BCI AVM

WHITE PAPER

INTRODUCTION

The Blockchain & Climate Institute (BCI) is a progressive think tank providing leading expertise in the deployment of emerging technologies for climate and sustainability actions. As an international network of scientific and technological experts, BCI is at the forefront of innovative efforts, enabling technology transfers, to create a sustainable and clean global future.

BCI AVM is an automated valuation model built for the UK real estate market specifically for the purposes of the Blockchain Climate Institute. Using advanced machine learning technology, it focuses on providing industry best-practice valuations which assist BCI asset risk models, CREM models, and others, which in turn provide a basis for residential real estate asset risks due to climate change, anywhere in UK. This paper explains the core concepts and functionality of BCI AVM.

COVERAGE

BCI AVM valuations are limited to single family homes, condominiums, apartments, and townhomes in the UK. Mobile homes, land, and commercial properties are not valued and are excluded from the database used during AVM development at this time, so that they do not bias value estimates of the included property types. BCI AVM provides valuations for 16M+ properties across all major UK cities and regions.

Individual valuations are based on local deviations between properties within the same postcode or the nearest comparable real estate market. In all cases, data used for development of the valuation model has been captured within 24 months of the AVM valuation date.

METHODOLOGY

AVM values are derived using technologically advanced machine learning methods, including:

- Feature engineering derived from small clusters of similar properties
- Gradient boosting algorithms

Both point values as well as value ranges are provided for each property that is valued. Value ranges are developed through a statistical bootstrapping approach that allows for asymmetrical ranges around each point value to be provided. The bootstrapping approach is flexible enough to accommodate reasonable BCI research quality standards for price recommendations. Thus, the upper and lower range price estimates indicate a range within which the most likely AVM error will be found. We derive this range by calculating the known errors found when testing against up to 15-100 randomly selected comparable properties.

CONFIDENCE SCORE AND FORECAST STANDARD DEVIATION

Every property receives a unique confidence score. This confidence score represents the precision of the AVM estimate and measures the deviation between the range of values and the point value itself. To provide the most transparent measure of variability in the estimates, we provide the larger of differences between the low and high value and point estimate. The low and high values represent the 16th and 84th percentile errors found during our bootstrap random re-sampling approach which are then applied to the point estimate, providing a 68% confidence interval around the point value, given the set of randomly generated comparable properties valued nearby. In this way, asymmetrical distributions of potential values are conservatively represented by the confidence score.

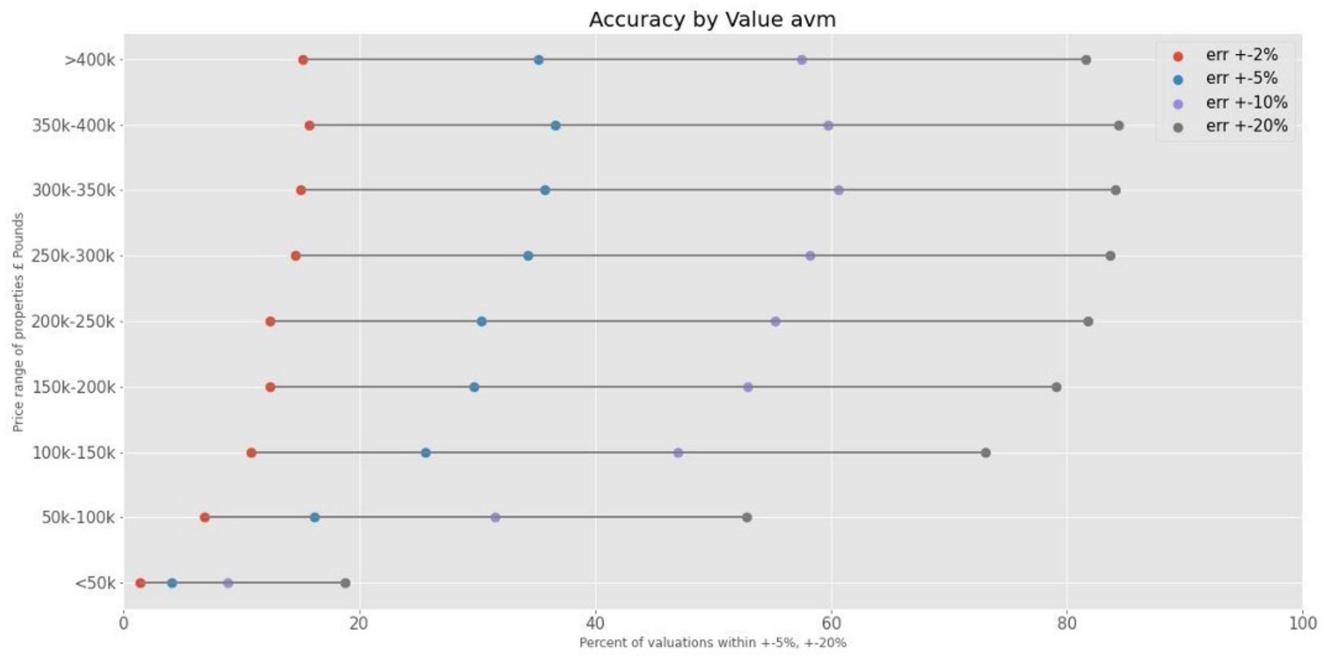
Confidence scores are calculated as 100 minus the forecast standard deviation, where the forecast standard deviation is the greater of the difference between the low range and the high range value and the point estimate, divided by the point estimate. This is an industry-standard way of computing confidence scores.

SCORING & ERROR METRICS

To gauge the error of BCI AVM we have compared our AVM values to over 100,000 properties listed on BCI over a recent period. These properties were selected to be representative of postcode, and local neighborhood characteristics such as property types, and sizes, and their proportional distribution over all of UK properties listed in EPC/Price data. For each new model version, these approx. 100,000+ properties are set aside at the outset of model training and development, in order to ensure that none of our production models ever sees these specific properties prior to AVM scoring.

For the most recent scoring session, the median difference between our AVM estimates and the true price was approx. £20012 POUNDS, meaning that one half of all valuations are within £20012 of the true price of the property. Additionally, 55%+ of valuations are within 10% of the true price, and 75%+ are within 20%.

MEDIAN ABSOLUTE ERROR	£33918
MEAN ABSOLUTE PERCENTAGE ERROR	%15.42
MEDIAN ABSOLUTE PERCENTAGE ERROR	%9.45
MAX ERROR	£920230
MEDIAN ABSOLUTE ERROR	£20012
R2	0.89
ROOT MEAN SQUARED ERROR	£56414
% OF VALUES WITHIN 5% OF TRUE PRICE	35%+
% OF VALUES WITHIN 10% OF TRUE PRICE	55%+
% OF VALUES WITHIN 20% OF TRUE PRICE	75%+



To provide transparency to the valuation process, the following outlines how a typical property is valued with BCI AVM.

