INTERNATIONAL REAL ESTATE REVIEW

2021 Vol. 24 No. 2: pp. 139 – 183

The Predictability of House Prices: "Human Against Machine"

Kristoffer B. Birkeland
NTNU Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology (NTNU), Gløshaugen, NO-7491 Trondheim, Norway. Email: kristoffer.birkeland@ntnu.no.

Allan D. D'Silva
NTNU Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology (NTNU), Gløshaugen, NO-7491 Trondheim, Norway. Email: allan.dsilva@ntnu.no.

Roland Füss
Swiss Institute of Banking and Finance (s/bf), University of St.Gallen, Unterer Graben 21, CH-9000 St.Gallen, Switzerland, Research Fellow at Center for Real Estate and Environmental Economics, NTNU Business School, Trondheim, Norway, and Research Associate at the Centre for European Economic Research (ZEW), Mannheim, Germany. Email: roland.fuess@unisg.ch.

Are Oust*
NTNU Business School, Norwegian University of Science and Technology (NTNU), Gløshaugen, NO-7491 Trondheim, Norway. Email: are.oust@ntnu.no.

We develop an automated valuation model (AVM) for the residential real estate market by leveraging stacked generalization and a comparable market analysis. Specifically, we combine four novel ensemble learning methods with a repeat sales method and tailor the data selection for each value estimate. We calibrate and evaluate the model for the residential real estate market in Oslo by producing out-of-sample estimates for the value of 1,979 dwellings sold in the first quarter of 2018. Our novel approach of using stacked generalization achieves a median absolute percentage error of 5.4%, and more than 96% of the dwellings are estimated within 20% of their actual sales price. A comparison of the valuation accuracy of our AVM to that of the local estate agents in Oslo generally demonstrates its viability as a valuation tool. However, in stable market phases, the machine falls short of human capability.

* Corresponding author

Keywords

AVMs; Housing Market; Machine Learning; Repeat Sales Approach; XGBoost.

1. Introduction

This paper develops an automated valuation model (AVM) that will leverage both historical transaction and attribute-specific data for real estate property valuation. First, our novel AVM approach combines the well-established repeat sales method (RSM) of Case and Shiller (1987) with four machine learning methods based on stacked generalization (Wolpert, 1992). Second, the study takes advantage of a comparable market analysis (Rattermann, 2007) to value dwellings based on transactions with close spatial and temporal proximities.

Hence, our empirical study could be situated in the research intersection among AVMs in real estate valuation, ensemble learning in the field of econometrics, and index construction methodology in the real estate markets. In the latter two research areas, we find extensive literature over the past five decades. Ensemble learning methods by Breiman (1996a) and Schapire (1989) have been successfully applied in a relatively few but growing number of econometric works (Graczyk et al. 2010; Inoue and Kilian, 2008).[1] The research work on the construction of real estate indices has had few improvements since Bailey et al. (1963) and Case and Shiller (1987). Construction methodologies include the use of median house price indices (e.g., Crone and Voith, 1992; Gatzlaff and Ling, 1994), hedonic pricing models (HPMs) (e.g., Balk et al., 2013; Geltner, 2015; Fisher et al., 1994), RSM (e.g., Calhoun, 1996), and hybrid models (e.g., Quigley, 1995; Meese and Wallace, 1997).

The four machine learning methods, which we combine with the RSM, are known as ensemble learning methods. Ensemble learning is a class of modern machine learning methods that combines multiple models into one model to increase its out-of-sample predictive power (Opitz and Maclin, 1999). The four sub-models are derived from two classes of ensemble learning techniques: bagging (Breiman, 1996a) and boosting (Schapire, 2013; Freund and Shapire, 1997). The use of these ensemble learning methods allows our model to learn patterns in the underlying data without making any assumptions about the data generation process. We hypothesize that, due to the amount of available data and model complexity, these ensemble learning methods are suitable for use in the residential real estate markets.

Over the last two decades, machine learning has found its way into the real estate sector. It has been proven as a tool for mass appraisal and tested for

[1] We particularly recommend Mullainathan and Spiess (2017) as an excellent overview of machine learning applications in econometrics.

different residential and commercial real estate markets. For instance, Antipov and Pokryshevskaya (2002) show that to predict prices in the residential apartment market of Saint-Petersburg, random forest (RF) is superior to techniques such as Chi-square automatic interaction detectors (CHAIDs), classification and regression trees (CARTs), multiple regression analyses, artificial neural networks (ANNs), and boosted trees. Peterson and Flanagan (2009) and Ma et al. (2015) demonstrate that ANNs outperform linear hedonic regressions in terms of pricing errors, out-of-sample pricing precision, and extrapolation of volatile pricing environments. Chiarazzo et al. (2014) use ANNs to illustrate how accessibility, land-use, and environmental quality variables improve the appraisal of home sales prices in the city of Taranto in Italy. Barr et al. (2017) construct home price indices based on a gradient boosted model. They show that their approach is superior to the traditional median sale prices or repeat sales indices. Manganelli et al. (2015) and Park and Bae (2015) demonstrate the effectiveness of modeling the relationship between the price and location of homes as well as between closing and listing prices based on genetic and classification algorithms, respectively. Furthermore, Kok et al. (2017) conclude on the superiority of AVMs over traditional appraisal approaches for the commercial real estate sector. Thereby, they emphasize the importance of location information.

We contribute to the recent literature on AVMs by replicating the behavior of estate agents who often use recent sales of comparable dwellings as well as earlier transaction prices of the same property as a starting point for valuation. More precisely, we propose a novel approach by combining ensemble learning methods with an RSM in an AVM. In addition to the prediction, the number of previous dwelling sales is included in the training data for the model stacking process. As frequently sold dwellings might have distinctive characteristics, this attribute is also likely to yield significant explanatory power. The RSM is trained on a separate dataset and aims to capture different market movements and implicitly proxy locational characteristics compared to the ensemble learning methods.

The four ensemble learning methods—bagging predictor (BP), RF, extra trees (ETs), and XGBoost (XGB)—are used to generate four independent value estimates. In addition, we leverage XGB as a stacking method. We carry out model stacking by using XGB as a meta-estimator with both exogenous indicators and value estimates from the individual models as the input variables. As a powerful stacking method, XGB can determine the performance of each base model and combine the underlying predictions accordingly. In our case, when both an array of ensemble learning methods and an RSM are selected, i.e., the underlying models are diverse, then stacking is highly effective. We employ a Norwegian dataset with 18,401 transactions in Oslo between August 2016 and April 2018. In the Norwegian real estate markets, existing residential dwellings are sold through English auctions. This market setting provides an exclusive opportunity to test AVMs and compare the price predictions with those of real estate agents. Norwegian real estate agents are not allowed to advertise

dwellings with a teaser price, i.e., an asking price lower than the reservation price of the seller. Hence, real estate agents are incentivized to set their asking price close to the expected market price. Our data also reflect this arm's length valuation behavior and thus, challenges our machine learning approach more than in other markets such as over-the-counter (OTC) markets or organized exchange markets.

Compared to commercial real estate, cash flow is absent in the housing markets and appraisals play a dominant role when valuing owner-occupied homes. Based on the sales comparison approach, appraisers go back in time in order to find comparable properties for the valuation. Since appraisals often rely on previous prices, this leads to the delayed recording of the market movement (see, e.g., McAllister and Tarbert, 1998). Matysiak and Wang (1995) find that the accuracy of appraisals vary in dynamic markets with an overestimation (underestimation) of prices when markets are falling (rising). The main reasons are that appraisers suffer from anchoring bias (see, e.g., Clayton et al., 2001) and partly adjust their estimate in response to new information (Quan and Quigley, 1991), as well as gradually incorporating non-transaction based information into their valuation. We therefore hypothesize that real estate agents perform in a similar way as machine learning approaches to predict house prices during quiet periods, but deteriorate in performance in more dynamic markets.

The AVM developed in this study shows several encouraging results. We evaluate our model by producing out-of-sample estimates for the 1,979 dwellings sold in Oslo in the first quarter of 2018. Our AVM values these dwellings with a median absolute percentage error (MdAPE) of 5.4%, with more than 96% of the dwellings being estimated within 20% of their actual sales price. We compare the results of the model to those obtained by traditional AVMs, estate agents, and the leading U.S. online real estate website, Zillow for commercial AVMs. The performance of our AVM is comparable to the accuracy of Norwegian estate agents, which, in Oslo, has been at an MdAPE of 5.3% over the past two years. Our AVM is also superior to the precision of Zillow for the selection of cities, for which official performance statistics are available. Specifically, when we compare the valuation accuracy of our AVM to that of local estate agents, the out-of-sample results are similar in accuracy as the estate agents in the training period of our model. This period includes a very dynamic development of prices in the Oslo housing market. However, when we directly compare the results for the first quarter of 2018, which consists of mostly stable house prices, the machine falls short of human capability.

The remainder of the paper is structured as follows: Section 2 describes our data from the residential real estate market in Oslo and provides an introduction on the data pre-processing steps in our AVM. In Section 3, we develop the technical framework of our AVM. Section 4 presents the results of our model and evaluates its performance. Section 5 summarizes the performance of our AVM and concludes.

2. Data

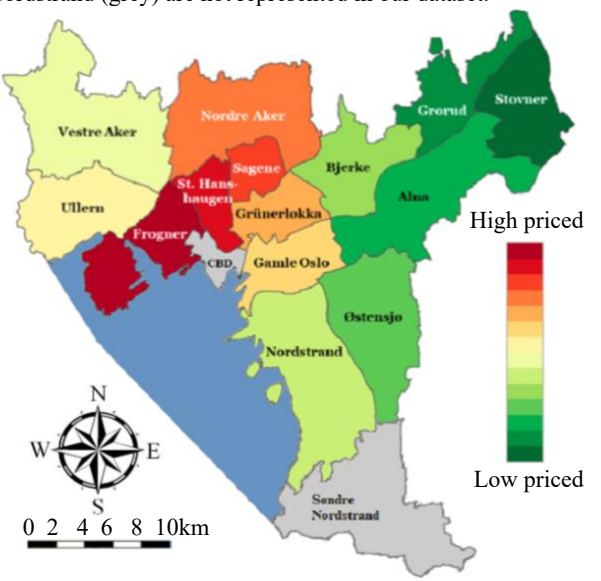
Real estate valuation models are known to be highly dependent on local conditions (Tsatsaronis and Zhou, 2004). Therefore, we seek to identify the factors that affect sales prices in the residential real estate market of Oslo. The identification of these factors allows us to adapt our AVM to the market conditions of Oslo.

2.1 The Residential Market of Oslo

The residential real estate market in Norway is characterized by a strong tradition of homeownership, with 84% of Norwegians living in a self-owned home (Eurostat, 2015). Oslo had a population of 673,468 in January 2018 (Statistics Norway, 2018). The city has experienced significant growth over the past few decades, and metropolitan Oslo has contributed to roughly 50% of the population growth of Norway (Statistics Norway, 2010). The dwellings located in the central parts of Oslo are typically characterized by four- and five-story brick apartment buildings. Historically, the western parts of Oslo have generally had larger, more expensive houses, while eastern parts have had smaller, less expensive apartments. As illustrated in Figure 1, Oslo is divided into 15 districts and the city center.

Figure 1 Districts in Oslo

Administrative districts of Oslo with price/m^2 ranking for 2017 (Oust et al., 2020). CBD and Søndre Nordstrand (grey) are not represented in our dataset.

