Alessandro Morosini
3149076
Bocconi University

House prices prediction in Italy
Machine Learning 30412

9/05/2023

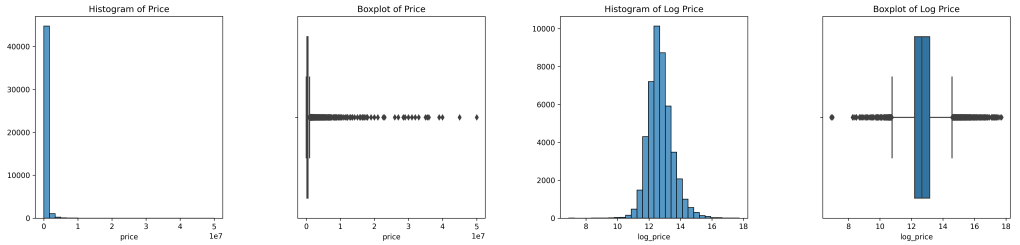
Rationale

The real estate market is a very complex system that is influenced by a huge variety of factors. Being able to predict house prices can be a challenging problem, but it can also be extremely useful (and fun), especially for people like us who live in Milan.
In this this project, we aim to use machine learning techniques in order to predict house prices in Italy.
We will first conduct some exploratory data analysis to gain insights and solve any potential issues with the given data, and then proceed to add some new features to the model that can be exploited at prediction time. Next, we will train and evaluate different machine learning algorithms and use cross-validation to select the best-performing one. We will finally conclude with a mention to some possible limitations for the analysis, as well as ideas for future improvement.

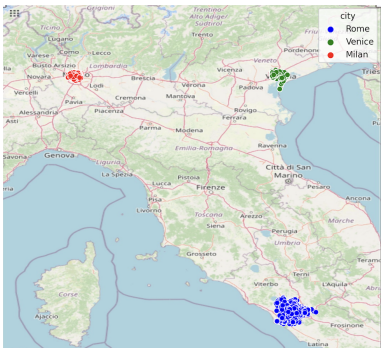
Exploratory Data Analysis

The dataset contains more than fourty-five thousand different house observations, each of which is associated a unique id and a sale price. The dataset also contains 15 different features; some of these are worth to be mentioned: the coordinates of the house - given as latitude and longitude, the proximity of the house to the center of the city, the the year of construction, the number of rooms and bathrooms, as well as the surface of the house, and many others.
The complete data can be seen on the notebook, while here are some of the key facts emerged from EDA:

- Price, our label, is very skewed to the left. A log transformation was taken both for better visualization and processing of the data.



- Overall, the dataset contains 17.69% of missing values. Most rows miss 1 to 4 values. Features like garden have more than 50% null values, while some such as coordinates miss a very few amount.
- Price has a good correlation with the surface of the house and the number of rooms or bathrooms, while features such as expenses or total floors seem to have an almost negligible correlation.
- The houses are clustered around three main cities: Rome (54%), Milan (30%) and Venice (16%).



Data Processing

This section aims to provide insights into how the main data challenges were addressed, namely missing values and outliers. The notebook provides specific justifications for each feature's handling.

Missing Values

The binary variables balcony, garden, and elevator, are among the ones with highest percentage of missing values. It is worth noting that the present values for these features are exclusively True values. This indicates that null values in the dataset represent the absence of these features in the house. After all, the data was scraped from the internet, and it is unlikely that sellers would explicitly state the absence of these features in their announcements. Therefore, imputing these missing values with False seemed appropriate.
Energy efficiency and expenses are other features with a significant number of missing values. Given their low correlation with price and the high number of outliers, median simple imputing was chosen as approach.
For the features with higher correlation, more in depth reasoning was carried out for the imputation strategies. For instance, missing number of rooms were imputed through a regression on surface area, given the strong linear correlation; other features such as construction year were imputed with 5-NN on the train dataset, as similar houses might have been built in similar periods. Some other features were instead imputed with a flag, as it is the case of the house conditions, and others simply with the median or the mode.

