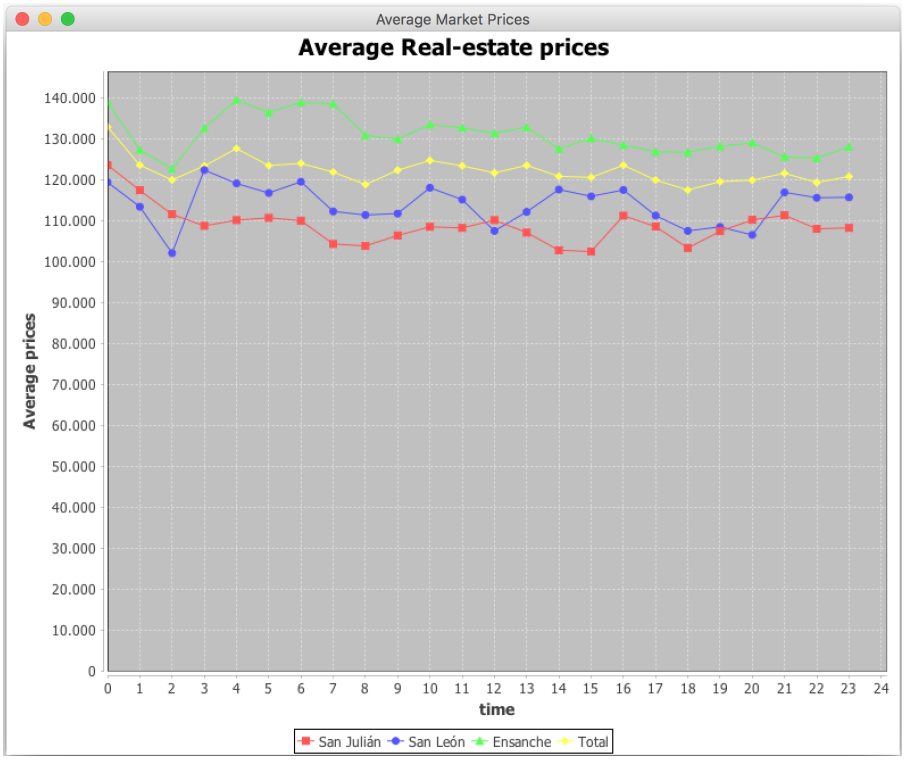

Fig. 4 Average price evolutions simulated by the ABS about real-estate markets.

the main interface such as success ratio of each buyer/seller strategy, the average price they bought/sold the house units, and the average waiting time in number of months.

The resolution of problem 1 is mainly aimed at generating the missing data for the initial set of houses for performing ABSs. This could lead to much more realistic evolutions of individuals from the beginning of simulations.

This estimation of house prices could also be useful in other contexts. For instance, the banks could consider this information for tuning the appraisal of houses for deciding different factors of mortgages. In addition, home buyers could use this information to determine which are the best offers by comparing their prices with the estimated ones. Sellers could use this information to set the initial prices of their houses. These applications could be useful for supporting decisions of human bankers, buyers and sellers, but could also be applied for simulating agent decisions in ABSs based on the estimation of the fair market price of each particular house.

Although the current approach for addressing this challenge is proposed and experienced in the real-estate market, it may be the precursor for estimating the initial missing data in the ABSs of other domains.

4 Study

We conducted an analysis of the appropriateness of different machine learning techniques combined with several reduction mechanisms in the context of the real-estate market. All the software developed for this study and the used dataset are available from a public research data repository (García-Magariño et al, 2017), in order to ensure its reproducibility and allow other researchers to use or extend them.

Section 4.1 mentions the sample of houses selected in the current study. Section 4.2 briefly introduces each of the machine learning and dimensionality reduction methods applied in the current research. The feature scales were adjusted as indicated in section 4.3, so that their values were comparable and the machine learning methods were able to be applied. Section 4.4 explicitly indicates the protocol applied in the current research. Section 4.5 determines the ranges of values that were used for the automatic calibration of parameters in the different methods.

4.1 Sample

In order to achieve certain statistical power, the minimum houses sample size was calculated a priori with the G*Power 3 tool (Faul et al, 2007). In particular, we calculated the minimum sample size for detecting medium effect sizes (i.e. 0.5) with a common significance level ($\alpha = 0.05$) and a power of 0.95 for a paired t-test. The obtained minimum sample size was 45. Thus, we selected a real sample that was larger than this statistically calculated minimum.

More concretely, this study used the houses of two neighborhoods of Teruel city that were collected from the Idealista website up to November 13, 2017. The neighborhoods were respectively the city center and Ensanche. In total, the sample included 89 houses.

This study considered the following features of each house: (1) the location with its neighborhood, (2) whether it is a flat, (3) whether it is a duplex, (4) whether it is an attic, (5) the number of bedrooms, (6) the area, (7) the floor, (8) whether the house has a lift, (9) whether it has a garage, (10) gas heating, (11) whether it is new/remodeled or not, (12) whether it has a box room, and (13) the number of bathrooms. The price was considered as the target value for being predicted.

There is a large variety of features used for real estate price prediction that can be reasonably linked to the output. The features selected in the present study can be found in the Idealista website and allowed us to gather a dataset without missing input features. Besides, they have also been considered in previous studies, directly or with a related characteristic (Park and Bae, 2015; Chiarazzo et al, 2014; García et al, 2008; Li, 2006; Nguyen and Cripps, 2001).

4.2 Methods

We tested several machine learning and dimensionality reduction methods, to address the problem of estimating the missing prices of a sample of houses. The former methods were related to regression, since the missing value was the price, which is a real numeric value. Each of the used methods is briefly introduced in a subsection of this section.

4.2.1 Linear regression

In a linear model, the predicted price of a house $\hat{p}$ is expected to be a linear combination of its features, $\mathbf{x}$. In particular, the price would be calculated with the following formula:

$$\hat{p}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 \tag{1}$$

where w_0 is the intercept, and $\mathbf{w}$ is a vector of coefficients (weights).

The intercept and weights are found by minimizing the sum-of-square error function. If $\mathbf{x}_n$ are the training feature vectors and p_n the true prices of the houses, then that sum is:

$$\frac{1}{2}\sum_n \|\hat{p}(\mathbf{x}_n) - p_n\|^2 \tag{2}$$

The solution of this problem can be found in closed form. Another advantage of the method is that it has no parameters other than the weights and intercept, which are found directly in the training phase. However, the model is not able to capture non-linear behavior and is very sensitive to random errors when the features have a high degree of collinearity.

