Housing Prices Prediction with a Deep Learning and Random Forest Ensemble

Bruno Klaus De Aquino Afonso
Universidade Federal de São Paulo
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Willian Dihanster
Universidade Federal de São Paulo
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Samuel Bruno da Silva Sousa
Universidade Federal de São Paulo
5 PUBLICATIONS **1** CITATION
SEE PROFILE

L. Berton
Universidade Federal de São Paulo
33 PUBLICATIONS **105** CITATIONS
SEE PROFILE

Housing Prices Prediction with a Deep Learning and Random Forest Ensemble

Bruno Klaus de Aquino Afonso[1], **Luckeciano Carvalho Melo**[2],
Willian Dihanster Gomes de Oliveira[1], **Samuel Bruno da Silva Sousa**[1],
Lilian Berton[1]

[1]Institute of Science and Technology – Federal University of São Paulo (UNIFESP)
São José dos Campos – SP – Brazil

[2]Technological Institute of Aeronautics (ITA)
São José dos Campos – SP – Brazil

isonettv@gmail.com,luckeciano@gmail.com,williandihanster@gmail.com

s-sousa@outlook.com,lberton@unifesp.br

Abstract. *The development of a housing prices prediction model can assist a house seller or a real estate agent to make better-informed decisions based on house price valuation. Only a few works report the use of machine learning (ML) algorithms to predict the values of properties in Brazil. This study analyzes a dataset composed of 12,223,582 housing advertisements, collected from Brazilian websites from 2015 to 2018. Each instance comprises twenty-four features of five different data types: integer, date, string, float, and image. To predict the property prices, we ensemble two different ML architectures, based on Random Forest (RF) and Recurrent Neural Networks (RNN). This study demonstrates that enriching the dataset and combining different ML approaches can be a better alternative for prediction of housing prices in Brazil.*

1. Introduction

A housing market can be understood as any market for properties which are negotiated either directly from their owners to buyers, or through the services of real state brokers. People and companies are drawn to this market, which presents many profit opportunities that come from housing demands worldwide. These demands are influenced by several factors, such as demography, economy, and politics. For this reason, the analysis of such markets has been challenging data scientists and ML engineers around the world, as they must take into account a wide range of scientific fields, each one addressing different kinds of data [Poursaeed et al. 2018], to come up with accurate results to customers and stakeholders [Fan et al. 2018].

The prediction of housing prices as they vary through time and place is a complex task, particularly when the available data is massive and divided into many different data types. As such, it was an appropriate choice for the Data Science Challenge at Engineering Education for the Future (EEF) 2019[1]. This competition was based on predicting property prices based on ML models. The data for training and testing the models was composed of 12,223,582 housing advertisements, collected from Brazilian websites from

[1] https://www.kaggle.com/c/data-science-challenge-at-eef-2019/

2015 to 2018. Every advertised home was for sale, none for rent. The data attributes include descriptions in plain text, dates, geographic coordinates, photos, as well as categorical and numerical features. The advertisements which compose the competition dataset were provided by Properati BR[2]. Properati is a market place for real state in Latin America based in Argentina, Mexico, and Brazil. It assists consumers and provides public databases for ML research. Currently, this company keeps data updated for more the 1.5 million properties, aiming to better understand the market dynamics of prices variation and urban growth.

In spite of being the largest real state market in Latin America, with a demand for about 7.77 million new housing units as of 2017 [Associação Brasileira de Incorporadoras Imobiliárias – ABRAINC 2017], the studies about this market in the country [Moreira de Aguiar et al. 2014, De Souza 1999] do not report the use of ML algorithms to predict housing prices of Brazilian properties for sale and rental. To leverage this promising under-researched topic, this work aims to combine supervised ML methods, Natural Language Processing (NLP) models, and Data Science (DS) techniques to predict housing prices in Brazil. The regression task was initially performed with the Random Forest (RF) algorithm making use of the numeric attributes found within the competition dataset. Google Maps API and the Brazilian Institute of Geography and Statistics metadata were used to enrich the original database. To process textual attributes, we employed Word embeddings, such as Word2Vec, and the deep architecture Bidirectional Long Short-Term Memory (BiLSTM). Furthermore, we used a model that applies transfer learning and reinforcement learning to extract image features. As a result, we provide a solution to the price prediction problem for Brazilian properties, whose performance is a strong baseline for future proposals regarding housing prices prediction.

