Automatic valuation model for Hotel property prices

Milestone Report 2

Preprocessing

I ended up dropping all observations with missing values rather than in imputing them. The reason for this is connected to the fact that, as described above, I only had access to the original data for the newer observations. The older observations already came in merged form, and unfortunately all observations with any missing values had already been dropped. Likewise, many variables that were seen as less important had already been dropped.
This is because the data were previously used for regression modeling, so the variables were reduced to a small number deemed most relevant by domain experts. To enable easier modeling, any rows with missing observations were dropped. (This is because the regression models were intended for statistical inference, which requires more complicated imputation than predictive modeling that is typically the goal and machine learning requires. In particular, in order to get correct standard errors, *multiple* imputation needs to be performed.)

I decided to perform the same processing on the newer data. This leaves us with 7,500 rather than close to 11,000 observations, and 32 rather than at least 100 usable features. Note that since gradient boosting does not have any trouble dealing with missing values and is able to use even a high number of features, this decision favors ordinary linear regression. Furthermore, the small size of the data set does not allow gradient boosting to reach anywhere near its full potential. This is intentional, because it creates what is sometimes called a "least likely case": If it can be shown that gradient boosting is preferable to linear regression even in such a case that is stacked against it, it makes a stronger case in favor of it if it prevails even in such a unlikely case. If we preprocessed the data specifically for gradient boosting (kept all possible features) and as we increase the number of observations (don't drop observations with missing values, and in general as the number of collected data increases over time), the superiority of gradient boosting will be more pronounced.

Next, let's talk about the encoding of categorical and ordinal variables. I took different steps here for different models: For gradient boosting, I performed one-hot encoding for real categorical variables, but mapped cortical variables that were in fact ordinal to integers corresponding to the correct order. This works because tree-based models do not distinguish between ordinal and numeric variables. By contrast, for linear regression, I had to perform one-hot encoding for both categorical and ordinal variables. Leaving ordinal variables as numeric would otherwise amount to the differences between categories being treated as equal, which is usually not the case.

Goal

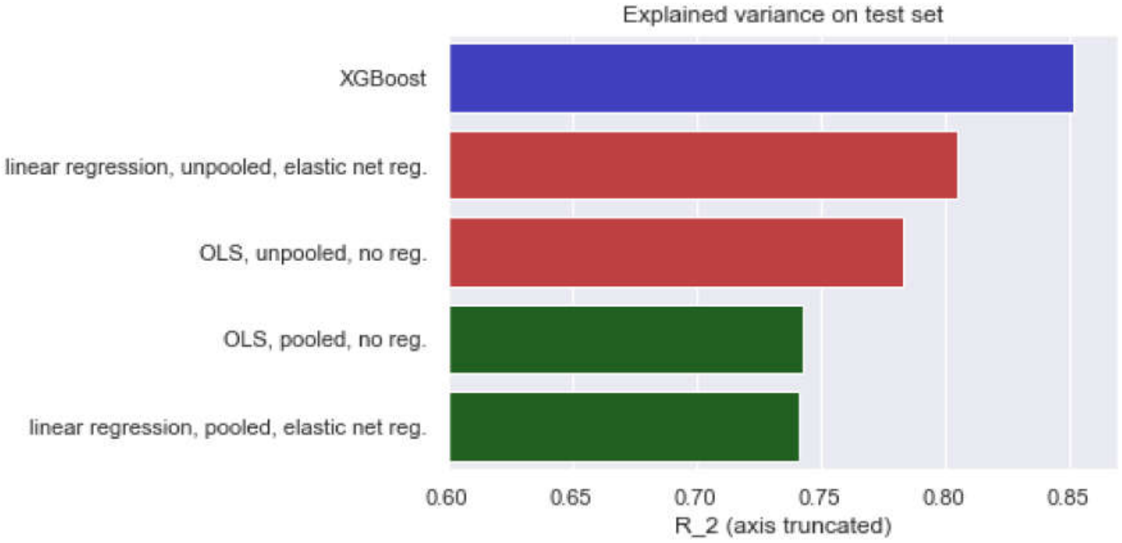
The main goal of this analysis is to show that models of hotel property prices should use machine learning techniques that are able to model more complex relationships between variables, rather than using the linear regression models that have been traditionally the main workhorse for statistical inference.
A secondary goal of mine is to show that if linear regression models are used – e.g. as a first step or a point of comparison – it still pays to borrow two related techniques from machine learning, regularization and the practice of assessing the performance of the model by measuring predictive accuracy on withheld data.

The predominant way of assessing the trustworthiness of a model in applied statistics is based on verifying theoretical assumptions combined with model checking, e.g. analysis of the residuals. By contrast, it is common practice in machine learning to instead verify the fit of a model through empirical means, namely by withholding part of the data and then assessing the predictive accuracy on the withheld data. This is preferable because it avoids relying on dubious assumptions about the distribution of the data. Furthermore, in practice model checking is often neglected, or the findings are not presented. Furthermore, the model checking that is done is biased towards not rejecting useless models. Surely this is a major cause of the replica place in crisis in science. For a great discussion of these issues, see [2 cultures in statistical inference]. The second technique borrowed from machine learning that should be standard practice in applied statistics is regularization. The purpose is to avoid overfitting our model to the data we train our model on. This problem is usually dealt with an applied statistics by practices such as using the adjusted R_2 for model selection. However, the adjustment done to the R_2 for the last degrees of freedom is not based on any principled way. By contrast, fit statistics based on information theory, such as AIC or BIC, *are* derived from a theoretical framework that tries to adjust for the overfitting to the training set. However, they are not only much less widely used than the adjusted R_2, but are also based on a lot of assumptions about the distribution of our data that are rarely completely satisfied in practice, so it is often hard to say how confident we can be in its prescriptions.