1. **Employ UK valuation model:** One pipeline, hereafter referred to as "predictive model" or "avm", which has been hyperparameter-tuned to achieve low error valuations for the entire UK area is deployed to create a point estimate for the target property.

$$\text{AVM} = \mathit{ML\ model\ prediction}\ (\mathit{the\ middle\ point\ value\ for\ the\ target\ property})$$

2. **Find 'comparable' properties.** Next, we select a set of 15-100 comparable properties with known true prices where the true price is a publicly recorded transaction which occurred during the previous 2-year period. Each "comparable" is based on the geographically nearest homes of the same or most similar type. When we are not able to identify a minimum of 15 comparable properties, we expand the criteria used to qualify what is a "comparable" property. In very rare cases, such as in rural settings, we may expand the geographical region until a property of similar type is found. Then, using a statistical random re-sampling procedure, we make predictions on the sample(s) of comparable properties. At the end of this step, we will have made predictions on all randomly sampled comparable properties with known true prices, and thereby calculated the error for each prediction.

$$\text{COMPARABLES ERROR}_{\mathit{list\ of\ each}\ \boldsymbol{avm}\ \mathit{error\ found\ for\ random\ sample\ of}\ n\ \mathit{comparables\ where}\ 15 \leq n \leq 100} = [\varepsilon_0, \varepsilon_1, \varepsilon_2 \ldots]$$

3. **Determine range estimate(s):** We then rank the errors of estimated values from low to high. To provide a confidence score for a high number of properties, we use the 5th and 95th percentile values as the low-end and high-end range of the value estimate, and we apply these error percentages to our original point value estimate as follows:

$$\omega_{LOW} = 5th\ Percentile\ Error\ in\ \text{COMPARABLES ERROR} = \frac{\text{AVM}_{\text{lower}}}{\text{AVM}} - 1$$

$$\omega_{HIGH} = 95th\ Percentile\ Error\ in\ \text{COMPARABLES ERROR} = \frac{\text{AVM}_{\text{upper}}}{\text{AVM}} - 1$$

$$\text{AVM}_{\text{lower}} = \text{AVM} + \text{AVM} \times \omega_{LOW}$$

$$\text{AVM}_{\text{upper}} = \text{AVM} + \text{AVM} \times \omega_{HIGH}$$

4. **Determine the FSD$_{\text{max}}$ & Confidence Score:** Next, we look at the percentage difference between the low (5th percentile) and high (95th) values and the original point value estimate. We take the larger of the two (the FSD Max) and subtract from 100 to create a confidence score. By taking the larger of the two, our confidence score accounts for skewed distributions of values and offers a conservative measure of our confidence in the AVM value.

$$\text{A} = \text{AVM} - \text{AVM}_{\text{lower}}$$

$$\text{B} = \text{AVM}_{\text{upper}} - \text{AVM}$$

$$\text{FSD}_{\text{max}} = \max(A\ or\ B)$$

$$\text{CONFIDENCE SCORE} = 1 - \left(\frac{\text{FSD}_{\text{max}}}{\text{AVM}}\right)$$

Versions: Prior to valuing properties with each new AVM version release, we conduct regular AVM scoring tests where we compare our AVM values to known true prices of properties that were recently listed on UK EPC/Price databases. Using this information, we then hyperparameter-tune model training parameters using search algorithms which seek to find those parameters which provide the lowest error. The most accurate data processing pipeline + model(s) get rolled-up into a python pipeline object which we refer to as our avm model.

PROPERTY TYPES NOT SUPPORTED

There are several property types we do not support, and don't include in our comparables sampling. We eliminate these property types for the following reasons: First, including them would have a negative impact on the model's predictive power for most other residential units. Second, these property types require different data that is not widely available.

1. Manufacturer homes: In the case of manufactured homes, a two-bedroom, two-bathroom unit would value much lower than most standard fixed foundation units of the same specifications (size, bedroom count, etc.)

2. Mansions & Ultra-Luxury Units: These properties have unique features that influence the value and impede intelligent comparable selection.

3. Residential units on farm or agricultural land: In this case, the land and its use influence the value more than the residential aspect of the unit itself. Because of this, the valuation requires other types of agricultural data which are outside the scope of this model's inputs.

4. Multi-units: These properties rely on rental data to determine value because they are most typically investment properties that are under multi-tenant lease. As such, they require different data and models.

MODEL EXPLAINABILITY – SHAP SUMMARY PLOT

The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The color represents the value of the feature from low to high. Overlapping points are jittered in y-axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance.

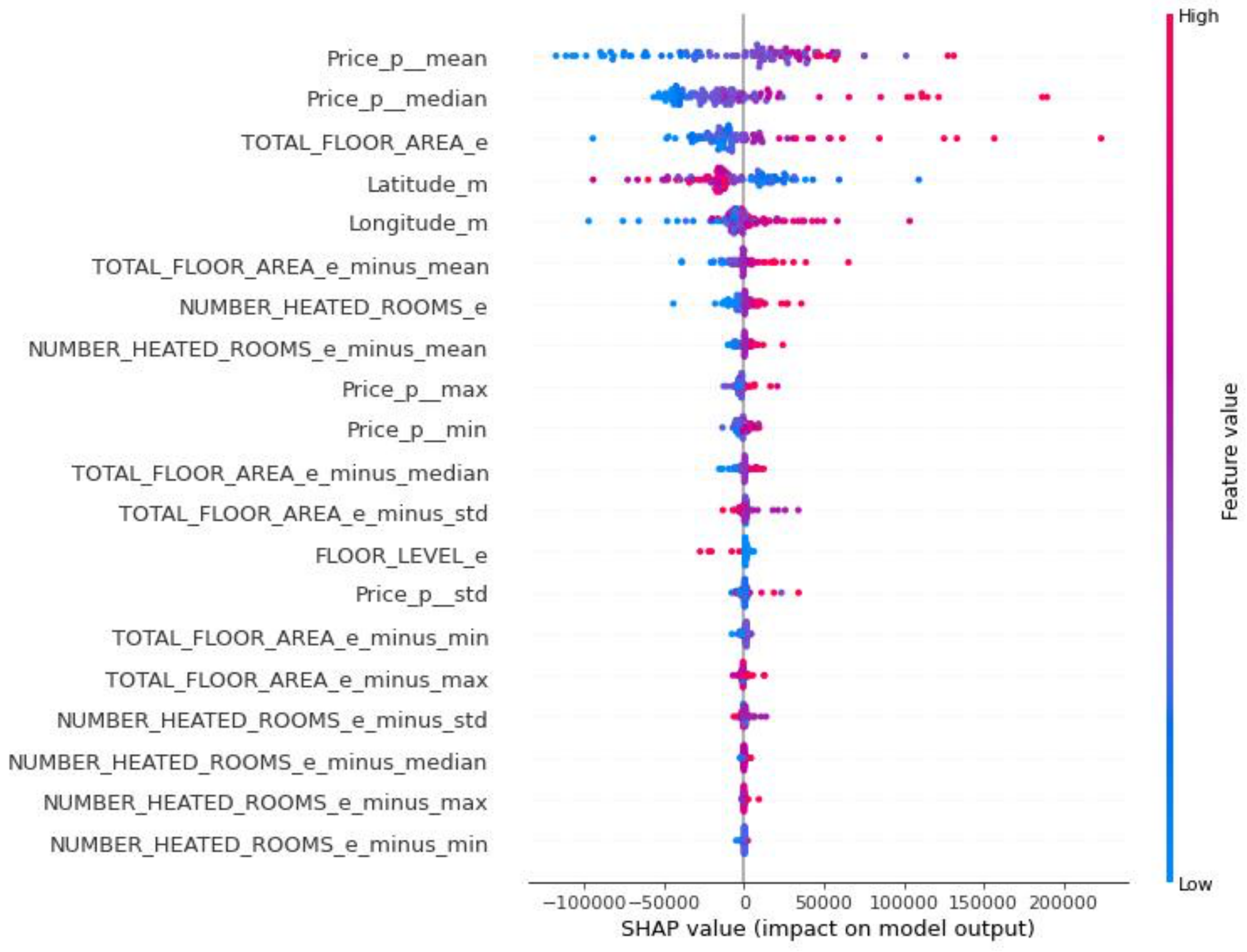


FIGURE: SHAP summary plot. Low value on Price_p__mean reduces the predicted avm valuation, a value increases the avm valuation. Reminder: All effects describe the behavior of the model and are not necessarily causal in the real world.