The remainder of this paper is organized as follows. In Section 2, an overview of the works on housing prices prediction is presented. The Section 3 details the dataset and the algorithms. The experiments and their setup are described in Section 4, whereas the results and drawbacks of our work are discussed in Section 5. Finally, in Section 6 we present the conclusions and future works.

2. Related Work

By the decade of 1990, the increasing popularity of the Internet made it suitable for hosting advertisements which previously were published in newspapers and magazines. As of today, there is a huge number of property advertisements in the Web and exploiting knowledge from them is a topic which is observed in the recent ML literature [Wu and Brynjolfsson 2015, Sirignano et al. 2016]. Due to the large amounts of data from these advertisements, deep learning approaches seem to perform feature extraction for housing prices prediction with good performance [Poursaeed et al. 2018].

Statistical models have been a common approach to analyze and predict property prices for a long time. In [Fik et al. 2003] a study to explain the housing prices variation was conducted by analyzing the influence of location features on the property prices. [Goodman and Thibodeau 2003] employed hierarchical linear models

[2] https://www.properati.com.br

[Raudenbush and Bryk 2002] for estimating housing prices in Dallas County, USA, using data extracted from 28,000 single-family transactions in the period between 1995 and 1997. Among the contributions of this study, [Goodman and Thibodeau 2003] argue that housing market is segmented by the quality of public education, which is not properly a property attribute.

Linear Regression (LR) and Artificial Neural Networks (ANN) algorithms were successfully applied to housing prices prediction by [Wu and Brynjolfsson 2015] and [Selim 2009], respectively. In the former, the authors collected data from Google searches to predict property prices based on advertisement frequency. On the other hand, the latter used an ANN architecture composed of input, hidden, and output layers that yielded better price prediction results than the statistical method of Hedonic Regression. Furthermore, [Selim 2009] compared the influence of many variables on housing prices variation. The most significant variables were found to be: water system, pool, type of house, type of building, number of rooms, house size, characteristic of the location.

Decision Trees, Random Forest (RF), and Support Vector Regression (SVR) have also been successfully employed on this task. In [Park and Bae 2015], the C4.5 decision tree, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Naïve Bayes, and AdaBoost were used to predict housing prices on data from Multiple Listing Service, historical mortgage rates, and public school ratings. As a result, RIPPER outperformed all the other models, due to its strategy of selecting more recurrent values as default outcomes at the same time as learning rules for the lowest values. Another multi-model comparison for housing prices prediction was performed in [Fan et al. 2018], using RF, SVD, Ridge Linear Regression, Lasso Linear Regression, and XGB algorithms.

Deep learning architectures are a rising research topic because of their high performances on massive amounts of data. In [Sirignano et al. 2016], a neural architecture with softmax function was proposed to predict the mortgage risk on a dataset composed of more than 120 million loan records in the United States of America from 1995 to 2014. This kind of ML method is well suited to analyze the variation of house values according to their internal and external appearances, as done by the framework proposed by [Poursaeed et al. 2018]. This framework uses the convolutional neural network DenseNet [Huang et al. 2017] to classify the images to categories related to parts of the house, such as kitchen, living room, or bathroom.

3. Materials and methods

This section presents the details concerning the data and tools used in the study. The choice of the algorithms to perform housing prices prediction has taken into account the nature of the dataset, since real-world data are not always balanced, complete, scaled, or easy to handle [Alpaydin 2009]. RF and BiLSTM were applied for regression and ultimately combined into a final prediction.

3.1. The dataset

The chosen dataset for the competition presents a few key challenges. First of all, there is a significant volume of data. More specifically, the database is composed of 12,223,582 total instances, such that 8,557,058 instances belong to the training set, whereas the remaining 3,666,524 are found in the test set. The advertisements refer to 2,300,079 distinct properties. As such, any approach that makes use of the whole data should be memory-efficient. Moreover, each instance comprises twenty-four features of five different data types: integer, alphanumeric, date, string, float, and image (see Table 1). Therefore, any suitable model must be able to deal with mixed data types. Lastly, the significant amount of missing values for some key attributes made it particularly hard for regression, and the competitors had to come up with strategies for replacing the many missing values.

