# A comparison of data mining methods for mass real estate appraisal

del Cacho, Carlos

Media Net Software

11 December 2010

# A comparison of data mining methods for mass real estate appraisal

Carlos del Cacho {delcacho@gmail.com}, December 11th, 2010

*Abstract*

**We compare the performance of both hedonic and non-hedonic pricing models applied to the problem of housing valuation in the city of Madrid. Urban areas pose several challenges in data mining because of the potential presence of different market segments originated from geospatial relations. Among the algorithms presented, ensembles of M5 model trees consistently showed superior correlation rates in out of sample data. Additionally, they improved the mean relative error rate by 23% when compared with the popular method of assessing the average price per square meter in each neighborhood, outperforming commonplace multiple linear regression models and artificial neural networks as well within our dataset, comprised of 25415 residential properties.**

******

Automated real estate valuation models are gaining attention both in academic and business circles as a result of the release of massive amounts of data over the Internet that were not previously available. Potential applications of mass appraisal systems are *ad valorem* taxation methods and fast prescreening for mortgage requests resulting in lower costs for the incumbent parties, as they do not require an inspection of the property under consideration. They can also be used by investors to determine which houses can be considered as potential investments for buy to let scenarios without resorting to expensive certified appraisals as a first step. Both a housing rents model and a housing sales model could be generated for that specific purpose.

## Introducing the dataset

Let us first present the variables measured. The city of Madrid is divided into 21 administrative districts. As real estate appraisers already know, location is a factor of paramount importance when determining the pricing of a property, aside of other considerations. Therefore we decided to train each regression model on each district individually. Furthermore, districts are subdivided in neighborhoods, which are fed into the model as dummy variables.

We collected all properties on sale for Madrid in one of the most widely trafficked real estate portals in Spain on the date of November 10$^{th}$ of 2010. The resultant data was parsed and relevant information was extracted and converted to Weka's ARFF file format. Our dataset will be made available for researchers to use in our web site.

Here is the breakdown of available properties by district:

| District | Number of properties |
|---|---|
| Arganzuela | 1198 |
| Barajas | 382 |
| Carabanchel | 1856 |
| Centro | 1904 |
| Chamartín | 1528 |
| Chamberí | 1274 |
| Ciudad Lineal | 1689 |
| Fuencarral | 1263 |
| Hortaleza | 1684 |
| Latina | 1276 |
| Moncloa | 1366 |
| Moratalaz | 355 |
| Puente de Vallecas | 1394 |
| Retiro | 973 |
| Salamanca | 1865 |
| San Blas | 1224 |
| Tetuán | 1589 |
| Usera | 872 |
| Vicálvaro | 247 |
| Villa de Vallecas | 617 |
| Villaverde | 859 |

The information for each property is expressive with regards to the different housing characteristics, and resulted in the following variables being encoded:

- **Price**: Quoted sales price in the web site. Numerical.
- **Area**: Built surface in meters. Numerical.
- **Bedrooms**: Number of bedrooms. Numerical.
- **Bathrooms**: Number of bathrooms. Numerical
- **Closets** : Number of closets. Numerical.
- **Garage**: Number of parking places. Numerical.
- **Terrace Area**: Surface of the terrace, if available. Numerical.
- **Floor With Elevator**: Positive number indicating the floor where the flat is located if the building has an elevator. Only relevant for flats, zero otherwise. Numerical.
- **Floor Without Elevator**: Positive number indicating the square of the floor where the flat is located if the building does not have an elevator. The rationale is that a first floor without an elevator is not nearly as worse as a fifth one, therefore the relationship is not likely to be linear. Only relevant for flats, zero otherwise. Numerical.
- **Parcel Area**: Surface of the housing lot. Only relevant for unifamiliar buildings, zero otherwise. Numerical
- **Garden Area**: Surface of the garden if available. Numerical.