MODEL EXPLAINABILITY – SHAP INTERESTING PARTIAL DEPENDENCE PLOTS

While the above SHAP summary plot gives a general overview of each feature, the below SHAP dependence plots show how the model output varies by feature value.

In the Dependence Plots, note that every dot is a unit valuation, and the vertical dispersion at a single feature value results from interaction effects in the model. The feature used for coloring is automatically chosen to highlight what might be driving these interactions. Later we will see how to check that the interaction is really in the model with SHAP interaction values. Note that the row of a SHAP summary plot results from projecting the points of a SHAP dependence plot onto the y-axis, then recoloring by the feature itself.

Below we give the SHAP dependence plot for each of the chosen features, revealing interesting but expected trends. Keep in mind the calibration of some of these values can be different than a real world effect, so it is wise to be careful drawing concrete conclusions.

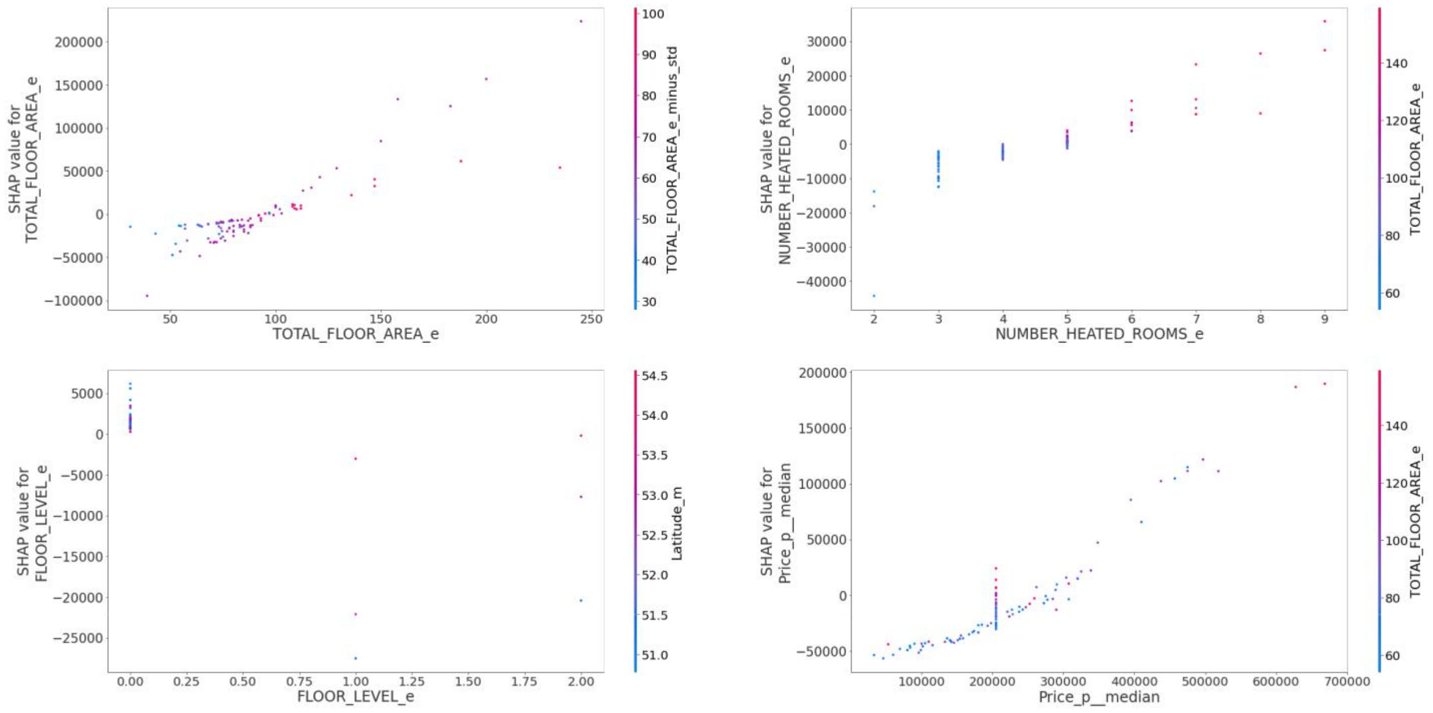


FIGURE: SHAP dependence plot for different combinations of features.

MODEL EXPLAINABILITY – UNDERSTANDING HIGH ERROR VALUATIONS

What about valuations with over 20% error? Is there any way we can drill down on these valuations to understand what is happening, and perhaps make some corrections to fix, or at the very least, avoid providing valuations when we know the model is most likely to have a high error?

We can in fact drill down on these high-error units. To do so, we first calculate the SHAP values for only the high-error valuation units. The below SHAP Summary Plot is produced considering *only* high-error valuations.

