ML Challenge
House Prices Prediction

Giacomo Cirò

May 9, 2023

Introduction

This project aims at predicting the prices of real houses using machine learning techniques, using a provided dataset which consists of information on various features of the houses, such as the number of rooms, bathrooms, square footage, and location.
There is a second dataset provided, *poi.csv*, which is a collection of coordinates of points of interest.
First of all, I will carry out an analysis of the data provided to well understand the distribution of the different features as well as their relative correlations.
Afterwards, I will exploit *poi.csv* to enrich the houses with new features that will make the predictions more precise and robust.
Eventually, I will train three machine learning models and compare them in order to determine the best one for predicting house prices.

Disclaimer: for convenience, the manipulation of the data frame according to the analysis described in the following section, has been performed entirely in the section "Process the Dataset" of the attached notebook

1 Data Analysis

The *train.csv* dataset contains information relative to 46,312 houses. In particular, there are fifteen columns, three non numerical and the rest numerical, and one column with the price.

1.1 Encoding

I started by encoding the non-numerical columns, so that I could carry out quantitative analysis:

- GARDEN, boolean column with only True and Null values, I assumed Null values to be representing False and one-hot-encoded the column;
- BALCONY, same as above;
- CONDITIONS, categorical column with labels referring to the house condition. I proceeded to encode it using an ordinal encoder.

Next, I looked at the percentage of null values and decided not to drop any feature, the highest percentage of null values was 42.9% in the conditions column, which is below the 50% threshold I decided to set.

1.2 Overview

After that, I plotted histograms of each column, and immediately noticed how skewed some features were. This is due to the large sample size, which causes outliers and extreme observations to be present, even though in minuscule percentage compared to the rest, and therefore skew the histogram.
Before inspecting each feature in detail, I plotted a heatmap of their respective correlations, as well as pairwise scatterplots. As expected, characteristics such as SURFACE, N_ROOMS and N_BATHROOMS are extremely correlated to the price and among them.
I noticed also a negative correlation between the year which the house was constructed in and its price.

1.3 Features

Finally, I started to carefully inspect each of the sixteen columns, starting from the target variable, to understand the distributions and deal with outliers and incorrect values:

1. PRICE, to better visualize, I plotted scatter plots and histograms of both the price and the log of the price. More than 99% of the observations are below 7,500,000; there are observations above 50 millions and below 1,000, which I decided to drop.
2. ENERGY_EFFICIENCY this attribute ranges from 0 to 3 millions, with only one observation at 12 billions, which I decided to set to null.

3. TOTAL_FLOORS, FLOOR the tallest residential building in Italy is Torre Solaria (37 floors), which I took as an upper bound for the total number of floors a building in this dataset could have.

4. EXPENSES according to money.it, the average "spesa condominiale" in Italy is 100 euros, whereas the mean of this database is 195. I looked at the highest expenses and found out there was something strange. The top two expenses were surprisingly high and exactly equal to the price of the corresponding house. I decided to set to null all those expenses that were greater or equal than the corresponding price, which were more than just these two. I also noticed that 8.8% of the observations report 0.0 euros of expenses, these must be houses and not apartments. For curiosity, I checked some of their coordinates on Google Maps and what showed up were streets with houses far from the city center, rather than apartments.

5. SURFACE I checked the distribution of this feature and noticed that a few houses (97 - 0.21%) reported 0.0 surface, which is not possible. I recalled the high correlation surface had with bathrooms and rooms, and I decided to set the 0.0 surfaces to null and then impute a value using a linear regression on the number of bathrooms and rooms, after I had made sure there were not too many houses with all the three attributes null. I tried also KNN to impute the missing surface value, but given the almost same loss and R squared I decided to stick with the simpler linear regression.

6. GARDEN two thirds of the houses appeared to have a garden, which I concluded did not refer to a private garden given that even houses above or below the ground had it. Initially surprised, I observed that the average energy efficiency of the houses with the garden was substantially higher than that of those without it. However, I realized that this was only due to the fact that the two largest and extreme observations of energy efficiency were labeled as having a garden. I tried dropping them and the mean dropped as well.

7. BALCONY similar to the previous case, the mean of balcony is biased by the energy efficiency outliers.

8. CONDITIONS to analyze this attribute and understand the semantic meaning of the labels, I encoded it back to the original categories. From the graph I observed that most of the houses were "in buono stato / abitabili" and that the houses labeled as "nuove / ristrutturate" had the highest mean, which is in line with what I was expecting.

9. CONSTRUCTION_YEAR the database collects houses built from 1000 to 2500, which is clearly a wrong entry. However, many houses are yet to be completed, some of them in 2025, and have been already sold or are on sale. From the histogram I understood that most of the houses in the dataset have been built in the 20th century, with most of them around the '60.

10. N_BATHROOMS, N_ROOMS given the similarity of these features, I decided to analyze them jointly. They both appear to be positively linearly correlated with the price, as well as with the surface as explained above.

11. LATITUDE, LONGITUDE given the importance of the location of the house for determining the price, I decided to drop any observation without these entries, fortunately only a few turned out to be missing. Thanks to a simple scatter plot, I was able to identify three main clusters where the houses were located, and identified these as the three main cities of Milan, Venice and Rome

12. PROXIMITY_TO_CENTER by plotting the houses in Milan with a colormap representing this attribute, I understood that it referred to center as meaning the entire city, and the distinction was between the houses in the city and those in the metropolitan area outside.

13. ELEVATOR the mean of the houses without the elevator is higher than that of those without it, this could be hinting at the fact that mostly of the cheapest houses are apartments, which in most of the cases have an elevator.