- **Air Conditioning**: If air conditioning is available. Binominal.
- **Alarm**: If the property has an alarm protection system. Binominal.
- **Basketball**: If the property has a basketball court. Binominal.
- **Clothes Line**: If the property has a clothes line. Binominal.
- **Concierge**: If the property has a hired concierge. Binominal.
- **Floor Material**: Type of floor. Nominal.
- **Football**: If the property has a soccer court. Binominal.
- **Furnished Kitchen**: If the kitchen is equipped. Binominal.
- **Golf**: If the property has a golfing area. Binominal.
- **Green Area**: If there are green areas nearby. Binominal.
- **Gym**: If the property has a gym. Binominal.
- **IntExt**: Attribute only relevant for flats that says if it is exterior on interior (facing the street or not). Binominal.
- **Neighborhood**: Name of the neighborhood where the property is located. Nominal.
- **Orientation**: Whether the property is facing north, south, east, etc. Nominal.
- **Padel**: Whether the property has a padel court or not. Binominal.
- **Reformed**: Whether according to the advertisement description the property has been reformed. Binominal.
- **Satellite Dish**: If the property contains a satellite dish. Binominal.
- **Security Door**: If the property has a reinforced door. Binominal
- **Squash**: If the property has a squash court. Binominal.
- **State**: State of the property. One of the following: New, Good or To Reform. Nominal.
- **Swimming Pool**: Whether the property has a swimming pool. Binominal.
- **Tennis**: Whether the property has a tennis court. Binominal.
- **Type**: One of the following: House, Duplex, Studio, Flat, and Attic. Nominal.
- **Year Built**: Categorical attribute indicating a rough estimate of the years of the building. One of the following: less than 5 years, between 5 and 10 years, between 10 and 20 years, between 20 and 30 years, more than 30 years. Nominal.

The data above, being rather comprehensive in itself, was further enhanced with GIS information coming from the street map Nomecalles of the Community of Madrid. Geospatial variables added were the following:

- **Metro Distance**: Distance in meters to the nearest subway station. Numerical.
- **Metro Station**: Name of the nearest subway station. Nominal.
- **Renfe Distance**: Distance in meters to the nearest railway station Numerical.
- **Renfe Station**: Name of the nearest railway station. Nominal.
- **Businesses**: Number of retail businesses in a 500 m radius from the property

## The contenders

The algorithms applied where the ones listed below. The validation procedure followed was tenfold cross-validation using stratified sampling, which is a well recognized benchmark for estimating data mining performance from small data sets. For the nearest neighbor algorithms leave one out cross-validation was used instead, given that no training time is required.

- Naïve Neighborhood
- Multiple Linear Analysis.
- Multilayer perceptron.
  - Unbagged
  - Bagged
- M5 Model trees.
  - Unbagged
  - Bagged
- K-Nearest Neighbors.
  - SVM
  - Genetic Algorithm
- Local Multiple Linear Analysis

Before advancing in the discussion, we briefly present the fundamentals of each and every single one of them and the way they were employed in our problem domain where applicable.