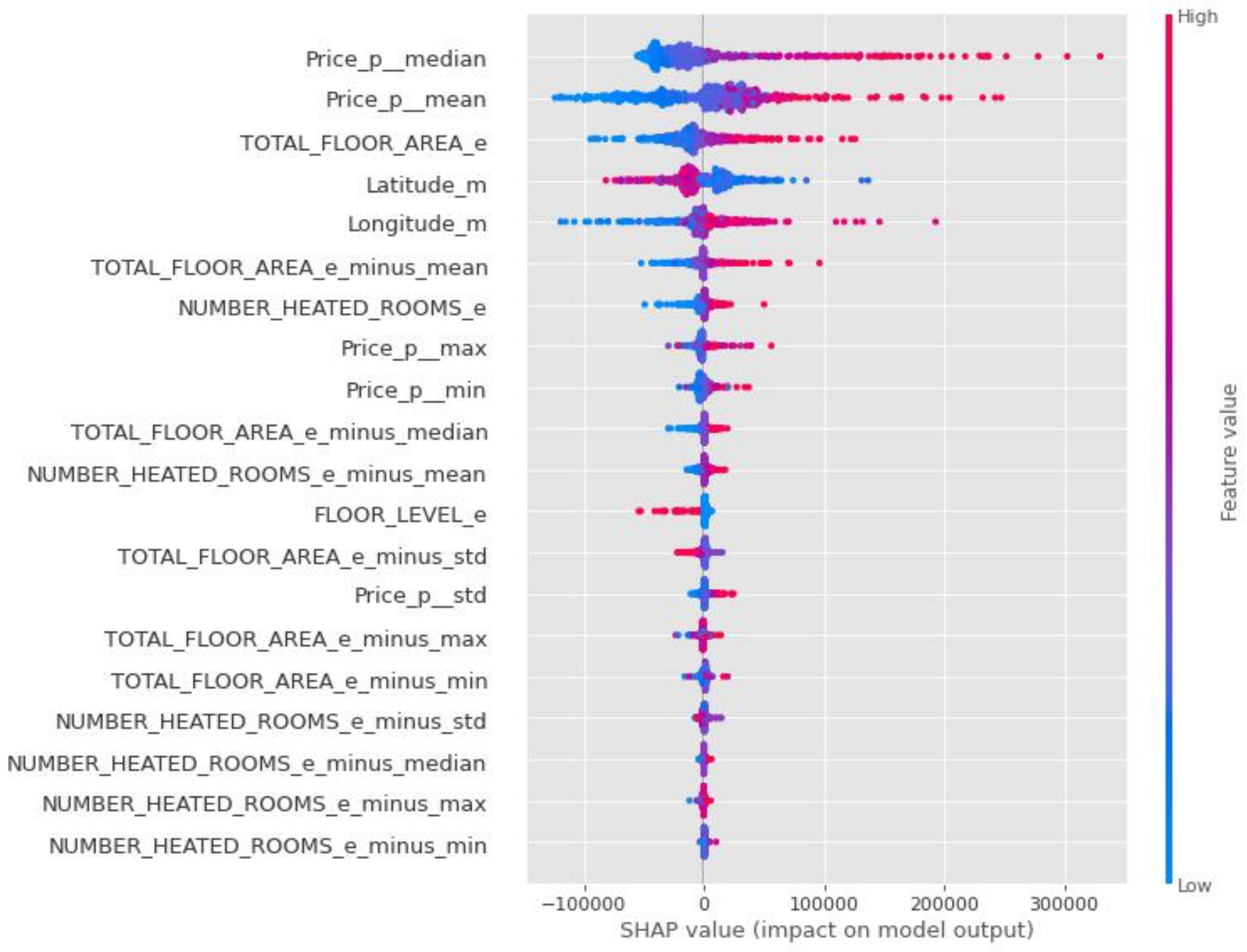
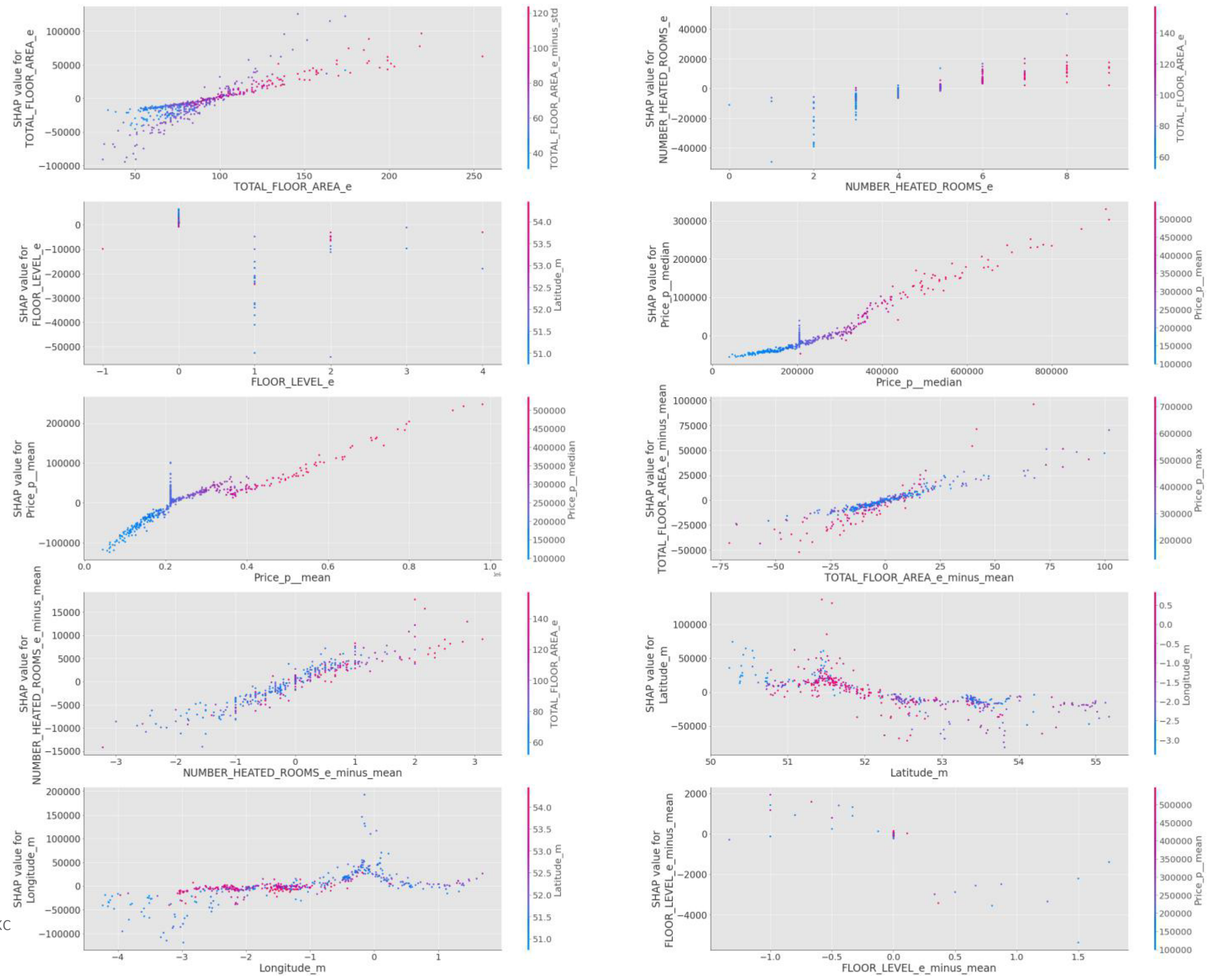


FIGURE: SHAP summary plot for only high-error valuations. Illustrates the most important features contributing to high-error valuations.

The feature that contributes most to high errors is Price_p__median which is an engineered (i.e. derived) feature built by looking at aggregate statistics about comparable units nearby the target property. In general, many of the engineered features play a large role in the high error valuations, but it's important to note that they also increase the performance of the model overall (i.e. without them, the model would perform worse). However, the overall model Summary Plots(s) and the high-error Summary Plot are very similar, and it's difficult to infer exactly what's happening from the Summary Plots alone.

MODEL EXPLAINABILITY – ABSOLUTE ERROR %

In the Dependence Plots, note that every dot is a unit valuation, and the vertical dispersion at a single feature value results from interaction effects in the model. The feature used for coloring is automatically chosen to highlight what might be driving these interactions.



