Milan House Prices

Federico Rausa

June 19, 2025

1 Introduction

The aim of this work is to estimate house prices in the city of Milan. The error metric that we need to minimize is the MAE. The other variables are common information about the house.

2 preprocessing

In the following lines, will use these new names for the variables of the dataset:

- selling price (price) : price of the house
- square meters (sqmtrs) : dimension of the house
- other feautures (of) : comprends 20 dummies, if the house include each of the 20 commodities (electric gate, garden, terrace, attic, closet, cellar, satellitetv, exposure, concierge, balcony, fireplace, kitchen, hydromassage, tavern, optic fiber and 5 types of window frames)
- condominium fees (cf) : additional costs of some houses. If is 0 then maybe the house is not inside a condominium.
- energy efficiency class (energy) : can be encoded as a group of 6 dummies or can codified as a numeric variable, with the classes a-g mapped to 1-7. Is higly correlated with the age of the house.
- heating centralized (hc) : dummy that 1 if is central
- year of construction (year) : this can be recoded in age = 2030 - year to get a gamma distributed variable.
- car parking (cp) : From this string variable are extracted 2 new variables, cp box and cp shared, that are, respectively, the number of car parks private and shared (each between 0 and 3)
- rooms number (rn) : Is considered both a categorical variable and a numeric variable, where 5+ is converted to 6.
- bathrooms number (bn) : Is considered both a categorical variable and a numeric variable, where 3+ is converted to 4.
- condition (cond) : Is considered a categorical variable, with four classes.
- total floors in building (tf)
- zone (zone): this variable can be encoded in 2 ways. The first is a group of dummies, with some categories that can be discarded if have very low frequency. The second is to encode it as a couple of coordinates, latitude and longitude, using the arcgis database. For linear models is better the first option, while using tree models is preferable the second encoding.

The other variables will keep the same names. Availability will not be used.

2.1 target variable

Here is applied a reparameterization of the target variable as the price of the house per square meter under natural logarithm. The reason of the division with the square meters is the fact that maybe square meters obscures the importance of the other variables. It has indeed a very strong correlation with the response, and, from domain knowledge, can be said that this correlation is mostly caused by the fact that in house market the total price is expressed as the sum of the square meter price. Since the ratio is gamma distributed, and is desirable to predict a variable with a symmetric distribution, is applied a log transform to it. This transform helps also to note that there are 7 outliers, which corresponds to 7 houses with sqmtrs = 1 (which is obviously false, if these houses have at least one bath and one room), so that 1 has to be interpreted as a NA.

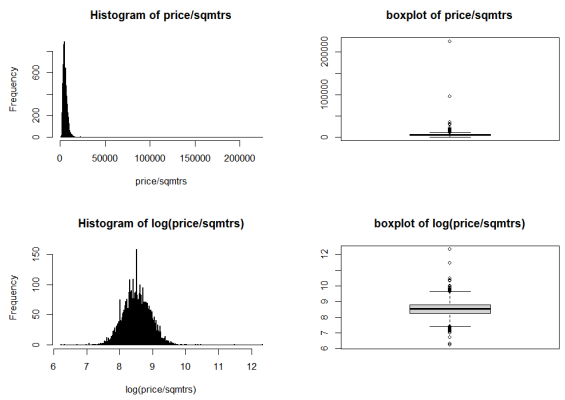

Figure 1: Distribution of target variable. 7 outlier, were the sqmtrs were set equal to 1, are removed. The log gives symmetry to the response variable distribution.

Response	R2 adj.	MAE test	MAPE test
log(sqmtrs)	0.78	16 (on sqmtrs)	0.17 (on sqmtrs)
log(2030 - year)	0.39	27 (on year)	0.76 (on year)
tf	0.15	2.9	0.76
log(1+cf)	0.23	109	0.61

Table 2: Regression results with continuous and categorical predictors. The choice of the predictors is omitted for simplicity

2.2 missing data imputation

To impute the missing values, two simple models are used: OLS regressions for continuous variables and contingency tables for categorical variables. The missing imputation follows a sequential path that helps to use the maximum number of optimal covariates. Each model is validated on a single test set, with MAE for the regressions and accuracy for the contingency tables. Some models don't perform well (e.g. the contingency table for heating centralized has an accuracy on test data that is 62% near to the random guessing), but anyway are accepted as no better way is found to impute that missing values in a reasonable amount of time. The predictor used in each contingency table is the one that presents the higher contingency coefficient with the response (and so the higher chi squared test statistic on the correlation between categorical variables) between the categorical variables present in the dataset. For the dummies belonging to the group of variables in other features (of) simply the mode is taken to replace the NA. Contingency tables report:

Response	Predictor	Contcoef	Acc test
bn (4 classes)	rn (6 classes)	0.65	0.73
cond (4 classes)	year_class (16 classes)	0.58	0.50
energy (7 classes)	year_class (16 classes)	0.67	0.45
lift (2 classes)	tf (24 classes)	0.48	0.75
hc (2 classes)	tf (24 classes)	0.39	0.62

Table 1: Classification results with categorical predictors. Contcoef is the contingency coefficient, Acc is the accuracy.

