Prediction with Machine Learning for Economists assignment 2: Predicting AirBnB prices in Hawaii

Mihaly Hanics, MS in Social Data Science

This report is modelling and predicting AirBnB prices on Hawaii, testing out three machine learning models. Note: The assignment asked for a city, however I found Hawaii's AirBnB prices to be pretty homogeneous, the difference in islands is similar to difference in neighborhoods in cities.

Data: The data is gathered from Inside AirBnB. I filtered for Hawaii, for model building to September 2023, and for testing to December 2022. All code is found on GitHub: me9hanics/DA3-phdma.

Data preprocessing: I had to choose only those data, that had 2-6 renters, and were apartment-like. Aside from these and the target variable price, I found the number of beds, bedrooms, property and room type, the having of specific amenities, host verification and profile picture, review ratings, minimum nights the most important. The importance of each variable is discussed later.

The models:

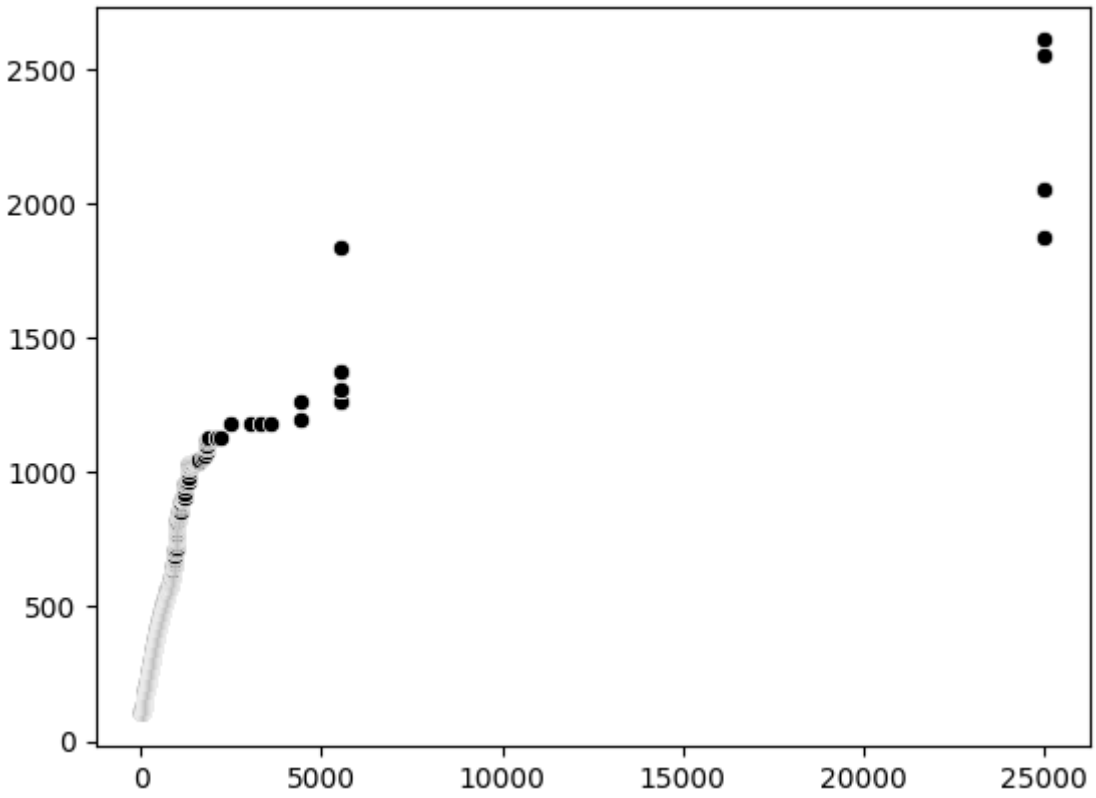
1) OLS: Before I fit a line on the data, I had to first take the logarithm of the prices for a more linear plot, and normalize the data. I also filtered out prices that were 0 (free) or more than 30000.
2) Random forest: I ran a cross validation to find the ideal maximum features parameter, and ideal minimum samples split. I used 200 trees for ensembling, and a max depth of 10. The best CV parameters were maximum features: 5, and minimum samples split: 10.
3) Gradient boosting: I used 1000 estimators with depth of 10.