Table 1. Features of the Housing Advertisements dataset.

Feature	**Data Type**
id, floors, rooms	Integer
created_on, colleted_on	Date
property_id, operation, property_type, place_name, place_with_parent_names, country_name, state_name, geonames_id, currency, description, title	String
lat_lon, lon, lat, surface_covered_in_m2, surface_total_in_m2, expenses, price	Float
image_thumbnail	Image

Dataset features The twenty-four features are listed in Table 1. The '*id*' attribute represents the advertisement identifier. On the other hand, '*property_id*' identifies each unique property, which may be advertised several times through time. '*Created_on*' holds the date the advertisement was initially published, while '*collected_on*' is the date when that particular (and possibly altered) version of the advertisement was collected. For this dataset, '*surface_covered_in_m2*' holds the indoor property area, and the whole property area (including outdoors) can be accessed via '*surface_total_in_m2*'. The '*property_type*' attribute details whether a property is a house, an apartment, a store, or a pharmacy. There were some redundant attributes: '*operation*' and '*geonames_id*' had constant values and were safely ignored. '*floors*' is the number of floors of the property;'*rooms*' refers to the number of bedrooms;'*expenses*' is the amount of condo taxes in the advertisements. The distinction among the advertisements currencies is made by the '*currency*' attribute. Many attributes were related to the location of a property: '*place_with_parent_names*' is the description of the full address.This description is further divided into '*country_name*','*state_name*'and '*place_name*'. All of '*lat*', '*lon*', and '*lat_lon*' provide the GPS coordinates of the property. The textual attributes are '*title*' and '*description*'. The last attribute, '*image_thumbnail*', is a collection of property photos. Finally, the target attribute '*price*' is the desired output.

Missing values and outliers As mentioned previously, many of the values in this dataset were missing, which follows from the seller being free to choose what information to make available. Figure 1 shows each missing value rate. Not included is '*place_with_parent_names*' and neither are its derived attributes, which were present for every instance. This turned out to be useful, as the roughly 50% missing latitude and longitude coordinates can be approximated employing geocoding, i.e. translating addresses to coordinates. The missing rate was extremely high (upwards of 90%) for the number

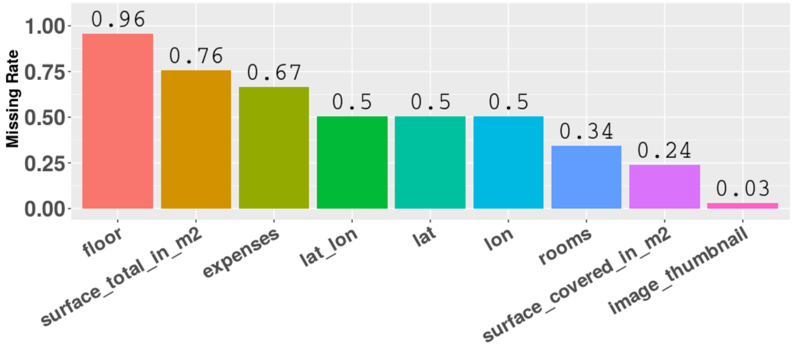


Figure 1. Frequency of missing values (from 0 to 100%) for each attribute.

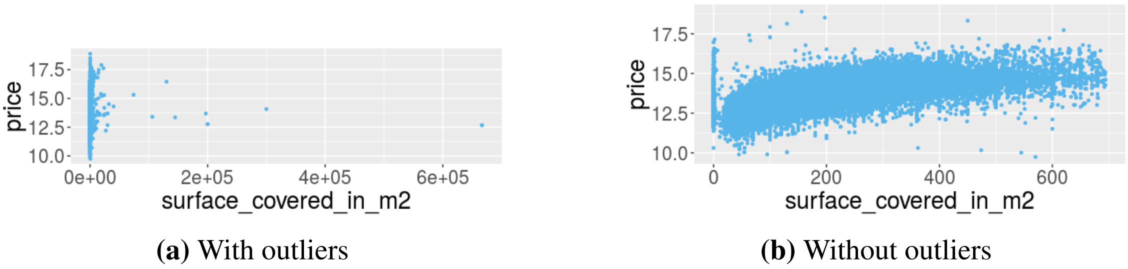
of floors, and moderate (around 34%) for the number of rooms. Most often, only the indoor area was given, as opposed to the total area (24% and 76% respective missing rates). Yet another problem within this dataset is the presence of outlier values for some attributes. Figure 2 illustrates the outliers for the *'surface_covered_in_m2'* attribute. A similar phenomenon occurs with the *'expenses'* atribute. Most importantly, the price does not increase enough to match the extremely high and isolated attribute values. Once these are removed, it is much easier to visualize the correlation between these variables and the target price.