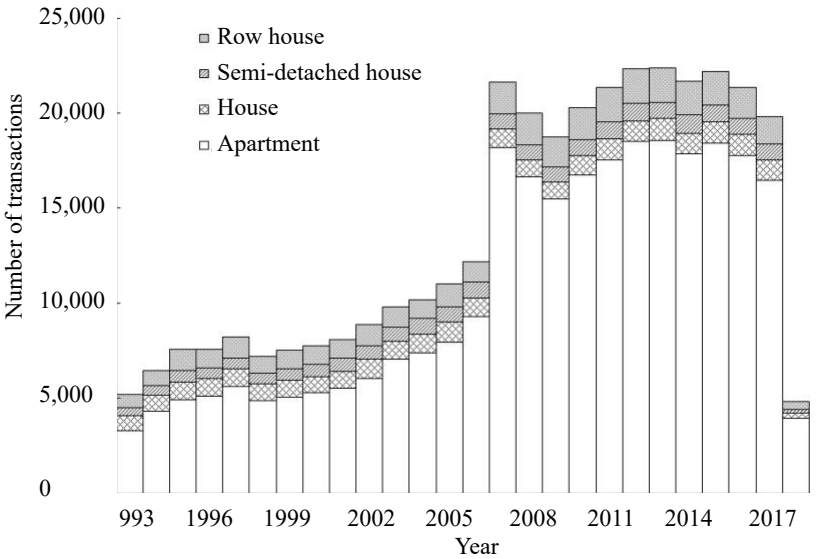


2.2 Merged Datasets

Our primary datasets are consolidated from two official land registers, the *Grunion* and *Matrikkelen*, and extended by using proprietary data.[2] *Alva Technologies*, a Norwegian real estate IT and computer services company, has provided us with three datasets: (i) the *address dataset* which consists of all dwellings in Norway, among which 276,780 are located in Oslo. The data contain the variables listed in Table 1; (ii) the *enhanced transaction dataset* which covers 18,401 transactions in Oslo between August 2016 and April 2018. This proprietary data takes into account improved quality and additional variables collected from the sale advertisements of dwellings; and (iii) the *historical transaction dataset* which includes all registered residential real estate transactions in Oslo between January 1993 and May 2018 as plotted in Figure 2. In total, 220,898 separate transactions are mapped to Oslo addresses. The data include a unique address identifier, the sales price, and sale date, but not common debt and usable square meters (USMs).

Figure 2 Number of Transactions

All recorded transactions by year of transaction in the dataset for 1993 to 2018. Sales from cooperatives added around 2007, which causes a sharp increase in the number of transactions. Note small bar for 2018 as the data only cover the period until May.



[2] *Matrikkelen* and *Grunnboken* (The Norwegian Mapping Authority) constitute the real estate property and ownership relations of the Norwegian real estate market. Information given on the dwellings listed in the *Matrikkelen* include their location and boundaries, size, property type and, in the case of apartments, the building in which they are located. *Grunnboken* describes the ownership relationships for both private properties and cooperatives.

Table 1 lists the attributes that are gathered from these datasets and used by the ensemble learning methods in Section 3.2. Both the geographical coordinates and districts are used to pinpoint the dwelling location. The size of an individual dwelling is defined by its USMs and the number of rooms. Dwellings in Norway are typically categorized as one of four main unit types: *apartments*, and *row*, *semi-detached*, and *detached houses*. While apartments are part of a larger building complex, the other three unit types denote variations of dwellings that are single-family homes. In the ensemble learning methods, we include unit type as a categorical variable by differentiating between apartments and non-apartments (i.e., row, semi-detached, or detached houses) due to their specific characteristics discussed in Section 2.4.[3]

Table 1 Overview of Exogenous Variables

Overview of the exogenous variables used by the attribute-based pricing methods. The attributes are classified as numeric and categorical variables.

Variable	Type	Description
USMs	Numeric	USMs of dwelling.
Log(USMs)	Numeric	Natural logarithm of USMs.
Coordinates	Numeric	Geographical coordinates of dwelling.
Story	Numeric	Story on which dwelling is located.
# of rooms	Numeric	Number of rooms in dwelling.
# of days since sale	Numeric	Number of days since occurrence of transaction.
Rank	Numeric	A measure (increasing with distance) of proximity to target dwelling.
Built year	Numeric	Construction year of dwelling.
Common debt	Numeric	Part of the debt of dwelling held by the group of properties.
Sold month	Categorical	Year and month of occurrence of transaction.
District	Categorical	District in which dwelling is located.
Unit type	Categorical	Unit type of dwelling.
Build type	Categorical	Type of the dwelling as per NS3457 (2013)
Ownership type	Categorical	Ownership type of dwelling; see Section 2.1.
Elevator	Categorical	Whether building of dwelling has elevator.

[3] The correlation structure among the different housing attributes ranges between -0.11 and +0.15, and thus, shows no clear linear relationships between the attributes in our dataset. The only exception is the positive correlation of 0.51 between *USM* and number of rooms, which is significant at the 1% level.

In addition to the datasets provided by *Alva Technologies*, we use a separate dataset to analyze the aggregate valuation precision of real estate agents and utilize this precision as a benchmark for our model. This dataset is obtained from *Finn.no*[4] and contains 15,786 transactions in Oslo from 2016 and 2017 and 3,009 transactions in Oslo in the first quarter of 2018. The data include both *asking prices* provided by estate agents and final *sales prices*.

2.3 Data Pre-Processing

Our sample shows missing data for some of the characteristics. For instance, the values are missing in roughly 30% of the records in the sample, such as the *number of rooms*. The explanatory variables *story* and *built year* have 3,191 and 755 missing values, respectively, out of a total of 18,073 observations.[5] In addition, the *historical transaction dataset* contains erroneous data due to the occurrence of sales under particular circumstances, inadequate data management, and imprecise matching of data from multiple sources.

To address the missing data, we remove all dwellings for which we do not have data on the *district*. By doing so, we simultaneously remove all data points with missing values for the *elevator*, *unit type*, *build type*, and *coordinates*. In total, this affects less than 1% of the dwellings. Next, we impute values for missing data points with the mean value of all remaining dwellings for the variables *story*, *built year*, and the *number of rooms*.

We further clean the *historical transaction dataset*. First, we remove transactions with a sales price below 100,000 NOK (12,035 USD) or above 70 million NOK (8,420,546 USD)[6], which account for less than 1% of the transactions. We then eliminate transactions in which the same property is sold twice within three months. These are most likely distressed sales or speculative transactions, and therefore, do not reflect the real appreciation or depreciation in that given period (Jansen et al., 2008). We also remove transactions where the ratio of the sales prices between two transactions is larger than five. Finally, we exclude dwellings that have more than ten previous transactions within the recorded time period. Such a high frequency of re-selling is unlikely, and the dwellings will typically not be representative as argued by Case and Shiller (1987). We note that our AVM is evaluated on a test set with transactions solely taken from the enhanced transaction dataset. Therefore, removing transactions from the historic transaction dataset will not bias the selection of our test set.

[4] Finn.no is an online Norwegian advertisement firm used by all of the real estate agent companies in Norway.
[5] For the following variables, the number of missing values (count) are: *District* (107), *Elevator* (78), *Unit Type* (77), *Build Type* (77), *Coordinates* (77), and *Common Debt* (3).
[6] Exchange rate NOK to USD per June 1, 2021: 1 USD = 8.313 NOK

2.4 Exploratory Data Analysis

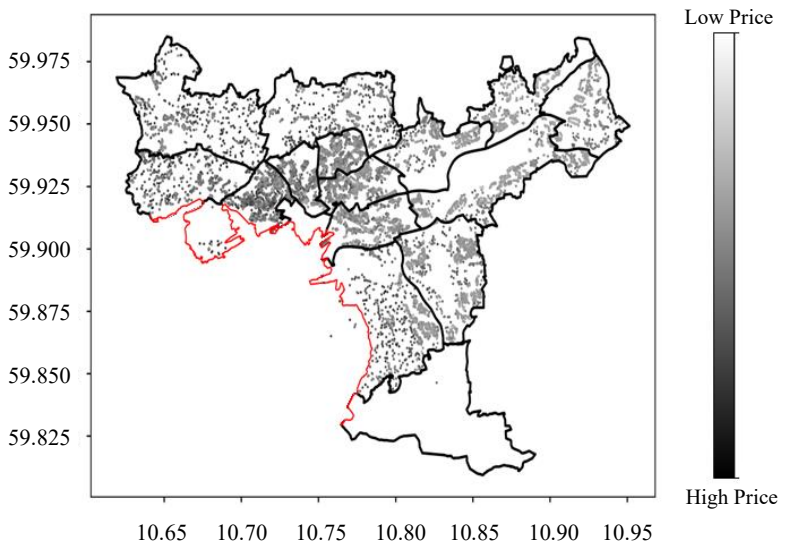
This section provides the summary statistics on our primary variables *location*, *sales price*, *common debt*, and *USMs* for the different units and ownership types. We provide a detailed report on the historical transactions as well as the spatial and temporal distributions of the most recent sales in Oslo.

Figure 3 shows the price per square meter (PPSM) for all of the transactions in the enhanced transaction dataset. Panel A of Figure 3 highlights the spatial variation in the prices, which motivates the use of districts as an explanatory variable (the administrative districts follow the historical east/west borders as mentioned in Section 2.1). There are also sizeable local price variations within the districts. Panel B of Figure 3 shows a high variation in the PPSM in the Nordstrand district, which is in proximity to the coastal line and areas with large high-rise buildings. Therefore, we include the geographical coordinates of a dwelling as a finer-grained measure of location.

Figure 3 Spatial Distribution of Total Transactions

Figure 3a shows all recorded transactions in Oslo from 2016-2018, which covers a total of 18,073 transactions. Panel B shows all recorded transactions in the district of Nordstrand from 2016-2018, which results in a total of 1,167 transactions. A darker color represents a *higher PPSM*. The black bold lines represent the district borders, while the red line denotes the coastline. *x*- and *y*-axes are longitudinal and latitudinal coordinates. The recorded transactions outside the coastline are transactions on the islands.

a) All Recorded Transactions in Oslo



b) All Recorded Transactions in the District of Nordstrand

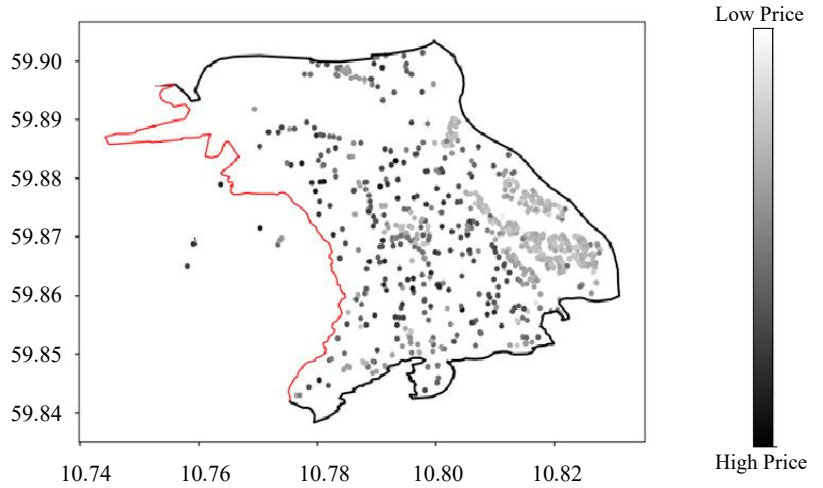


Table 2 presents the descriptive statistics for the ownership types that are discussed in Section 2.1 which are derived from the enhanced transaction dataset. The parameters indicate significant differences in the amount of common debt for the two ownership types. While the median-sized dwellings in the two ownership types are almost identical in size, the median amount of *common debt* for condominiums is less than 3% of that of cooperatives. The low level of common debt for condominiums implies that the historical transaction dataset can be used without concern of the lack of common debt data. Noticeable differences also exist in the *sales price.* These variations might be due to factors other than the ownership type (e.g., differences in location, built year, etc.).

We have registered transactions on 185,961 of the dwellings in our dataset, of which 100,268 are sold at least twice. We define two consecutive transactions of a dwelling as a *repeat sale*. Figure 2 illustrates the number of dwellings sold each year. We note that we do not have data on the sales of *cooperatives* for the period of 1993 to 2004. Also, in the period of 2005 to 2006, only a small number of cooperative sales are registered. Hence, we observe a spike in Figure 2 in 2007, when sales from cooperatives are added, and at that point, an increasing number of transactions are observed each year. During the last ten years, the number of transactions in Oslo has doubled. Similarly, the large total number of repeat sales indicates that previous sales contain useful information for the valuation of dwellings.

