Article title: Property valuation by machine learning for the Norwegian real estate market
Authors: Amandip Sangha[1]
Affiliations: Askin, Norway[1]
Orcid ids: 0000-0002-8058-1751[1]
Contact e-mail: amandip.s.sangha@gmail.com

DOI: 10.14293/S2199-1006.1.SOR-.PP0TP9I.v1
Preprint first posted online: 13 September 2021
Keywords: property valuation, machine learning, real estate, artifical intelligence, house price prediction, property price prediction, automated valuation model, property price prediction

Property valuation by machine learning for the Norwegian real estate market

September 13, 2021

Contents

1 Abstract

1.1 Purpose

We train a machine learning model on large data set for predicting property values in the Norwegian real estate market. Our model is a gradient boosted regression tree. The data set is the largest market data set of properties in Norway considered in the research literature. We achieve state of the art accuracy.

1.2 Methodology

A large scale market data set of real estate properties is collected from sales and rental ads on publicly accessible internet sites. The property advertisements show

property features and appraisal values made by real estate brokers. We train a gradient boosted regression tree model on selected features of the data set. This is a multivariate regression model built with supervised learning. We do 5-fold cross validation to assess the accuracy and robustness of the model.

1.3 Findings

The gradient boosted regression tree models are already known to give the best prediction accuracy on real estate price valuations. We achieve state of the art prediction accuracy using a minimal feature set and only publicly and freely available sales advertisement data.

1.4 Originality

The novelty of our work lies in the fact that we use a minimal feature set in our model, and we have the largest data set in the research literature, and moreover we have used only freely and publicly accessible data which are simple to obtain. This shows that useful estimation models with high accuracy can be built with quite simple resources.

2 Introduction

We explore a recent and large data set of Norwegian real estate properties for the purpose of price prediction. In our model we use intrinsic property features such as property size, year built, number of bedrooms, floors and shared costs, and geographical attribute such as the location given by postal code, and one temporal attribute, namely the date of publication of the ad. We have intentionally kept the number of features in our model to a minimum, as we wish to assess the level of prediction accuracy possible to attain with the most minimal feature set.

There are many similar works that explore and compare the accuracy of machine learning models for predicting property values on data sets from countries around the world. We now give a brief background on these related studies and the relevant research literature. One of the most recent additions to the literature is [2]. There the authors investigate six different estimation methods for valuation based on a large data set of properties in Switzerland. The prediction accuracy, robustness and volatility are assessed, and the main conclusion is that the gradient boosting method yields the greatest accuracy.

In [4] the authors compare the prediction performance of various machine learning algorithms on data from the residential property market in Santiago, Chile. The comparison is between ordinary least squares regression, support vector machines, a neural network and random forest. The results of the analysis show that the random forest method outperforms the rest in terms of accuracy.

The article [5] discusess automated valuation models (AVM) versus the traditional real estate appraisal approach. Their data is collected from California,

Florida and Texas, USA. The methods they compare, are linear regression, random forest, gradient boosting, and extreme gradient boosting. The latter is concluded to be the superior method.

In [3] the authors use the classical hedonic regression method, in their case a multivariate linear regression with normally distributed error term. The data set originates in the city of Lublin, Poland, and is a small data set consisting of 1211 observations. Their model consists of 8 features, and statistical analyses are conducted to establish variable significance. Finally the authors conclude that the model is rather poor as it has a low R^2 score 0.239566 even though estimation error residuals seemed somewhat acceptable.

In [9] the authors consider a multilayered artificial neural network (ANN). Their data set is fairly large, consisting of 464467 residential properties in North Carolina, USA. Their ANN model achieved an R^2-score in the range 0.70 to 0.80.

The article [7] seems to be the first systematic application of random forests to the task of property valuation. The study compares numerous methods, such as KNN, decision trees, random forests, multi layer perceptron, linear regression, to mention a few.

In [10] the authors consider market data from Norway. They consider a much smaller data set of 15786 records restricted to Oslo between 2016-2017. They also use many more features in their model, namely up to 14 features as compared to our 5-7 features for the various property types. Their model is also a gradient boosted tree model which is built by stacking several submodels. We achieve comparable accuracy in the present work, though a direct comparison is difficult as their data set is much more restricted in volume and context.

Many similar studies on automated valuation models are to be found in the research literature, that compare the prediction accuracy of various models. We do not elaborate more on these here, but refer the interested reader to the research literature.