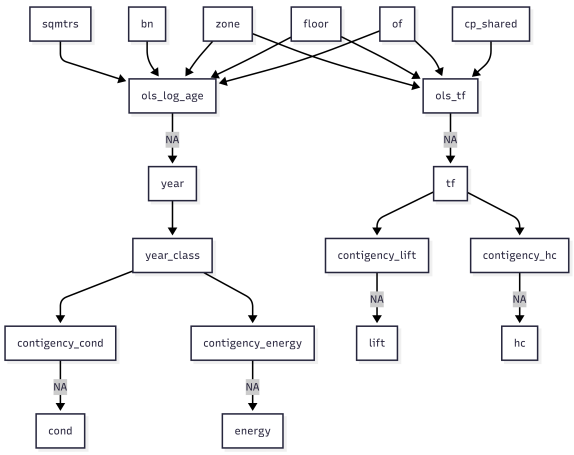

Figure 2: Missing imputation chain representation, after the imputation of square meters. For continuous variables (i.e. log of (year-2030), tf) OLS regressions are used. of (other features) represents a group of dummies, and in some cases just few of them are used. To impute the year missing, is estimated a new variable $log(age) = log(2030 - year)$. The log helps to model a linear relationship with the covariates. Year class is a new categorical variable, that is the segmentation of year in the intervals with breaks c((1000, 1800, 1850, seq(1900, 2030, by=10)). Its contingency table with cond (conditions of the house, with 4 classes) and energy (with 7 classes) gives the respectively

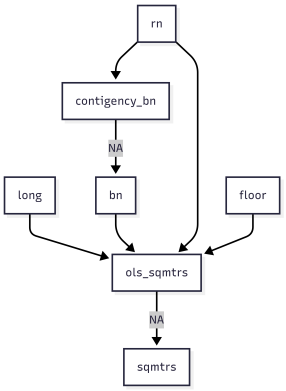

Figure 3: Missing imputation of bathrooms number and square meters, using a contingency table for the first and an OLS regression model for the second. bn and rn are included in the model as groups of dummy variables.

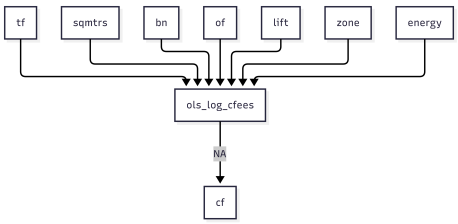

Figure 4: Missing imputation of condominium fees, transformed as y = log(1+cf).This imputation was made at the end, after the imputation of the missing its predictors. The 1 is added to avoid infinite when cf=0.

2.3 final dataset

The final dataset has 40 columns. Many of them are categorical or ordinal, and could be used to build new groups of dummy variables. Since the main modeling choice is to use regression trees, and not linear models, to predict the response, this conversion would not be applied, withouth lack of information, thanks to the property of the regression tree.

3 modeling

3.1 comparing regression trees and linear models

The main model used is an ensemble learning method over regression trees. Regression trees, i.e. CART, use the recursive binary splitting algorithm [Ize08] to find at each iteration the best variable that separates the mean of the response in two different groups. In general, ensemble learning methods such, i.e. bagging and boosting, aren't effective for linear models, since the both the forms of this models and of this ensemble methods are additive. When methods such as bagging and boosting are adopted with CART, the performance of regression trees is often consistently improved, but it comes at 2 main costs:

- higher computational cost (that depends on the hyperparameter choices)

- lack of interpretability (that can be partially compensated by variable importance tecniques for ensemble methods over trees [Has09b])

For this specific dataset, the presence of many dummy and categorical ordinal variables, a regression tree seems to be a more appropriate choice, since it gets better than linear models (that assumes linearity over the predictors) the interaction effects between covariates. For example, the variables latitude and longitude, extracted from zone, aren't so useful (withouth any transformations) for a linear moldel such as OLS or elastic-net, because there is a non linear relationship between them and the response, but can be used by a regression tree, that can segment the space in a discrete way exploiting the interaction over them. Another advantage of regression tree is the minor work requested to preprocess the continuous variables: if a linear model is used, then there would be some transformations of the continuous (and also of poisson distributed) variables, such as logarithms, polynomials, dummy interval segmentation or other interaction terms. With regression trees this choices haven't to be made, since the building process of a single split over a variable and the one over any of its infinitely possible univariate transformations led to the same result.