Table 2 Descriptive Statistics for Cooperatives and Condominiums

Descriptive statistics on *USMs*, *sales price* in millions of NOK (MNOK), and *common debt* in thousands of NOK (kNOK) for 4,089 cooperatives and 3,076 condominiums based on the enhanced transaction dataset. Exchange rate NOK to USD per June 1, 2021: 1 USD = 8.313 NOK

	Cooperative			Condominum		
	USM	Sales Price (MNOK)	Common Debt (kNOK)	USM	Sales Price (MNOK)	Common Debt (kNOK)
Mean	58	3.7	230	67	5.1	41
St.Dev.	18	1.0	318	27	2.0	72
Min	14	1.3	0	16	1.1	0
25%	46	3.0	76	49	3.6	0
50%	59	3.4	141	60	4.5	4
75%	69	4.0	228	82	6.1	58
Max	183	12.9	2,900	339	18.0	1,200

The dwellings of the four types of units (apartments, and row, semi-detached and detached houses) differ in both size and sales price. Table 3 lists these differences for all of the recorded transactions from 2016 to 2018. An important aspect here is that roughly 90% of the transactions are from apartments, while only 10% of the transactions are from the non-apartments.

Table 3 Descriptive Statistics on Unit Type

Descriptive statistics on the size, measured by *USMs*, and the *sales price* of different dwelling types based on enhanced transaction dataset. Both measures are provided for apartments and non-apartments (row, semi-detached and detached houses) denoted in millions of NOK (MNOK). Exchange rate NOK to USD per June 1, 2021: 1 USD = 8.313 NOK. We show the parameters for mean, standard deviation (St.Dev.), median, as well as 1st and 2nd quartiles.

	Apartment		Row House		Semi-detached House		Detached House	
	USMs	Sales Price (MNOK)	USMs	Sales Price (MNOK)	USMs	Sales Price (MNOK)	USMs	Sales Price (MNOK)
Mean	63	4.2	137	6.3	137	8.5	192	11.6
St.Dev.	24	1.9	45	2.9	45	2.9	58	4.0
25%	48	3.0	105	4.3	105	6.2	156	9.0
50%	62	3.6	135	5.7	135	8.5	187	11.2
75%	73	4.8	128	7.9	165	10.4	222	13.6

Table 3 shows that apartments are much smaller in size compared to the non-apartments. For instance, the 75th percentile USMs for apartments is smaller compared to the 25th percentile USMs for row houses. Semi-detached houses

are more expensive than row houses, and detached houses are by far the largest and most expensive. The median-sized house is more than three times the size of the median-sized apartment. We observe smaller differences between the categories in *sales price* than in *USMs*, thus indicating that there might be a decreasing marginal sales price for a larger dwelling size.

The dynamics of the housing market often cause large short-term fluctuations in real estate prices. Therefore, the precise date of a transaction is critical when modeling these prices. The *sold date* is the date when the buyer and seller agree upon a sales price, whereas the *official date* is the date when the dwelling is handed over to the buyer. In our data, we find the average difference between the sold and official dates to be 46 days. We argue that the sold date is the most interesting when modeling house prices as it indicates the time at which the sales price was determined. All of the transactions in the enhanced transaction dataset have a recorded sold date. For some of the historical transactions, only the official registration date is available. Therefore, for these observations, we use the official date as a proxy for the sold date.

3. Methodology

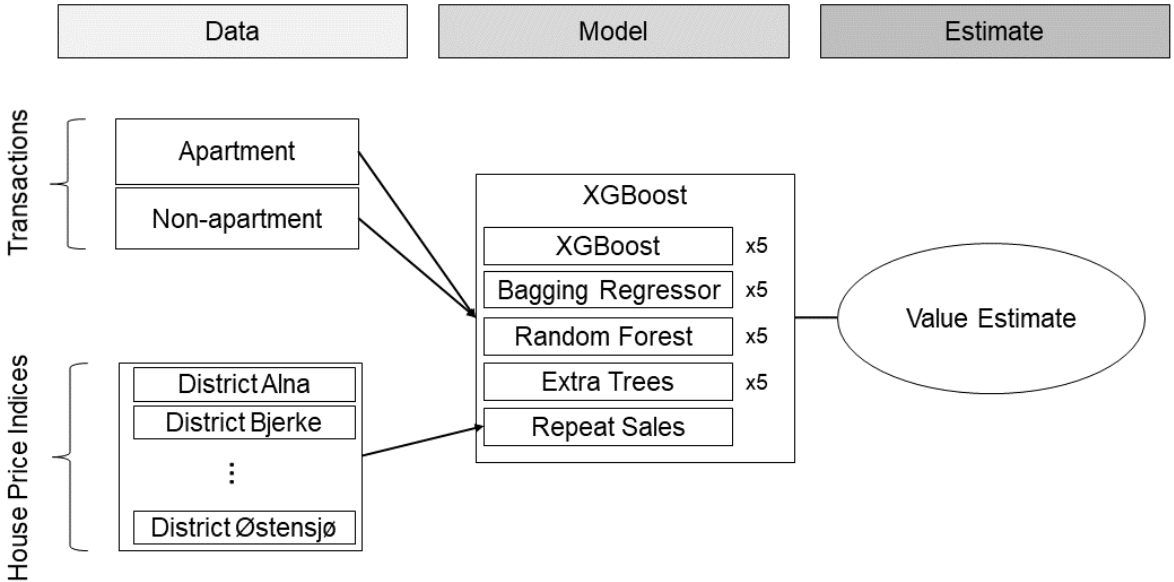
In this section, we develop our AVM. First, we provide an introduction on the two key concepts of our AVM – *stacked generalization* and *comparable market analysis*. We then describe the attribute-based pricing methods as the most prominent value estimation technique for AVMs. The four recently developed and widely-used machine learning methods aim to circumvent some of the shortcomings of traditional parametric attribute-based pricing methods. Furthermore, we describe the RSM and its application to real estate valuation. Finally, we present our AVM as a combination of the concepts mentioned above.

3.1 Stacked Generalization Scheme and Comparable Market Analysis

When producing a value estimate for a dwelling, our AVM uses multiple underlying *sub-models* to create individual value estimates for every dwelling in the training data. Subsequently, a *stacked model* as a separate model analyzes the individual value estimates *in-sample* to determine the *out-of-sample* prediction for the dwelling. This technique is referred to as stacked generalization, which is pioneered by Wolpert (1992) and refined by Breiman (1996b). The model stacking process uses the training data and individual predictions to identify and reduce the biases made by the underlying algorithms when predicting out-of-sample. The stacking procedure is outlined in Figure 4 and will be detailed in Section 3.4.

Figure 4 Automated Valuation Model

AVM is trained for each produced individual value estimate. The model extracts n comparable transactions, where n = 10,000 if the dwelling is an apartment; otherwise n = 2,000. The RSM requires pre-processed indices based on the historical transaction dataset, while the other four ensemble learning algorithms are trained on five folds of the training data. The XGB-algorithm combines the underlying models to produce one value estimate.



Our AVM tailors the *training data* for each value estimate based on the dwelling that is to be valued and fits a unique model to each particular dwelling. The training data are selected to model the dwelling as closely as possible. This concept, known as comparative market analysis, is a prevalent valuation principle often used in real estate valuation (Rattermann, 2007). Specifically, estate agents use recent sales nearby as a starting point when valuing a dwelling. AVMs can replicate this behavior by tailoring their source data to include transactions of dwellings in close geographical proximity to the dwelling in question. This fundamental concept of our model is described in Section 3.4. Moreover, in a recently conducted meta-analysis on the results of 70 papers, Valier (2020) compares advanced machine learning techniques to traditional linear regression methods. The study finds 57 cases that support the hypothesis that machine learning methods provide accurate predictions of dwelling values, while in 13 cases, the study finds support for the superiority of linear regression modeling.

3.2 Attribute-Based Pricing Methods

Traditionally, HPMs have been prevalently used in the academic literature for residential real estate valuations (Balk et al., 2013). HPMs build on the assumption that goods are typically sold as a bundle of inherent attributes and their implicit price can be estimated from the observed prices of the characteristics associated with them (Rosen, 1974). Using these implicit attribute prices, one can predict the sales price of a dwelling from the value of its underlying attributes. However, due to the potential existence of multicollinearity, non-linearity, and omitted variable bias, traditional HPMs may suffer from model misspecification (Wheeler and Tiefelsdorf, 2005; Balk et al., 2013).

We propose an alternative approach, where four ensemble learning methods are used to generate four independent value estimates. Each method creates multiple sub-models, fitted to independently sampled input data. When predicting the value of an unseen dwelling, the methods combine the prediction of each sub-model to determine the new sales price. We apply the four widely used methods, BP, RF, ETs[7], and XGB. Each of the four ensemble methods builds multiple sub-models known as *decision trees*, which combine the prediction of each tree to produce one value estimate. The number of trees in each ensemble model and the rules for building each tree are critical for the success of the ensemble method.

[7] Extra trees is actually an acronym derived from **ext**remely **ra**ndomized trees.

3.3 Ensemble Learning - Key Concepts

In this section, we provide a brief introduction on the underlying machine learning concepts that the ensemble methods rely upon, namely decision trees and bootstrapping. In addition, we explain how these methods are adjusted to the datasets through hyperparameter tuning. In the following sections, we present our selected ensemble learning methods.

A *decision tree* is a simple but powerful tool for prediction modeling (Lior and Maimon, 2015). Informally, a decision tree has a tree-structure that resembles a flowchart as illustrated in Figure 5, where each internal node denotes a test of an attribute. The subsequent branching represents the outcome of the test, and each leaf node holds a prediction.[8] Specifically, each internal node in the decision tree splits the dataset into two disjoint sets based on a particular binary test related to a *cut-point* value of a given attribute. The attribute and its cut-point are chosen to minimize an objective function, typically the mean squared error, of each branch. The predictions in the leaf nodes of our decision trees are determined by the average PPSM of the dwellings in the training data that directs into that branch.[9]

There are several hyperparameters on the construction of decision trees in each ensemble method, which determine the strength of the model. Two essential hyperparameters are the number of decision trees to build and the depth of each decision tree. Another more subtle hyperparameter is the option to use bootstrapping. By using bootstrapping, the methods pick random transactions from the training data with replacements. This sampling procedure produces separate datasets for each decision tree. When determining the hyperparameters above, a trade-off has to be considered between the explanatory power of the model and its computational complexity. In general, increasing the number of trees will yield a higher explanatory power; however, this also increases the computational burden. A model with high explanatory power captures the prevalent relationships in the data (i.e., prevents underfitting), while at the same time, reduces the identification of non-existing relationships between the attributes (i.e., prevents overfitting).

Figure 5 exemplifies a binary decision tree. The XGB method produces several of such binary decision trees, which are grown sequentially to improve on the residual of the previous tree as described in Section 3.2. Below, we present an example of the first tree for a randomly selected dwelling. The dwelling is a three-room, cooperative apartment in the district of Østensjø. The attribute names are given as $f0, f1, \ldots$ due to the nature of our selected graphing tool. We note a few of the important attributes: longitude ($f0$), latitude ($f1$), USMs ($f2$), number of rooms ($f4$), days since sale ($f6$), built year ($f7$), and common debt ($f8$).

[8] The dataset is typically divided into leaf notes based on binary choices, where predictions are given based on the training set.
[9] As mentioned in Section 3.2, we model the PPSM of the dwellings rather than the sales price.

Figure 5 Example of a Decision Tree for the XGBoost-Method

Binary decision tree produced by the XGB-method for a randomly selected dwelling with three-rooms and is a cooperative apartment in the district of Østensjø. The attribute names are: longitude (*f*0), latitude (*f*1), USMs (*f*2), number of rooms (*f*4), days since sale (*f*6), built year (*f*7), and common debt (*f*8). The vertical distance between the nodes indicate the attribute importance.