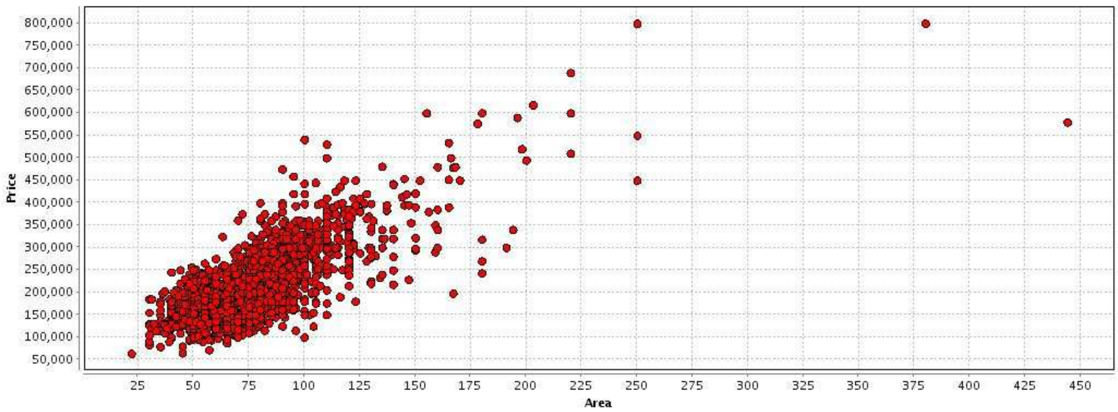
### Naïve Neighborhood

Online real estate portals typically display the price per square meter in each neighborhood as a coarse measure to indicate whether a property is overpriced or underpriced with regards to its peers. Given the mean price per square meter we can predict the price just by multiplying this quantity by the built surface of the home. For the lack of a better name, this process was called the Naïve Neighborhood algorithm, which performs surprisingly well given its simplicity. It is included as a comparison because most web sites report their statistics in this fashion.

### Multiple Linear Analysis

MLA is the industry standard in mass appraisal and has been extensively applied for this task during the last couple of decades. We introduced it in the study as a comparison baseline to see if other algorithms provided better results.

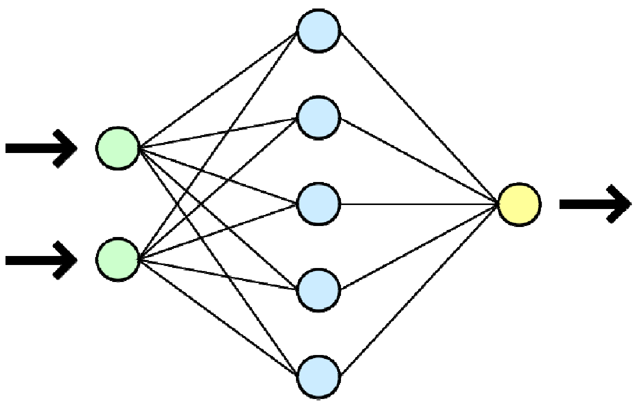
The idea is that we can decompose the pricing of an item into several constituents and estimate the relative weight of each of them in the final valuation. Such a way of proceeding is often called and hedonic pricing model. The price is estimated as a linear combination of these factors with the overall goal of minimizing the quadratic error. The parameters can be then estimated using the ordinary least square method. We used the Linear Regression learner as well as the Pace Regression from Weka. As can be seen from the scatter plot for the properties in the district of Carabanchel, the connection between the area and the price follows a linear relationship quite well. It stands to reason that an hedonic model can therefore approximate the final sales price.




The attractiveness of MLA stems from the fact that it creates a model comprehensible even for the lay man and gives a disaggregated output where each factor is assigned a specific price, contributing a fixed and known amount to the final output.

## Multilayer perceptron

The MLP is a biologically inspired feed forward kind of artificial neural network composed of several levels of neurons, typically three, where each neuron is connected to all the neurons in the preceding and succeeding layers. It internally combines all its inputs by processing them through an activation function and propagates the results to the neurons in the next layer. The weights can be learned through gradient descent by means of the back propagation method.



References to the use of MLPs in mass appraisal systems in the literature yield disparate results. Some claim that they increase accuracy when compared to multiple regression analysis, while others report otherwise. The attractiveness of MLPs is that they are able to estimate outputs that show non-linear relationships with regards to its inputs, but critics often state that they act as a black box and the way they arrive to their prediction is not easily understood by humans.

MLPs can be used both for classification tasks, where there is one neuron in the output layer per class instance, and for regression, with only one neuron as the output. For our setup a network with five neurons in the hidden layer was used. To avoid overtifitting, twenty percent of the training set was used to trigger an early stopping criterion if the quadratic error was not reduced for a fixed number of epochs.

To reduce the impact of random weight initialization we found out that the performance could be improved significantly by using ensemble learning, more concretely the technique known in the data mining community as Bagging (a contraction for bootstrap averaging). Bagging is a process whereby a learner is applied to random subsamples of the training set, resulting in N different models, and the final prediction is arrived at through averaging the output of those said models. It is a way to reduce variance and avoid overfitting, a malaise common to the usage of neural networks. In this case the networks were deliberately overfitted.