In the current work, the linear regression was applied with the ordinary least squares algorithm (Dismuke and Lindrooth, 2006) using the Python library Scikit-learn (Pedregosa et al, 2011). This library has also been applied for the other machine learning techniques and the dimensionality reduction methods used in this study.

4.2.2 KNN

KNN (Maltamo and Kangas, 1998) is based on storing a set of training samples. When predicting the price of a new house, it searches for the k most similar houses in the training set. Then, it returns an average of its prices.

A key aspect of the method is the definition of similarity between exemplars. In order to determine the most similar houses, the Euclidean distance has been considered in this study. However, the features are very different in nature. Thus, they were preprocessed beforehand (see section 4.3) so that the Euclidean distance was meaningful.

The technique is conceptually simple, yet it can yield good results in practice. There is no need to find parameters in a training step since it is a purely data driven method, which does not make any assumption about the shape

of the regression function. However, there should be training vectors covering all the input region so that the k-neighbors are truly similar. Besides, it can be slow at making predictions if there are many samples in the training set. The problem of finding the size of the neighborhood, k, can be solved by cross-validation.

4.2.3 SVR

SVR is an adaptation of the Support Vector Machine (SVM) classifier to regression problems (Bishop, 2006), so that it shares some of its characteristics. In this case the predicted price of a house is found by means of a function given by:

$$\hat{p}(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{3}$$

where $\mathbf{w}$ is a vector of weights, b a bias parameter and ϕ denotes a feature space transformation.

Given a set of feature vectors, $\mathbf{x}_n$, and their true prices, p_n, the weights and bias parameters are found by minimizing a regularized error function:

$$C \sum_n E_\epsilon(\hat{p}(\mathbf{x}_n) - p_n) + \frac{1}{2}\|\mathbf{w}\|^2 \tag{4}$$

C plays the role of an inverse regularization parameter and E_ϵ is an ϵ-insensitive error function instead of the typical quadratic error:

$$E_\epsilon(\hat{p}(\mathbf{x}) - p) = \begin{cases} 0, & if\ |\hat{p}(\mathbf{x}) - p| < \epsilon \\ |\hat{p}(\mathbf{x}) - p| - \epsilon, & otherwise \end{cases} \tag{5}$$

As for SVM, the problem can be transformed into the dual formulation (for details see (Bishop, 2006)) whose goal is to maximize the following function with respect to the variables a_n and a_n^*:

$$\begin{aligned} -\frac{1}{2} \sum_n \sum_m (a_n - a_n^*)(a_m - a_m^*) k(\mathbf{x}_n, \mathbf{x}_m) \\ -\epsilon \sum_n (a_n + a_n^*) + \sum_n (a_n - a_n^*) p_n \end{aligned} \tag{6}$$

subject to the constraints: $0 \leq a_n \leq C$, $0 \leq a_n^* \leq C$. $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ is the kernel function.

The prediction function is set as:

$$\hat{p}(\mathbf{x}) = \sum_n (a_n - a_n^*) k(\mathbf{x}, \mathbf{x}_n) + b \tag{7}$$

Some practical aspects of the method are the following:

- The transformation function $\phi()$ is not required explicitly. Instead, the kernel $k(\mathbf{x}, \mathbf{x}')$ can be defined directly. In this work, the popular radial basis function kernel has been used: $k(\mathbf{x}, \mathbf{x}') = exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2)$. The Euclidean distance between two vectors, $\|\mathbf{x}-\mathbf{x}'\|$, is found after preprocessing the data (section 4.3).
- The method can be applied to high dimensional spaces as far as a kernel is defined.
- The E_ϵ error function leads to sparse solutions: for many points both a_n and a_n^* are zero, so that the prediction, equation (7), depends on a reduced set of training points (the support vectors). These are the points whose predictions lie at a distance higher or equal to ϵ from the true value.
- The parameters of the method can be found by cross-validation: C (the inverse of the regularization parameter), ϵ (in the error function E_ϵ) and γ in the kernel function.

4.2.4 Artificial neural networks

Artificial neural networks were somehow inspired by the biological neural networks in an attempt to find a mathematical representation of their information processing. From a mathematical point of view, a typical feed forward neural network consists of a set of layers in which each layer feeds the next one. In each layer there are several nodes (neurons), which compute their outputs depending on a weighted sum of their inputs (Bishop, 2006). To be specific, consider a network with a single hidden layer and an output layer. In the hidden layer, each node j computes the combination for a given feature vector $\mathbf{x}$ as:

$$a_j = \mathbf{w}_j^T \mathbf{x} + w_{j0} \tag{8}$$

where $\mathbf{w}_j$ is a vector of weights and w_{j0} a bias parameter. The dimension of $\mathbf{w}_j$ is determined by the dimension of the feature vector.

Then, the output of each hidden node, z_j is given by:

$$z_j = h(a_j) \tag{9}$$

where $h()$ is the activation function. It is usually a differentiable nonlinear function, for instance a sigmoid.

The outputs of the hidden nodes, represented as a vector $\mathbf{z} = (z_1, \ldots, z_M)^T$, are combined in the output layer. In our case, the prediction is a single value, the house price, so that there is a single output node with the predicted value $\hat{p}$ as:

$$\hat{p} = \mathbf{v}^T \mathbf{z} + v_0 \tag{10}$$

where $\mathbf{v}$ is a vector of weights in the output layer, and v_0 the corresponding bias. The dimension of $\mathbf{v}$ is determined by the number of nodes in the hidden layer, M.

This output can also be transformed with an activation function, but for regression problems the activation function is the identity, leading $\hat{p}$ unchanged.

Given a set of feature vectors, $\mathbf{x}_n$, and their true prices, p_n, the weights and biases in the model, $\mathbf{w}_j$, w_{j0}, $\mathbf{v}$ and v_0 are found by minimizing an error function:

$$\frac{1}{2}\sum_n \|\hat{p}_n - p_n\|^2 \tag{11}$$

in which, $\hat{p}_n$ depends on the input vector $\mathbf{x}_n$ throughout equations 8, 9 and 10. The corresponding algorithm for minimization is called backpropagation.

Neural networks with non-linear activation functions are known to have great expressive power. In classification, they can implement arbitrary decision functions and in regression they can approximate any continuous function provided the number of hidden nodes is sufficiently large. However, an increasing number of weights and biases in the model can lead to overfitting, which would be the case if the network included many hidden layers or many nodes in them. In this study, a structure with a single hidden layer was selected and the number of nodes was tuned with cross-validation.

4.2.5 NMF

Machine learning algorithms can improve its performance when the high-dimensionality of the dataset is reduced by some technique. This section introduces one of these techniques: NMF.