3.4 Selected Ensemble Learning Methods

The ensemble method, *bagging,* trains the underlying decision trees in parallel and independently. We utilize three bagging methods, specifically the *BP*, *RF*, and *ETs*. RF is an extension of BP, while ETs extend RF further. The methods have subtle but distinct variations in the process of building decision trees. The most general bagging method is the BP in Breiman (1996a), which builds a fixed number of independent decision trees by sampling the training data with bootstrapping. When constructing each decision tree, the method searches over each attribute and each cut-point to find the attribute that best splits the data at a given node. When calculating the sales price of an unseen dwelling, the BP averages the estimates from all the decision trees.

RF differs from the BP in terms of the method used for "growing" the underlying decision trees. RF builds the trees by sampling from only a randomly selected sub-sample of the attributes at each node split. This approach is known as *feature bagging*. As noted by Breiman (2001), the prediction error of the ensembles of tree predictors depends on the strength of the individual trees as well as the correlation among them. By using feature bagging at each node split, the RF will tend to reduce the correlation between trees, thus yielding a more robust model for out-of-sample predictions. In addition, feature bagging has the added benefit of being less computationally burdensome.

Instead of using feature bagging, as in the RF method, the *ETs* method in Geurts et al. (2006) randomizes the choice of cut-points for each attribute to learn about decorrelated trees. In so doing, the ETs method arbitrarily chooses a value (cut-point) for each attribute when splitting the trees in lieu of trying all possible cut-points. As a result, this method increases the randomness by simultaneously reducing the computational burden of the algorithm.

The ensemble technique, *boosting,* which is first introduced by Schapire (1989), trains the underlying decision trees *sequentially* with each tree being fitted to reduce the errors made by the preceding trees. As with the bagging method, bootstrapped training data are used to train the underlying trees. In contrast to the bagging method, each new tree improves on the predictions of the previous tree by attempting to reduce its "shortcomings". The boosting algorithm, *AdaBoost,* was first proposed by Freund and Schapire (1997) and then generalized into the *Gradient Boosting Machine* by Friedman (2001).[10] *XGB* is a recently developed boosting method that has proven successful in a variety of machine learning competitions.[11] The algorithm minimizes an objective function that consists of a loss function and a regularization term. At each

[10] See Schapire (2013) and Chen and Guestrin (2016) for a detailed overview on the development of the boosting methods.

[11] Nielsen (2016) provides a comprehensive analysis of the methodology. For implementation details, we refer to Chen and Guestrin (2016).

iteration, the regularization term is added to prevent the model from overfitting. The objective function is defined as:

$$Obj = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (1)$$

where $l(\cdot,\cdot)$ is the loss function, $\Omega(\cdot)$ is the regularization term, f_k is the kth decision tree, and $\hat{y}_i$ and y_i are the predicted and actual sales prices of the ith dwelling, respectively.

The trees are built by splitting at each leaf node on the attribute value (cut point), which minimizes the pre-specified objective function. This is done recursively until the trees reach a pre-specified maximum depth. In our implementation, the loss term of the objective function is chosen as the *mean absolute error*. Each leaf contains a weight, determined by the first- and second-order differentials of the loss function and the regularization term.[12] Estimates continually improve by training each decision tree on the residuals from the previous iteration. When creating a value estimate for a dwelling x_i, the XGB sums up the selected weight for each tree:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (2)$$

For the construction of decision trees in each ensemble method, several *hyperparameters* are available to determine the strength of the model. We employ the bootstrapping methods to pick random transactions from the training data with replacement. This sampling procedure produces separate datasets for each decision tree.

We note a trade-off between the explanatory power of the model and its computational complexity when determining the hyperparameters. In general, increasing the number of trees will yield a higher explanatory power but is more computationally burdensome. A model with high explanatory power captures the prevalent relationships in the data (avoidance of *underfitting*) while simultaneously avoids identifying non-existing relationships between the attributes (avoidance of *overfitting*).

Both the bagging and boosting algorithms (XGB) require tuning of the hyperparameters in order to fit any particular dataset. This is due to the differences in the amount of available data, number of attributes in the dataset and structure of the dataset. Some hyperparameters are common to all of our algorithms, like the number of decision trees to create and the maximum depth

[12] For the technical rationale, see Equation (5) and the related descriptions in Chen and Guestrin (2016).

of each tree, while others are specific to each model. Panels A and B of Table 4 provide the different hyperparameters and their tuned values.[13]

Table 4 Hyperparameters

Descriptions and tuned values of the bagging hyperparameters: BP, RF and ETs in Panel A. Descriptions and tuned values of the hyperparameters of XGBoost in Panel B, both as a model stacker and underlying method. Running the algorithm repeated times shows that the tuned values provide the highest parameter stability.

Variable	Applicable for	Description	Tuned Value
Panel A: Bagging hyperparameters			
Number of trees	BP, RF, ETs	Total number of decision trees created.	250, 150, 100
Bootstrapping	BP, RF, ETs	Whether subsamples picked with replacement.	True
Maximum depth	RF	(Maximum) Depth of each tree.	50
Share of attributes	RF	Share of attributes used when creating a split.	0.33
Panel B: XGBoost hyperparameters			
Learning rate		Step-size shrinkage used in each update.	0.005
Number of iterations		Number of trees to build.	1,000
Gamma		Minimum loss reduction required to make a split.	0
Maximum depth		(Maximum) Depth of each tree.	5
Subsample		Share of data points used when building a tree.	0.8
Colsample by tree		Share of attributes used when building a tree.	0.8
Evaluation Metric		Loss function that algorithm aims to minimize.	MAE

The optimization of the parameters for the BP, RF and ETs methods is based on Hauck (2014), who provides an in-depth analysis of the parameter tuning that our optimizer is based on.[14] When tuning the parameters for XGB, we follow DMLC (2016), a guide provided by the machine learning community who developed XGB. The parameter tuning is implemented by using the CrossValidation package in SKLearn in order to search over a set of possible parameters. The search is done iteratively and by decreasing the range for each

[13] We refer to the documentation from SKLearn and XGB for further descriptions of the parameters and their default values (Pedregosa et al., 2011; Chen and Guestrin, 2018). We use the default values for the remaining hyperparameters in XGB.

[14] The cross-validation procedure divides the training data into five separate training and validation sets and runs each model with a given set of hyperparameters on each dataset. Subsequently, the prediction errors on the validation sets are averaged and the optimal combination of hyperparameters based on a scoring function is chosen.

parameter successively to find the optimal value. The cross-validation procedure divides the training data into five separate training as well as validation sets, and runs each model with a given set of hyperparameters on each dataset. It then averages the prediction errors on the validation sets and chooses the optimal combination of hyperparameters based on a scoring function. We use the mean absolute error as the scoring function when selecting the optimal hyperparameters. Note that hyperparameter tuning is done on the training data and not the testing data (i.e., the data from the first quarter of 2018).[15] Hyperparameter tuning on the testing data would lead to overfitting, and thus, would overstate our performance estimates.

The ensemble methods as non-parametric models do not require any rich *a priori* knowledge regarding the underlying data generating process. This omission allows the method to adapt to the underlying data and the potential non-linear relationships among the variables. In contrast, traditional HPMs do not model non-linear relationships and make limiting assumptions about the data such as linearity and homoscedasticity. Furthermore, ensemble learning methods are ideal for the amount of relevant and available training data. More complex non-linear models, such as ANNs, often require many more degrees of freedom to yield high predictive power (Bishop, 2006), which may quickly result in *overfitting* in the case of limited data availability. Similarly, simpler models, such as HPMs based on ordinary least squares (OLS), may not be able to fully utilize the dataset, due to their limited functional form. We discuss the use and demonstrate the performance of OLS and ANNs in Sections 4.3 and the Internet Appendix, respectively.

We also note that the selected ensemble learning methods require minimal feature engineering[16], especially concerning the grouping of attributes into categorical variables, but also transforming the continuous variables. The underlying decision trees are constructed to learn such patterns without requiring significant domain knowledge. Thus, our model is simpler to implement and more robust to applications in dynamic real estate markets. On the other hand, the ensemble methods do not provide the same degree of transparency as the simpler OLS-based HPMs, which yield price estimates of individual attributes. In Section 4.3, we calculate *attribute importance* for the underlying characteristics to provide a sufficient degree of transparency for the application of our AVM.

[15] We refrain from using a validation set approach due to the limited time dimension in our sample. The division between training and validation is more common for techniques like ANNs which are more likely to be affected by the problem of overfitting.
[16] This process transforms the attributes to create new attributes, which capture the domain-knowledge of the prediction problem. An example could be to create an attribute for the availability of an elevator in the dwelling's building *and* the dwelling being in the third story or higher.

As ensemble learning methods do not make any underlying assumptions of the distribution of the data, it might be difficult to use them for generalization beyond the observed training data. Therefore, ensemble learning methods are heavily reliant on an extensive dataset to yield robust results. We do not expect this to be an issue for our model as we use a comparable pricing approach to select and engineer the training data (see Section 3.4). More precisely, we select training data, which are tailored to suit the dwelling in question. Thus, this ensures that the model is highly likely to generalize to the given dwelling.

3.5 Repeat Sales Method

We use the RSM introduced in the seminal paper of Bailey et al. (1963) to create a separate index for each district in Oslo to capture the appreciation or depreciation in the market over time. The indices are used to produce price estimates for all previously sold dwellings in our dataset by adjusting the previous sales price of each dwelling with the appropriate index. The RSM has the advantage of isolating actual increases in the price of a dwelling without requiring detailed information about the characteristics of individual properties. As discussed in Section 2, there are several explanatory variables omitted in our dataset.

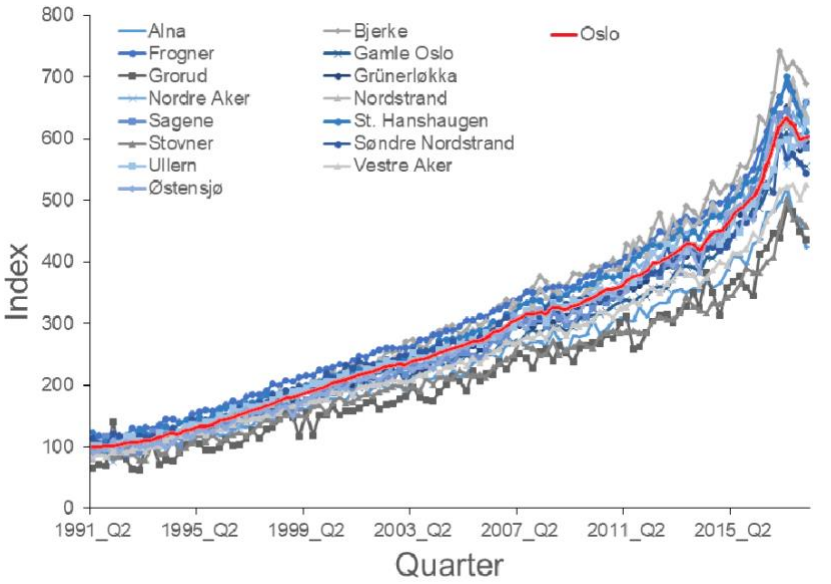
The core idea behind the method is that the ratio of the sales prices for the same dwelling at two different periods of time can be thought of as a ratio of an (unobservable) index for the local area on the two selling dates. This idea is justified under a constant-quality assumption, i.e., the quality of a dwelling does not deteriorate or improve substantially over time. In theory, the RSM provides unbiased results of the price appreciation by controlling for all attributes that do not change over time, and in particular, for the micro-location. In our dataset, which spans more than a quarter of a century, we are prone to find dwellings that have been upgraded or significantly altered in some form or manner. Hence, assuming that there is a consistent quality will possibly bias our results. However, by using the RSM as proposed in Case and Shiller (1987), repeat sales with longer periods between them are potentially given less weight in the estimation of the model. Although there are other implementations of the RSM, we note that the RSM in Case and Shiller (1987) is deemed more favorable both in the academia and industry (Balk et al., 2013; S&P Dow Jones Indices, 2021).[17] The model is stated as a weighted least squares, with the logarithms of the ratios of the sales prices as the endogenous variables and the selling times as the exogenous variables. We only consider condominiums for index construction, because the *historical transactions dataset* has missing data on common debt, which has a significant share in cooperatives as described in Sections 2.2 and 2.4. Although this may lead to partially biased indices, we find

[17] Two prominent alternatives for creating real estate indices are proposed by Bailey et al. (1963) and Calhoun (1996), which only vary in their assumptions of the heteroscedasticity of the underlying regression problem.

that this alternative provides a better fit for our data. Figure 6 shows in Panels A and B the repeat sales indices for Oslo (from 1991Q1 to 2018Q1) and its districts (from 1993Q2 to 2018Q1), respectively.