Outliers

Relatively to the presence of outliers, we opted for a very conservative approach. The data was processed mainly following common sense and looking at boxplots, but some online researches were also needed.
Houses with a log price lower than 8, i.e. a price of lower than 3000 euros, were dropped from the dataset. Similarly, houses with a price exceeding 50 million euros were also removed, as the most expensive house in Italy is valued at around 45 million euros. These thresholds were chosen to err on the side of caution.
As for the features, only values that seemed physically unfeasible were removed. Examples could be a floor higher than 50, which is the highest number of floor in Italy (Isozaki Tower), or a house whose construction year is in future (2500). Some other less radical values, such as houses with construction year in 2024, were kept in the dataset as they were found to be consistent with the data.
An important mention goes to the discovery of some 'suspicious' values. Examples are houses with a surface area of zero or floors higher than the total number of floors in the building. One could have considered these observations outliers and drop them, or could have also considered them as missing values and impute them. However, a good correlation was found between 'suspicious' values and price. Thus, the final decision was to keep them in the data with a flag, as there may be some hidden factors at play.

Feature Engineering

The data is now processed and as clean as possible. However, to enhance the accuracy of the price predictions we need to introduce some more pieces of information into our dataset. In this section, the focus is on adding new features with the hope of increasing the explanatory power of the dataset. Additionally, some time will be devoted to the selection of the most informative features among the collected ones.

Feature Augmentation

The feature space of the model will be improved by exploring data related to points of interest and creating interaction variables from existing features. The coordinates of the house will also receive important attention. A brief overview of the steps taken to process this information is provided below.

Geographic Information

As noted in the EDA section, houses can be clustered into three main cities: Milan, Rome and Venice. This information was leveraged to introduce a categorical variable for the city in which the house is located. Another feature was also introduced to represent the distance of the house to the center of the city.
Furthermore, the coordinates will be used in a second moment to establish a clustering of houses within each city, in order to capture information at a neighborhood-level.

Point of Interest (POI)

The point of interest dataset includes valuable public information about cities such as amenities and transports, all of them located through their set of coordinates. This dataset was slightly pre-processed in another notebook,

and then exploited thanks to a function that retrieves the distance of the house from the closest 'poi'.
The extracted data includes information on different amenities, such as universities and restaurants. It also includes information on public transports, such as subways, trains, airports and some others. Moreover, information about tourism was also extracted, such as museums and hotel locations: the rationale is that a house located near to a tourism destination could have have a higher price.

Interaction Variables

In this part, by taking some combinations of existing features we give birth to some new features that might have a significant correlation with price. Examples of interaction variables that were introduced include the "qualityöf the house, derived from a combination of the construction year and the condition of the property, the "floor ratio", which combines the house floor and total floors in the building, a "noisy"flag, which is true if the house is in proximity to an airport or train station. Many other features were introduced, and the rationale behind the combinations is described more in detail within the notebook.

Extra Information

As already mentioned, some flags were introduced to represent the particular cases of outliers which have an unexplained correlation with the price. In particular, the case in which a a house whose floor is higher than the total number of the building floors, or the case in which the house has a zero surface area.
Some other information that was introduced at this stage is the average GDP per capita in the specific city, sourced from Wikipedia (link in the notebook).

Encoding

The augmented dataset contains two categorical features, namely the conditions of the house and city. The conditions of a house have a natural ordering, so ordinal encoding has been applied to this feature. For the city feature, one-hot encoding was preferred, creating for each city a binary column which indicates whether the house is located in that city or not.

Feature Selection

The feature space was extensively extended, and the dataset now contains more than 40 features, some of which have a strong correlation with the price while others do not and only risk to introduce noise. Moreover, some features are also correlated with each other, which could cause problems when using certain models for prediction. To address this issues, this section aims to reduce the dimensionality of the feature space. The main strategy employed was to drop all features with a correlation with the price below 0.05 and manually removing features that are likely to convey the same information. By doing this, insignificant features that would also add noise were removed, and collinearity was reduced.
However, this approach did not completely solve the multicollinearity problem. Indeed, PCA was then performed to to create a reduced copy of the data. This 'algorithmically' reduced dataset was only used for models that are particularly affected by multicollinearity, such as Linear Regression.