Given a non-negative matrix $A \in \mathbb{R}^{n\times m}$ and a positive integer $p < \min(m, n)$, NMF seeks to find two non-negative matrices $U \in \mathbb{R}^{n\times p}$ and $V \in \mathbb{R}^{p\times m}$ such that $\|A - UV\|_F$ is minimal and $A \approx UV$. It is clear that $rank(UV) \leq p$. Since p is chosen to be smaller than n and m, U and V are smaller matrices than A. Hence UV is a compressed version of the original matrix A.

NMF is applied to a matrix $A \in \mathbb{R}^{n\times m}$ where m is the number of samples of the dataset and n is the number of features. Taking into account the non-negativity constraints in U and V, NMF provides a representation of each sample as a non-subtractive combination of parts. Hence, non-negativity constraints provide a representation of an object as a sum of localized features. Let us see this in detail. The approximation $A \approx UV$ can be written column by column as $a \approx Uv$, where a and v are the corresponding columns of A and V. In other words, each vector a, i.e, a sample in A, is approximated by a linear combination of the columns of U, weighted by the entries of v. So, the columns of U represent a basis of the localized features of the type of object represented by the dataset considered. Summarizing, the columns of U represent a basis of the localized features of the samples of the dataset and each column of V contains the encoding of the sample represented by the corresponding column of A. In that sense, Lee and Seung (1999) demonstrate an algorithm for NMF that is able to learn parts of faces and semantic features of text. Other methods like principal component analysis learn holistic, not part based representations, in contrast to NMF. In the context of faces, the j-th column of the matrix A contains n non-negative pixel values corresponding to the j-th face of the database of images. In that sense the p columns of matrix

U are called basis images. The j-th column of the matrix V is an encoding of the corresponding face.

In the literature, several algorithms for NMF has been presented. These algorithms can be classified into three categories: (a) multiplicative update algorithms, like in the works of Lee and Seung (1999) and Simovici (2012) in page 757, (b) based on gradient descent algorithms, as for example in (Lee and Seung, 2001), and (c) alternating least square algorithms, like in (Paatero and Tapper, 1994).

NMF has been applied in machine learning and data mining in text mining and spectral data analysis among other domains. A brief survey on NMF can be seen in section 14.8 of (Simovici, 2012).

4.2.6 Feature Selection

FS is used to improve estimator accuracy scores or to boost their performance on very high-dimensional datasets. FS tries to simplify a model by selecting a subset of representative features. So redundant or irrelevant features are ruled out. Feature selection methods are used to rule out irrelevant and redundant attributes from data that do not contribute to the accuracy of a model. Feature selection methods are compatibles with other techniques for reducing the dimensionality. In this section we summarize two methods for feature selection: RFE and FS based on a variance threshold.

Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of RFE is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained. Then, the least important features are ruled out from the current set of features. That procedure is recursively repeated on the pruned set until a good accuracy for the predictive model is obtained using the smallest number possible of features. RFE can be considered an embedded method since it performs feature selection as part of the model construction process. In particular, the current work has used the linear regression as the estimator.

The FS mechanism based on a variance threshold is a filter method since the selection of features is independent of the classifier used. It calculates the variance of each feature for all the houses and all the features whose variance is below an established threshold are discarded. By default, all the features that have the same value in all samples are removed by this technique.

In the experiments of the current approach, FS mostly selected the following house features: (a) the location, (b) whether it was a duplex, (c) whether it was an attic, (d) the floor, (d) the lift, (e) the garage, (f) the gas heating, (g) whether it was new/remodeled or not, and (h) whether it had a box room. On the other hand, RFE selected (1) whether it was a flat, (2) the number of bedrooms, (3) the area, (4) the floor, (5) the lift and (6) the number of bathrooms. The differences can be due to some redundancy of data, in which one method selected one feature and the other method selected the opposite ones. For instance, we only classified a house as a flat only when it was neither

duplex nor an attic. FS selected duplex and attic features, while RFE selected the flat feature. In addition, both methods agreed that the floor and the lift were relevant for being selected. All the features were selected by at least one of the two reduction methods. Thus, all the analyzed features influenced the prices of houses according to our experiments.

4.3 Feature pre-processing

Feature pre-processing is very common for most practical applications, see a comparative evaluation of pre-processing techniques in a recent work about analysis of Twitter datasets (Symeonidis et al, 2018) as an example. The original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve (Bishop, 2006). A simple linear rescaling of the input variables is very useful for example if different variables have typical values which differ significantly. So, in some applications, depending on the units in which each of the input variables is expressed, they may have values that differ by several orders of magnitude. But, the typical sizes of the inputs may not reflect their relative importance in determining the required outputs. By applying a linear transformation, one can arrange all the inputs to have similar values (Duda et al, 2012). In addition, pre-processing is also useful for guaranteeing that the input data satisfy the requirements of the methods applied, such as the restriction of having nonnegative data for NMF (Chung et al, 2018).

Most of the previously introduced machine learning and dimensionality reduction methods require that features are scaled so that their values are comparable. For this purpose, each feature is divided by its average value obtained from all the houses of the training sample. It is worth noting that all the features used in this study are non-negative. Therefore, the average of a given feature is positive and represents a relevant scale of the data.

In the selected mechanism, zero values are converted into zero and the average values are converted into one. The other values are proportionally transformed. In this way, different features are comparable.

This mechanism has been selected as all non-negative values keep been non-negative. This is useful for applying NMF. Oppositely, this property would not be satisfied when subtracting the mean and dividing by the standard deviation, which is another common method of pre-processing.

4.4 Protocol

Firstly, the houses sample was obtained from Idealista. Most features of the houses were directly extracted from the fields of each house. Other information was taken from the natural-language description of the house. In some cases, a person watched the photos to fill the missing information such as the kind of heating. In this manner, we assured that all the information was complete and coherent for each house.

The dimensionality of the house features was reduced respectively with NMF, RFE and FS. In the experiment, we considered each reduced dataset and the original one. For each dataset, this work respectively applied different machine learning regression methods, which are linear regression, SVR, KNN and a MLP neural network. A set of prices was predicted for each combination of dimensionality reduction and regression methods.