Figure 6 Aggregated Index for Oslo

Index for the residential market of Oslo and its administrative districts based on RSM in Case and Shiller (1987).



3.6 Comparable Pricing and Stacking of Models

The ensemble learning sub-models of the AVM are trained separately for each value estimate. That is, the ensemble learning methods are trained on *comparable recent sales*, which are selected based on the location and type of the dwelling. Specifically, we select the n geographically nearest transactions[18] of dwellings for each of the unit types (apartments and non-apartments) separately. Furthermore, we add a *ranking* variable to these transactions, i.e., $rank \in [1, 2, \ldots, n]$ based on the proximity, i.e., the variable increases with distance from the target dwelling.[19]

[18] If the dwelling is an apartment, then n = 10,000, and if not, n = 2,000. The distinction is made due to availability of transactions for the two types of dwellings.

[19] The distance between two points in spherical geometry is given by the Haversine formula which adjusts for the earth's curvature, particularly becoming relevant for distances more than 50 km (Sinnott, 1984).

The comparable pricing approach enables the ensemble methods to explicitly recognize geographically close and recent transactions when valuing a dwelling. We note that the comparable transactions are selected from the enhanced transaction dataset, and thus, are all from August 2016 and onwards. Therefore, we do not eliminate any transactions based on the sold date when selecting the comparables.

In addition to using the XGB as an underlying method, we leverage the XGB as a stacking method or *model stacking process*. As described, the XGB is used as a meta estimator with both the exogenous variables and the value estimates from the individual models as an input. This stacking process is described in detail as follows.

Automated Valuation Modelling Process[20]

Step 1: The current sales price of the dwelling to be predicted is labelled as u.

Step 2: n comparable transactions are selected, as described above, and a set of these transactions is denoted as *training data* or X.

Step 3: For each model $m \in [XBoost, Random\ Forest, Extra\ Trees, Bagging\ Predictor]$:

a) The training data are divided into $k = 5$ random subsets of equal size X_i with $i = 1 \ldots k$. For each of the subsets X_i:

i) m observations are fitted to the training data *not* included in X_i to obtain a fitted model m_i.

ii) m_i is used to obtain estimates of the sales price, $\hat{P}_i$, for the data in X_i and the training data X_i are extended with $\hat{P}_i$.

iii) m_i is used to obtain an estimate of the sales price of u which is denoted as $\hat{u}_i$.

b) the k predictions of the price of dwelling u are averaged to obtain and extend the dwelling data u with $\hat{u}$.

Step 4: For each data point in the training data X and the dwelling u:

a) all previous sales for the dwelling are identified, and

b) the repeat sales price indices are used to generate price estimates based on each of the previous sales.

c) the data for the relevant dwelling are extended with an average of the repeat sales price estimates as well as the number of predictions. If there are no previous sales, the data are extended with a 0 for sale and resale.

Step 5: Another XGB model is fitted to the now extended training data X, and the model is used to predict a sales price for u.

Using estimates from a diverse set of estimators enables our AVM to deliver robust results with high predictive power. Although it is a more recent state-of-the-art approach in the field of econometrics, the idea of stacking different

[20] The Python Code for the procedure can be found in Internet Appendix 2.

ensemble learning methods is gaining popularity in both the academia (Campos et al., 2017) and data sciences (Adam-Bourdarios et al., 2015; Alves, 2017). We choose the powerful stacking method XGB due to its ability to detect the performance of each base model to combine the underlying predictions accordingly. Stacking is highly effective when the underlying models are diverse as in our case, i.e., when both an array of ensemble learning methods and an RSM is selected. The drawback of combining stacked generalization with a comparable market analysis is that the latter requires one instance of the model to be trained for each value estimate. By stacking multiple individual models, the estimation becomes increasingly complex. Specifically, the model encompasses five ensemble learning methods, of which four are trained five times. The combined effect of these choices leads to increased model training time. We analyze and discuss the practical implications of our approach in Section 4.6.

3.7 Out-of-Sample Prediction

When evaluating our model, we divide the data into two disjoint sets, one which is used as a *training set* and the other being the *test set.*[21] We simulate a real-life scenario, where the model is trained on data recorded up to a given day and produces value estimates on possible transactions for the next day. We perform this split monthly, while in practice, one would update the data every day. Hence, the results of our model can be interpreted as conservative in terms of both estimates and out-of-sample predictive power. When evaluating our model in Section 4, we make three such partitions using the following *splits*: January 1st, 2018, 00:00, February 1st, 2018, 00:00, and March 1st, 2018, 00:00.

For each dwelling in the test set, we choose the comparable transactions from the corresponding training set, as discussed in Section 3.4. Similarly, when applying the RSM to predict previously sold dwellings in the test set, we use indices built solely on the training set. As we have the attribute *sales month* in our dataset, we set this to be equal to the previous month for all dwellings in the test set. Thus, when making predictions with both the ensemble learning methods and the RSM, we are predicting the sales price as though the sales date is the first day of the month.

4. Empirical Results

In this section, we evaluate the prediction accuracy of our AVM and its sub-models. Subsequently, we discuss the essential model choices and potential challenges.

[21] This gives an approximate 90-10 split between the training and the test sets.

4.1 Performance Evaluation of the AVM

To analyze the performance of our AVM, we examine the distribution of its value predictions and compare the distribution to the valuation provided by real estate agents and industry leaders. Our AVM achieves an overall MdAPE of 5.4% in the first quarter of 2018 (see Panel A of Table 5). When comparing the performance of our AVM with the precision of the estate agents in Panels A and B of Table 5, we find that the performance of our AVM is very similar to that of the estate agents in 2016/17 (the training dataset period). Both the quantiles and the MdAPEs are almost identical, while the mean absolute percentage error (MAPE) of our AVM is slightly lower than that of the estate agents. Looking at the more directly comparable first quarter of 2018, the real estate agents perform better than the AVM along the forecast performance ratios. The main reason that the real estate agents outperform the model in the first quarter of 2018 compared to 2016/17 is probably due to the dynamics in the market in 2018. While the house price development in the first quarter of 2018 was quite stable, house prices in Oslo increased by 23% in 2016 before they fell by 6% in 2017.

Furthermore, we compare the performance of our AVM with that of the hedonic regressions in Oust et al. (2020) and Zillow. Panel C shows the percentage of predictions with different hedonic regressions in accordance with Oust et al. (2020). The data cover Oslo and are taken from the same data provider. They are therefore the same data used in this study, with the same starting point, but an end date of December 2017 for the sample. They also employ a randomly drawn training and testing sample instead of making a time separation. This allows the model to train on the same time period as the model is tested on, which reduces its exposure to price changes over time since this is already included in the model. The ordinary hedonic regression shows a percentage of prediction within a range of 10% of the correct values at 59.6%. In contrast, when Oust et al. (2020) allow the model to form its own geographical districts (using *K*-means), the within 10% performance is 61.6%. Adding repeated sale on top of the *K*-means, the within 10% performance becomes 66.0%. Employing mixed autoregressive (administrative and K-means districts) and spatial autoregressive models, the within 10% prediction accuracy is at 63.7%, 63.6%, and 67.4%.

Zillow creates similar value estimates for more than 100 million dwellings in the U.S. and publishes their aggregated performance for a handful of selected cities. Panel D of Table 5 illustrates some of these performances. The MdAPEs of Zillow vary between 3.3% and 8.2%, which is both considerably better and worse than the performance of our model. In addition, we observe that none of the cities have a higher percentage of estimations that are within 20% of the sales price than Denver, which has a value of 94.5%. Here, our model outperforms Zillow, with more than 96% of the estimates being within 20% of the sales price. In addition, Zillow has data that describe roughly 1-2 million dwellings in each of the presented cities, which is a considerably higher volume of training data than we can access. While the comparative analysis illustrates

the value-added by our AVM, we must point out that such a comparison of performances is limited due to the apparent differences between markets and data availability.

Table 5 Prediction Accuracy: Machine versus Human

Prediction accuracy for the AVM, estate agents, and Zillow. Evaluations for the AVM in Panel A and *real estate agents* in Panel B are based on the share of the predictions within the range of 5%, 10%, and 20% of the correct value or actual sales price, respectively, as well as the MdAPE and MAPE. For the AVM, the out-of-sample performance refers to 2018Q1. For the real estate agents, the data are from *Finn.no*, including 15,786 transactions in Oslo in 2016 and 2017 as well as 3,009 transactions for 2018Q1. Panel C shows the share of predictions with different hedonic regressions in accordance with Oust et al. (2020), who use data from Oslo from the same data provider as in Panel A. Panel D shows the share of *Zestimates* from Zillow within 5%, 10%, and 20% of the actual sales price and the overall MdAPE for a selection of U.S. cities as reported by Zillow. Data obtained from www.zillow.com/zestimate/ on May 27, 2018.

	Within 5%	Within 10%	Within 20%	MdAPE	MAPE
Panel A: Comparable prediction accuracy of the AVM					
2018Q1	46.9%	76.4%	96.3%	5.4%	7.2%
Panel B: Comparable prediction accuracy of real estate agents					
2016/2017	47.8%	74.0%	96.5%	5.3%	7.6%
2018Q1	70.6%	93.8%	99.5%	3.0%	5.1%
Panel C: Comparable prediction accuracy of hedonic models (Oust et al., 2020)					
Ordinary hedonic regression (Administrative districts)		59.6%			
Ordinary hedonic regression (K-means districts)		61.6%			
Ordinary hedonic regression (K-means districts and repeated sales)		66.0%			
Autoregressive Model (Administrative districts)		63.7%			
Autoregressive Model (K-means districts)		63.6%			
Autoregressive Model (K-means districts and repeated sales)		67.4%			
Panel D: Comparable prediction accuracy of Zillow					
Baltimore, MD	54.6%	73.6%	85.1%	4.3%	-
Boston, MA	53.9%	78.1%	89.9%	4.5%	-
Charlotte, NC	52.3%	72.3%	84.1%	4.7%	-
Chicago, IL	56.7%	76.8%	88.5%	4.1%	-
Cincinnati, OH	46.4%	68.4%	84.0%	5.5%	-
Cleveland, OH	44.8%	65.4%	80.2%	6.0%	-
Dallas-Fort Worth, TX	33.1%	57.2%	79.6%	8.2%	-
Denver, CO	65.5%	86.1%	94.5%	3.3%	-
Detroit, MI	50.9%	71.9%	85.6%	4.8%	-

On a final note, we highlight a behavioral finance aspect of the comparisons made in this section. The predictions of both the real estate agents and Zillow are made (and published) prior to establishing the sales price; thus, they are likely to influence the buyer and seller. We believe that this can have two effects: real estate agents might be inclined to price a dwelling lower than the expected sales price to attract more potential buyers and hence, start a bidding war. We observe that roughly 61% of the value estimates (asking prices) by estate agents are lower than the final sales price in the 2016/2017 data, and roughly 43% lower in the 2018Q1 data. In addition, according to the anchoring-and-adjustment heuristic as presented in the psychology literature[22], such reference points are prone to bias the valuation of market participants. Thus, the final sales price is likely to be insufficiently adjusted away from the anchor.

4.2 Performance Evaluation of the Stacked Model

To justify the application of stacked generalization in our AVM, we perform a thorough evaluation of the effect of stacking. First, we compare the accuracy, measured by the MdAPE, of the stacked model with that of the selected ensemble learning methods, the RSM, and a simple OLS-based HPM. Then, we analyze the performance against the training time of our models, to find the optimal number of comparables to use at each valuation. Finally, we rationalize the choice to stack the four ensemble learning methods.