A few important remarks on the aforementioned literature is in order. It is noteworthy that most of this research is consistently pointing to the best prediction models being those that are based on some form of decision trees, e.g. random forest, often accompanied by some form of boosting, e.g. gradient boosting. Not surprisingly, our experiments show the same evidence as well. The aforementioned studies do however differ somewhat in their model configurations and features explored. Why tree based methods are consistently coming out on top, is indeed an interesting question. We elaborate on this question in the concluding remarks below.

It is also interesting to note that our best models reach a fairly high level of accuracy even with a small number of features. For instance, our best model for apartment price estimation has an R^2-score of 0.92 and shows test results where the mean error corresponds to approximately 7% of the average sales price. A central goal of our investigation is to keep the number of input features to a minimum. The motivation behind this constraint, is that when predictive analytics functionality is to be deployed into production systems, the availability of relevant data features can often be poor. It is thus of practical importance that predictive models are able to function on minimum input features.

Let us also remark here that we do not consider neural network models in the present paper, the reason being that there exists no "canonical" neural network architecture so that the neural network topology or architecture becomes a salient property itself. The search for an optimal, or even adequate, neural architecture becomes a demanding task in its own right.

The originality of the paper is two fold: 1. Our data set is the largest Norwegian real estate property data set to be considered in the research literature, for the purposes of developing and testing machine learning based price estimation. The data set consists of 384347 properties from all over Norway. The sheer volume of our data set in combination with the broad coverage of both sale and rental properties of various types, makes our data set and model analysis novel.

2. Our modelling is minimalistic in the number of features used, giving a greater accuracy-to-feature-dimensionality trade off. We thus illustrate how it is possible to base price estimation on a rather low-dimensional data set for real estate price estimation. The benefits of such a small-feature data set are that the analysis and prediction becomes easier and more robust. Moreover, such a data set is much easier to obtain, e.g by direct online capture.

3 Data

The data set consists of publicly available Norwegian property sales ads published on real estate agent websites. The sales price data was collected in the time period between March 2018 and September 2021. Note that our data consists of ads which means that the prices are ad listings of the property appraisal values by real estate brokers, and not actual sales transaction data.

Property type	**Number of records**
Freehold apartment sale	51146
Housing cooperative apartment sale	45942
Shared ownership apartment sale	3603
Freehold house sale	69600
Cabin sale	7797
Apartment rental	206259

Table 1: Data volume

We have restricted our attention to a small yet essential set of features which are intrinsic to the property, except for *postal code* which encodes geographical placement of the property, and *date* of publication which encodes the point in time for the price of the property object in question. Geographical coordinates (WGS 84) were also tested instead of postal codes, but they did not yield a significantly higher accuracy in the model. Note also that we do not one-hot encode the postal code variable. On the contrary, decision trees can handle the integer valued postal code quite well as a standalone feature. As the decision tree partitions the space, the partitioning on the postal code value simply entails a large number of intervals

for the postal code. These decision rules based on the postal code dimension are able to correctly encode the neighborhoods and closeness of varying pricing areas geographically.

Feature	**Data type**
Primary area (m^2)	integer
Year built	integer
Number of bedrooms	integer
Shared costs	integer
Floor	integer
Postal code	integer
Date of publication	timestamp (float)

Table 2: Features - apartment sales price model

Feature	**Data type**
Primary area (m^2)	integer
Year built	integer
Number of bedrooms	integer
Postal code	integer
Date of publication	timestamp (float)

Table 3: Features - house sales price model

Feature	**Data type**
Property type	categorical
Primary area (m^2)	integer
Number of bedrooms	integer
Postal code	integer
Date of publication	timestamp (float)

Table 4: Features - rental price model

4 Model

The property value estimation problem is posed as a multivariate regression problem (supervised learning). We use *gradient boosted decision trees* for our regression model. *Decision trees* (or more precisely *regression trees* in our setup) partition the feature space into rectangles and fit a simple model such as a constant within each rectangle. The main work thus consists in finding a good partition of the feature space. Various strategies are employed, such as information gain criteria. Decision trees are often visualized as binary trees with a single rule or decision taking

place at each node. The function can thus be visualized as being one full traversal of the tree, starting at its root node. More precisely, the decision tree function is a linear combination of set indicator functions, where the sets are merely the rectangles in the partition. Decision and regresion trees are conceptually simple yet powerful, and can model highly non-linear relationships.