In order to avoid overfitting and validate the estimation power of the model, the current work has applied a 10-fold cross-validation for each predicted set. 10-fold cross-validation divides randomly the training dataset into 10 equal sized subsets. One of the subsets is used as the validation dataset and the remaining 9 subsets are used as training data for fitting the parameters. The cross validation is then repeated 10 times, with each of the 10 subsets used exactly once as validation dataset. The results are usually averaged to produce a single estimation. Thus, in each validation, the data had 90% of houses with real prices and 10% of houses with estimated prices facing real prices in the comparison. In each step, the calibration of the algorithm parameters was performed with the training cases in an automatic way considering certain ranges (see section 4.5). After this, the technique predicted the unused tenth of the sample. This was repeated ten times until all the sample was predicted for each case of dimensionality reduction and regression methods.

In order to asses each set of predicted price values, these were compared with the real ones. The errors were measured as the absolute differences between prices divided by the real ones. We calculated the mean squared errors (MSEs) as the effect size of the comparison between each predicted set and the real one. The distributions of errors were analyzed with boxplots. A paired t-test was conducted for comparing each predicted set of prices with real ones.

A common way of initializing the individuals of a simulator is to assign all of them with the average (García-Magariño and Plaza, 2017). For this reason, we used a basic mean estimator as a control mechanism in another set of comparisons. We compared the errors of the presented combinations of dimensionality reductions and regressions with the errors of this basic estimator. More concretely, we applied t-tests to determine which combinations significantly improved this basic estimator. The effect size was measured as the difference of MSEs for each comparison.

Furthermore, we measured the execution times for each combination of dimensionality reduction and regression methods to compare them.

4.5 Calibration of the parameters of the experiments

The applied dimensionality reduction techniques need to determine either the target number of dimensions or certain threshold. More concretely, in the current study NMF and RFE were applied to obtain 6 values from the 13 features for each house, in order to reduce the input information to its half. In the case of FS, the dimensions were reduced considering a variance threshold of 0.8.

	Linear Regression	SVR	KNN	MLP	Average	Minimum
None	0.161	0.090	0.167	0.168	0.147	0.090
NMF	0.333	0.185	0.190	0.169	0.220	0.169
RFE	0.105	0.105	0.151	0.212	0.143	0.105
FS	0.302	0.133	0.291	0.207	0.233	0.133
Average	0.226	0.128	0.200	0.189		
Minimum	0.105	0.090	0.151	0.168		

Table 1 Results of MSEs for the different regression and dimensionality reduction methods

Most of the used machine learning methods needed to calibrate certain parameters on their configuration. The selection was performed with an internal cross-validation using only the training data. Several parameters were tested and those that obtained the best performance were selected. This work has used some common ranges of values for each method. More specifically, the range was [2, 20] for the number of neurons of the hidden layer of MLP. We calibrated k in KNN by scanning it in the interval [1, 5]. In SVR, there are three parameters (Pedregosa et al, 2011): C, γ and ϵ. Following Chang and Lin (2011), the values were scanned to find the best ones in a logarithmic scale from 10^{-3} to 10^{3}. In addition, there was no specific parameter to be calibrated in the case of the linear regression.

5 Results

Table 1 indicates the MSEs of the different combinations of regression and dimensionality reduction methods. This table indicates the minimum and the averages for separately each regression and each dimensionality reduction technique. The minimum MSE values were useful to know the best results for each regression technique considering all the dimensionality reduction techniques. Similarly, it shows the best results for each dimensionality reduction approach. By contrast, the averages were useful to compare more robustly two regressions or two dimensionality reductions mitigating the bias from the good results obtained by chance for a very specific combination of regression and dimensionality reduction with the used data. Figure 5 presents these results with a chart. In addition, figure 6 shows the boxplots of the errors for these combinations of regression and dimensionality reduction methods.

All the simulated prices were compared with the real prices with paired t-tests for each combination of regression and dimensionality reduction methods. All the tests had 88 degrees of freedom. Table 2 shows the t statistics and the two-tailed significance for each test. As one can observe, there were no significant differences between the real and the simulated prices for any combination of dimensionality reduction methods and the linear regression, SVR and KNN. MLP obtained no significant differences between the real and the simulated prices only when it was used without dimensionality reduction.

The errors of the different method combinations have been compared with a very basic estimator (predicting each price as the known mean of the sample).

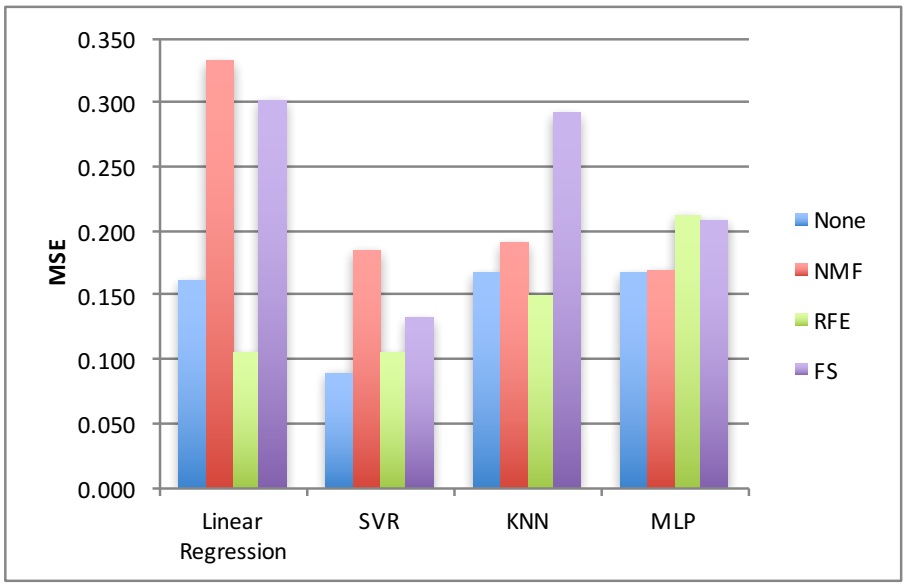

Fig. 5 Comparison of MSEs for the different regression and dimensionality reduction methods

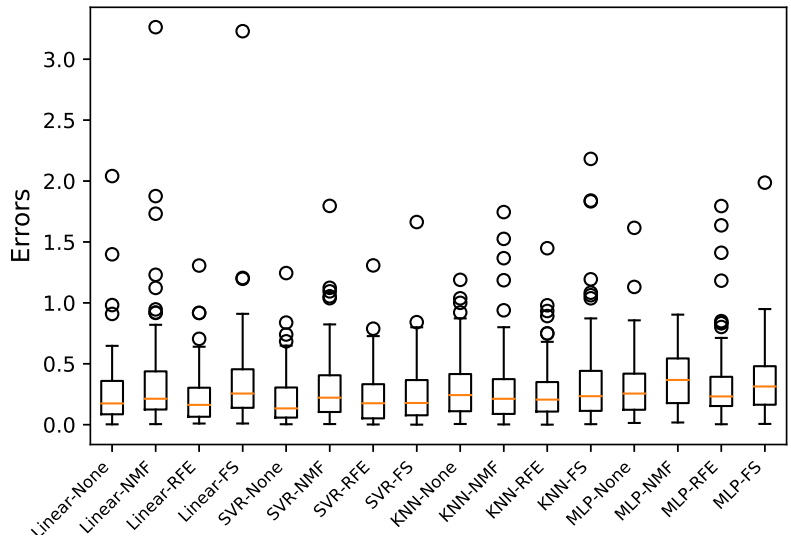

Fig. 6 Boxplots of the errors for the different regression and dimensionality reduction methods