## M5 Model Trees

Model trees are a special kind of decision tree that approximate a function at the leaves through multiple linear regression. They are ideally suited to the task of mass appraisal with disjoint clusters because they identify segments of similar properties along the decision path and are able to learn non-linear relationships by applying regression multiple times for different variable ranges. Acciani et al [1] applied them to estimate the land value of vineyards in South Italy with promising results.

Other authors have witnessed non linear relationships between area and price, and model trees can exploit these relationships by branching on the area variable. We confirm these findings. The strength of the connection appears to be higher for the top quintile of the area distribution in all districts. Also higher priced districts result in higher correlations between price and area on average. Computing the correlation between price and area independently for each area quintile we get the following table:

| District | 0-20% | 20-40% | 40-60% | 60-80% | 80-100% |
|---|---|---|---|---|---|
| Salamanca | 0.574 | 0.353 | 0.146 | 0.288 | 0.866 |
| Chamartín | 0.417 | 0.315 | 0.458 | 0.536 | 0.768 |
| Chamberí | 0.494 | 0.395 | 0.388 | 0.335 | 0.809 |
| Retiro | 0.632 | 0.430 | 0.367 | 0.334 | 0.872 |
| Centro | 0.538 | 0.255 | 0.328 | 0.304 | 0.706 |
| Moncloa | 0.379 | 0.515 | 0.423 | 0.275 | 0.623 |
| Hortaleza | 0.170 | 0.246 | 0.158 | 0.699 | 0.705 |
| Arganzuela | 0.478 | 0.268 | 0.213 | 0.226 | 0.948 |
| Tetuán | 0.501 | 0.218 | 0.314 | 0.439 | 0.768 |
| Fuencarral | 0.327 | 0.324 | 0.430 | 0.700 | 0.805 |
| Ciudad Lineal | 0.267 | 0.177 | 0.232 | 0.383 | 0.814 |
| Barajas | 0.466 | 0.252 | 0.274 | 0.444 | 0.482 |
| San Blas | 0.154 | 0.338 | 0.161 | 0.439 | 0.814 |
| Villa de Vallecas | 0.287 | 0.208 | 0.137 | 0.245 | 0.444 |
| Moratalaz | 0.219 | 0.270 | 0.361 | 0.508 | 0.783 |
| Carabanchel | 0.198 | 0.121 | 0.249 | 0.206 | 0.617 |
| Vicálvaro | 0.200 | 0.333 | 0.119 | 0.150 | 0.856 |
| Latina | 0.310 | 0.147 | 0.171 | 0.201 | 0.94 |
| Usera | 0.361 | 0.108 | 0.124 | 0.176 | 0.431 |
| Puente de Vallecas | 0.396 | -0.007 | 0.097 | 0.178 | 0.567 |
| Villaverde | 0.154 | 0.388 | 0.397 | 0.227 | 0.549 |

Correlation between price and area for each area quintile

While model trees are a robust regression method, we found ensemble learning to be helpful. Bagging unpruned decision trees provided the best performance over all the algorithms tested, as shown in the results table. As with neural networks, bagged model trees are not readily apprehensible and they have to be inspected through sensitivity analysis to gauge how the different factors affect prices.

## K-Nearest Neighbors

The K-nearest neighbors model is similar to the way human appraisers approach their estimates when relying on market values. They search for homes similar to the one being appraised (known as comparables) that have been sold recently, make adjustments to make up for the differences, and arrive at a final value. The main difference is in the way they obtain their set of comparables, relying mostly on their personal judgment, resulting in an error prone system.

In stark opposition, in this algorithm the set of comparables is determined automatically, which eliminates human bias and subjectiveness from the equation. The price is set to a weighted average of the comparables (in our case K=9) where the weight of each comparable depends on a distance measure indicating how similar the property is to the assessed one.

McCluskey and Anand [4] showed good results in mass appraisal by employing an Euclidean based K-nearest neighbor where the weights were determined by using a genetic algorithm.

