rbe
REVISTA BRASILEIRA DE ECONOMIA

DOI 10.5935/0034-7140.20210003
ISSN 1806-9134 (online)

FGV
FUNDAÇÃO GETULIO VARGAS

A note on real estate appraisal in Brazil

Thiago Marzagão[*]
Rodrigo Ferreira[†]
Leonardo Sales[‡]

Contents

1. Introduction 29
2. Data 31
3. Training the model 32
4. Testing the model 33
5. Discussion 34
6. Pictures 34
7. Future directions 35

Keywords
real estate, hedonic pricing, market behavior

JEL Codes
R30, L110, D40

Abstract · Resumo

Brazilian banks commonly use linear regression to appraise real estate: they regress price on features like area, location, etc, and use the resulting model to estimate the market value of the target property. But Brazilian banks do not test the predictive performance of those models, which for all we know are no better than random guesses. That introduces huge inefficiencies in the real estate market. Here we propose a machine learning approach to the problem. We use real estate data scraped from 15 thousand online listings and use it to fit a boosted trees model. The resulting model has a median absolute error of 8.16%. We provide all data and source code.

1. Introduction

How do we know the market value of real estate? The Brazilian Association of Technical Standards (ABNT) advises the use of econometric models for the valuation of urban property (NBR 14653-2 – "Appraisal of urban real estate"). Many appraisers follow that recommendation. They find real estate similar to the target property—say, other residential apartments in the same city—, collect data on those properties, and regress price on features like area, location, number of bedrooms, and the like. The appraiser then uses the estimated model to find the market value of the target property.

[*] Controladoria-Geral da União, Coordenação-Geral de Inteligência de Dados. Brasília, DF, CEP 70050-904, Brasil.
0000-0003-0395-3985

[†] Controladoria-Geral da União, Coordenação-Geral de Inteligência de Dados. Brasília, DF, CEP 70050-904, Brasil.
0000-0002-3143-8385

[‡] Controladoria-Geral da União, Coordenação-Geral de Inteligência de Dados. Brasília, DF, CEP 70050-904, Brasil.
0000-0001-6772-823X

thiago.marzagao@cgu.gov.br rodrigo.p.ferreira@cgu.gov.br leonardo.sales@cgu.gov.br

Received on Jul 15th, 2019
Final Acceptance on Feb 17th, 2020

The ABNT guidelines tell the appraiser to check the estimated model for linearity, heteroskedasticity, autocorrelation, multicollinearity, normality of residuals, presence of outliers, and for the statistical significance of each coefficient and of the model as a whole. The guidelines also say to check the model fit, by observing the R^2. If no serious problems are found, and if the R^2 is not considered too low (the guidelines do not specify a threshold), the work is done.

That approach is flawed. All of the samples are used to fit the regression line. No samples are left out to test the performance of the model. Hence we cannot know how good or bad the model is. The model may have an unacceptably high mean or median error. For all we know, the models created today by Brazilian appraisers are no better than random guesses.

In other words, the current approach is an econometric solution to a machine learning problem. In real estate appraisals we are not interested in the effect of swimming pools on house prices. We are interested in finding the market value of an individual house. We do not care what the coefficient of "has swimming pool" is or whether it is statistically significant.

In fact we do not even care about linear regression at all. This being a machine learning problem, the way to attack it is to try different algorithms—like random forest, support vector machines, and neural networks —, test each algorithm's predictive performance, then use the winning algorithm to estimate the price of the target property.[1]

The current reliance on linear regression is possibly a result of NBR 14653-2 discussing it in detail. It is an unfortunate accident, as linear regression requires us to hypothesize the model's functional form. For instance, maybe the effect of each additional squared meter on the price depends on the property's location. But there are just too many possible interactions for any appraiser to consider. An algorithm like random forest, support vector machines, or neural networks, on the other hand, learns the interactions from the data. The modeler does not need to specify any interactions beforehand.