To justify the use of stacked generalization in our AVM, we compare the accuracy of the stacked model to that of the underlying models. The comparison is made out-of-sample for January, February, and March 2018, as well as in aggregate for the three months. The results are presented in Table 6. We observe that, on average, the AVM performs significantly better than the individual models. We also note that the RSM performs considerably poorer than all of the other underlying methods.

We further examine the correlation between the individual model residuals to compare their value estimates and discuss the potential gain of stacking. Table 7 illustrates the correlations between the out-of-sample residuals of the ensemble learning methods.[23] An apparent observation is the highly positive correlation between the methods. We choose to include all four methods since no single method yields strictly better results and believe that the stacking algorithm should be able to choose their optimal combination. Figure 7 shows

[22] Seminal works by Slovic and Lichtenstein (1971) and Kahneman and Tversky (1972) provide an introduction and discuss this heuristic, while Northcraft and Neale (1987) consider the heuristic empirically in the setting of the residential real estate market in Tuscon, Arizona. They find that the "*subject populations were significantly biased by listing prices"* (Northcraft and Neale, 1987, p. 95)

[23] We do not include the RSM here, because it only covers 77% of the transactions.

that the model-stacker (XGB-S) can detect relationships in the data not captured by the individual methods; hence, yielding better out-of-sample results.

Table 6 Sub-model Accuracy Compared to the AVM

MdAPE of the AVM compared to the ensemble learning methods and the RSM (see Section 3.5), as well as an OLS-based hedonic pricing method for the out-of-sample performance period of 2018Q1 (see Internet Appendix 3). MdAPE for the RSM is only calculated for dwellings with previous sales, which constitutes 77% of the dwellings in the test set.

	Ensemble Learning				**Repeat Sales (RSM)**	**Traditional OLS**	**Stacked Model XGB-S**
	BP	**RF**	**ET**	**XGB**			
January 2018	6.23%	6.31%	6.21%	6.13%	8.93%	8.95%	**5.49%**
February 2018	5.60%	5.70%	5.71%	5.56%	8.86%	8.85%	**4.94%**
March 2018	5.81%	5.47%	5.64%	5.90%	9.42%	8.46%	**5.50%**
Total	5.95%	5.90%	5.92%	5.99%	9.05%	8.77%	**5.36%**

Table 7 Correlations of Sub-models

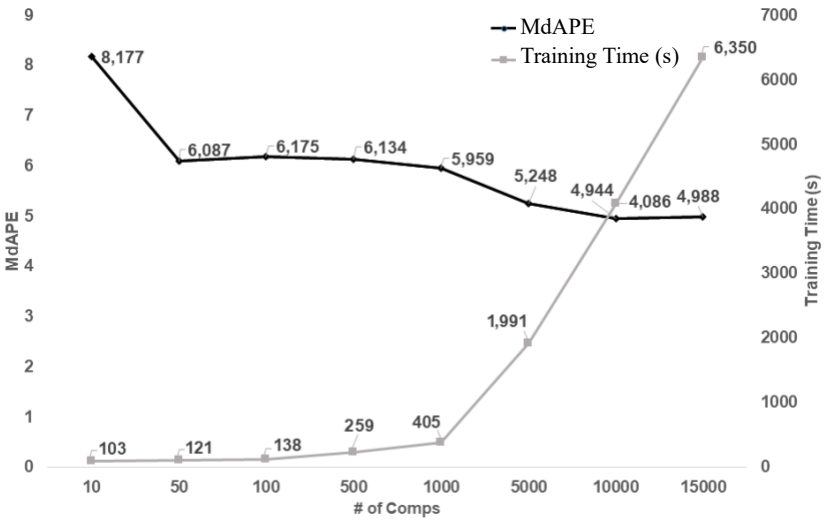
Pearson correlation coefficients of the residuals of the AVM (denoted by XGB-S) and the submodels XGB, BP, RF and ETs. A lighter color, and higher positive number, indicate a higher positive correlation. We observe high correlations between the residuals.

	XGB-S	BP	RF	ETs	XGB
XGB-S	1	0.82	0.81	0.8	0.8
BP	0.82	1	0.97	0.95	0.94
RF	0.81	0.97	1	0.98	0.97
ETs	0.8	0.95	0.98	1	0.96
XGB	0.8	0.94	0.97	0.96	1

Figure 7 shows how training time and accuracy, represented by the MdAPE, increase with the number of comparable transactions for each valuation for the period of February 2018. We observe that the training time is more or less linear in the number of comparable transactions. With only 50 comparable transactions, the model can score an out-of-sample MdAPE of about 6%. However, a further gain towards 5% MdAPE is computationally burdensome. As we do not observe further improvements after including 10,000 comparable transactions and due to training time constraints, we choose this number of comparables in our final model.

Figure 7 Model Performance of the AVM for Various Comparables

Accuracy and training time of the AVM for various comparable sales. The results are for the out-of-sample period February 2018 (470 transactions). Note that the *x*-axis is not linear in the number of comparable sales. MdAPE is shown on the left *y*-axis, whereas training time is plotted on the right. Note that the stated training time is a result of our computer and software. The point is to show the trade-off between improving MdAPE and the training time.



4.3 Performance Evaluation of the Attribute-Based Pricing Methods

Table 6 shows the out-of-sample performance of the selected ensemble learning methods and our AVM compared to a conventional OLS-based HPM. We note that the latter uses the same set of attributes as the ensemble learning methods. The performance of the ensemble learning methods is superior to OLS in each of the test months, as they outperform the HPM by more than 30% on average. We acknowledge that the HPM might suffer from the lack of feature engineering and might obtain better results by including transformation and grouping attributes in a pre-processing step.

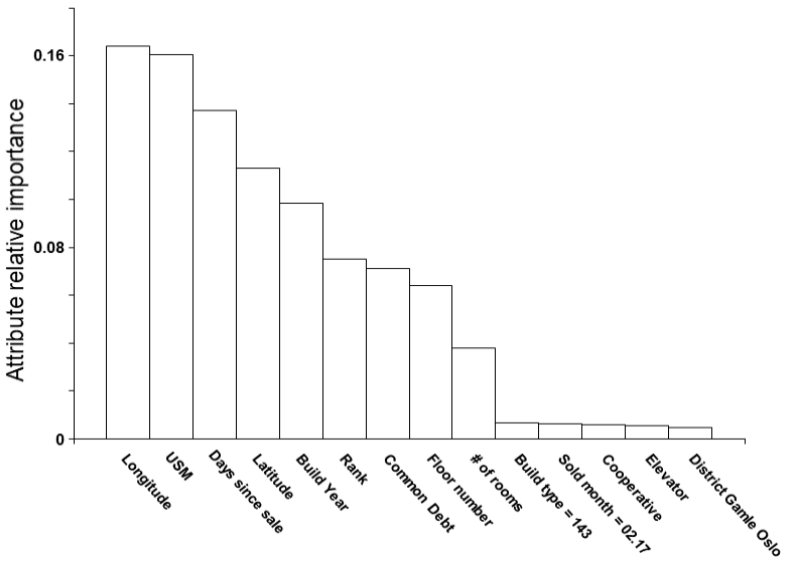
As stated in Section 3.2, AVMs that use ensemble learning are more difficult to interpret than OLS-based HPMs. However, this shortcoming can be addressed by analyzing the underlying decision trees of the models, either by showing the individual trees or analyzing aggregated statistics from each model, such as *attribute importance*. Thereby a score is given to each attribute based on its importance in enhancing the performance of the model. To calculate this ratio, several methods are found based on the aggregate frequency of occurrence and

the placement of attributes in the decision trees.[24] We give an example of analyzing attribute importance for the XGB-algorithm by using the definition given in Chen and Guestrin (2018). For the XGB-algorithm, attribute importance is given as the share of predictive power brought by including a particular attribute in the decision trees (Chen and Guestrin, 2018). These scores are denoted as $\alpha_i \in [0,1]$, where $\sum_i \alpha_i = 1$.

Since we build a separate model for each dwelling, we analyze the model built for one particular apartment sold in Oslo in March 2018. The apartment is a condominium situated in the Gamle Oslo district with 82 USMs. The dwelling was constructed in 2013, has four rooms, and is located on the fifth story. Figure 8 illustrates the importance of the most relevant attributes when building the decision trees for the XGB-algorithm of the discussed apartment. We observe that the numerical attributes are considered to be far more critical than the categorical attributes when building the decision trees. Specifically, the location, represented by longitude and latitude, the size of dwelling, represented by USMs, and the date of the transaction, represented by the *days since sale*-attribute, are the most significant attributes. We observe some variables with a *zero* attribute importance score, such as districts far from the selected dwelling.

Figure 8 Attribute Importance

Relative importance of the most important attributes when valuing a dwelling sold in Oslo in March 2018. The apartment is a condominium situated in the district of Gamle Oslo and is 82 USMs. Constructed in 2013 with four rooms on the fifth story. The relative importance is calculated as a ranking score of the underlying XGB-model.



[24] We refer to Chapters 10.13 and 15.3 of Friedman et al. (2001) for an overview of variable importance estimations.

These observations are not only reasonable but also confirmed by real estate theory. The numerical variables have a larger number of possible *cut-points* than the binary categorical variables; therefore, they can be used more often to adequately partition the dataset. The variables that have received the highest attribute scores are those most often associated with the sales prices of dwellings. The fact that some attributes receive a score of *zero* for one particular model does not hinder the attribute from being necessary in another model. This variation illustrates the ability of the model to adapt to the underlying data.

The choice of using ensemble learning methods as our attribute-based pricing methods was made on an extended exploratory analysis of the state-of-the-art machine learning techniques. We acknowledge that traditional hedonic methods, such as those based on OLS, might yield favorable results in some situations. Given a smaller dataset, less training time, or more detailed inference statistics, we believe that traditional hedonic methods could be applied with adequate precision (see Table 6 for the comparable results).

Moreover, we consider the use of an ANN as a sub-model in our AVM. Due to a large number of hyperparameters and low interpretability, our findings demonstrate their high-level complexity for real-world applications. We provide a summary of our results when including ANN as a sub-model for our AVM in the Appendix. In short, we find the process of designing the network to be a highly specialized engineering process, with many design choices and only a handful of established guidelines.

4.4 Performance Evaluation of the Repeat Sales Method

As shown in Table 6, the RSM has a considerably poorer performance compared to the other individual models and similar performance to the OLS-based HPM. Therefore, one might question the inclusion of the model in the AVM. However, as argued in Section 3.5, the RSM is trained on a separate dataset and aims to capture different market movements than those of the ensemble learning methods. Hence, we believe that the inclusion of the RSM in the AVM is advantageous. To empirically justify this decision, we run the AVM as described in the process in Section 3.6—without Step 4—and compare the performance to that of the AVM. We present this comparison in Table 8 and note that the MdAPE increases by 8% overall when the RSM is excluded from the model. This decline in predictability indicates the considerable benefit of including the RSM. We note that the number of previous sales for the dwelling is included in the training data for the model stacking process, in addition to the RSM prediction. This attribute is also likely to yield significant explanatory power, as frequently sold dwellings often have distinctive characteristics. These results support the findings documented in Oust et al. (2020) and indicate that

the RSM is able to capture some information about the dwellings that are not included in the hedonic characteristics.

Table 8 Prediction Accuracy of AVM with and without RSM

Share of the predictions within the range of 5%, 10%, and 20% of the sales price, MdAPE, and MAPE derived from AVM with and without the RSM for the out-of-sample performance period of 2018Q1.

	Within 5%	Within 10%	Within 20%	MdAPE	MAPE
AVM incl. RSM	46.89%	76.36%	96.31%	5.36%	7.17%
AVM excl. RSM	45.52%	73.23%	95.08%	5.81%	7.43%

4.5 Model Discussion

In this section, we discuss and critique our model, both with regards to the hypothesized model rationales in Section 3 and in the context of the empirical results provided above. We begin by reiterating three of the main argued strengths of our model and elaborate on these in turn. First, we argue that applying stacked generalization would allow our model to combine the predictions of several submodels with improved results. In the results of Tables 6 and 8, we find this to be the case, as the performance of the stacked model clearly improves on the individual methods. Second, while developing the model, we hypothesize that the use of a comparable market analysis would be beneficial as the model would benefit the most from nearby transactions. However, as Figure 7 shows, we observe that up to 10,000 transactions, with the additional ranking variables, yield superior results.