MODEL EXPLAINABILITY – High Error % Overestimation

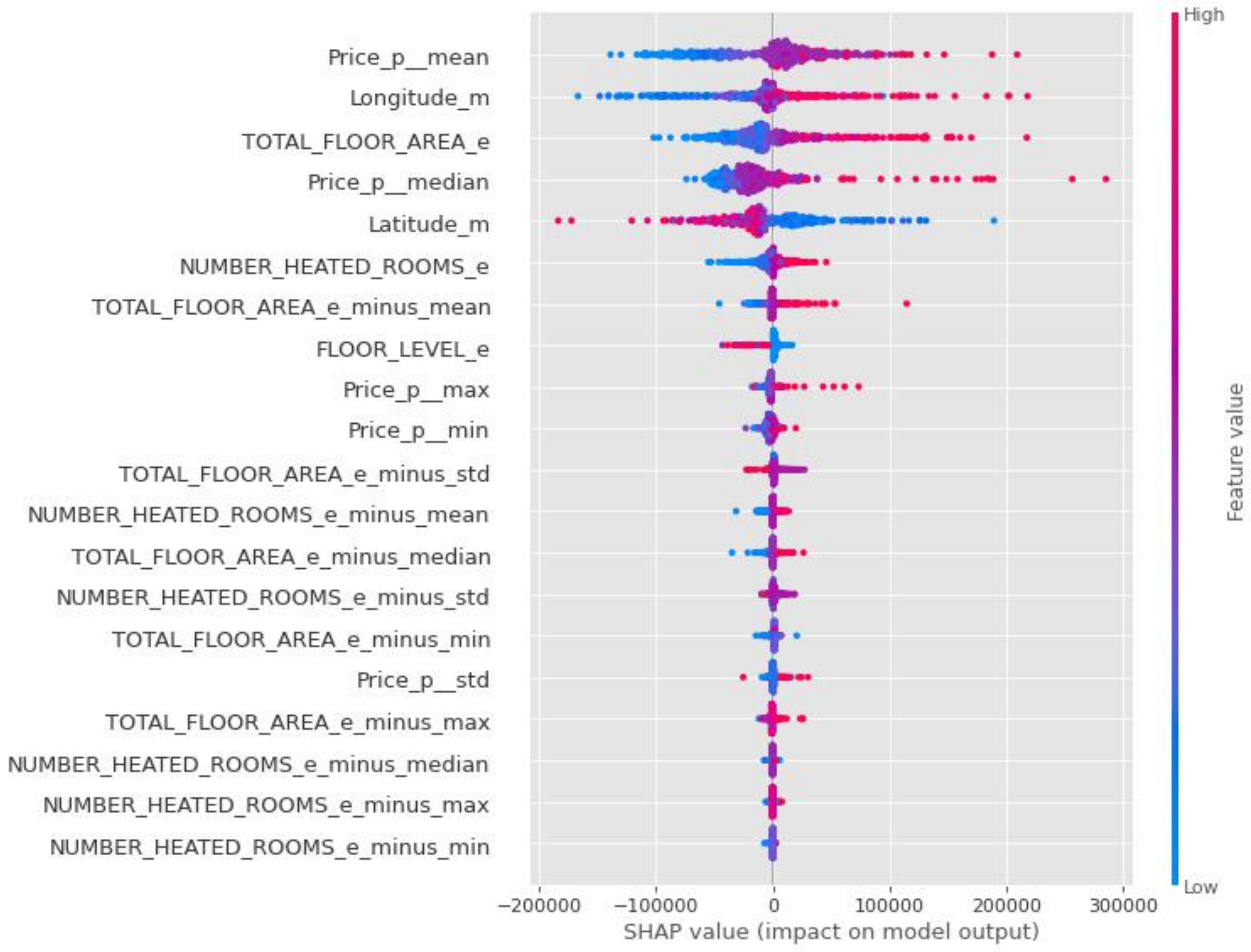


FIGURE: SHAP summary plot for only high-error valuations where the error-type was *overestimation*. Illustrates the most important features contributing to high-error valuations.

MODEL EXPLAINABILITY – Insights - Overestimation Error

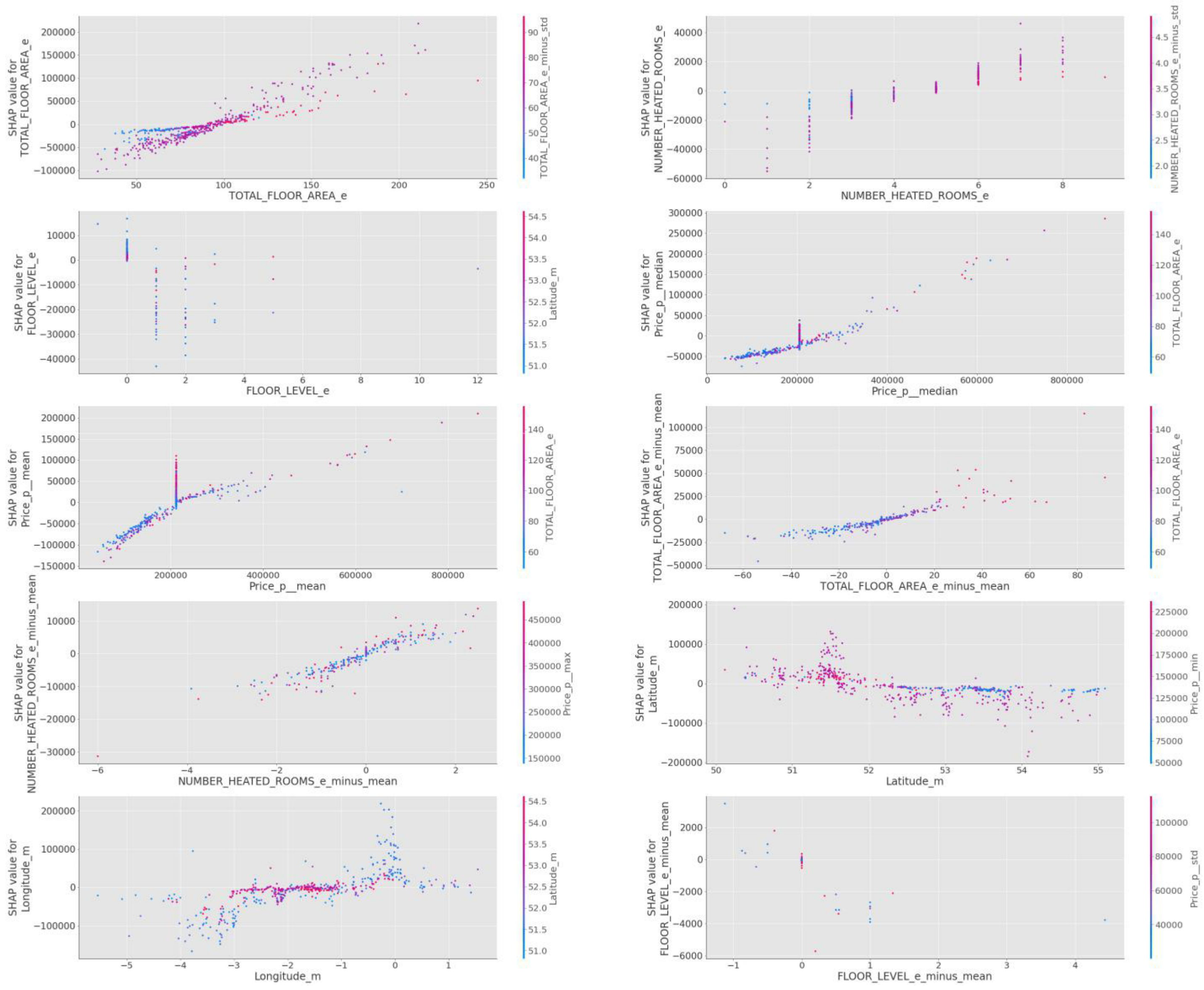


FIGURE: SHAP dependence plots for only high-error valuations where the error-type was *overestimation*. Illustrates the most important features contributing to high-error valuations with overestimation.