3.2 ensemble learning of regression trees

Bagging for regression consists in making a mean of the predictions of different models, which predicts all the same response but are trained on a different subset of observations (or on the same dataset with different weights for each of them). Least squares boosting, or boosting for regression, consists on iteratively generate predictions with a new model over the residuals of the predictions of a previous model, and update the final prediction with a learning rate factor. Boosting [Has09a] fits an additive model in a set of elementary basis functions, like:

$$f_M(x) = \sum_{m=1}^{M} \beta_m b(x; \gamma_m)$$

were β_m can be seen as the learning rate, that usually is constant, but can also decrease along the iteration process, and $b(x;\gamma_m)$ are some basis functions (like regression trees, or any other predictive model) with their specific parameters γ_m. Both methods can include column-subsampling (that consists in select randomly a fixed number of predictors to use at each iteration-model) to avoid overfitting. The Random Forest is a variant of classical bagging that reduces the variance of the predictions with the inclusion of column-subsampling [Bre01, Bre96].

3.3 error prediction and hyperparameters

To predict the log of price/sqmtrs, the 8000 rows of the dataset were splitted in 70% of training, 20% of validation, to manually adjust the optimal configuration of the hyperparameters, and the last 10%was left to the test set, to compare the different models. Here are listed the MAE obtained on the test set by each one of the models. For simplicity, the results are rounded to the nearest hundred.

- a OLS regression: test MAE of 85600 when the response is the log(price/sqmtrs) (and 80200 when the response is log(price), that is not considered with the tree based models). Using a penalized model, i.e. elastic-net, for lambda¿0 the MAE error is worse than the one of OLS.

- a regression tree : test MAE of 95100. The best configuration was given by a small penalization, that can be given alternatively lowering the cost complexity pruning parameter or increasing max depth (the maximum number of sequential split nodes), the minimum number of observations in a note (since it can be used to build a new split) , and the minimum number of observations that should be present in a leaf node. This very same hyperparamters are present in the function gbm, to make boosting over trees, and in ranger, to make random forest.

- boosting over regression trees: test MAE of 80500. The best model uses approximately 100 full trees (with maximum max depth and no regularization) and a learning rate of 0.1.

- random forest, with all the predictors (i.e. bagging of trees): test MAE of 73100. The best model uses approximately 100 trees, sample with replacement of all the observations and uses all the variables, so the bagging of trees is preferable to random forest.

- boosting over bagging of trees: test MAE of 69000. This model uses 100 iterations of boosting, with learning 0.1, with bagging of trees models, each trained on just the 90% of the training, and with the same hyperparameters of the model described above (100 full trees with all the variables and sampling with replacement of the entire data).

The better ratio between performance and simplicity was showed by the tree bagging model, which reached a MAE of 73100. The application of boosting to multiple versions of it leds to a MAE of 69000, that seems a significat improvement (and came at the price of a much higher computational cost).

3.4 R implementation

The algorithm used for the bagging of trees was the R ranger package (that is a more efficient version of the classical randomForest package available in R). The boosting over bagging of trees, that results in the best predictor, consist in the application of an auto-implemented boosting function [ens] to the random forest function available in ranger.

4 conclusion

OLS model works better when the response is price and square meters is included as a predictor, but this assumption seems to be wrong. The usage of tree based methods leds to a better result when the response is the ratio between the price and the square meters of the house.

References

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[ens] boosting and bagging r functions. https://github.com/Fede-Rausa/R_coder_tools/blob/main/functions/regression_models/ensemble_models.R.

[Has09a] Tibshirani R. Friedman J. Hastie, T. Boosting and additive trees. *elements of statistical learning*, pages 337–387, 2009.

[Has09b] Tibshirani R. Friedman J. Hastie, T. Variable importance. *elements of statistical learning*, pages 592–594, 2009.

[Ize08] Alan J. Izenman. Recursive partitioning and tree-based methods. *Modern multivariate statistical techniques*, pages 281–314, 2008.