	Linear Regression		SVR		KNN		MLP	
	t	sig.	t	sig.	t	sig.	t	sig.
None	-0.575	0.567	1.079	0.284	-0.396	0.693	0.993	0.323
NMF	-0.444	0.658	0.724	0.471	0.196	0.845	6.158	0.000*
RFE	-0.114	0.910	1.063	0.291	0.417	0.610	3.773	0.000*
FS	-0.524	0.601	0.989	0.325	-0.883	0.380	2.570	0.012*

Table 2 Comparison of real and simulated values with paired t-test for each combination of methods. * Significant with a 0.05 level.

	Linear Regression		SVR		KNN		MLP	
	t	sig.	t	sig.	t	sig.	t	sig.
None	3.147	0.002*	3.756	0.000*	-2.344	0.021*	2.274	0.025*
NMF	1.550	0.125	2.288	0.025*	-1.291	0.200	1.798	0.076
RFE	3.925	0.000*	3.622	0.000*	-1.750	0.084	2.053	0.043*
FS	1.435	0.155	2.911	0.005*	-1.971	0.052*	1.599	0.113

Table 3 Comparison of errors between each combination of methods and a basic mean estimator with paired t-tests. * is significant with a 0.05 level.

	Linear Regression	SVR	KNN	MLP	Average
None	0.410	0.481	0.404	0.403	0.432
NMF	0.238	0.386	0.381	0.402	0.351
RFE	0.466	0.466	0.420	0.359	0.428
FS	0.269	0.438	0.280	0.364	0.338
Average	0.345	0.443	0.371	0.382	

Table 4 Effect sizes between each combination of methods and the basic estimator, considering the differences of MSEs

This basic estimator obtained an MSE of 0.571. It is worth mentioning that all the combinations of machine learning and dimensionality reduction methods obtained better results, since the MSEs in all these methods ranged from 0.090 to 0.333.

A paired t-test was conducted for comparing each combination of methods with this basic estimator. Table 3 shows the t statistics and the two-tailed significances. As one can observe, all the machine learning methods significantly reduced the MSEs when not applying dimensionality reduction. All the machine learning methods significantly reduced MSEs when applying some of the dimensionality reductions. In particular, SVR significantly reduced MSEs for all the dimensionality reduction methods. In addition, MLP significantly reduced the errors with RFE. The lack of significance in some of the combinations may be due to either a relative small sample (N=89) or that these combinations may not be completely appropriate. Normally developers would select only one combination for their corresponding ABS.

In order to determine the effect sizes of the reduction of errors, table 4 shows the differences of the MSEs for each combination of methods and the basic estimator. Notice that the improvements of MSEs ranged from 0.238 to 0.481.

This work has measured the execution times for predicting the house prices based on their features with the aforementioned 10-fold cross validation including the automatic calibration of the parameters, after each dimensionality

	Linear Regression	SVR	KNN	MLP	Average
None	32.1	11251.3	285.9	301381.1	78237.6
NMF	10.9	11088.6	292.4	293855.8	76311.9
RFE	7.0	11665.5	288.6	290087.1	75512.0
FS	30.6	11665.5	285.1	293121.2	76275.6
Average	20.2	11417.7	288.0	294611.3	

Table 5 Results of execution times (ms) for the different regression and dimensionality reduction methods

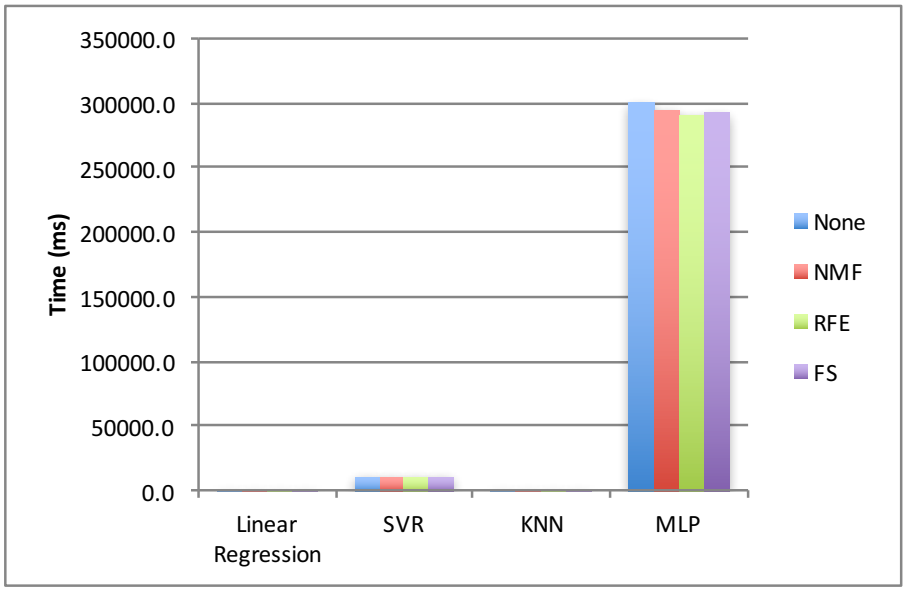

Fig. 7 Comparison of execution times of the different regression methods after applying the corresponding dimensionality reduction methods

reduction had been applied. Table 5 shows these execution times, and figure 7 compares them with a chart.