3.2. Finding a suitable metric for housing prices prediction

Before going into the inner workings of the model, a loss function must be chosen. Although the mean squared error is one of the most widely known and used regression metrics [Alpaydin 2009], we used the root mean squared log error:

$$\text{RMSLE} = \sqrt{\frac{1}{N}\sum_{i}(\log y_i - \log \widetilde{y_i})^2} \tag{1}$$

where y_i, $\widetilde{y_i}$, and N are, respectively, the true value, the predicted one and the number of samples. The percent change approximates the difference of logarithms well when the percent changes are small. Also, it must be noted that RMSLE is less punishing when the change in percent is huge. The logarithmic transformation is further justified because it makes it so that relations between the price and some independent variables become closer to linear, as one can observe in Figure 2.



(a) With outliers

(b) Without outliers

Figure 2. Logarithm of price values, given the indoor area.

3.3. Random Forest Regression

Random Forest (RF) [Breiman 2001] is an ensemble ML algorithm based on multiple decision trees whose outcomes are merged, leading to gains in performance and stability while still being faster than other ensemble methods, such as Adaboost since this method is robust to noise in the data, as well as able to tackle high bias and high variance [Breiman 2001]. RF works in two steps: first, for every tree, we obtain a sequence of instances, sampled randomly with replacement from the training set. Each sequence of instances corresponds to a random vector Θ_k characterizing a particular tree. As the sequences will be slightly different from one another, so will the decision trees constructed from them. The prediction of the k-th tree for a given input **x** is given by:

$$h_k(\mathbf{x}) = h(\mathbf{x}, \Theta_k), \forall k \in \{1, 2, \ldots, K\} \tag{2}$$

where K is the number of trees. Each split of a tree uses a random selection of features to further avoid correlation. There are many ways to split a node S into two subsets. Assume a threshold c is chosen for the selected feature, splitting S into S_1,S_2 according to each feature value v_i. For a regression task, one can use a c that minimizes the difference in the sum of squared errors [Alpaydin 2009]:

$$SSE = \left(\sum_{i \in S_1} (v_i - \frac{1}{|S_1|} \sum_{i \in S_1} v_i)^2 + \sum_{i: i \in S_2} (v_i - \frac{1}{|S_2|} \sum_{i \in S_2} v_i)^2 \right) \tag{3}$$

The prediction of any subtree can be obtained as the mean (or median) output of instances that follow the same decision rules. The final prediction is simply an average of each tree's output:

$$h(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} h_k(\mathbf{x}) \tag{4}$$

3.4. Bidirectional LSTM

Recurrent Neural Networks (RNNs) are neural networks that process sequential data such as text. Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber 1997] is a RNN designed to lead with long-term dependencies by using memory elements composed of memory cells and multiplicative units (input, output, and forget gates), whose functions are related to reading, writing, and resetting operations. Bidirectional LSTM [Schuster and Paliwal 1997] is an example of LSTM that accesses long-range context in both sides of the input. For a given time step t, the input is $\mathbf{X}_t \in \mathbf{R}^{n \times d}$, in which n is the number of examples and d is the number of inputs; and the hidden layer activation function is $\emptyset$. The forward and backward hidden states for this time step are, respectively, $\overrightarrow{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$ and $\overleftarrow{\mathbf{H}}_t \in \mathbb{R}^{n \times h}$, and h represents the number of hidden units. These states are computed as follows:

$$\overrightarrow{\mathbf{H}}_t = \emptyset(\mathbf{X}_t \mathbf{W}_{xh}^{(f)} + \overrightarrow{\mathbf{H}}_{t-1} \mathbf{W}_{hh}^{(f)} + \mathbf{b}_h^{(f)}), \tag{5}$$

$$\overleftarrow{\mathbf{H}}_t = \emptyset(\mathbf{X}_t \mathbf{W}_{xh}^{(b)} + \overleftarrow{\mathbf{H}}_{t-1} \mathbf{W}_{hh}^{(b)} + \mathbf{b}_h^{(b)}), \tag{6}$$

where the weight parameters $\mathbf{W}_{xh}^{(f)} \in \mathbb{R}^{d \times h}$, $\mathbf{W}_{hh}^{(f)} \in \mathbb{R}^{h \times h}$, $\mathbf{W}_{xh}^{(b)} \in \mathbb{R}^{d \times h}$, and $\mathbf{W}_{hh}^{(b)} \in \mathbb{R}^{h \times h}$, as well as the bias parameters $\mathbf{b}_h^{(f)} \in \mathbb{R}^{1 \times h}$ and $\mathbf{b}_h^{(b)} \in \mathbb{R}^{1 \times h}$ are all model parameters [Zhang et al. 2019].

The hidden forward and backward states aforementioned are concatenated to obtain a new hidden state $\mathbf{H}_t \in \mathbb{R}^{n\times 2h}$, which is fed to the output layer. In bidirectional deep architectures, it is instead passed on to the next bidirectional layer. Thereafter, the output $\mathbf{O}_t \in \mathbb{R}^{n\times q}$ is computed by the output layer as:

$$\mathbf{O}_t = \mathbf{H}_t\mathbf{W}_{hq} + \mathbf{b}_q, \tag{7}$$

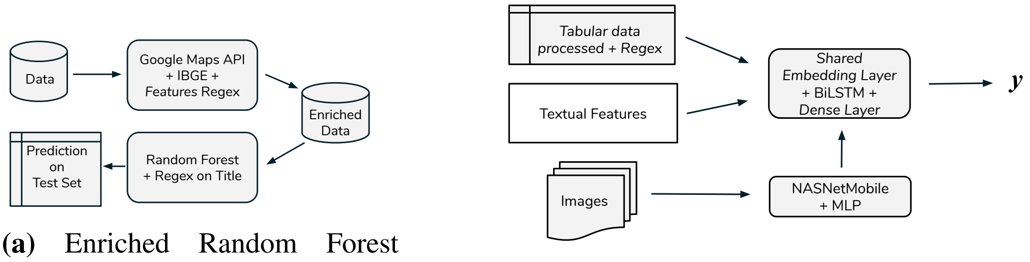
in which q is the number of outputs, and $\mathbf{W}_{hq} \in \mathbb{R}^{2h\times q}$ and $\mathbf{b}_q \in \mathbb{R}^{1\times q}$ are model parameters for the output layer. BiLSTM, as well as other bidirectional RNNs, can have different numbers of hidden units [Zhang et al. 2019].

3.5. Architectures for Housing Prices Prediction

To predict the property prices, we developed two different architectures, which are depicted in Figure 3. The first architecture (Fig. 3a) is based on the enrichment process of the location features combined to Regex on textual attributes. After these steps, the RF algorithm was used to predict property prices. The KISS Model (Fig. 3b) is a deep architecture which uses a shared embedding layer fed with the enriched tabular data, the textual attributes, and processed image features. The '*description*' and '*title*' attributes were extracted by two BiLSTM networks and the image features were obtained from a NASNet-Mobile network [Zoph et al. 2018] combined to a Multilayer Perceptron (MLP). A BiLSTM layer of size 50 and ReLU (Rectified Linear Unit) activation [Nair and Hinton 2010] was employed with the following setup:

- Epochs = 3;
- Batch Size = 2048;
- Learning Rate = 0.01;
- Loss Function = RMSLE;
- (Adam optimizer) β_1 = 0.9;
- (Adam optimizer) β_2 = 0.999;
- (Adam optimizer) ϵ = 1e-08;
- Dropout Rate = 0.1.