Third, we emphasize the benefits of ensemble learning techniques in the field of econometrics. The non-parametric nature of the methods makes them applicable to a wide range of tasks. Even though ensemble learning methods are novel tools in econometrics, they often provide superior results to traditional methods and therefore becoming increasingly popular.

The major drawback of applying stacked generalization in a model is the training time demanded due to the required predictions made on the training data by the individual folds. Depending on the quality of the implementation,[25] the model runs for 60-400 seconds for a single prediction. For practical applications, this imposes certain constraints on the design of the service. However, we argue that the value of a robust model exceeds the drawback of time complexity. Furthermore, we note that the model can easily be adapted to the demands of training time. In practice, several instances of the model could

[25] The quality of the implementation largely depends on choices on programming language, parallelization and pre-processing. See Appendix 2 for an overview and discussion of our implementation.

be run in parallel to give the user improved results as the different instances are completed.

A challenge for the non-parametric ensemble learning methods is that they require sufficiently diverse training data to provide high out-of-sample predictive force. Specifically, they do not generalize well outside the observed range of attribute values, as they do not make any prior assumptions about the underlying data. However, as we find in the results above, the model is not only able to predict with high accuracy, but also with high precision. That is, in addition to a low MdAPE, the model has a large number of predictions within a 20% deviation. This is compelling evidence of its ability to generalize the given dataset well, and consequently, its suitability for use in AVMs.

The model also lacks of transparency and, to a certain extent, of underlying model assumptions. Although we have explained how the model may be visualized with the use of attribute importance and by showing the underlying decision trees, we concede that this may be insufficient for their use in public policy. A major transition from traditional HPMs to ensemble learning AVMs is the shift from a theory-driven to a data-driven approach, where an econometrician defines fewer of the model's assumptions.

5. Conclusion

This study develops an AVM by stacking four different ensemble learning methods and the RSM. We evaluate the predictive power of our model on transaction data of the residential real estate market in Oslo. This novel approach of combining ensemble learning and real estate indices in the structure of a stacked generalization leads to substantial improvements compared to the individual methods. The approach shows that the use of a comparable market analysis provides valuable information as the model benefits the most from nearby transactions. Thus, our findings support the understanding of how the quality of an AVM can be enhanced to become a valuable instrument for commercial use.

Our AVM estimates the value of 1,979 dwellings sold in 2018Q1 with an MdAPE of 5.4%. This performance is comparable to the accuracy of Norwegian estate agents and superior to the precision of Zillow for a selection of cities, for which official performance statistics are available. In summary, we conclude that in very dynamic markets, the valuation accuracy of our AVM is similar to that of real estate agents. However, in more stable market phases, the machine falls short of human capability.

A drawback of applying stacked generalization in a model is the training time demanded due to the required predictions made on the training data by the individual folds. Another challenge for the non-parametric ensemble learning

methods is that they require sufficiently diverse training data to achieve high out-of-sample predictive power. Specifically, they do not generalize well outside the observed range of attribute values, as they do not make any prior assumptions about the underlying data. However, our model not only predicts with high accuracy but also with high precision. In addition to a low MdAPE, a large number of the predictions fall within 20% of the actual sales price, which indicates that the model has good ability to generalize the given dataset.

A significant shift from traditional HPMs to ensemble learning AVMs is one from a theory-driven to a data-driven approach. Hence, another well-known drawback is the lack of transparency of the model and, to a certain extent, the lack of underlying model assumptions. We have addressed how the available tools, such as attribute importance and decision trees, have been designed to enhance the interpretability of the ensemble learning methods. To achieve our goal of increasing the validity of AVMs in real-life applications, we opt to design a model with substantial computational complexity. At the same time, we have shown that one this complexity could be significantly reduced at the cost of lower modeling precision.

Acknowledgement

For helpful comments we are grateful to Jon Olaf Olaussen and Alois Weigand. We are also indebted to *Alva Technologies* for providing the data to us.

References

Adam-Bourdarios, C., Cowan, G., Germain, C. and Guyon, I. (2015). The Higgs Boson Machine Learning Challenge. NIPS 2014 Workshop on High-energy Physics and Machine Learning. *PMLR,* 42, 19–55.

Alves, A. (2017). Stacking Machine Learning Classifiers to Identify Higgs Bosons at the LHC. *Journal of Instrumentation*, 12(05), T05005.

Antipov, E.A., and Pokryshevskaya, E.B. (2002). Mass Appraisal of Residential Apartments: An Application of Random Forest for Valuation and a Cart-Based Approach for Model Diagnostics, *Expert Systems with Applications*, 39, 1772-1778.

Bailey, M.J., Muth, R.F. and Nourse, H.O. (1963). A Regression Method for Real Estate Price Index Construction, *Journal of the American Statistical Association* 58(304), 933-942.

Balk, B., De Haan, J. and Diewert, E. (2013). Handbook on Residential Property Prices Indices (RPPIs). World Bank Group.

Barr, J., Ellis, E., Kassab, A. and Redfearn, C. (2017). Home Price Index: A Machine Learning Methodology, *International Journal of Semantic Computation*, 11, 111-133.

Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Neural Network: Tricks of the Trade*, Montavon, G., Orr, G.B. and Muller, K.-R. (Eds.), Springer-Verlag.

Bishop, C.M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics), Vol. 4. Secaucus, NJ: Springer-Verlag.

Breiman, L. (1996a). Bagging Predictors, *Machine Learning* 24(2), 123-140.

Breiman, L. (1996b). Stacked Regressions, *Machine Learning* 24(1), 49-64.

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

Calhoun, C.A. (1996). OFHEO House Price Indexes: HPI Technical Description, Office of Federal Housing Enterprise Oversight, 20552.

Campos, R., Canuto, S., Salles, T., de Sá, C.C. and Gonçalves, M.A. (2017). Stacking Bagged and Boosted Forests for Effective Auto-Mated Classification. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 105-114). New York, NY: Association for Computing Machinery.

Case, K., and Shiller, R. (1987). Prices of Single Family Homes Since 1970: New Indexes for Four Cities, *New England Economic Review*, (September/October), 45-56.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (pp.785-794), San Francisco, CA: Association for Computing Machinery.

Chen, T. and Guestrin, C. (2018). Understand Your Dataset with XGBoost – xgboost 0.71 documentation. from: https://xgboost.readthedocs.io/en/latest/R-package/discoverYourData.html#creation-of-new-features-based-on-old-ones accessed: July 15. 2019.

Chiarazzo, V., Caggiani., L., Marinelli, M. and 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.

Clayton, J., Geltner, D. and Hamilton, S. (2001). Smoothing in Commercial Property Valuations: Evidence from Individual Appraisals, *Real Estate Economics*, 29(3), 337-360.

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, 2(4), 303-314.

Crone, T. M., and Voith, R. (1992). Estimating House Price Appreciation: A Comparison of Methods. *Journal of Housing Economics*, 2(4), 324-338.

DMLC (2016). Notes on Parameter Tuning – xgboost 0.71 documentation, from: https://xgboost.readthedocs.io/en/latest/tutorials/param_tuning.html accessed: July 15. 2019.

Eurostat (2015). Distribution of Population by Tenure Status, Type of Household and Income Group, EU – SILC survey.

Fisher, J. D., Geltner, D. and Webb, R.B. (1994). Value Indices of Commercial Real Estate: A Comparison of Index Construction Methods. *Journal of Real Estate Finance and Economics*, 9(2), 137-164.

Freund, Y. and Schapire, R.E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences*, 55, 119-139.

Friedman, J., Hastie, T. and Tibshirani, R. (2001). The Elements of Statistical Learning, in *Springer Series in Statistics* (Vol 1). New York: Springer.

Friedman, J.H. (2001). Greedy Function Approximation: A Gradient Boosting Machine, *The Annals of Statistics*, 29(5), 1189-1232.

Gatzlaff, D.H. and Ling, D.C. (1994). Measuring Changes in Local House Prices: An Empirical Investigation of Alternative Methodologies. *Journal of Urban Economics*, 35(2), 221-224.

Geltner, D. (2015). Real Estate Price Indices and Price Dynamics: An Overview from an Investments Perspective. *Annual Review of Financial Economics*, 7(1), 615-633.

Geurts, P., Ernst, D. and Wehenkel, L. (2006). Extremely Randomized Trees, *Machine Learning*, 63(1), 3-42.

Glorot, X. and Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks, *Proceedings of Machine Learning Research (PMLR)*, 9, 249-256.

Glorot, X., Bordes, A. and Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 15, 315-323.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep Learning, *Nature*, 521(7553), 800.

Graczyk, M., Lasota, T., Trawiński, B. and Trawiński, K. (2010). Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal, In: *Intelligent Information and Database Systems, LNCS* (Vol. 5991), Nguyen, N.T., Le, M.T. and Świątek, J. (eds.), Springer Heidelberg.

Hauck, T. (2014). Scikit-Learn Cookbook. Packt Publ. (2nd ed.)

Inoue, A. and Kilian, L. (2008). How Useful Is Bagging in Forecasting Economic Time Series? A Case Study of U.S. Consumer Price Inflation, *Journal of the American Statistical Association*, 103, 482-511.

Jansen, S.J., De Vries, P., Coolen, H.C., Lamain, C.J. and Boelhouwer, P.J. (2008). Developing A House Price Index for the Netherlands: A Practical Application of Weighted Repeat Sales, *Journal of Real Estate Finance and Economics*, 37(2), 163-186.

Kaggle (2018). Kaggle|Zillow Prize: Zillow's Home Value Prediction (Zestimate) - Kernels. Available from: https://www.kaggle.com/c/zillow-prize-1

Kahneman, D. and Tversky, A. (1972). Subjective Probability: A Judgment of Representativeness, *Cognitive Psychology*, 3(3), 430-454.

Kingma, D.P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization, *ICLR conference paper*.

Kok, N., Koponen, E.-L. and Martinez-Barbosa, C.A. (2017). Big Data in Real Estate? From Manual Appraisal to Automated Valuation, *The Journal of Portfolio Management*, 43(6), 202-211.

Lior, R., and Maimon, O. (2015). Data Mining with Decision Trees: Theory and Applications (Vol. 81), World scientific.

Ma, H., Chen, M. and Zhang, J. (2015). The Prediction of Real Estate Price Index Based on Improved Neural Network Algorithm, *Advanced Science and Technology Letters*, 81, 10-15.

Manganelli, B., De Mare, G. and Nesticò, A. (2015). Using Genetic Algorithms in the Housing Market Analysis, In *International Conference on Computational Science and Its Applications* – ICCSA 2015 (pp. 36-45), Gervasi, O., Murgante, B., Misra, S., Gavirlova, M.L., Rocha, A.M.A.C.R., Torre, C., Taniar, D. and Apduhan, B.O. (eds.), Banff, AB, Canada: Springer.

Matysiak, G. and Wang, P. (1995). Commercial Property Prices and Valuations: Analyzing the Correspondence, *Journal of Property Research*, 12, 181-202.

McAllister, P. and Tarbert ,H. (1998). Price or Appraisal Discovery? An Analysis of Lead/ Lag Relationships in the Property Market, RICS Cutting Edge Conference, de Montfort University.

Meese, R.A., and Wallace, N.E. (1997). The Construction of Residential Housing Price Indices: A Comparison of Repeat-Sales, Hedonic-Regression, and Hybrid Approaches. *Journal of Property*, 14(1/2), 51-73.

Mullainathan, S., and Spiess, J. (2017). Machine Learning: An Applied Econometric Approach, *Journal of Economic Perspectives*, 31(2), 87-106.

Nielsen, D. (2016). Tree Boosting with XGBoost: Why Does XGBoost Win "Every" Machine Learning Competition? *NTNU Tech Report*, (December), 2016.