With an algorithm like random forest or support vector machines there are no coefficients to speak of. They are non-parametric algorithms: we are not estimating anything, so there are no underlying assumptions that could be violated. With neural networks there are weights associated with each neuron and these weights resemble regression coefficients. But these weights are rarely interpretable. And there are usually too many weights for anyone to try to interpret them in any case (as computing power increases, neural networks are becoming larger; it is now common for neural networks to have hundreds of thousands of weights).

An appraiser might respond that a high R^2 is an indication of good performance. But as long as that R^2 is based on the same samples that were used to fit the regression,

[1] For an introduction to machine learning and to random forest, support vector machines, and neural networks, see Trevor Hastie's popular textbook Hastie, Tibshirani, and Friedman (2009).

it tells us nothing about the model's predictive performance. In fact a high R^2 might be the result of overfitting, in which case it comes *at the expense* of the model's performance on unseen samples.

We could not find hard numbers on what proportion of appraisals currently rely on econometric approaches (as opposed to other approaches, like simply computing the average price of similar properties). But we downloaded dozens of appraisal reports from different sources and of the ones that did rely on econometric tools all used linear regression, and all did so in the manner described above—i.e., without any consideration given to the model's predictive performance. We could not find a single appraisal report that separated the samples between training and testing.

It is particularly troubling that even the appraisal reports by *Caixa Econômica Federal*—a state-owned bank that concentrates 70% of the mortgage market in Brazil—incurred the same mistake. *Caixa* usually outsources appraisals to other companies, but it supposedly reviews and approves each appraisal report individually. None of the *Caixa* reports we found tested the performance of the models.

In short, it is possible that billions of *reais* in real estate transactions are based on models whose performance is completely unknown. To help fix this, in the rest of this paper we show a better way to precify real estate.

2. Data

Today's appraisals are based on small samples, sometimes as small as $n = 25$.[2] But nowadays there are thousands of online listings in websites like ZAP[3], wimoveis[4], and Viva Real[5]. Popular programming languages like Python or R have packages that makes it easy to download data from web sources.

Here we chose wimoveis as our data source. We scraped 18,387 wimoveis listings of residential apartments located in the state of Goiás and in the Federal District. The code used for this is split in three scripts: one that scrapes each page of results (each page contains up to 20 listings),[6] one that extracts each listing's URL from each page of results,[7], and one that finally scrapes each individual listing.[8] The scripts are in Python and require the packages *requests*, *BeautifulSoup*, and *pandas*.

[2] See, for instance, https://www.brameleiloes.com.br/principal/pub/Image/20181025040759LAUDO_CAIXA.pdf

[3] https://www.zapimoveis.com.br

[4] https://www.wimoveis.com.br

[5] https://www.vivareal.com.br

[6] https://gist.github.com/thiagomarzagao/2ef1316d7179f33211503cf1ba4c90be

[7] https://gist.github.com/thiagomarzagao/6ad89ac2ba9908e79af65883eee29465

[8] https://gist.github.com/thiagomarzagao/fd21b8e2bca553f90485ae515b6edbb2

We limited ourselves to wimoveis and to residential apartments in Goiás and in the Federal District due to time restrictions (it took approximately one week to scrape all the 18,387 listings). But the approach we propose here should generalize to other states and property types.

We discarded 2,760 of the 18,387 samples. Some of these we discarded because they had obviously incorrect data (say, condo fees of millions of *reais* or asking prices of R$ 0). Others were discarded because they had too much missing information (private area, location, etc). That left us with a dataset of 15,627 samples.

The dataset includes the following variables:

- ID of the listing
- asking price
- private area
- city and state
- number of bathrooms
- number of bedrooms
- number of suites
- number of garage spaces
- age of the building
- number of floors in the building
- condo fee
- property tax
- barbecue area?
- swimming pool?
- playground?
- sauna?
- gym?
- 24-hour doorman?
- security system?
- party room?
- game room?

The dataset is available for download from https://thiagomarzagao.s3.amazonaws.com/wimoveis.csv

3. Training the model

We want to predict a property's asking price from the other features in the dataset: private area, location, number of bathrooms, etc.