KISS is an acronym to 'keep it simple, stupid', whose insight to develop it was to combine all the features into a unique model without losing the sequence of text, since the meaning of words changes according to their position. Therefore BiLSTM [Schuster and Paliwal 1997] was chosen because of its robustness to maintain the order of textual attributes at the same time as it analyses their inverse order.


Data
Google Maps API + IBGE + Features Regex
Enriched Data
Random Forest + Regex on Title
Prediction on Test Set
Tabular data processed + Regex
Textual Features
Images
Shared Embedding Layer + BiLSTM + Dense Layer
NASNetMobile + MLP
y

(a) Enriched Random Forest Model.

(b) KISS Model.

Figure 3. Workflow for the two models implemented for housing prices prediction.

4. Experimental Methodology

This section outlines the experiments performed to predict the value of properties on the housing prices dataset. The pre-processing step was split into four routines respectively related to data enrichment, data processing, text processing, and images processing.

Thereafter, the two architectures developed to perform the regression task were ran over the processed features set[3].

4.1. Data Enrichment

By the analysis of the Spearman's rank correlation on the numeric attributes (Figure 4), it can be noticed that the price is correlated to the number of rooms, the covered area and total area. This kind of correlation is weak between price and latitude or longitude, even though those turned out to be important features. Also, Figure 1 shows us that covered area, total area, number of rooms, latitude and longitude are some of the attributes with a significant amount of missing data. Therefore, since these attributes are important for determining the price of real estate and have many missing values, approximations have been made. In this case, only latitude and longitude data can be approximated. Specifically the Google Maps API[4] was used to obtain a bounding box according to the '*place_with_parent_names*' attribute. We can obtain coordinates as the center of this box, and we also use its area as yet another attribute. The data was further enriched by getting information that was available in the Brazilian Institute of Geography and Statistics website. Namely, we enriched the data with the average monthly salary and GDP per capita of each city.

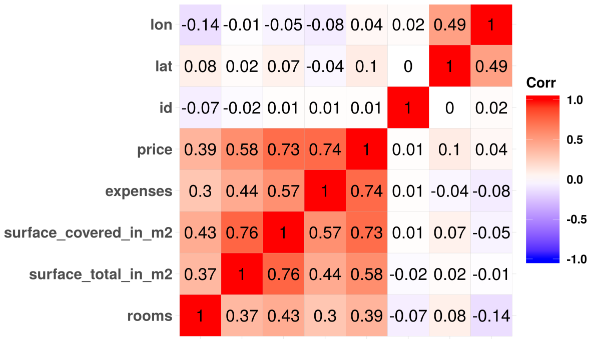


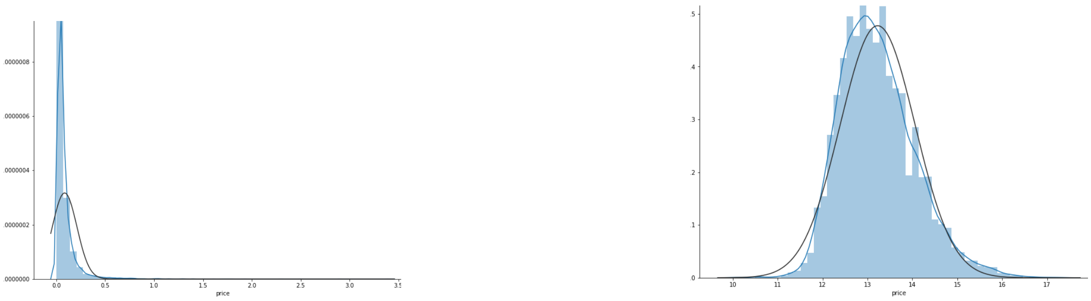
Figure 4. Matrix showing Spearman's rank correlation for property characteristics and financial features.