Nielsen, M.A. (2015). Neural Networks and Deep Learning. Determination Press.

Northcraft, G.B. and Neale, M.A. (1987). Experts, Amateurs, and Real Estate: An Anchoring-And-Adjustment Perspective on Property Pricing Decisions, *Organizational Behavior and Human Decision Processes*, 39(1), 84-97.

NS3457 (2013). NS3457 Standard, Available from https://www.standard.no/nettbutikk/produktkatalogen/produktpresentasjon/?ProductID=665100.

Opitz, D. and Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study, *Journal of Artificial Intelligent Research* 11, 169-198.

Oust, A., Hansen, S.N. and Pettrem, T.R. (2020). Combining Property Price Predictions from Repeat Sales and Spatially Enhanced Hedonic Regressions. *Journal of Real Estate Finance and Economics*, 61, 183-207

Park, B. and Bae, J.K. (2015). Using Machine Learning Algorithms for Housing Price Prediction: The Case of Fairfax County, Virginia Housing Data, *Expert Systems with Applications*, 42(6), 2918-2934.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Van-derplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay E. (2011). Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, 12, 2825-2830.

Peterson, S. and Flanagan, A.B. (2009). Neural Network Hedonic Pricing Models in Mass Real Estate Appraisals, *Journal of Real Estate Research*, 31(2), 147-164.

Quan, D. and Quigley, J.M. (1991). Price Formation and the Appraisal Function in Real Estate Markets, *Journal of Real Estate Finance and Economics*, 4, 127-146.

Quigley, J.M. (1995). A Simple Hybrid Model for Estimating Real Estate Price Indexes. *Journal of Housing Economics* 4, 1-12.

Rattermann, M. (2007). Valuation by Comparison: Residential Analysis and Logic. Appraisal Institute.

Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition, *Journal of Political Economy* 82(1), 34-55.

Schapire, R.E. (1989). The Strength of Weak Learnability (Extended Abstract), *Machine Learning*, 227(October), 28-33.

Schapire, R.E. (2013). Explaining AdaBoost. Available from: https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf.

Sinnott, R.W. (1984). Virtues of the Haversine. *Sky and Telescope* 68(2), 159.

Slovic, P. and Lichtenstein, S. (1971). Comparison of Bayesian and Regression Approaches to the Study of Information Processing in Judgment, *Organizational Behavior and Human Performance*, 6(6), 649-744.

S&P Dow Jones Indices (2021). S&P Corelogic Case-Shiller Home Price Indices, Available from: https://www.spglobal.com/spdji/en/index-family/indicators/sp-corelogic-case-shiller/sp-corelogic-case-shiller-composite/#overview.

Statistics Norway (2010). Befolkningsvekst rundt Oslo, [Population growth in the Oslo area], Available from https://www.ssb.no/befolkning/artikler-og-publikasjoner/befolkningsvekst-rundt-oslo.

Statistics Norway (2018). Population And Population Changes – Quarterly. Available from: https://www.ssb.no/en/folkemengde/.

Tsatsaronis, K. and Zhu, H. (2004). What Drives Housing Price Dynamics: Cross-Country Evidence, *BIS Quarterly Review*, (March), 65-78.

Valier, A. (2020). Who Preforms Better AVMs vs Hedonic Models, *Journal of Property Investments and Finance*, 38(3), 213-225.

Wheeler, D., and Tiefelsdorf, M. (2005). Multicollinearity and Correlation Among Local Regression Coefficients in Geographically Weighted Regression, *Journal of Geographical System*, 7(2), 161-187.

Wolpert, D.H. (1992). Stacked Generalization, *Neural Networks*, 5(2), 241-259.

Appendices

Appendix 1 Evaluation of Artificial Neural Networks as AVMs

ANNs are a class of machine learning techniques that model learning tasks by combining a collection of units—known as *artificial neurons*—each of which applies a simple threshold function—known as *activation functions*—to its input. These neurons are typically organized in layers, with connections between neurons in adjacent layers which can transmit the outputs of a layer as inputs to the following layer, adjusted by a certain *weight*. The activation functions, learning method, and structure of the neurons and layers are determined by the user, while the model learns the weights of the network from the relevant training data. Although ANNs are known to have the ability to approximate any finite mathematical function[26], they can be challenging to apply to many learning problems. This is due to their complex training procedure and non-parametric structure.

Implementation: We conceptually follow Bengio (2012) and Goodfellow et al. (2016) to design our ANN and use the *MLPRegressor* package by *scikit-learn* for Python in the implementation. There are a few generally accepted practices for applying ANNs, such as normalization of the input data and the use of mini-batches.[27] For the hyperparameters of the model, we use a selection of default and recommended parameters[28], as well as experience combined with a grid-search and cross-validation. The most challenging task is the choice of the structure of the neurons; i.e., the number of layers, number of neurons per layer, and connections between each layer. The choices are numerous, and the use of a grid-search alone to determine the appropriate choice is infeasible due to the exponential increase in computational requirements. Our final model is the most stable with reasonably consistent results. We provide the selected hyperparameters in Table A.1.

Results and Discussion: The results of our work are presented in Table A.2, which also include the stacked AVM for comparison purposes. We see that our implementation of an ANN results in a far poorer performance than the AVM. Although there might be superior implementations of ANNs for this problem, we cannot determine this by any structured approach. Instead, we rely primarily on experience and grid-searches. When considering the solutions used in

[26] This is known as the universal approximation theorem (Cybenko, 1989).

[27] Mini-batches are randomized subsamples of the training data, which allow the network to train faster with less memory.

[28] The activation functions of the neurons are set to *ReLU* (Glorot et al., 2011) and the weight optimizer is set to *Adam* (Kingma and Ba, 2014). Both choices are well-established for regression problems, although even these have several prominent alternatives.

several Kaggle-competitions (Kaggle, 2018), we find that ANNs are prevalent as submodels, but rarely used without a model-stacker. Furthermore, the recent literature shows that the inherent struggle of training ANNs is an established issue.[29]

Table A.1 ANN Hyperparameters

Variable	Description	Selected Value
Optimizer	Solver used for weight-optimization.	Adam (Kingma and Ba, 2014)
Activation function	Activation function used in the neurons.	ReLU (Glorot et al., 2011)
# of hidden layers	Number of layers.	4
Hidden layers	Number of nodes per layer.	[64, 64, 32, 32]
Maximum number of iterations	Maximum number of iterations of the training data	1,000
Learning rate	Step-size used to update weights.	0.01
Alpha	Regularization parameter to prevent overfitting of data	0.0001

Table A.2 Prediction Accuracy of Artificial Neural Network

	Within 5%	Within 10%	Within 20%	MdAPE	MAPE
AVM	46.89%	76.36%	96.31%	5.35%	7.17%
ANN	24.20%	46.18%	77.30%	10.96%	13.99%

We conclude from this exploration that, although ANNs are universal approximators, they require a great deal of experience to apply with success and lack a structured engineering process. Therefore, we believe ensemble learning methods are more consistent and straightforward to apply in practice. However, we believe that given a sufficient amount of training data, ANNs could be incorporated into an AVM.

[29] See Chapter 5 of Nielsen (2015) for a conceptual understanding of some of the challenges in designing ANNs, and Glorot and Bengio (2010) for a more technical treatment of the reasons.

Appendix 2 Implementation Overview

We provide a brief overview of the implementation of the AVM, with an emphasis on the choices related to the programming details, external dependencies, and hardware. Our stacked AVM is implemented in Python. The implementation is carried out with the aim to produce many estimates in parallel, to be able to quickly create value estimates for all dwellings in a given month, and therefore not optimized to create single estimates quickly. We make use of mpi4py[30], a standardized API for parallel computing, to divide the estimates into a given number of parallel cores, all running one instance of the AVM. To fully exploit the capabilities, we run the implementation on a HP bl685c G7 server computer, using four 2.2 GHz AMD Opteron 6274 CPUs, each with 16 logical cores.

We rely on two different libraries for the individual methods; SKlearn and XGBoost. SKlearn is a large library, consisting of packages for many commonplace statistical methods. We use:
i) **BaggingRegressor** - Contains all required methods for the BP algorithm
ii) **RandomForestRegressor** - Contains all required methods for the RF algorithm
iii) **ExtraTreeRegressor** - Contains all required methods for the ETs algorithm
iv) **GridSearchCV** - Contains the cross validation algorithm to search for hyperparameters
The XGB library is provided by an open source community, and contains all the necessary methods to both tune and run the XGB-algorithm.

[30] See http://mpi4py.readthedocs.io/en/stable/ for an overview of this package.

Figure A1 Stacked Generalisation – Python Code for the automated valuation modeling process

Python code used to implement Steps 3-5 in the automated valuation modeling process.

```
class Ensemble(object):
    def __init__(self, n_splits, stacker, base_models):
        self.n_splits = n_splits
        self.stacker = stacker
        self.base_models = base_models

    def fit_predict(self, X, y, T, comp_transer, test_unit):
        X = np.array(X)
        y = np.array(y)
        T = np.array(T)

        folds = list(KFold(n_splits=self.n_splits,
            shuffle=True).split(X, y))
        S_train = np.zeros((X.shape[0], len(self.base_models)))
        S_test = np.zeros((T.shape[0], len(self.base_models)))

        for i, clf in enumerate(self.base_models):
            S_test_i = np.zeros((T.shape[0], self.n_splits))

            for j, (train_idx, test_idx) in enumerate(folds):
                X_train = X[train_idx]
                y_train = y[train_idx]
                X_holdout = X[test_idx]
                y_holdout = y[test_idx]

                clf.fit(X_train, y_train)
                y_pred = clf.predict(X_holdout)[:]

                S_train[test_idx, i] = y_pred

                predictions = clf.predict(T)[:]
                S_test_i[:, j] = predictions
            S_test[:, i] = S_test_i.mean(axis=1)

        if run_RS:
        rs_train_pred = []
        for id, unit in comp_transer.iterrows():
            rs_train_pred.append(list(prepred_rs_dict[id].values()))
```

(*Continued…*)

(Figure A1 Continued)

```
        rs_test_pred =
            [list(prepred_rs_dict[test_unit.name].values())]
        S_train = np.concatenate((S_train, rs_train_pred), axis=1)
        S_test = np.concatenate((S_test, rs_test_pred), axis=1)
        x_and_s_train = np.concatenate((S_train, X), axis=1)
        x_and_s_test = np.concatenate((S_test, T), axis=1)

        self.stacker.fit(x_and_s_train, y)
        stack_ppsm = self.stacker.predict(x_and_s_test)[:]

        return stack_ppsm, S_test
```

Appendix 3 Hedonic Model (OLS) as AVMs

HPMs build on the assumption that goods are typically sold as a bundle of inherent attributes and their implicit price can be estimated from the observed prices of the characteristics associated with them (Rosen, 1974). The method recognizes that even though individual dwellings are inherently heterogeneous, their price might be predicted on the basis of the underlying attributes. These underlying characteristics describe both the structure and location of the dwelling and the time of the transaction. We use an HRM to obtain prices for the underlying characteristics of dwellings, including the changes in prices over time, and apply the model to predict the current selling price of any dwelling.

The model is specified as:

$$\ln p_h^t = \beta_0 + \sum_{\tau=1}^{T} \delta^\tau D_h^\tau + \sum_{k=1}^{K} \beta_k z_{hk}^t + \epsilon_h^t ,$$

where p_h^t is the price of the dwelling, the time dummy variable D_h^τ equals 1 if the transaction occurs in time period τ and 0 if not, and z_{hk}^t captures one of K location and structural characteristics for dwelling h and time t. We apply variables for sales year and month, dwelling size (log (USMs) and log(log(USMs), dummies for construction year with 10 year intervals, dummy sold as new, location dummies (administrative city districts, see Figure 1), dummies for the number of stories in the building (2 or less, 3 to 4 and more than 4 stories), dummies for location of apartment in the building (basement, ground floor or loft), dummy if the number of apartments in the building is more than 10 and dummy for elevator if the dwelling is located on a story higher than 3. The result of the predictability of the OLS model is reported in Table 6.