In our setup, pure optimization through genetic algorithms resulted in poor results, because given the number of attributes to optimize for the number of generations required to reach a good approximation makes the search impractical in terms of time required for training. As an enhancement, the initial weights were set as variations of those determined by a support vector machine in a preprocessing step. All attributes were previously normalized. As an example, here are the top five attributes identified by the SVM as affecting the prices for the district of Ciudad Lineal, along with their respective weights:

| Attribute | Weight |
|---|---|
| 1. Area | 1.00 |
| 2. Bathrooms | 0.51 |
| 3. Garage | 0.21 |
| 4. Terrace Area | 0.20 |
| 5. Neighborhood = Arturo Soria | 0.17 |

For comparison purposes we include the results of the K-nearest neighbors algorithm with the support vector machine weights alone without further refining. While minor improvements are obtained through the usage of genetic algorithms, it is probably not worth the effort given the substantial additional time required to go through in the weighting process.

## Local Multiple Linear Analysis

Appraisers often perform linear regression on top of their small set of comparables. Unfortunately they also often display their error rates and the attained correlations within their training set and generalize from there, which is a highly misleading practice from a statistical point of view. Error reporting must always be done from a validation set independent from the training set being used.

As a variant of the algorithm presented above and to research whether linear regression on top of a set of comparables was more accurate than linear regression over the whole training set, we decided to use the K-nearest neighbor approach to identify the closest set of properties to the one being appraised and perform linear regression afterwards to arrive at the final estimate. The setup was the following: the 25 most salient features as identified by the support vector machine were used and the 100 closest neighbors determined the regression equation. We found out that as the number of comparables was reduced the error rates increased, hence deriving conclusions from a handful of properties by means of linear regression is a dubious method, because this also means that the degrees of freedom are lessened and we must use a smaller subset of the available attributes, thus decreasing the benefits of having a large dataset to begin with.

## Discussion

Firstly we present correlation figures. While undoubtedly important, they must be interpreted with caution. If we devise an estimation mechanism whereby we systematically fall short 50% from the final price, we will arrive at a correlation of 1, meaning that the two variables considered are linearly dependent to perfection. However, the algorithm would be a poor regressor indeed. A pathological example is exemplified by the neural network result in the district of Hortaleza, where it simultaneously attained a correlation of 0.913 and a mean relative error of 62.09% in out of sample data. If we were to trust correlation alone, we may conclude that the performance was rather good, when in fact the opposite is true. We now present a summarized view of the results. The full tables can be obtained from the appendix.

| Correlation Ranking | > 0.90 | 0.85 to 0.90 | 0.80 to 0.85 | < 0.80 |
|---|---|---|---|---|
| 1. ModelTreeBagged | 13 | 4 | 2 | 2 |
| 2. ModelTree | 12 | 5 | 3 | 1 |
| 3. PaceRegression | 11 | 4 | 3 | 3 |
| 4. LinearRegression | 11 | 5 | 2 | 3 |
| 5. NeuralNetworkBagged | 11 | 7 | 3 | 0 |
| 6. K-NearestNeighborsGA | 10 | 5 | 2 | 4 |
| 7. K-NearestNeighborsSVM | 9 | 6 | 3 | 3 |
| 8. NaiveNeighborhood | 11 | 5 | 0 | 5 |
| 9. NeuralNetwork | 4 | 7 | 4 | 5 |
| 10. LocalLinearRegression | 2 | 10 | 2 | 7 |

| Relative Error Ranking | Mean | Best District | Worst District |
|---|---|---|---|
| 1. ModelTreeBagged | 15.25% | 12.34% | 19.00% |
| 2. K-NearestNeighborsGA | 15.61% | 11.53% | 19.72% |
| 3. K-NearestNeighborsSVM | 15.83% | 11.52% | 19.77% |
| 4. ModelTree | 17.51% | 13.45% | 32.32% |
| 5. NeuralNetworkBagged | 18.97% | 12.79% | 33.76% |
| 6. NaiveNeighborhood | 19.82% | 16.03% | 24.87% |
| 7. PaceRegression | 21.59% | 14.14% | 32.51% |
| 8. LinearRegression | 22.39% | 14.08% | 40.36% |
| 9. LocalLinearRegression | 27.25% | 14.69% | 42.41% |
| 10. NeuralNetwork | 31.65% | 17.58% | 62.09% |