The results:

Mean-squared error on the training set:

OLS: 278k before filtering out outliers, 24k after. The predictions:

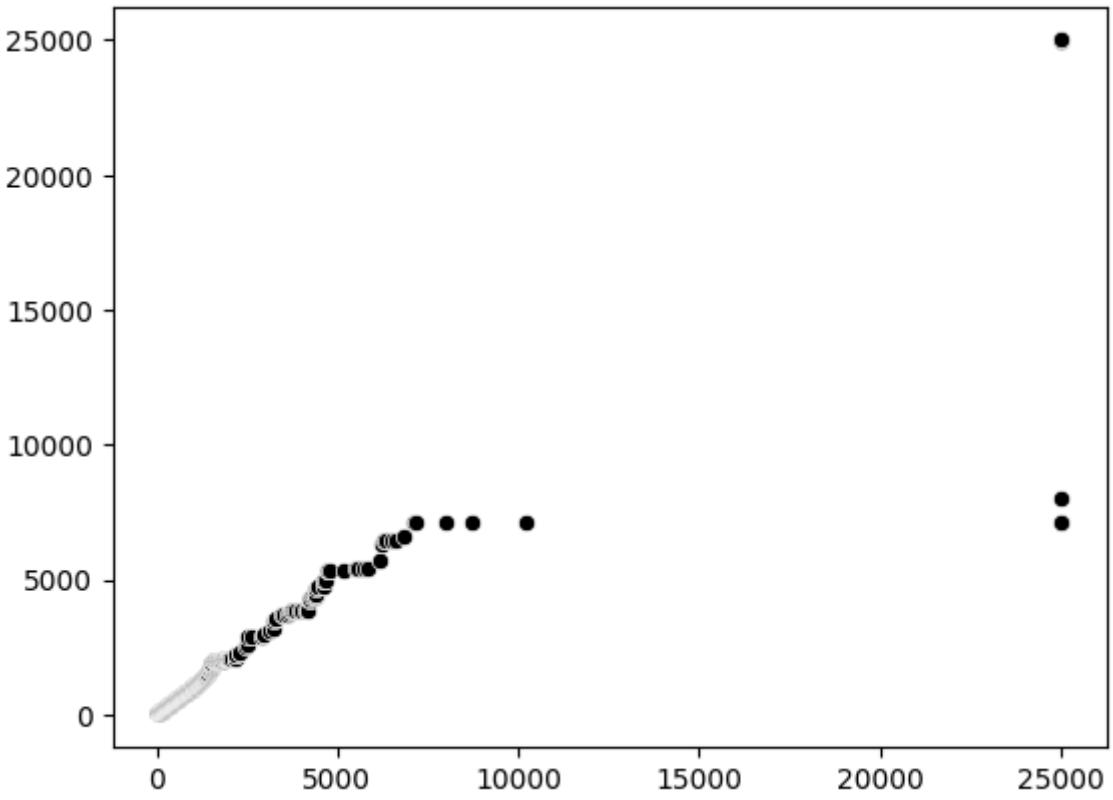
Random forest: 245k before filtering out outliers, 11k after.



As we see, the line in both cases in the beginning is very steep, then plateaus (OLS for longer), and fails to any accurately predict outliers.

Gradient boosting: 138k before filtering, 577 (!) after filtering.

This is a clear winner, to have only MSE of 577 after removing compared to 10-20k in the other models is fantastic. Indeed, the prediction is much closer to a 45° line.



However, as we will see, the results are nowhere this good, as these are just predictions on the set that we trained our models on.

Mean squared error on the December data:

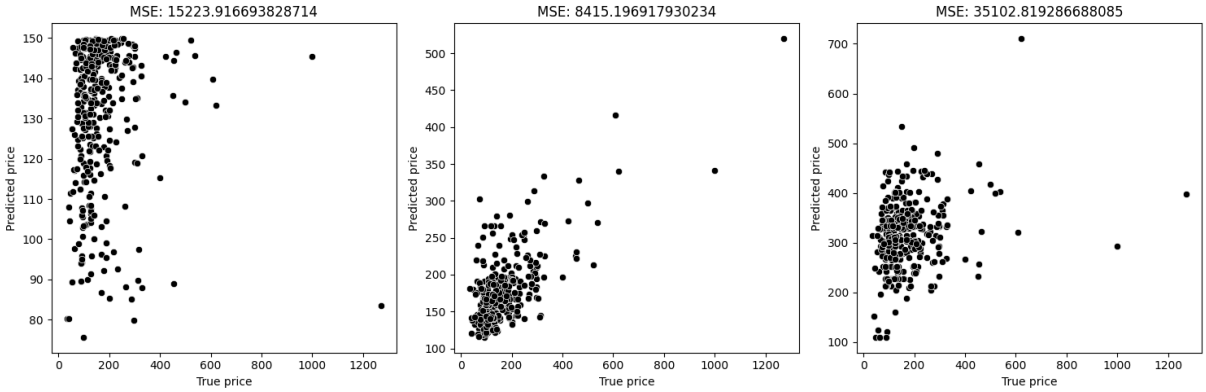
OLS: 444k, terrible, but expected.