Gradient boosting is an ensemble of decision trees, but here the model is built iteratively by adding more trees ("weak learners") to the model successively. The model is an additive ensemble of distinct trees where in each step an optimization problem is solved by gradient descent by minimizing the difference between the true target and the prediction at the previous step. In our gradient boosting model, we have 5000 decision trees.

It should be clear from our feature set which consists of intrinsic characteristics of the real estate property, that we have no explicit macroeconomic feature variables such as population growth rates, real estate development rates, interest rates and so on.

We build our models by cross validation training on the data set, where we have used 80% of the data for training and the remaining (held out) 20% for testing, for each iteration. Each model is trained in 5 separate cross validation training runs. The error statistics are then averaged over these 5 independent training runs, wherein the training-test split (80-20) is done independently in each run.

5 Results

We present the results in tabular form. Each model is given with the *mean absolute error* (MAE), standard deviation of the MAE and the R^2 coefficient of determination. Recall that the *absolute error* between the true value y and the estimated value $\hat{y}$ is just the absolute value of the difference $|y - \hat{y}|$. The mean absolute error (MAE) is then calculated as the mean of all the absolute errors on the test set. The R^2 coefficient is a much used measure of goodness of fit. It measures the proportion of variance in the target variable that has been explained by the independent variables in the model, and is defined as

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

where the sum is taken over the test set, and $\bar{y}$ is the mean of the test set true values. The best possible R^2 score is 1.0. A constant model always predicting the mean value, would get a score of 0.0. A model which is worse than the constant mean model, would give a negative score.

For each model, we also show the error percentiles: 10th, 25th, 50th, 75th and 90th. The reason we include these percentiles, is due to their practical importance. It is of great interest to know "how far off" the price estimate might be. Clearly any practical application of such a prediction would need to be quantified in terms of error margins. These percentiles serve that purpose. Note that the error figures are calculated on the test set which are samples previously unseen by the trained model. When the test set has reached a fairly large size, we may make

statements such as: "In 10% of the cases, the model makes no greater price estimation error than 40 000", and "In 50% of the cases, the model makes no greater price estimation error than 200 000". Such statements illustrate the accuracy of the model in an easy to grasp manner which is highly relevant to the practical case at hand, namely valuation of real estate. Thus, the error percentiles give the most important information about the model error for practical purposes. The following results show that the best models here are viable for practical use cases. Indeed, the errors seem not much greater than what a human real estate agent might produce, given the limited input features. The accuracy for house sales prices is lower than for apartments. Note that we have used even fewer features for houses than for apartments. For instance, we have not included how many stories a house has, whether there is a garden, parking spaces for cars, and so on. The main reason for omitting these features, is data availability and consistency.

Property type	MAE	MAPE	RMSE	R^2
Freehold apartment sale	288333	0.087015	406041	0.902707
Housing cooperative apartment sale	209072	0.099069	288587	0.922291
Shared ownership apartment sale	271880	0.075492	405264	0.913126
Freehold house sale	589291	0.186160	845908	0.826615
Cabin sale	518582	0.373133	697988	0.628049
Apartment rental	1238	2.460752	1952	0.751623

Table 5: Property value prediction performance

Property type	10p	25p	50p	75p	90p
Freehold apartment sale	36121	91612	206128	385964	637224
Housing cooperative apartment sale	27987	70703	153538	283503	458687
Shared ownership apartment sale	31762	80667	180902	350273	560147
Freehold house sale	75277	190286	421301	778566	1255311
Cabin sale	61097	162238	376824	696774	1066528
Apartment rental	160	407	883	1591	2517

Table 6: Prediction error percentiles

Property type	Average price	Average price error
Freehold apartment sale	4051324	7.11%
Housing cooperative apartment sale	2674448	7.81%
Shared ownership apartment sale	3968498	6.85%
Freehold house sale	4340743	13.57%
Cabin sale	2112316	24.55%
Apartment rental	10482	11.81%

Table 7: Average price prediction error

5.1 Sample size and accuracy

It is interesting to ascertain how prediction accuracy relates to data volume. We can see how much data is needed before the model achieves a certain level of accuracy. We approach this question for the gradient boosting model by random subsampling at intervals of 10% of the total data volume iteratively. At each subsample level, we train the gradient boosting model with 5 folds of cross validation where the train-test split is done randomly each time, and then average the resulting R^2 and MAE to see how these metrics develop with sucessively increasing data volume. We note that the MAE is not significantly approved relative to the property prices after including 50% of the data set.