Bagged model trees came out on top in terms of correlation and relative error rates, confirming them as a firm contestant for mass appraisal purposes, even though they are often neglected in the literature. Neural networks performed poorly unless bagged, despite the fact that an early stopping criterion was used to avoid overfitting. Other network topologies aside from the multilayer perceptron can be tried. Activation functions other than the sigmoid can be expected to perform better, at least for the Area variable, which is highly correlated to the Price.

One point that must be stressed from the data is that the linear regressors (both the default one and pace regression) did relatively well on correlation rates, though they lagged behind other algorithms in relative error rates. Furthermore, of particular significance is the fact that they performed well below their mean for the whole city in highly priced districts. In the top seven districts ranked by price per square meter they averaged 28.14% and 26.14% mean relative error, respectively. The reasons behind this need to be investigated further.

## Conclusions and future work

In this paper we compared several algorithms for the problem of housing valuation. We found ensembles of model trees to be a competitive method for mass appraisal in urban areas, improving upon widely spread linear regression and neural network models. Nonetheless, given the good performance of the simplistic Naïve Neighborhood algorithm, encoding neighborhood and area together as was done for the floor and elevator attributes could yield an improvement and must be further investigated. The K-nearest neighbors approach, the second best performing algorithm in terms of relative error, could also benefit from other distance computation methods aside from the Euclidean one.

While an average of 15% of deviation from the quoted price in out of sample data may seem excessive at first sight, increases in accuracy are to be expected if data for several months is used instead of a snapshot at one given period, adjusting it properly for inflation to eliminate the influence of time. We must also account for the fact that offering prices were used instead of actual sales information, and as was verified by manual inspection, the asking price for many properties is clearly out of the market and they are therefore unsellable at their stated pricing points. It would be interesting to follow such outliers over the course of time to see if properties that the model judges to be overpriced effectively reduce their offering price over the upcoming months.

## References

[1] Acciani, Claudio et al (2008) - Model Tree: An application in real estate appraisal

[2] Bourassa, Steven (2002) - Do Housing Submarkets Really Matter?

[3] Limsombunchai, Visit et al (2004) - House Price Prediction: Hedonic Price Model vs. Artificial Neural Network

[4] McCluskey, William and Anand Sarabjot (1999) - The application of intelligent hybrid techniques for the mass appraisal of residential properties

[5] Parker, David (2006) - Automated Valuation Models: A Practitioner Perspective

[6] Peterson, Steven - Neural Network Hedonic Pricing Models in Mass Real Estate Appraisal

[7] van Wezel ,Michiel et al (2005) - Boosting the Accuracy of Hedonic Pricing Models