MODEL EXPLAINABILITY – Underestimation Error

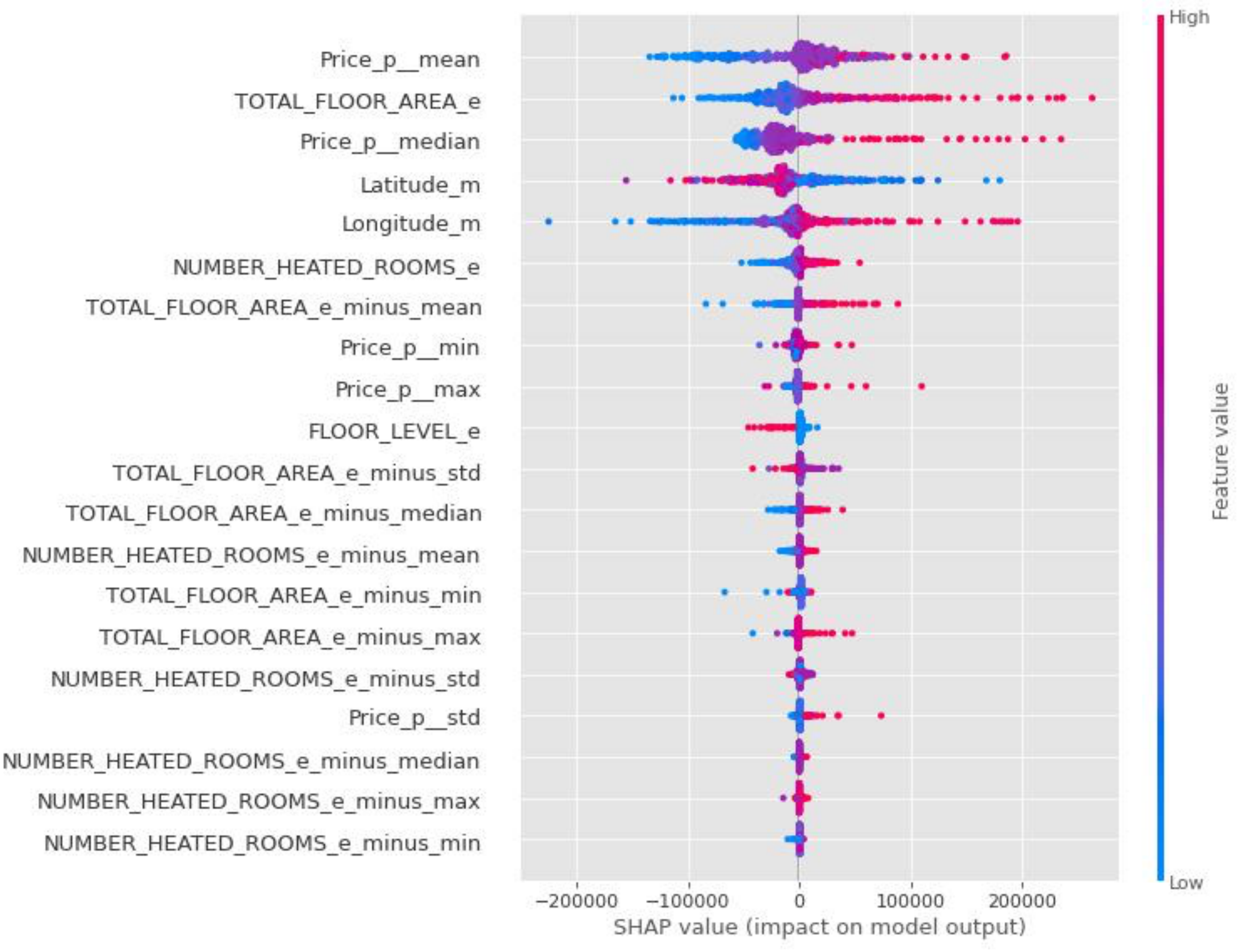


FIGURE: SHAP summary plot for only high-error valuations where the error-type was *underestimation*. Illustrates the most important features contributing to high-error valuations with *underestimation*.

MODEL EXPLAINABILITY – Insights - Underestimation Error

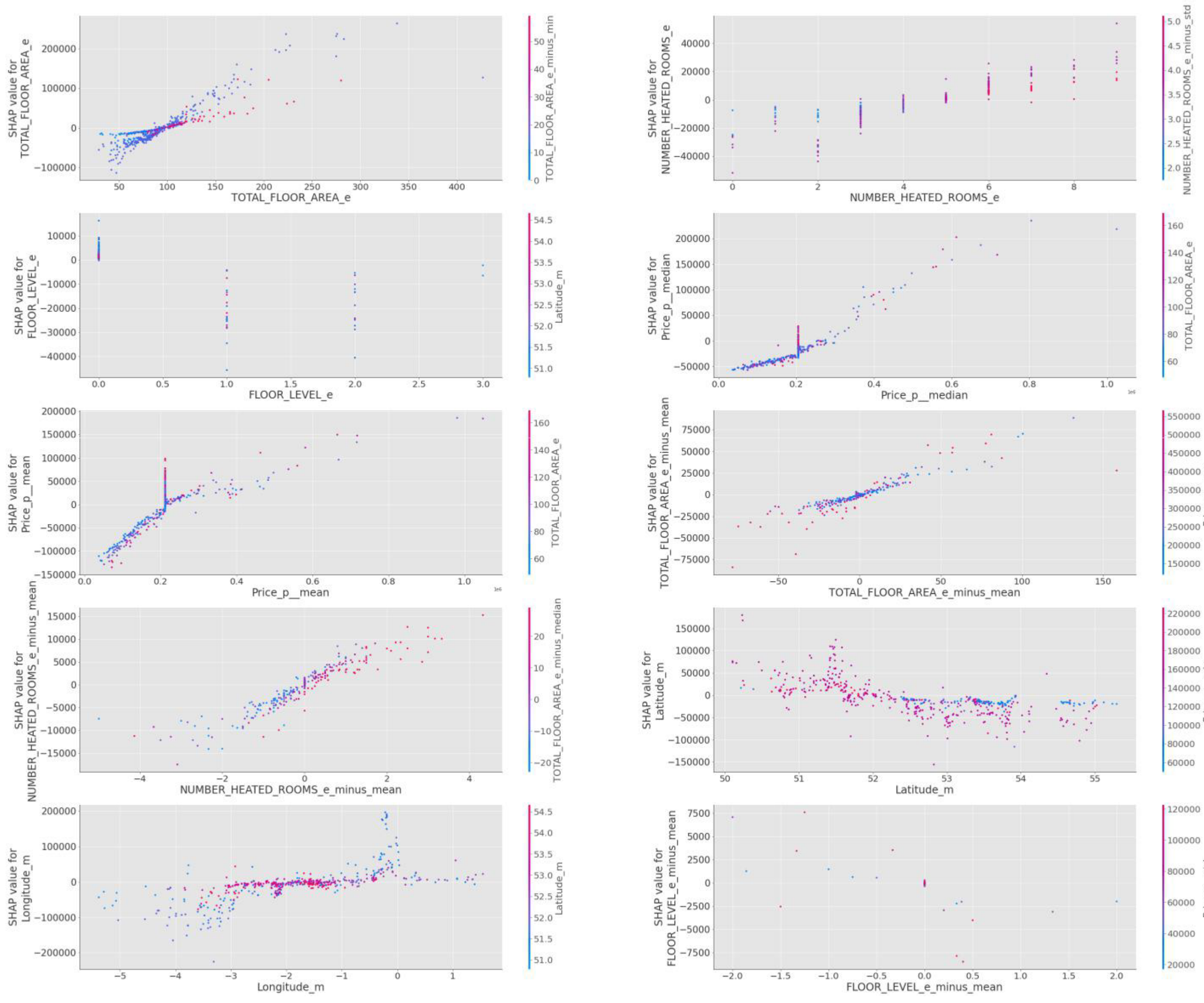


FIGURE: SHAP dependence plots for only high-error valuations where the error-type was *underestimation*. Illustrates the most important features contributing to high-error valuations with *underestimation*.