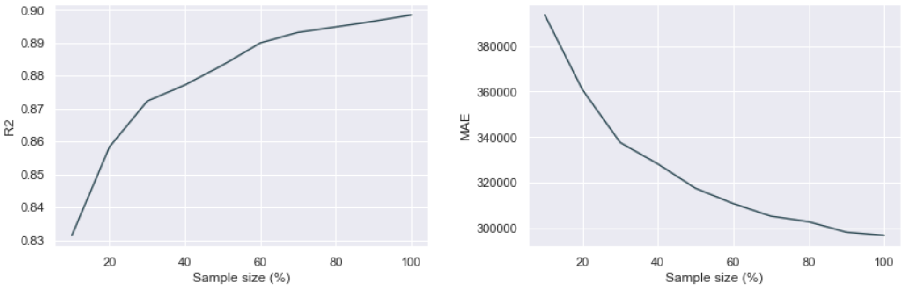

Figure 1: Apartment sales price - gradient boosting - R^2 and MAE

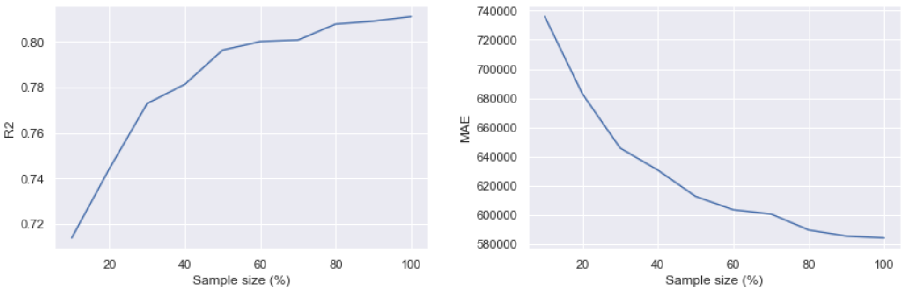

Figure 2: House sales price - gradient boosting - R^2 and MAE

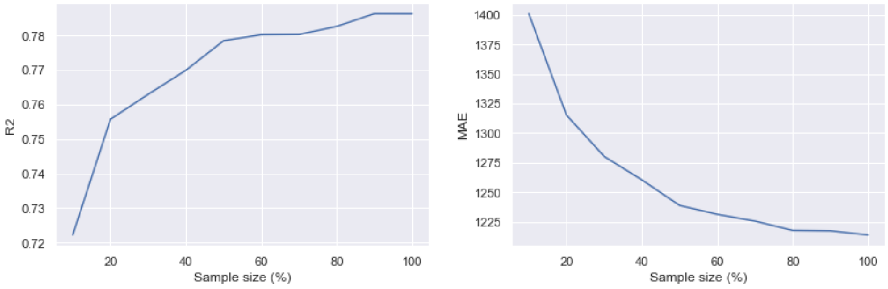

Figure 3: Rentals - gradient boosting - R^2 and MAE

5.2 Feature importances

An important facet of machine learning and indeed of statistical modelling in general, is *explainability*. Generally speaking this concerns any insight into the internals of a predictive model: how and why the model is predicting the results. It is also highly relevant to the practical case of property valuation, to be able to see which attributes of a given property are affecting the price estimate. *Feature importance* is one such metric which explains the significance each of the features hold for the predictions. Recall that the gradient boosting regressor is an ensemble of regresion trees. Hence, the feature importances in our model are averages of the importances contributed from the trees in the ensemble. For each such regresion tree, the importance of a feature is calculated as the total reduction in node *impurity* brought by the feature. Impurity refers to some chosen criterion by which splitting is selected. As we are dealing with a regression problem, our criterion is variance reduction by MSE (mean square error). Hence, at each node, a split is selected which reduces the variance the most at that node. The feature importance is then calculated as the decrease in node impurity weighted by the probability of reaching said node, summed over all nodes and normalized. These are the Gini importances, also called *mean decrese impurity*.