## Appendix

Mean correlation in out of sample data from tenfold Cross-validation

| | Arganzuela | Barajas | Carabanchel | Centro | Chamartín | Chamberí | Ciudad Lineal | Fuencarral | Hortaleza | Latina | Moncloa |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NaiveNeighborhood | 0.924 | 0.894 | 0.783 | 0.880 | 0.911 | 0.925 | 0.938 | 0.918 | 0.865 | 0.921 | 0.859 |
| LinearRegression | 0.913 | 0.904 | 0.887 | 0.896 | 0.924 | 0.922 | 0.947 | 0.926 | 0.911 | 0.727 | 0.891 |
| PaceRegression | 0.911 | 0.903 | 0.887 | 0.897 | 0.924 | 0.928 | 0.946 | 0.926 | 0.913 | 0.706 | 0.891 |
| NeuralNetwork | 0.807 | 0.873 | 0.802 | 0.802 | 0.907 | 0.900 | 0.90 | 0.859 | 0.913 | 0.866 | 0.820 |
| NeuralNetworkBagged | 0.888 | 0.875 | 0.879 | 0.901 | 0.938 | 0.928 | 0.943 | 0.929 | 0.931 | 0.866 | 0.874 |
| ModelTree | 0.918 | 0.824 | 0.877 | 0.896 | 0.914 | 0.925 | 0.95 | 0.93 | 0.931 | 0.874 | 0.905 |
| ModelTreeBagged | 0.924 | 0.835 | 0.885 | 0.91 | 0.934 | 0.94 | 0.952 | 0.942 | 0.916 | 0.882 | 0.91 |
| K-NearestNeighborsSVM (*) | 0.657 | 0.914 | 0.875 | 0.882 | 0.914 | 0.917 | 0.944 | 0.933 | 0.904 | 0.761 | 0.897 |
| K-NearestNeighborsGA (*) | 0.661 | 0.918 | 0.875 | 0.882 | 0.915 | 0.918 | 0.944 | 0.935 | 0.909 | 0.759 | 0.897 |
| LocalLinearRegression | 0.898 | 0.885 | 0.818 | 0.853 | 0.883 | 0.88 | 0.79 | 0.904 | 0.887 | 0.857 | 0.874 |

(*) Performance estimated from leave one out Cross-validation

Mean correlation in out of sample data from tenfold Cross-validation (Cont.)

| | Moratalaz | Puente Vallecas | Retiro | Salamanca | San Blas | Tetuán | Usera | Vicálvaro | Villaverde | Villa Vallecas |
|---|---|---|---|---|---|---|---|---|---|---|
| NaiveNeighborhood | 0.903 | 0.746 | 0.954 | 0.922 | 0.911 | 0.918 | 0.706 | 0.869 | 0.776 | 0.767 |
| LinearRegression | 0.919 | 0.773 | 0.915 | 0.926 | 0.869 | 0.932 | 0.848 | 0.843 | 0.882 | 0.793 |
| PaceRegression | 0.924 | 0.79 | 0.922 | 0.926 | 0.804 | 0.933 | 0.847 | 0.849 | 0.882 | 0.736 |
| NeuralNetwork | 0.854 | 0.736 | 0.874 | 0.821 | 0.858 | 0.899 | 0.743 | 0.791 | 0.798 | 0.797 |
| NeuralNetworkBagged | 0.904 | 0.812 | 0.918 | 0.904 | 0.910 | 0.93 | 0.816 | 0.829 | 0.876 | 0.871 |
| ModelTree | 0.918 | 0.812 | 0.901 | 0.912 | 0.916 | 0.933 | 0.849 | 0.874 | 0.877 | 0.75 |
| ModelTreeBagged | 0.938 | 0.829 | 0.932 | 0.934 | 0.94 | 0.939 | 0.78 | 0.899 | 0.878 | 0.782 |
| K-NearestNeighborsSVM (*) | 0.912 | 0.803 | 0.896 | 0.812 | 0.932 | 0.932 | 0.833 | 0.772 | 0.868 | 0.859 |
| K-NearestNeighborsGA (*) | 0.916 | 0.802 | 0.907 | 0.796 | 0.934 | 0.932 | 0.835 | 0.772 | 0.870 | 0.859 |
| LocalLinearRegression | 0.916 | 0.735 | 0.874 | 0.805 | 0.893 | 0.794 | 0.763 | 0.77 | 0.78 | 0.79 |