1.4 Null and missing values

I decided to drop all the rows with more than 45% (7 or more columns out of 15, which is half of the columns without the price) of null values, which turned out to be around 200 observations.
Then, I trained a linear regression model to fill the surface missing values using the number of bathrooms and number of rooms. Ultimately, I filled the remaining null values using a simple imputer with median strategy, which is more robust to outliers than using mean strategy.

2 Feature Engineering

After a detailed analysis of the available features, I decided to test possible ways of augmenting the information available.

2.1 City center

First of all, I wanted to cluster the different houses into three groups corresponding to the three main cities and also quantify how close to the city center each house was, which I thought to be one of the key factors driving the price. To do so, I looked up on Google Maps the coordinates of what I considered to be the city center of each city (Duomo, Colosseo and Piazza San Marco), then I defined a function geo_distance to compute the Haversine distance in km from two points given their coordinates. I preferred to define a function myself because the geopy.distance.distance does not support vectorized operations, and I wanted to avoid too many loops. At this point I computed the distance of each house from each of the two city center and created two new columns in the original dataframe:

- CITY argmin of the distances (0,1,2 depending on the closest city center);
- FROM_CITY_CENTER the distance in km from the closest city center.

2.2 School

While looking through the columns of *poi.csv* I was thinking about what might impact the price of a house and immediately I came up with the distance from public transport as well as from schools.
I defined an auxiliary function add_closest, which takes as input the data frame of the houses and a data frame of coordinates of points of interest and adds to the former the distance of each from the closest point of interest.
Surprisingly, I found that none of the schools in the dataset were located near Venice, and so the distances of the houses in Venice from the closest school were all hundreds of km. However, I didn't want to drop this feature, and I decided to use replace th actual distance with a flag which tells whether or not the closest school is within a radius of 1km from the house (this is automatically done by the function if given the attribute radius).

2.3 Public transport

Furthermore, I had some troubles when adding the closest public transport. Initially I wanted to add one column for the subways, one for the buses and one for the tramways. However, Venice didn't have subways, and I've already lost the distances in the schools column, so I decided to make a unique column with the distance from the closest public transport, either a bus, subway or tramway.

2.4 University / Kindergarten

For universities and kindergarten the process was straightforward and I didn't have to do anything but retrieve the dataset from poi.csv and pass it to the function.

2.5 Street / Zip Code

I tried to exploit the poi database to add also the street which the house was located in as well as its zip code. In order to achieve this, I found each house's nearest neighbor from the database, and assigned it the same street and postal code. However, this process was extremely costly and absolutely not worth it, so I dropped these two features from the final training data frame.

3 Model Selection

I decided to develop, train and test three different models.

- Linear Regression, because I thought I could exploit the strong correlation of price with surface and other features;
- Neural Network, because I am fascinated by this model and I wanted to try and implement one myself, even though it might not be the optimal choice for predicting house prices.

- Random Forest Regressor, because I thought it could be efficient due to its structure. I thought that similar houses are likely to have similar features, and random forest regressor could be interpreted as creating many different clusters of houses which share common features and averaging their price to make a prediction. Furthermore, according to Isaac Ake's master thesis this is the best model to predict house prices.

3.1 Linear Regression (8e11)

I tried to implement a linear model with the three features that were mostly correlated with the target variable, hence SURFACE, N_BATHROOMS,N_ROOMS.
The resulting mse loss (8.43e11) is quite high compared to what I managed to achieve with the other models.
Additionally, some of the values of the surface have been determined as a linear function of the other two regressors. Similarly, some of the values in each of the three columns are the median of the respective column, as a result from the null values imputing performed.These facts combined cause dependency between the regressors and break the OLS assumptions.

3.2 Neural Network (6e11)

I trained a two-layers neural network with ReLU activation functions, optimized with the Adam version of stochastic gradient descent.
I tried playing around and tweaking the parameters, but I concluded that the trade-off between training cost and efficiency was not worth it compared to Random Forest, and therefore I focused on optimizing the latter.

3.3 Random Forest Regressor (4e11)

To train this model I used all the features available. Initially I trained the model with the default settings, in particular 1 as the minimum sample in the leaf and 30 as the number of regressors in the ensemble. However, I noticed high variance in both the train and test loss across different fits of the model, with the train loss being significantly smaller than the test loss, which led me to try and tune the parameters better. To reduce overfitting, I set 5 as the minimum number of samples in the leaf, and to reduce variance I raised the number of regressors to 250. Eventually, I managed to stabilize both the train and test loss at about 4.0e11, with the test loss oscillating more.

4 Conclusion

The goal of this challenge was to create a model which best predicts the house price given different features.
In order to address this, I started by looking at the data I was provided to get an overview and understand general patterns.
I then focused on cleaning the dataset, and I found many null observations, as well as extreme values and errors, which I fixed with what I deemed to be the best strategy.
Once I was done with the data cleaning, I turned to augmenting the features available. I exploited the information provided by the coordinates of each house to quantify the distance from the closest city center as well as the city it belongs to.
Next, I used the data set of the points of interest to add the distances from relevant locations, such as universities and public transports, as well as a flag to mark the presence of a school within a radius of 1 km from the house.
I also tried adding other features, which however revealed to not be worth the cost of inferring them.
Finally, I created and trained three different models, the first very easy which in fact turned out to be not that good.
The second one, on the other hand, turned out to be the best, even compared to the more complex neural network.
In conclusion, predicting house prices could be seen as the "hello world" of machine learning, but nonetheless it is a very complex task with many aspects to take into account and many obstacles to overcome and it requires a deep understanding of the data, the context and the problem you are dealing with.
I found this assignment to be a great opportunity to challenge myself and exploit the things I studied from the books to try and solve a real-world problem.
I'm sure this is just the beginning of what I want to be a long and stimulating journey in the world of machine learning and artificial intelligence.