Models

Model Selection

At this stage, the dataset was scaled and split intro train and validation. Cross-validation was first used to find the optimal hyper-parameters for each algorithm, and then used to analyze the different model performances. The algorithms that were tried include linear ridge regression, K-nearest neighbors, support vector regression and random forest. Additionally, a shallow neural network was also implemented in PyTorch. Linear regression did not perform very well, which may be attributed to the lack of linearity of the relationship of the features with price. Unsurprisingly, the addition of regularization did not help, as the model was not complex enough to capture the underlying relations. SVR performed even worse, even when non-linear kernels such as polynomial or radial basis function were used. K-NN definitely performed better, and a number of neighbors greater than 20 seemed enough to allow the model generalize well. However, random forest was by far the one with the best performance. More information about the model performances can be read on the notebook.
Overall, random forest seems the most suitable choice. Indeed, not only it has the lowest training error, meaning that the model captures well the structure in the train data, but it also has the lowest validation loss, meaning that it has good ability to generalize to unseen data. The optimal parameters were found to be: 2 minimum samples required for the split of an internal node, 5 minimum samples required the split of a leaf node, and no fixed maximum depth. Moreover, a choice of 250 ensamblers seemed a good balance between

efficiency costs and performance. The following table[1] reports train and validation losses from the different models.

Model	Train Loss (billions)	Validation Loss (billions)
Linear Regression	686.10	682.27
Ridge Regression	690.26	685.51
K-NN Regression	566.46	615.53
SV Regression	789.32	787.56
RF Regression	**286.69**	**584.44**

Model Extension

The final model has been chosen. However, during the course of the project, the geographic information about the city has only been included as a one hot encoded feature, and one might have the suspect that this approach deos not fully exploit the predictive power coming from the geographic information.
To address this issue, the dataset was split by city, resulting in three sub-datasets for Milan, Rome, and Venice. Moreover, given that the area of a city could also be another key factor, K-means clustering was then performed for each sub-dataset to obtain houses clusters within each city, which could resemble of neighborhoods. The centroids for each cluster were chosen based on the center of each city, resulting in 12 neighborhoods for Milan, 14 for Rome, and 6 for Venice. These numbers were chosen based on the performance of the K-means algorithm and the distribution of clusters on the map. The city datasets were eventually split in neighboorhood sub-datasets accordingly to the clusters.
After obtaining the city-specific and neighborhoods-specific datasets, random forest regression was performed for each sub-dataset to obtain predictions at differrent geographic granularity, and the final results were then joined. The following table reports train and validation losses of the three model extensions at different geographic granularities.

Model Extensions	Train Loss (billions)	Validation Loss (billions)
Nation-wide	286.69	584.44
City-wide	**330.52**	**473.40**
Neighborhood-wide	419.20	623.89

It is evident that, as the dataset granularity increases, the random forest model struggles to fit the training data. This can be attributed to the algorithm working with smaller sets of observations, resulting in less data to discern the underlying relationships. Despite a high validation loss at the neighborhood level, the city-level validation loss actually decreases compared to the baseline nation-wide model. This suggests that the previous model was overfitting the training data and not generalizing well.
In conclusion, the optimal approach is to run the random forest model separately on the three datasets, each representing a main city, and then join the predictions.

Limitations & Future Improvements

In conclusion, our random forest model has achieved good performances in predicting house prices, but there are still several aspects under which it could be improved.
One limitation of our approach is that we assumed that random forest is still the best model even after splitting the dataframe into sub-datasets. It would have been worth to explore different algorithms.
Also, our neural network only had two layers with 256 neurons each; however, we have a good amount of data and could potentially benefit from a more complex architecture.
Another area for improvement is the incorporation of additional features. For example, we did not include information such as crime rates or quality of life. Moreover, the neighborhoods were just obtained through k-means clustering, but using some real coordinates would probably lead to better divisions.
Finally, we did not fully exploit the information contained in the points of interest data. While we used the distance from the closest point of interest as a feature, this is often highly correlated with distance from the city center. Future work could explore other features such as the quantity of points of interest in the surroundings of a given location.

[1]The losses reported are calculated on the best choice of hyper-parameters for each model. The otpimal parameters are reported in the notebook, with no need of running GridSearchCV again.