(*) Performance estimated from leave one out Cross-validation

Mean relative error in out of sample data from tenfold Cross-validation

| | Arganzuela | Barajas | Carabanchel | Centro | Chamartín | Chamberí | Ciudad Lineal | Fuencarral | Hortaleza | Latina | Moncloa |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NaiveNeighborhood | 17.79% | 18.98% | 20.97% | 20.11% | 20.92% | 19.54% | 17.52% | 17.48% | 21.58% | 18.80% | 24.87% |
| LinearRegression | 15.10% | 18.13% | 14.16% | 21.89% | 28.70% | 28.77% | 17.77% | 22.72% | 30.48% | 17.57% | 33.46% |
| PaceRegression | 14.98% | 17.44% | 14.15% | 20.90% | 26.87% | 27.15% | 17.52% | 19.94% | 27.74% | 17.31% | 32.51% |
| NeuralNetwork | 18.69% | 21.53% | 20.52% | 32.92% | 38.66% | 34.70% | 24.24% | 34.81% | 62.09% | 23.11% | 54.60% |
| NeuralNetworkBagged | 13.92% | 18.85% | 15.15% | 20.55% | 21.04% | 20.67% | 17.44% | 17.84% | 24.52% | 15.97% | 33.76% |
| ModelTree | 13.45% | 32.21% | 14.37% | 18.42% | 19.62% | 18.81% | 14.56% | 16.88% | 18.81% | 15.04% | 21.40% |
| ModelTreeBagged | 12.34% | 17.13% | 13.71% | 17.19% | 16.99% | 15.24% | 14.10% | 13.03% | 16.82% | 14.38% | 19.00% |
| K-NearestNeighborsSVM (*) | 12.81% | 15.21% | 14.36% | 18.17% | 18.06% | 17.33% | 14.87% | 14.76% | 16.57% | 14.34% | 19.77% |
| K-NearestNeighborsGA (*) | 12.77% | 14.08% | 14.38% | 18.17% | 18.03% | 16.78% | 14.82% | 14.61% | 16.05% | 14.13% | 19.72% |
| LocalLinearRegression | 14.69% | 19.08% | 19.14% | 29.29% | 36.43% | 35.47% | 28.07% | 25.18% | 42.41% | 17.99% | 40.92% |

(*) Performance estimated from leave one out Cross-validation

Mean relative error in out of sample data from tenfold Cross-validation (Cont.)

| | Moratalaz | Puente Vallecas | Retiro | Salamanca | San Blas | Tetuán | Usera | Vicálvaro | Villaverde | Villa Vallecas |
|---|---|---|---|---|---|---|---|---|---|---|
| NaiveNeighborhood | 16.61% | 21.83% | 16.03% | 19.48% | 20.52% | 19.43% | 21.29% | 16.69% | 19.24% | 18.93% |
| LinearRegression | 14.91% | 17.31% | 29.25% | 27.06% | 20.12% | 19.08% | 15.24% | 15.31% | 14.08% | 40.36% |
| PaceRegression | 14.32% | 16.94% | 28.27% | 26.44% | 27.25% | 18.49% | 15.36% | 15.80% | 14.14% | 27.04% |
| NeuralNetwork | 18.25% | 24.62% | 33.06% | 44.72% | 26.77% | 22.74% | 20.21% | 17.86% | 21.02% | 17.58% |
| NeuralNetworkBagged | 15.64% | 16.47% | 18.80% | 20.80% | 17.20% | 17.02% | 16.25% | 15.85% | 15.24% | 12.79% |
| ModelTree | 13.54% | 16.57% | 17.01% | 20.12% | 16.52% | 17.67% | 15.12% | 14.91% | 14.17% | 25.40% |
| ModelTreeBagged | 12.51% | 15.71% | 14.08% | 16.18% | 13.26% | 14.98% | 16.53% | 12.73% | 13.81% | 16.74% |
| K-NearestNeighborsSVM (*) | 13.42% | 15.64% | 15.71% | 18.68% | 13.34% | 15.15% | 15.30% | 11.52% | 14.20% | 12.78% |
| K-NearestNeighborsGA (*) | 12.73% | 15.61% | 15.08% | 18.00% | 13.18% | 15.23% | 15.06% | 11.53% | 13.98% | 12.21% |
| LocalLinearRegression | 14.75% | 22.09% | 29.24% | 33.76% | 21.32% | 27.49% | 19.46% | 21.08% | 18.80% | 17.20% |

(*) Performance estimated from leave one out Cross-validation