4.2. Data Processing

To process the data, first a logarithmic scale was applied to the prices, which can be visualized in Figure 5a and 5b. The feature outliers were identified by calculating the 99% percentile. For the KISS model, these outliers were set to the value of the 99% percentile. We also standardized numerical attributes. This is not necessary for the random forest model because the splits are unchanged by the transformation. The KISS model also had the dates categorized into month, year and day. It also had the *currency, property_name, place_name, state_name* attributes categorized as well. Whenever there was missing data for an attribute, it was set to one of the following: a constant value of 0 when the attribute had missing rate greater than 90%; the center of the bounding box given by the GoogleMaps API for the latitude and longitude (only for the random forest model); a value of -1 (random forest) or the mean value of that attribute (KISS) .

[3] Codes available at https://github.com/luckeciano/kaggle-eef-house-prediction/.

[4] https://developers.google.com/maps/documentation/



(a) Before

(b) After

Figure 5. Distribution of prices before and after applying the logarithmic scale.

4.3. Text Processing

To extract information from the title and description, we followed the regular Natural Language Processing pipeline, which encompasses tokenization (splitting sentences into separate words), removal of common unhelpful words such as prepositions (stopwords removal), and lemmatization (reduction of words to their root forms). We tested three different approaches to convert the words into numeric attributes. The first one used a BiLSTM with padding sizes of 150 for the title, and 300 or 500 for the description. The second one consisted in extracting features based on the frequencies of the words, using the TF-IDF model. Thirdly, we performed an insightful feature engineering by exploiting a data leakage: price values encountered in the title or description, which could be extracted via regular expressions.

4.4. Image Processing

We extract image features by using transfer learning from a mobile version of NASNet model [Zoph et al. 2018]. This architecture is obtained via Neural Architecture Search [Zoph and Le 2016], where a RNN controller network optimizes convolutional architectures by using a reinforcement learning algorithm called Proximal Policy Optimization [Schulman et al. 2017]. The final NASNet architecture achieved state-of-the-art results on ImageNet, CIFAR-10 and COCO datasets and learned image features that are generically useful and that can be transferred to other computer vision problems [Zoph et al. 2018].

Firstly, we resized the image to the dimensions of 256×256, and applied the pretrained NASNetMobile architecture available in Keras library[5]. Then, we applied average poolings in the output tensor to map features to a latent space of 264 dimensions. We fed the KISS model with such final features for each image in the housing prices prediction database.

5. Results and Discussion

After processing the different features, we started running the models to predict housing prices on the Brazilian properties dataset. To measure the influence of enriched attributes on the regressor performance, we have run RF with and without them and computed the RMSLE for each running. On the training set, the enriched features from Google and

[5] https://keras.io/applications/nasnet/

IBGE led to a decrease of 34.84 % for the RMSLE. Furthermore, when we ran this same model on the validation set, the difference between the original features and the enriched features had 20.07 % lower RMSLE. In Table 2, the results for the enriched RF model are shown. The experiments were validated by 10-fold cross-validation. Note that the enriched RF model submitted to the competition used slightly more estimators for the random forest and made use of data leakage (i.e. detecting prices in the advertisement title via regular expressions), and so the test error was slightly less than the validation error of Table 2.

Table 2. Results for Enriched Random Forest with 10-Fold Cross-Validation

Set of Features	**RMSLE on Training**	**RMSLE on Validation**
Google + IBGE	0.19725 ± 0.00043	0.30273 ± 0.00148
Original Features	0.29194 ± 0.00050	0.37877 ± 0.00157

In Table 3, the results for the two architectures are presented. The KISS model was the best architecture for housing prices prediction, since it shared the embedding layer among all the textual attributes, without losing information by training models separately for these features. KISS hit a RMSLE score of 6.55% lower than the one hit by the enriched RF model, which was the second-best performing architecture.

Table 3. Results for the Two Architectures and their Ensemble

	Enriched RF	***KISS Model***	***Ensemble Model***
RMSLE	0.27967	0.26135	**0.23847**

Even though each architecture achieved a RMSLE score lower than 0.3, we aimed to reduce the error further. We used a weighted average of the output given by the Enriched RF and KISS models, as shown in Figure 6. The weighted average coefficients for the final prediction were chosen according to a separate, held out set. The result of this ensemble was 8.75 % lower than the ones provided by the KISS model.

Table 4 presents a comparison between our method and the other competitors. We got the 1st place in the competition Data Science Challenge at Engineering Education for the Future (EEF) 2019[6] by having the smallest RMSLE, which indicates a better housing prices prediction.