MODEL EXPLAINABILITY – Interesting Feature Interactions

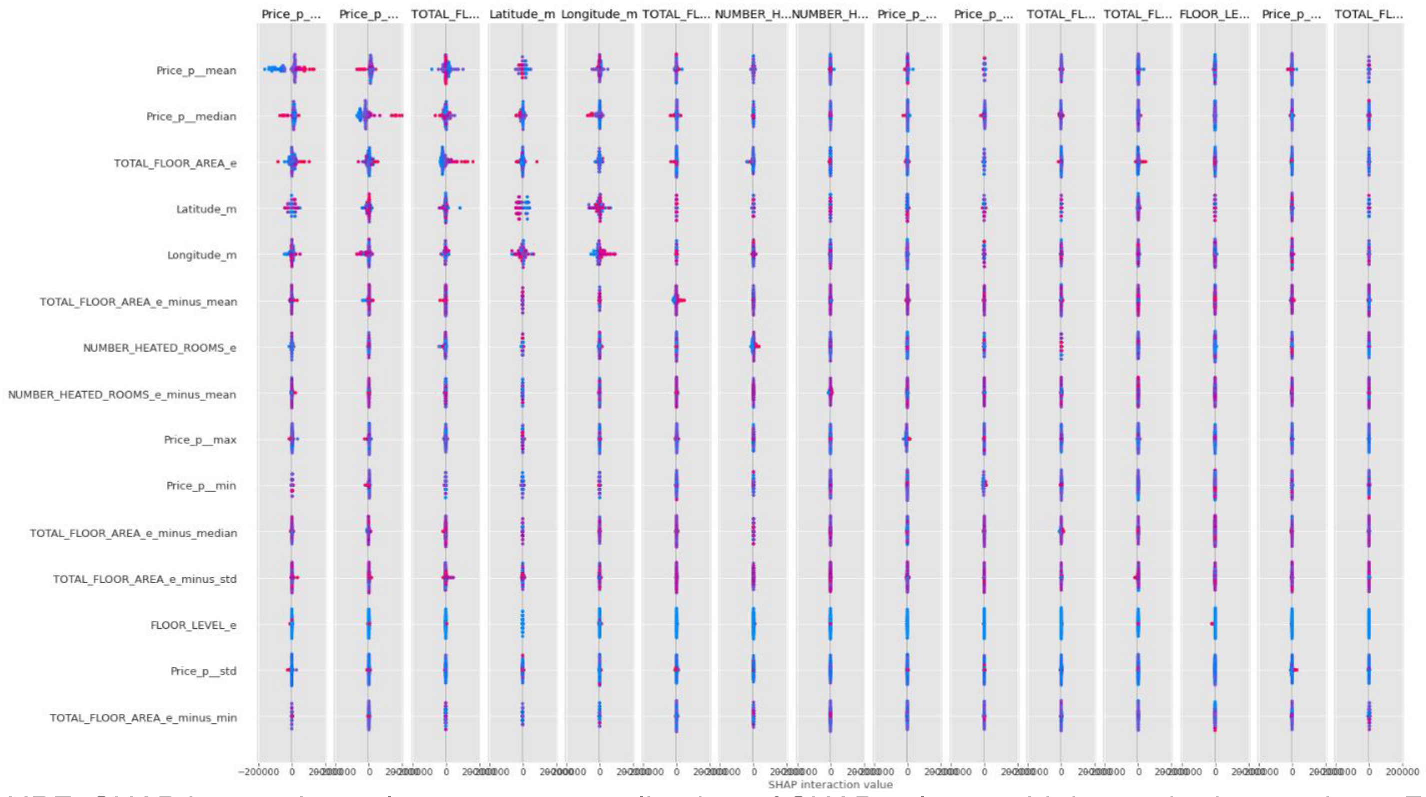
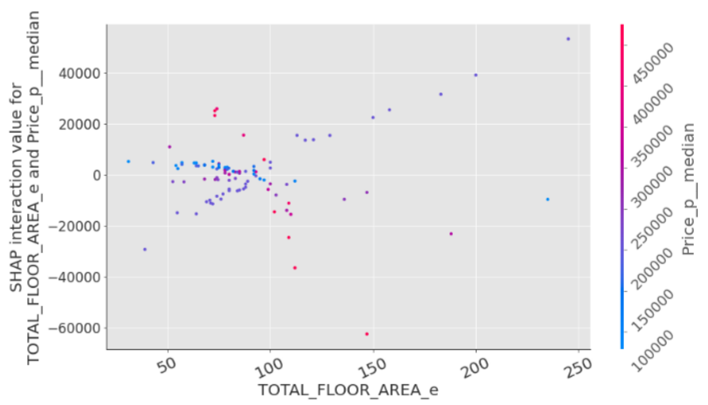
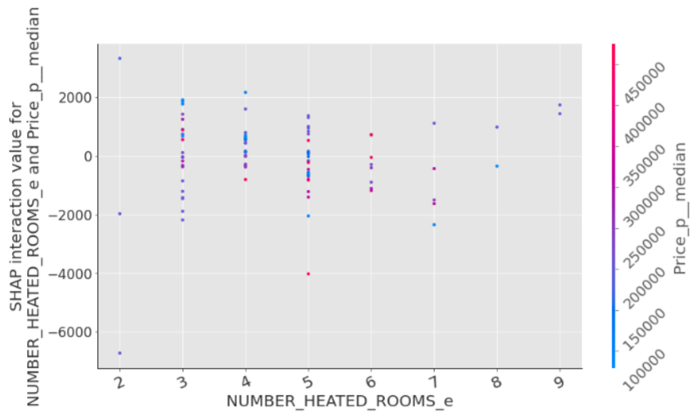


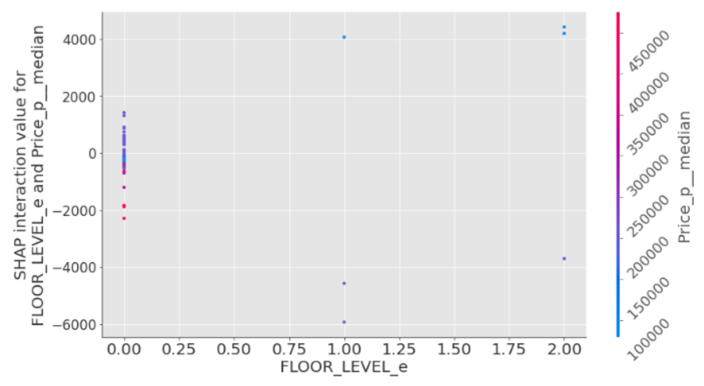
FIGURE: SHAP interaction values are a generalization of SHAP values to higher order interactions. Fast exact computation of pairwise interactions are implemented in the latest version of XGBoost with the pred_interactions flag. With this flag XGBoost returns a matrix for every prediction, where the main effects are on the diagonal and the interaction effects are off-diagonal. The main effects are similar to the SHAP values you would get for a linear model, and the interaction effects captures all the higher-order interactions are divide them up among the pairwise interaction terms. Note that the sum of the entire interaction matrix is the difference between the model's current output and expected output, and so the interaction effects on the off-diagonal are split in half (since there are two of each). When plotting interaction effects the SHAP package automatically multiplies the off-diagonal values by two to get the full interaction effect.