Random forest: 421k. Much worse than before, but the best result.

Gradient boosting: 450k. Compared to the expectations, it did worse, MSE is bigger than for our random forest regressor and even for OLS, which is underwhelming.

If we'd look at the price plots on these predictions, we'd see that there is a big underprediction for vastly all values. I tried re-doing the normalization, but I realized that this is an apparent problem even for predicting the data we were training on. This is very interesting and would be interesting to figure out how to not underpredict the values. I had the feeling, that the problem is an unbalanced dataset: we have too few large values, and many small values. Thus, our models are great at predicting small values, and bad at predicting big values.

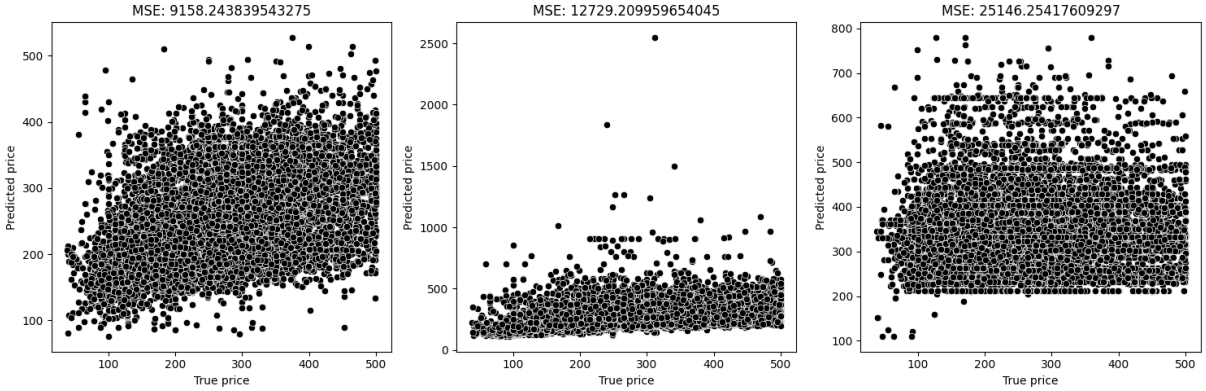
Some “random” small values:



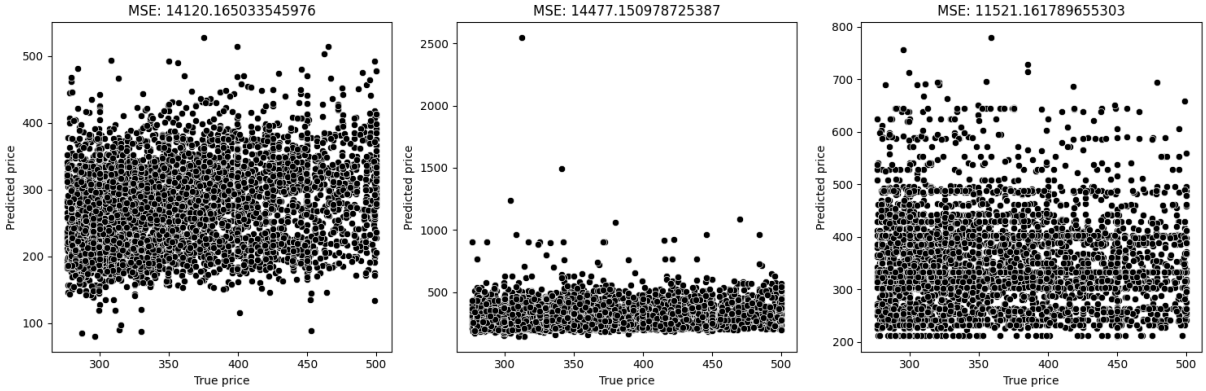
Here we finally see that the underguessing is not apparent that much anymore. In fact, gradient descent rather overguesses in this region. Random forest guesses well, OLS underguesses, so if we filter the bottom values too, we’ll have a higher OLS MSE.