Table 4. Results for Our Ensemble compared to the competition leaderboard

Model	**RMSLE**
1st Place Model - Our Ensemble	**0.23847**
2nd Place Model	0.25460
3rd Place Model	0.36015
Mean of the 10 best results	0.45141

[6] https://www.kaggle.com/c/data-science-challenge-at-eef-2019/

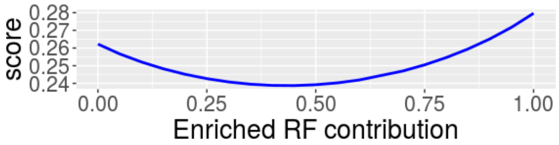


Figure 6. RMSLE score (on a separate, held out set) of weighted average combinations between the Enriched RF and KISS models

6. Conclusion and Future Work

Predicting housing prices from online advertisements in Brazil is a task which requires insight into the data combined with powerful ML algorithms. In this work, we applied two different methods for this task, and combined them into a final prediction. We provided a strong baseline to overcome, as our final ensemble hit a score of 0.23847 RMSLE. The two architectures (enriched RF and KISS) separately also presented good results and can be used as baselines for data enrichment, stacking models, or configuring RNNs.

The Enriched RF works well with numeric features, as it can derive rules not only depending on the value of an attribute but also on its presence or absence. However, it cannot handle raw image or text data. KISS, on the other hand, can represent all kinds of data through the embedding layers. But it did not handle numeric attributes as well as the random forest, and had to rely on more data types to edge it out. Combining the two methods yielded the best result, combining the different strengths of each approach.

As future work, feature selection algorithms can also be employed to leverage the training speed for the models. Another technique which we aim to apply on this dataset is weak supervised learning with pseudo-labeling to increase the number of training data instances for deep learning.

7. Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001; São Paulo Research Foundation (FAPESP) grants #2018/15014-0, #2018/09465-0, and #2018/01722-3; and National Council for Scientific and Technological Development (CNPq) grant #123380/2018-9. We also thank Intel Inc. for the computational resources provided.

References

Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.

Associação Brasileira de Incorporadoras Imobiliárias – ABRAINC (2017). Análise das necessidades habitacionais e suas tendências para os próximos dez anos. `https://www.abrainc.org.br/wp-content/uploads/2018/10/ANEHAB-Estudo-completo.pdf`.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

De Souza, F. A. (1999). Land tenure security and housing improvements in recife, brazil. *Habitat International*, 23(1):19–33.

Fan, C., Cui, Z., and Zhong, X. (2018). House prices prediction with machine learning algorithms. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 6–10. ACM.

Fik, T. J., Ling, D. C., and Mulligan, G. F. (2003). Modeling spatial variation in housing prices: a variable interaction approach. *Real Estate Economics*, 31(4):623–646.

Goodman, A. C. and Thibodeau, T. G. (2003). Housing market segmentation and hedonic prediction accuracy. *Journal of Housing Economics*, 12(3):181–201.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang, G., Liu, Z., v. d. Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

Moreira de Aguiar, M., Simões, R., and Braz Golgher, A. (2014). Housing market analysis using a hierarchical–spatial approach: the case of belo horizonte, minas gerais, brazil. *Regional Studies, Regional Science*, 1(1):116–137.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

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):2928–2934.

Poursaeed, O., Matera, T., and Belongie, S. (2018). Vision-based real estate price estimation. *Machine Vision and Applications*, 29(4):667–676.

Raudenbush, S. W. and Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods*, volume 1. SAGE Publications, Inc., 2 edition.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Selim, H. (2009). Determinants of house prices in turkey: Hedonic regression versus artificial neural network. *Expert systems with Applications*, 36(2):2843–2852.

Sirignano, J., Sadhwani, A., and Giesecke, K. (2016). Deep learning for mortgage risk. *SSRN Electronic Journal*, pages 1–75.

Wu, L. and Brynjolfsson, E. (2015). The future of prediction: How google searches foreshadow housing prices and sales. In *Economic analysis of the digital economy*, pages 89–118. University of Chicago Press.

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2019). *Dive into Deep Learning*. `http://www.d2l.ai`.

Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.