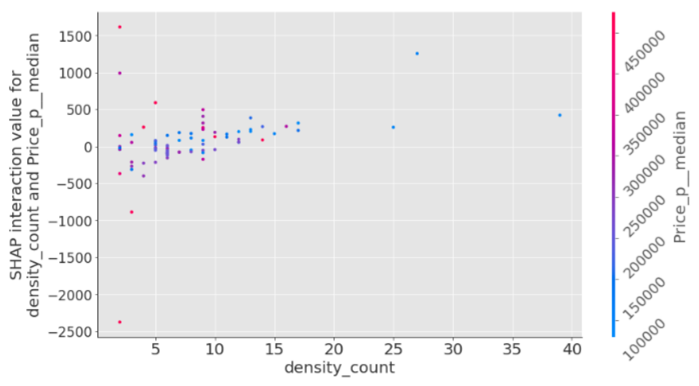
The figure above shows a summary plot of a SHAP interaction values. It plots a matrix of summary plots with the main effects on the diagonal and the interaction effects off the diagonal.

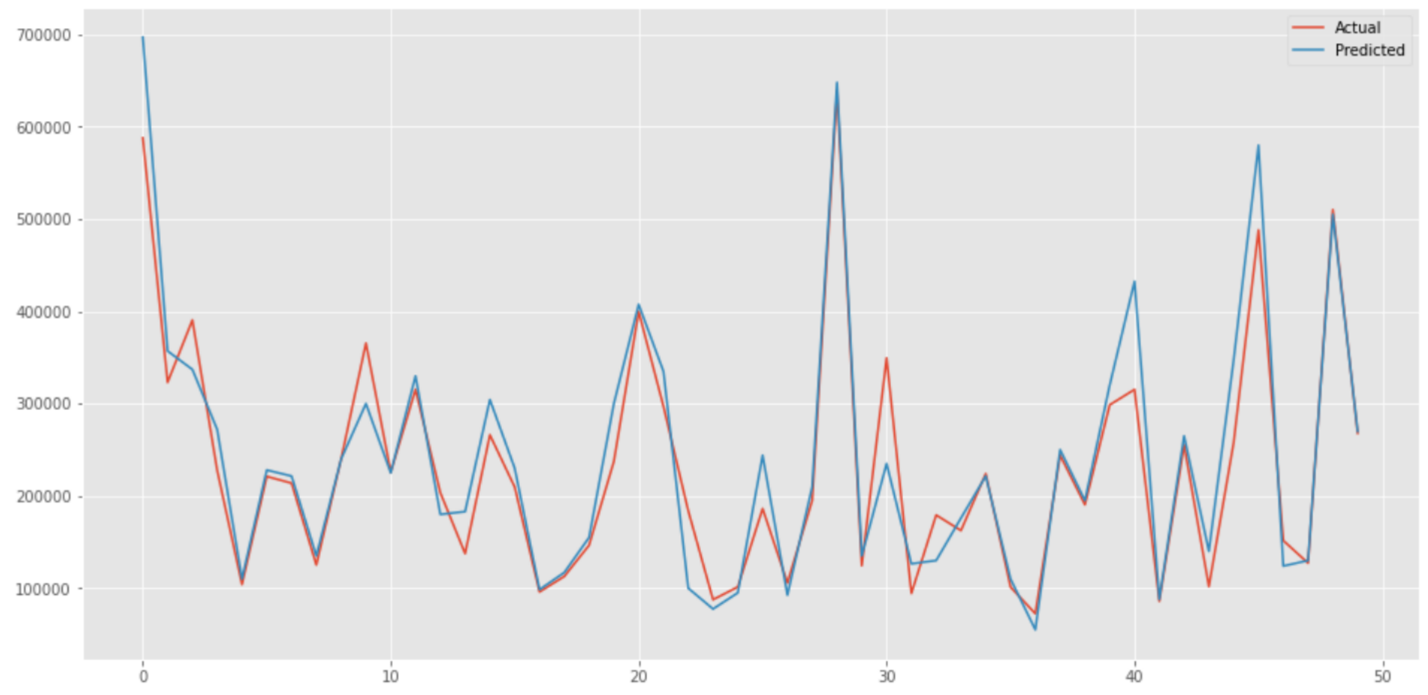
MODEL EXPLAINABILITY – Interesting Interactions Cont'd

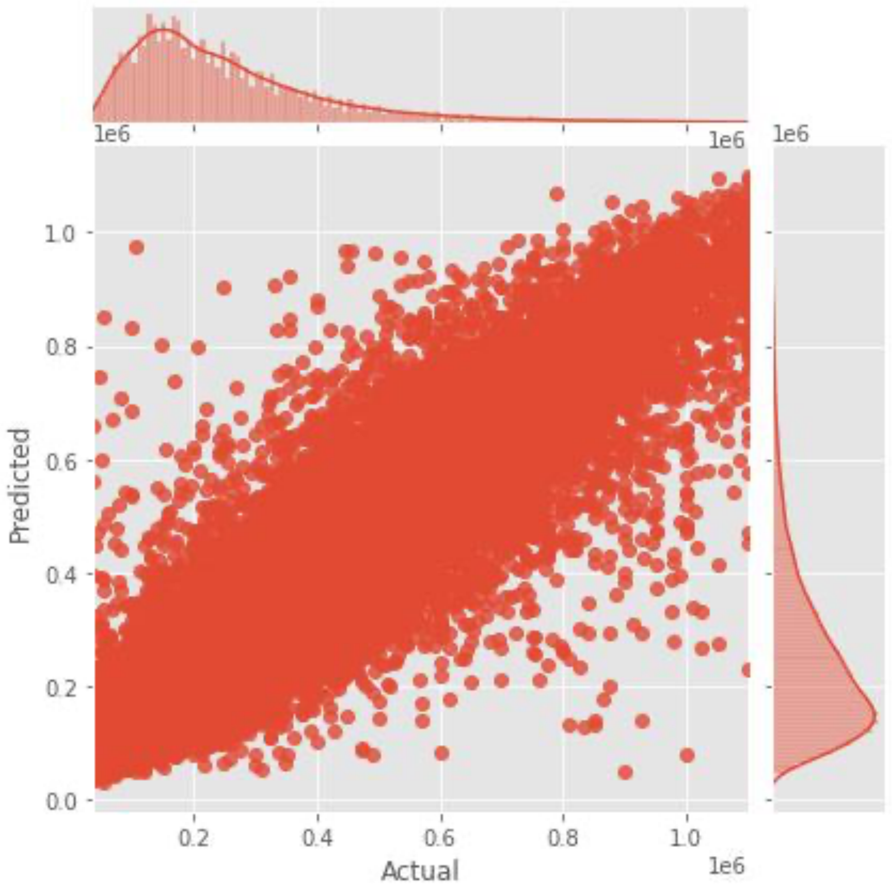











1e6
1e6
1e6
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Predicted
0.2
0.4
0.6
0.8
1.0
Actual