6 Discussion

Simulations of real-estate transactions have motivated the current work for their opacity in prices. Housing units have many properties, and some of these are difficult to find. To begin with an example, in cold places such as Teruel (used in the experimentation of this work), the existence and the type of heating are really relevant. Notice that the heating type is directly related with the energy consumption and consequently economic consumption. Real-estate websites like Idealista have this feature as optional, so owners decide whether to mention it or not in their advertisement. This makes the assessment of the house hard, especially in an automated way. Sometimes, one can estimate the heating type by looking at the pictures of the housing unit, but this is not always certain and is very hard to automate. Major real-estate websites

include all or almost all the information about the prices of all the housing units. However, not all the housing units are advertised in these major websites. Sometimes, owners just hang up some physical signs about their property offer (Galey, 2005), including a telephone but a lot of information is missing such as the prices. The purpose of owners is generally to receive calls asking for information such as the price. In this way, owners can talk with the people for convincing that the offer is good. Although by calling the owner you can usually get all the information you want, from a macro-level view point, it is really hard to obtain global large measures. Moreover, real-estate agencies sometimes may delay removing cheap advertisements to receive potential calls about these, and to suggest the people to buy/rent similar houses but no so cheap, possibly starting negotiation processes (Urbanavičiene et al, 2009). Thus, the available information may not be accurate for a given time. Furthermore, there is no centralized mechanism for accessing all the information of real-estate market. Old owners usually select local real-estate agencies to advertise their housing units. Each local real-estate agency has its own database, and normally agencies do not share information among each other. There are several major real-estate websites in Spain such as Idealista and FotoCasa[3], and while some housing units are advertised in several websites, other housing units are only available in one of these. In addition, some transactions are performed between familiars and friends, so other people can never know about them. For all these reasons, the real-estate market is opaque and difficult to be automatically processed.

In the current experimentation in Teruel, we used 89 housing units in two different neighbors available in Idealista. Based on the observation of FotoCasa and physical signs, we roughly estimate that these number of houses only represented the 55% of the existing houses. This percentage could be even lower based on the housing units that are completely unknown by the authors. In addition, we estimate that the price was not provided in the 30% of the houses for sale, since these houses were only advertised by physical signs. However, this is our estimation based on the information we know, since given the opacity of this market, it is impossible for us to provide a more accurate percentage. Regarding the features, the existence of lift was properly tagged in 85% of the houses, the existence of heating was tagged in 42% of the houses, the floor number was tagged in 95% of the houses, and the existence of garage was properly tagged in 75% of the houses. However, with the natural-language description and by observing the images, we could extract/estimate all the missing information in these tagging for almost every house. In some cases, we assumed that a house did not have some property as it was neither mentioned in the description nor shown in the pictures.

This work detects a problem that might be relevant in other domains. This problem is to determine an effective and realistic way of initiating the simulations. In fact, most simulators are intended to achieve similar results at the end of the simulations, while others are also interested in the evolution.

[3] https://www.fotocasa.es/es/ (last accessed October 22, 2018)

Some simulators start the initial population with the average value. For these cases, the current work proposes applying some of the existing learning machine techniques to supply some of the unknown values by training from the known values.

Furthermore, this work has considered the reduction of dimensionality, which can be useful in big data contexts with high amounts of simulated data and features. The dimensionality reduction allowed the current approach to reduce information without losing prediction capability considerably. This can be observed when comparing the MSEs of all the regression methods and the datasets after applying each dimensionality reduction method. In addition, both the original data and the reduced datasets allowed most of the regression methods to obtain simulated values that had no significant differences with the real ones.

In the particular context of real-estate market, some combinations of regression and dimensionality reduction methods outperformed the basic mean estimator in terms of similarity between real and simulated values. Concerning regression techniques, SVR showed to be better than the other regression methods, since it outperformed the basic estimator with all the dimensionality reduction methods, table 3. Concerning dimensionality reduction techniques, RFE was the best one since it outperformed the basic mean estimator with three regression methods, being this the maximum number in the current experiments. The lowest MSE and significance level was obtained with the combinations of SVR and RFE.

It is also remarkable that a simple linear predictor can achieve good results for either the original input space or the selected features obtained from RFE. This is not surprising, since RFE selects features based on regression performance, and the linear regressor was used in this process as explained in section 4.2.6. This avoids finding any structural parameters but gives an advantage compared to other algorithms. On the other hand, FS and NMF perform dimensionality reduction only with the information of the input dataset, without relying on a particular regressor.

Despite having attracted a lot of attention in recent years, NMF is not the best reduction technique among these combinations of regressors and dimensionality reduction methods. One difference with previous studies in which it was applied is the dimension of the input space, which is lower in the present study. In (Žibert et al, 2016), (Maruyama et al, 2014) and (Chen et al, 2014), the input dimensionality was at least several hundreds, far larger than in our dataset. Thus, it might be possible that NMF is more suitable for such problems. Besides, RFE and FS differ from NMF in a key aspect. While RFE and FS directly select features, NMF selects vectors in the original space. We have restricted this number to six to get a considerable dimensionality reduction. However, a technique like SVR also predicts new values with a subset of vectors, the support vectors, in this case among those of the training set. In our experiment this number ranged between 50 and 84, depending on the dimensionality reduction strategy and the cross-validation run. Although this number cannot be compared directly to the number of vectors selected in

NMF, there appears to be a large difference in the number of vectors used to predict values.

With respect to execution time, its measurement allowed determining whether the proposed combinations were efficient in terms of response time. The linear regression and KNN obtained high performance in comparison to the others.

Even though in this particular dataset linear regression and SVR both with RFE obtained the lowest average error with dimensionality reduction, this study points out some interesting remarks and positive aspects when applying MLP with NMF. Considering the NMF dimensionality reduction technique, the lowest error was found when combining it with MLP. In addition, MLP is the machine learning technique that had the lowest increasing ratio with respect to the original space (with no dimensionality reduction), as shown in figure 5. Furthermore, as one can observe in figure 6, MLP with NMF was the only combination without outliers. In addition, it had the lowest worst case error. Thus, taking these positive aspects into account, it would be worth investigating exhaustively the application of MLP with NMF to other datasets.

7 Conclusions and future work

The current work has addressed the problem of estimating unknown prices of houses in the initial populations of ABSs. For this purpose, we have proposed combining machine learning and dimensionality reduction methods. This approach outperforms the use of the basic estimator based on simply initiating the individuals with the average price. This work might open a relevant line of research, which is the application of machine learning and dimensionality reduction methods for supporting the initialization of ABSs for estimating the unknown information. The application of dimensionality reduction might be a step forward for applying ABSs with big data.

This initialization of agent population may provide more realistic simulation evolutions from the beginning in different domains. The experimentation results show that dimensionality reduction has allowed managing smaller amounts of information in the training phase providing results that had no significant differences with the real ones. The experiments showed that SVR was the best regression method (outperformed the basic estimator with all the dimensionality reduction techniques), and RFE was the best dimensionality reduction method (outperformed basic estimator with three regression methods, being this the highest number compared to other dimensionality reductions). In addition, the combination of SVR and RFE obtained the lowest MSE. Despite the simplicity of the linear predictor, it obtained especially good results with RFE, since RFE is based on linear regression for eliminating features. NMF did not perform as well as expected, and the reason might be that our approach reduced from 13 to 6 dimensions, and other studies had larger input number of dimensions (i.e. at least several hundreds). The linear regression and KNN obtained the highest performance in terms of response time. MLP achieved the lowest error when considering only the combinations with NMF.