We note a very interesting observation on the feature importances of the models. Note that the *postal code* feature is the most important feature for the sales prices, by a great margin over the other model features. The postal code encodes the geographical location of the property. We can thus observe that the location of a property is a key factor in determining the desirability of the property, and hence its market value. Our predictive model quantifies this belief, or domain knowledge, explicitly in terms of feature importance given in percentage. The feature importances are useful tools in analyzing how the various property attributes impact the valuation. It is interesting to note that even though the human real estate agent knows the attractiveness of the various locations and areas in a city, it is by way of a predictive model we can assign statistically valid measures to these location desirability beliefs.

Feature	Importance (%)
Postal code	48.17
Area (m^2)	29.15
Shared costs	11.00
Year built	6.31
Number of bedrooms	2.28
Floor	2.13
Date	0.95

Table 8: Feature importance for apartment sales

Feature	Importance (%)
Postal code	64.08
Area (m^2)	22.11
Year built	10.44
Date	2.69
Number of bedrooms	0.67

Table 9: Feature importance for house sales

Feature	Importance (%)
Area (m^2)	38.69
Postal code	28.09
Date	23.41
Number of bedrooms	6.85

Table 10: Feature importance for rentals

6 Discussion and conclusions

The gradient boosting model gives the lowest MAE and the highest R^2. This conclusion is thus in line with similar studies, where also tree models with boosting have proven to be the most successful prediction models for real estate valuation. We have restricted to a very small set of features. Including more features can potentially lead to more accurate models, but it is crucial to note that some features may turn out to be reduntant in terms of importance. Hence feature pruning is of relevance. While a rich feature set is certainly an attractive trait, the model becomes more demanding to use in practice. If such a model is to be deployed as a tool, more features mean more input information is required by the system. This information can often be hard to obtain for the properties in question. This is a strong point of the minimum-feature models.

It is interesting to observe that tree models give the best accuracy for predicting property values, quite broadly in the research literature. Regression trees are able to model these non-linearities quite well, as trees are combinations of step functions. Basically, when a real estate property lies within a certain range of structural values (size, bedrooms, location), the property's value then also lies within a certain range. Combinations of step functions, and hence a regression tree, are able to model this situation well.

Real estate markets are characterized by relative illiquidity and low turnover. The transactions occur at irregular time steps. These factors make the application of data driven machine learning models difficult in this domain. From the data point of view, the valuation of a specific property is, among other things, a matter of finding recently traded properties of comparable attributes. As the data set is quite sparse because properties are not traded so often, machine learning models do tend to lack rich training data. Nonetheless, the machine learning approach to property valuation should be further developed to provide a more robust, scalable and accurate method of property valuation, over the traditional hand crafted structure models of simple linear types.

References

[1] Rosen S. (1974) "Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition". The Journal of Political Economy, Vol. 82, No. 1., pp. 34-55

[2] Mayer M., Bourassa S. C., Hoesli M. and Scognamiglio D. (2019) "Estimation and updating methods for hedonic valuation", Journal of European Real Estate Research, Vol. 12 Issue: 1, pp.134-150

[3] Belniak S. and Wieczorek D. (2017) "Property valuation using hedonic price method – procedure and its application", Czasopismo Techniczne Vol. 6

[4] Masias V. H., M. Valle A., Crespo F., Crespo R. (2016) "Property Valuation using Machine Learning Algorithms: A Study in a Metropolitan-Area of Chile", Selection at the AMSE Conferences-2016

[5] Kok N., Koponen E-L., Martinez-Barbosa C. A. (2017) "Big Data in Real Estate? From Manual Appraisal to Automated Valuation", The Journal of Portfolio Management Special Real Estate Issue 2017

[6] Graczyk M., Lasota T., Trawinski B. (2009) "Comparative Analysis of Premises Valuation Models Using KEEL, RapidMiner, and WEKA", ICCCI 2009: Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems pp 800-812

[7] Antipov E. A., Pokryshevskaya E. B. (2012) "Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics", Expert Systems with Applications Volume 39, Issue 2, 1 February 2012, Pages 1772-1778.

[8] Robson G., Downey M. L. (2008) "Automated Valuation Models: an international perspective". RICS Automated Valuation Models Conference: AVMs Today and Tomorrow

[9] Petersen S., Flanagan A. B. (2009) "Neural Network Hedonic Pricing Models in Mass Real Estate Appraisal", Journal of European Real Estate Research Vol. 31 No. 2

[10] Birkeland K., D'Silva, A. D. "Developing and Evaluating an Automated Valuation Model for Residential Real Estate in Oslo". NTNU, 2018