We began by discarding some outliers (private areas larger than 50,000 m^2, for instance). That left us with 15,552 samples. We then randomly chose 15% of the 15,552 samples to be used purely for testing the model. That is a total of 2,333 samples, which left 13,219 for model training and validation.

We tried different algorithms: linear regression, random forest, boosted trees, support vector machines (SVM), and several different neural network architectures. For each algorithm we tried several different combinations of parameters. For instance, we tried increasing and decreasing regularization (say, by adjusting the pruning of the random trees and the dropout rate of the neural networks); we tried different learning rates when applicable (neural networks, boosted trees); and so on.

To assess the performance of each algorithm and corresponding parameter set we used 10-fold cross-validation (with the 13,219 samples randomly chosen for training and validation). That is, we randomly split the samples in ten (roughly) equal parts (the folds), used nine of them to train the model, used the trained model to predict the asking prices of the fold that was left out, computed the prediction errors, then repeated the procedure nine more times, each time leaving a different fold out. For each algorithm and corresponding parameter set we computed the median absolute error of the predictions, both in absolute terms (i.e., in R$) and in proportional terms (i.e., in %).

Table 1 reports the lowest median absolute errors obtained with each algorithm.

Table 1. Median absolute error.

	in R$	in %
linear regression	58,733.80	13.49
SVM	50,540.04	11.36
neural nets	46,741.42	11.08
boosted trees	43,140.12	9.49
random forest	36,978.18	8.14

The winning algorithm was random forest (with 1,000 trees and no pruning).

We inspected the relationship between the errors and each of the quantitative features: private area, condo fee, property tax, etc. We found no patterns. In other words, the model's performance does not seem to differ much across different types of properties.

4. Testing the model

Now that we have a winning algorithm (and parameter set) it is time to test the performance of our model. We trained it again, now using all 13,219 training samples, and used the resulting model to predict the prices of the 2,333 samples that were left out.

The result was a median absolute of error of R$ 35,463.24, or 8.16%. That is close to the median absolute of error of 7.9% obtained by Zestimate, the best known property valuation model currently in use (see Poursaeed & Belongie, 2018).

Again we inspected the relationship between the errors and each of the quantitative features (private area, condo fee, property tax, etc) and found no discernible patterns.

We computed variances for the 2,333 estimates using the infinitesimal jackknife method proposed in Wager, Hastie, and Efron (2014). For 94.68% of the test samples the true asking price was within one standard deviation of the predicted asking price.

Figure 1 shows the model fit. The vertical bars represent one standard deviation.

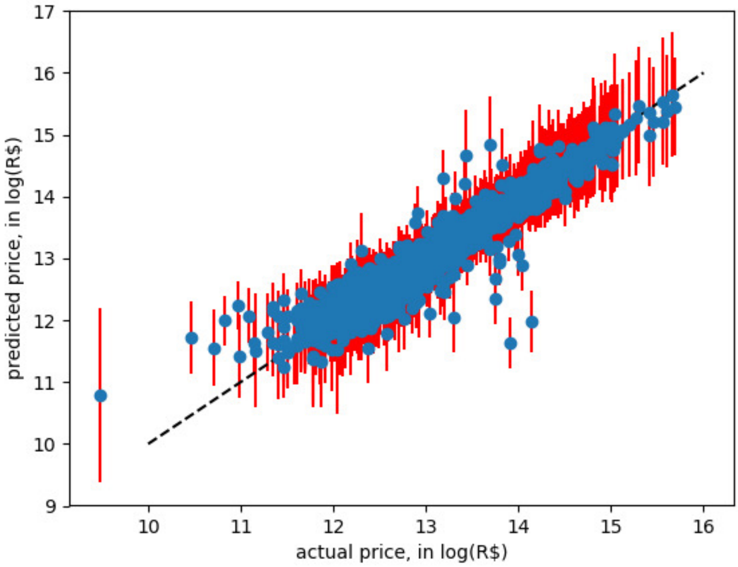

Figure 1. Actual *vs* predicted prices (w/error bars).