One might think that for our sample size of over 7000, regularly station might not be that important for a simple linear model. However, remember that we had 3 categorical variables with several hundred different categories. If we perform one-hot encoding on these, we thus end up with about 1000 predictors. Since two out of these three variables referred to the location of the property (e.g., the Metropolitan statistical area), we can expect that it would indeed be y useful if we could include these variables. The only good way to find out if it is worth losing so many degrees of freedom is to estimate the model on part of the data and then seeing which model performs better on the withheld data.

Results

The below table sums up the explained variance on the test set:



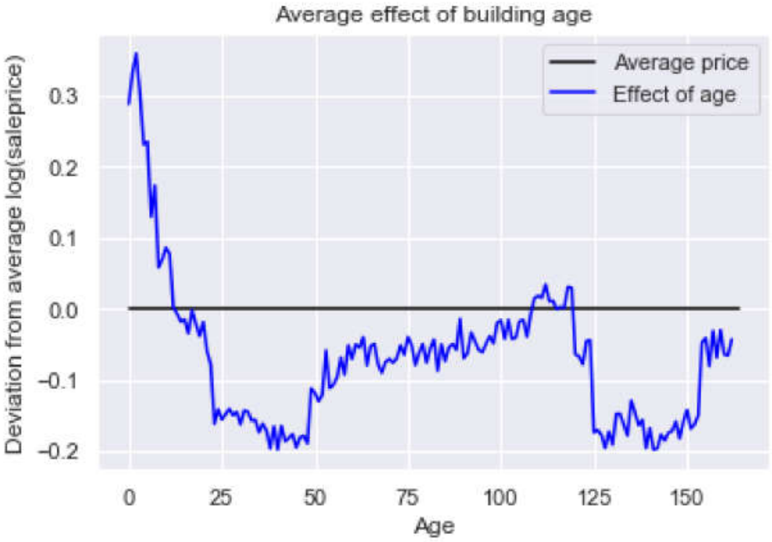
Overall, we see that gradient boosting gives us the best performance on new data, with an R_2 of 0.85. By contrast, OLS only achieves 0.78 and 0.74 for the cases unpooled and pooled case, respectively, if we do not use regularization. Adding elastic net regularization improves the performance in the unpooled case from 0.78 to 0.81. By contrast, regularization does not make any noticeable difference for the pooled case. This is not too surprising: In the unpooled case, adding dummies for hundreds of locations and affiliations of the sold hotel led to a proliferation of variables, so regularization is necessary to prevent overfitting. By contrast, in the unpooled case, since we have about 7000 observations and only 37 variables, a linear model seems to have almost reached its capacity: The number of observations is high enough that there is basically no overfitting, and as a result regularization does not yield any noticeable improvement.

Thus, this analysis successfully demonstrates that a state-of-the-art machine learning model is It is preferable to OLS, even in situations that seem to be stacked against it, such as a small data set (7000 observations) and not a lot of variables (35). While gradient boosting is more complex to implement, there have been a lot of recent developments that make advanced machine learning techniques more accessible, such as the rise Automated machine learning (AutoML). Furthermore, precisely because machine learning models are more complex, there has been a greater emphasis on developing tools to make model checking easier. In particular, the practice of measuring predictive performance on withheld data offers a simpler and more reliable alternative to applied statistics' heavy use of test statistics that are based on a lot of mathematics as well as often dubious assumptions.

This brings us to the 2nd goal of this analysis, namely to show that, independently of whether more sophisticated models from machine learning such as gradient boosting are adopted, there are also gains to be made from adapting practices from machine learning such as regularization and the use of a separate test set. While using regularization does not always yield noticeable improvements (see the pooled case), it is good practice to always do so, because it is not always easy to say beforehand whether it well. Furthermore, the additional effort involved is small, and it actually saves time in other ways, because we do not have to think as hard about which variables to include and which to exclude. Speaking of which, the reason that regularization did not yield any improvement in the pooled case may also be due to the fact that I'm using a data set in which the variables have already been pre-selected to optimize it for use with OLS. If I had access to the full data set, it is likely that regularization would have an edge, because it is able to intelligently select which variables contain useful information in which mainly contain noise.

Likewise, this analysis demonstrated how important it is to use a second central technique from machine learning, namely the practice of withholding data when training the model and then evaluating the model on these data. Without this, for example, it would have been impossible to say whether the pooled or unpooled model is preferable, or exactly how much better gradient boosting is likely to perform on new data.

Finally, let me say a few words about interpretability. As mentioned above, I argue that it is not the "fault" of more complex machine learning models that they lack easy interpretability; rather, reality is too complex to fit well into simple linear models. Thus, better fitting models will usually be less interpretable for the human mind. While interpretability is good to have, it is not worth sacrificing accuracy and fit.
I illustrate this problem by looking at the effect of age. Gradient boosting models both the nonlinear impact of the feature, as well as the interaction between features. In order to make the effect of age more interpretable, I abstract from the interaction, and simply calculate the effect of age, averaged over all situations. To do so, use a resampling technique: I draw a large number of samples, but randomly assign age values to them. Then, I generate predictions, and plot how the average prediction varies by age.



The resulting graph shows the nonlinear shape that would be impossible to generate by a lower-degree polynomial. This high capacity to adapt to the actual shape of the data is one of the reasons why models such as gradient boosting outperform linear models. As the number of observations grows, this advantage of gradient boosting becomes even more pronounced.