In addition MLP and NMF is the combination that had the lowest increasing ratio in MSE with respect to the case of no dimensionality reduction.

This work could have practical implications for house buyers and sellers. Both stakeholders could use the prediction models to know if they are going to buy or sell a house at a fair price, check for opportunities, identify market tendencies, and so on. In the case considered in this paper (a small town), the model can predict prices with an average MSE of 0.105 for the combination SVR-RFE. In this study, we have also detected the special behavior of NMF and MLP: despite not being the best in terms of average MSE, this combination had no outliers and got the lowest worst case error. Thus, this deserves further research in the future.

The current work is planned to be extended for solving other two problems in ABS initialization. The first problem is to generate an unknown realistic sample of a particular size from an existing one. In the context of real-estate market, it would be necessary to generate both the features and prices of the missing houses from an existing set. The second problem is to obtain a realistic past sample from which only the average price is known. The features of the houses would be generated to be similar to the current ones. The prices of the houses would be calculated by supervised learning from the present houses, but aiming at fitting a different average (the past average price).

In addition, the current approach could be further assessed with higher amounts of houses. It is also planned to be enhanced by taking more house features into consideration, such as their energy class. Furthermore, our future work will apply generalized additive models (GAMs) to this enhanced dataset, to assess their utility in the proposed approach.

Acknowledgements This work has been supported by the program "Estancias de movilidad en el extranjero José Castillejo para jóvenes doctores" funded by the Spanish Ministry of Education, Culture and Sport with reference CAS17/00005. This work also acknowledges the research project "Diseño de actividades de aprendizaje colaborativas con Big Data" with reference PIIDUZ_16_120 funded by University of Zaragoza. We acknowledge the research project "Construcción de un framework para agilizar el desarrollo de aplicaciones móviles en el ámbito de la salud" funded by University of Zaragoza and Foundation Ibercaja with grant reference JIUZ-2017-TEC-03. We also acknowledge support from "Universidad de Zaragoza", "Fundación Bancaria Ibercaja" and "Fundación CAI" in the "Programa Ibercaja-CAI de Estancias de Investigación" with reference IT1/18. This work was partially supported by the Spanish Research grant MTM2015-65433-P (MINECO/FEDER), Gobierno de Aragón, and Fondo Social Europeo. Furthermore, we acknowledge the "Fondo Social Europeo" and the "Departamento de Tecnología y Universidad del Gobierno de Aragón" for their joint support with grant number Ref-T81.

Conflicts of interest

The authors declare that there is not any conflict of interest about this work.

References

Anya O, Moore B, Kieliszewski C, Maglio P, Anderson L (2015) Understanding the practice of discovery in enterprise big data science: An agent-based approach. Procedia Manufacturing 3:882–889

Bárcena Ruiz MJ, Menéndez P, Palacios MB, Tusell Palmer FJ (2011) Measuring the Effect of the Real Estate Bubble: a House Price Index for Bilbao. Biltoki 5463, http://hdl.handle.net/10810/5463 (last accessed July 19, 2017)

Becker T, Illigen C, McKelvey B, Hülsmann M, Windt K (2016) Using an agent-based neural-network computational model to improve product routing in a logistics facility. International Journal of Production Economics 174:156–167

Bishop CM (2006) Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc., Secaucus, NJ, USA

Borges F, Gutierrez-Milla A, Luque E, Suppi R (2017) Care HPS: A high performance simulation tool for parallel and distributed agent-based modeling. Future Generation Computer Systems 68:59–73

Bosch M, Carnero MA, Farré L (2015) Rental housing discrimination and the persistence of ethnic enclaves. SERIEs 6(2):129–152

Brown JM, Phelps JJ, Barkwith A, Hurst MD, Ellis MA, Plater AJ (2016) The effectiveness of beach mega-nourishment, assessed over three management epochs. Journal of Environmental Management 184:400–408

Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2:27:1–27:27, software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm (last accessed July 19, 2017)

Chang CC, Chao CH, Yeh JH (2016) The role of buy-side anchoring bias: Evidence from the real estate market. Pacific-Basin Finance Journal 38:34–58

Chasco Yrigoyen C, Le Gallo J (2012) Hierarchy and spatial autocorrelation effects in hedonic models. Economics Bulletin 32(2):1474–1480

Chen J, Feng S, Liu J (2014) Topic sense induction from social tags based on non-negative matrix factorization. Information Sciences 280:16–25

Chiarazzo V, Caggiani L, Marinelli M, Ottomanelli M (2014) A neural network based model for real estate price estimation considering environmental quality of property location. Transportation Research Procedia 3:810 – 817, DOI https://doi.org/10.1016/j.trpro.2014.10.067, URL http://www.sciencedirect.com/science/article/pii/S2352146514002300, 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain

Chung H, Badeau R, Plourde E, Champagne B (2018) Training and compensation of class-conditioned nmf bases for speech enhancement. Neurocomputing 284:107–118

Cicirelli F, Furfaro A, Giordano A, Nigro L (2011) HLA_ACTOR_REPAST: An approach to distributing RePast models for high-performance simulations. Simulation Modelling Practice and Theory 19(1):283–300

Cui G, Zhuang G, Lu J (2016) Neural-network-based distributed adaptive synchronization for nonlinear multi-agent systems in pure-feedback form. Neurocomputing 218:234–241

Davidsson P (2002) Agent based social simulation: A computer science view. Journal of artificial societies and social simulation 5(1)

Dismuke C, Lindrooth R (2006) Ordinary least squares. In: Chumney E, Simpson N K (eds) Methods and Designs for Outcomes Research, American Society of Health-System Pharmacists: Bethesda, MD, pp 93–104

Duda RO, Hart PE, Stork DG (2012) Pattern classification. John Wiley & Sons

Faul F, Erdfelder E, Lang AG, Buchner A (2007) G* power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behavior research methods 39(2):175–191

Galey M (2005) System and method of online real estate listing and advertisement. US Patent App. 10/896,331

García N, Gámez M, Alfaro E (2008) Ann+gis: An automated system for property valuation. Neurocomputing 71(4):733 – 742, DOI https://doi.org/10.1016/j.neucom.2007.07.031, URL http://www.sciencedirect.com/science/article/pii/S0925231207003505, neural Networks: Algorithms and Applications 50 Years of Artificial Intelligence: a Neuronal Approach