If we exclude the highest 12% of price values, we actually get much better results:

MSE: OLS~9k(!), Random forest~12k(!), Gradient descent~25k



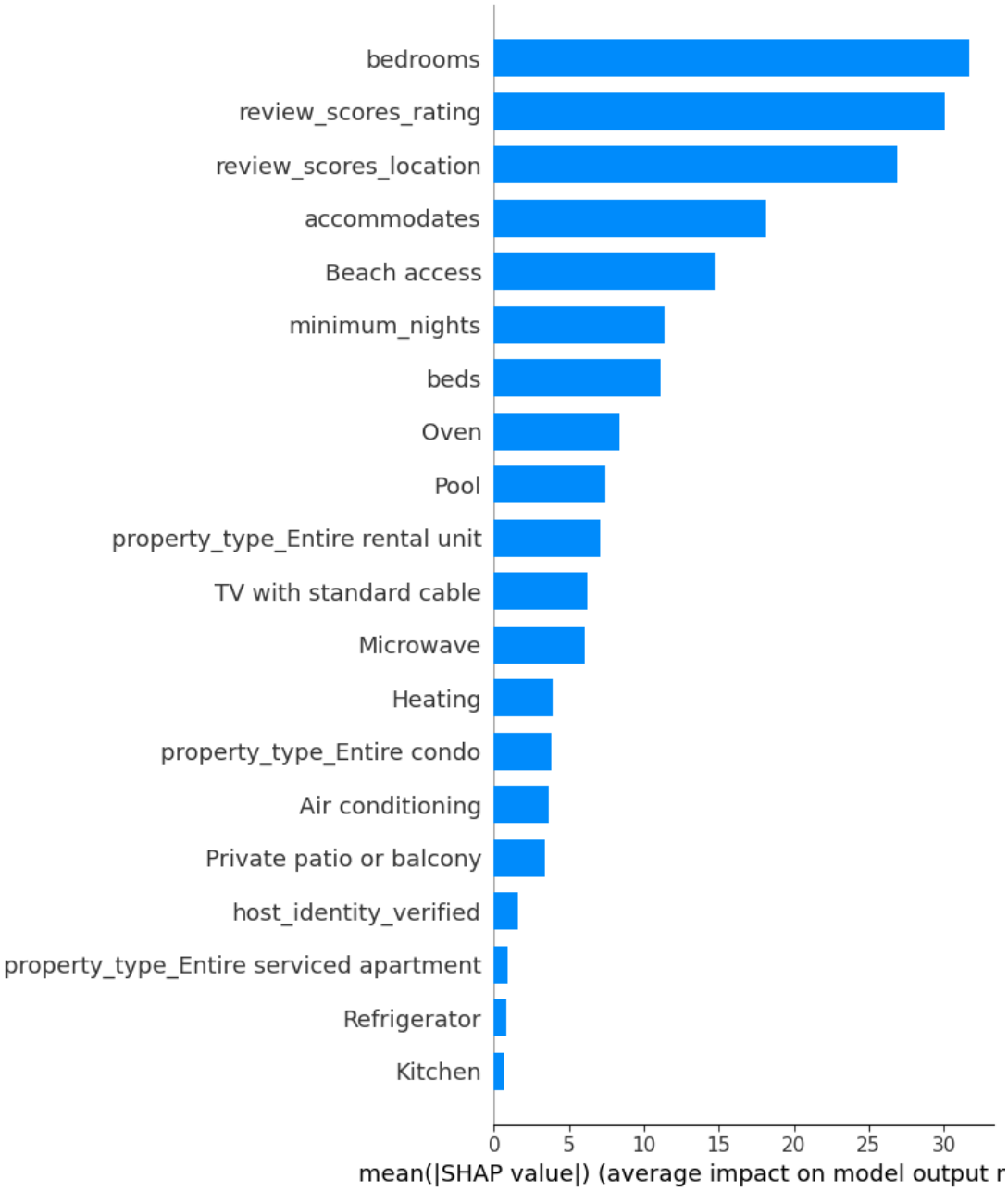
OLS surprisingly does best, Random forest also does well and would be the best predictor in the price range around 100-200$, gradient descent overshoots again. However, for the “median values” (near the 6000th value), the gradient descent model does best:



Which shows, that each model has its “best ranges”. The best model we could get is actually by combining all models correctly.

Which attributes matter the most?

The last task in the assignment was to find which attributes added up to the predictions the most. A common way to compute this is by using Shapley values. For this, I imported the SHAP library, and used it to generate feature importances.



The most important feature was bedrooms. Interestingly, beds, and refrigerator ranks quite low, whilst beach access ranked high. It can be important in Hawaii, but is typically not a feature you'd expect to have impact. Minimum nights rightly impacted the value, the longer you stay, the better deal you get. Reviews play an important role too.

Overall, we can say that building a right, accurate predictor for AirBnB prices is not an easy task. The best model may even vary from city to city, island to island.