5. Discussion

The best known property valuation model currently in use is Zestimate, a proprietary model that has a median absolute error of 7.9%. Zestimate was trained on a much larger number of samples—110 million properties in the United States—and it uses a much richer feature set, including transaction prices (as opposed to asking prices) and detailed tax information (like actual taxes paid). Our model achieves a similar performance—8.16% of median absolute error—with a dataset that's only 0.014% of Zestimate's and using a much poorer feature set. In other words, our model approximates state-of-the-art performance but at a fraction of the data collection cost and of the computational cost. On top of that, our model is not proprietary; both the data and the source code are publicly available on GitHub.

6. Pictures

Online listings usually have pictures. As shown in Poursaeed and Belongie (2018) and Bappy, Barr, Srinivasan, and Roy-Chowdhury (2017), these images can sometimes improve model performance. We tried to incorporate the pictures in our model in three ways. First we extracted the three dominant colors for each listing's set of pictures, then used each color's RGB values as features (which added nine extra features: three colors times the corresponding values of red, green, and bue).[9] The idea was to capture colors associated with newer and/or superior flooring, tiles, etc. That did not improve the model though.

Second, we tried object detection. We used a pretrained neural network (VGG16,[10] pretained on the popular ImageNet and Places365 datasets) to detect objects like A/C units, bathtubes, etc. That did not improve the model either. Finally, we used a manually labeled dataset (Bappy, Barr, Srinivasan, & Roy-Chowdhury, 2017) to train a model capable of classifying a picture as "bedroom", "living room", "kitchen", "bathroom", or "other". We then used that model to label each picture in our dataset and identify, for each wimoveis listing, one picture of each room type: bedroom, living room, kitchen, and bathroom.

Finally, we featurized each picture into a 1 × 50 vector[11] and concatenated the four vectors (one for each room type) to the vector that contained the features used before (private area, location, etc). That too did not improve the model.

We believe that the images did not improve the model because they lack standardization: some listings have several pictures of the living room but none of the kitchen, others have no pictures of the façade, many of the pictures have poor lightning, many of the pictures were shot from a bad angle that does not properly show the room, and so on.

7. Future directions

Here we scraped data from a single source (wimoveis), which only has listings from the state of Goiás and from the Federal District. Also, we only scraped apartament data, which leaves out houses, office spaces, industrial plants, etc. The next step is to add more sources and cover more regions and more property types.

References

Bappy, J. H., Barr, J. R., Srinivasan, N., & Roy-Chowdhury, A. K. (2017). Real estate image classification. In *2017 ieee winter conference on applications of computer vision (wacv)* (pp. 373–381). http://dx.doi.org/10.1109/WACV.2017.48

[9] To do this we clusterized the pixels of each listing's pictures, using k-means (DBSCAN would have been more appropriate, as the clusters may have widely different sizes and shapes, but the computational cost proved prohibitive. We used the elbow technique to find the optimal number of clusters, which turned out to be three for almost all pictures. Each centroid corresponds to a dominant color.

[10] See Simonyan & Zisserman, 2015.

[11] We passed each of the selected pictures through a VGG16 network pre-trained on the ImageNet and Places365 datasets and extracted the features corresponding to the block5pool layer, which is a 1 × 25088 vector. We then used Singular Value Decomposition (SVD) to reduce that to a smaller vector, of 1 × 50.

Bappy, J. H., Barr, J. R., Srinivasan, N., & Roy-Chowdhury, A. K. (2017). Real estate image classification. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 373–381). Santa Rosa, CA, USA. http://dx.doi.org/10.1109/WACV.2017.48

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer. https://link.springer.com/book/10.1007/978-0-387-84858-7

Poursaeed, O., & Belongie, T. M. . S. (2018). Vision-based real estate price estimation. *Machine Vision and Applications*, *29*(4), 667–676. http://dx.doi.org/10.1007/s00138-018-0922-2

Simonyan, K., & Zisserman, A. (2015, Sep). *Very deep convolutional networks for large-scale image recognition.* https://arxiv.org/abs/1409.1556 (arXiv:1409.1556v6)

Wager, S., Hastie, T., & Efron, B. (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, *15*(1), 1625–1651. http://dx.doi.org/10.5555/2627435.2638587