García M (2010) The breakdown of the spanish urban growth model: Social and territorial effects of the global crisis. International Journal of Urban and Regional Research 34(4):967–980

García-Magariño I, Lacuesta R (2017) Agent-based simulation of real-estate transactions. Journal of Computational Science 21:60–76

García-Magariño I, Plaza I (2017) ABS-MindHeart: An agent based simulator of the influence of mindfulness programs on heart rate variability. Journal of Computational Science 19:11–20

García-Magariño I, Gómez-Rodríguez A, González-Moreno JC, Palacios-Navarro G (2015) PEABS: a process for developing efficient agent-based simulators. Engineering Applications of Artificial Intelligence 46:104–112

García-Magariño I, Medrano C, Delgado J (2017) Python code for the estimation of missing prices in real-estate market with a dataset of house prices from Teruel city. Mendeley Data, v2 http://dx.doi.org/10.17632/mxpgf54czz.2

Gilbert N, Terna P (2000) How to build and use agent-based models in social science. Mind & Society 1(1):57–72

Gómez-Sanz JJ, Fernández CR, Arroyo J (2010) Model driven development and simulations with the INGENIAS agent framework. Simulation Modelling Practice and Theory 18(10):1468–1482

Hassan S, Garmendia L, Pavón J (2010) Introducing uncertainty into social simulation: using fuzzy logic for agent-based modelling. International Journal of Reasoning-based Intelligent Systems 2(2):118–124

Houari R, Bounceur A, Kechadi MT, Tari AK, Euler R (2016) Dimensionality reduction in data mining: A copula approach. Expert Systems with

Applications 64:247–260
Jalalimanesh A, Haghighi HS, Ahmadi A, Soltani M (2017) Simulation-based optimization of radiotherapy: Agent-based modeling and reinforcement learning. Mathematics and Computers in Simulation 133:235–248
Jayaram D, Manrai AK, Manrai LA (2015) Effective use of marketing technology in Eastern Europe: Web analytics, social media, customer analytics, digital campaigns and mobile applications. Journal of Economics, Finance and Administrative Science 20(39):118–132
Jiang GM, Hu ZP, Jin JY (2007) Quantitative evaluation of real estate's risk based on AHP and simulation. Systems Engineering-Theory & Practice 27(9):77–81
Khalil KM, Abdel-Aziz M, Nazmy TT, Salem ABM (2015) MLIMAS: A Framework for Machine Learning in Interactive Multi-Agent Systems. Procedia Computer Science 65:827–835
Lee D, Seung H (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791
Lee D, Seung H (2001) Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems 13:556–562
Li ZX (2006) Using fuzzy neural network in real estate prices prediction. In: 2007 Chinese Control Conference, pp 399–402, DOI 10.1109/CHICC.2006.4347291
Maltamo M, Kangas A (1998) Methods based on k-nearest neighbor regression in the prediction of basal area diameter distribution. Canadian Journal of Forest Research 28(8):1107–1115
Maruyama R, Maeda K, Moroda H, Kato I, Inoue M, Miyakawa H, Aonishi T (2014) Detecting cells using non-negative matrix factorization on calcium imaging data. Neural Networks 55:11–19
Nguyen N, Cripps A (2001) Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks. Journal of Real Estate Research 22(3):313–336, URL https://ideas.repec.org/a/jre/issued/v22n32001p313-336.html
North MJ, Collier NT, Ozik J, Tatara ER, Macal CM, Bragen M, Sydelko P (2013) Complex adaptive systems modeling with Repast Simphony. Complex adaptive systems modeling 1(1):1
Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5:111–126
Park B, Bae JK (2015) Using machine learning algorithms for housing price prediction: The case of fairfax county, virginia housing data. Expert Systems with Applications 42(6):2928 – 2934, DOI https://doi.org/10.1016/j.eswa.2014.11.040, URL http://www.sciencedirect.com/science/article/pii/S0957417414007325
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al (2011) Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12(Oct):2825–2830

Provost F, Fawcett T (2013) Data science and its relationship to big data and data-driven decision making. Big Data 1(1):51–59
Pyhrr SA (1973) A computer simulation model to measure the risk in real estate investment. Real Estate Economics 1(1):48–78
Reiser L, Mueller LA, Rhee SY (2002) Surviving in a sea of data: a survey of plant genome data resources and issues in building data management systems. In: Functional Genomics, Springer, pp 59–74
Sabarina K, Priya N (2015) Lowering data dimensionality in big data for the benefit of precision agriculture. Procedia Computer Science 48:548–554
Simovici D (2012) Linear Algebra Tools for Data Mining. World Scientific Publishing
Sun Y, Wen G (2017) Cognitive facial expression recognition with constrained dimensionality reduction. Neurocomputing 230:397–408
Symeonidis S, Effrosynidis D, Arampatzis A (2018) A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. Expert Systems with Applications 110:298–310
Tratalos J, Haines-Young R, Potschin M, Fish R, Church A (2016) Cultural ecosystem services in the uk: lessons on designing indicators to inform management and policy. Ecological Indicators 61:63–73
Urbanavičiene V, Kaklauskas A, Zavadskas EK (2009) The conceptual model of construction and real estate negotiation. International Journal of Strategic Property Management 13(1):53–70
Wang R, Hou J, He X (2017) Real estate price and heterogeneous investment behavior in China. Economic Modelling 60:271–280
Wang S, Wan J, Zhang D, Li D, Zhang C (2016) Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. Computer Networks 101:158–168
Wojtusiak J, Warden T, Herzog O (2012) Machine learning in agent-based stochastic simulation: Inferential theory and evaluation in transportation logistics. Computers & Mathematics with Applications 64(12):3658–3665
Yaqoob I, Hashem IAT, Gani A, Mokhtar S, Ahmed E, Anuar NB, Vasilakos AV (2016) Big data: From beginning to future. International Journal of Information Management 36(6):1231–1247
Zhang L, Wang Z, Sagotsky JA, Deisboeck TS (2009) Multiscale agent-based cancer modeling. Journal of mathematical biology 58(4-5):545–559
Zhuge C, Shao C, Gao J, Dong C, Zhang H (2016) Agent-based joint model of residential location choice and real estate price for land use and transport model. Computers, Environment and Urban Systems 57:93–105
Žibert J, Cedilnik J, Pražnikar J (2016) Particulate matter (pm10) patterns in europe: An exploratory data analysis using non-negative matrix factorization. Atmospheric Environment 132:217–228