1. Project Title and Team

Residential property price estimation using property features and spatial-temporal data
by Syeda Jaisha (syedajaisha), Rahim Rasool (rahimrasool), and Carly Schippits (cschippits)

2. Executive Summary

House valuation is critical to the fair and accurate estimation of property tax bills. Traditional methods used by different state and local jurisdictions are outdated and may result in certain parts of the income spectrum disproportionately bearing the property tax burden.

We use housing price data from Ames, Iowa to find a machine learning model that can predict a house's value more fairly and accurately. The dataset contains information on 83 house and neighborhood-level demographic features and sale price for 1,460 houses in Ames, Iowa.

In this paper, we experiment with regression-based models and optimize the relevant hyperparameters to find a model that can not only accurately predict the sale price of a house but is also fair in overvaluing and undervaluing properties across the home price spectrum. For all our models, we use five-fold cross validation to measure the variance of the model across different portions of the training set. For linear regression models, we run a combination of simple, regularized, and polynomial expansion models. We also use support vector regressors and gradient boosting frameworks to build on our analysis of finding the best-performing model. Finally, we stack our best-performing models in an attempt to further reduce the bias.

Our results show that our most accurate model is the stacked model that takes the Ridge Regression, XGBoost Regressor, and LightGBM models and blends them over with a Random Forest Regressor. The stacked model yields a mean absolute error (MAE) of $16,364.

Based on variance, the best-performing model is the Elastic Net model, with a standard deviation of $1,331 for the MAE statistic across the cross-validation folds.

Based on fairness, the best-performing model is the linear model with three-degree polynomial features. This model overvalues low- and high-priced homes in the test set at approximately the same rate.

3. Background and Overview of Solution

3.1. Background

Every year, state and local governments collect property taxes from homeowners. Specific tax amounts depend on a home's assessed value. While home values can easily be gleaned if a home was recently sold, state and local governments must come up with a method for estimating the value of homes that have not recently sold. This estimated value is then used to calculate the amount of property taxes owed by the homeowner. Ensuring a good prediction is important, because underpredictions result in less revenue for state and local governments, while overpredictions may result in homeowners being overly burdened by property taxes.

Conventional models used by property tax officials rely on the average value of attributes for the homes that have sold in a region. Our aim is to calculate a more accurate and fair estimate of a home's value using factors that impact its market value such as the home's square footage, number of bedrooms, location, etc.

3.2. Fairness & Ethical Concerns

Property assessments tend to undervalue higher-priced homes and overvalue lower-priced homes. This is because higher-priced homes tend to have internal features (e.g. newly remodeled kitchens) that make buyers willing to pay more for these homes. However, these features—because they are internal—may not be observable to property assessors. Likewise, a lower-priced home may look identical to a higher-priced home on the outside, but on the inside, the lower-priced home may lack high-end details. Due to the immense number of home features in our dataset (there are a total of 77 features, many of which capture internal home characteristics), we believe that our analysis may better capture the qualitative differences between homes and therefore produce more accurate estimates for homes at all price points.

Overvaluing low-priced homes and undervaluing high-priced homes is a fairness concern because it leads to owners of low-priced homes (who tend to have lower incomes) paying more than their fair share of property taxes, while owners of high-priced homes pay less than their fair share in taxes. This phenomenon perpetuates a pre-existing imbalance in society because it causes the poor to become poorer and the rich to become richer. Due to this concern, we will consider various fairness metrics when evaluating our models.

Another ethical consideration for our study is lack of transparency. Using more sophisticated machine learning algorithms may help us arrive at models that are more accurate and fair; however, it may also reduce the interpretability of property valuation methods. This may lead to a reduction in transparency and accountability of assessors, because more complex models are more difficult to communicate to the average citizen.

4. Data

Our primary data source is the Ames, Iowa housing sales dataset from Kaggle. The dataset contains data for 1,460 homes, spans 25 neighborhoods, and covers the years 2006 through 2010. Each row of our dataset represents a home. The dataset includes information on sale price, date of sale, and a host of home attributes including square footage, number of bedrooms, and 74 other features for each home.

We merged our primary dataset with neighborhood-level data on school quality, crime, walkability, median income, and the unemployment rate. We sourced the school quality data from GreatSchools.org, which assigns a numerical rating of 1 to 10 for each school district. We sourced the crime data from www.Realtor.com, which rates neighborhood crime levels on a scale from 1 to 5. Next, we sourced the walkability data from WalkScore.com. For each neighborhood, we pulled both the walk score and the bike score, which respectively measure a neighborhood's walkability and bike-ability on a numerical scale from 0 to 100. Finally, we sourced median income and unemployment rate data from the 2019 five-year American Community Survey. Combining the neighborhood-level data with our primary dataset yields a total of 83 features, which we will use to predict property values.

4.1. Data Exploration

4.1.1. Predicted Variable: Sale Price

The predicted variable--the observed sale price--ranges from $34,900 to $755,000 in our training set. We plot the distribution of sale price in Figure 4.1 (see appendix).

4.1.2. Home Features

A number of housing features in our dataset appear to have a strong correlation with sale price. One of these features is house quality, an ordinal variable measured on a scale of 1 to 10. In Figure 4.2, we present a plot that depicts a positive correlation between sale price and house quality.

Another feature that exhibits a strong positive correlation with sale price is square footage. This is true of both above- and below-ground square footage. In Figure 4.3, we present a plot that depicts the relationship between sale price and above-ground square footage. In Figure 4.4, we plot the relationship between sale price and below-ground (basement) square footage.

Another feature that appears to be strongly correlated with sale price is the number of full bathrooms (i.e. the number of bathrooms containing a shower/tub). The positive correlation between sale price and number of full bathrooms is plotted in Figure 4.5. Interestingly, the relationship between sale price and number of bedrooms does not follow the same monotonically increasing pattern (see Figure 4.6). As the number of bedrooms increases from zero to four, the sale price increases, but the sale price steeply drops off after the number of bedrooms increases to five and above.

The degree of the irregularity of the property shape also seems to be positively related with the sale price (see Figure 4.7). House style also appears to be an important feature in explaining the variation in sale price, as is evident in Figure 4.8. Many other features--including but not limited to the quality of the heating system, condition of the basement, and quality of the material on the exterior of the home--exhibit a general positive trend with sale price. The plot for heating quality is given in Figure 4.9.

4.1.3. Neighborhood Features

In this section, we consider the relationship between sale price and neighborhood-level features.

Figure 4.10 depicts the relationship between neighborhood-level median income and sale price. Sale price appears to have a positive relationship with median income, but with a few neighborhoods that have relatively high median income but relatively low home prices.

Next, Figure 4.11 plots the relationship between sale price and the neighborhood-level unemployment rate. We can see that the sale price exhibits a weak negative correlation with the unemployment rate. This makes sense as one might expect neighborhoods with more unemployed individuals to also have lower home prices. A few neighborhoods constitute notable exceptions, exhibiting both a low unemployment rate and low housing prices.

The crime rating is measured on a scale of 1 to 5. Interestingly, we can see from Figure 4.12 that sale price appears to be weakly decreasing in crime level.

Next, Figure 4.13 plots the neighborhood-level walk score, which is measured on a scale of 0 to 100. The relationship between walk score and sale price is a bit ambiguous. At walk scores below 20, the sale price is low. As the walk score increases from 20 to 30, sale price increases substantially, but at higher walk scores above 30, there is no noticeable pattern in the relationship between sale price and walk score.

Unlike the walk score, the neighborhood-level bike score appears to exhibit a stronger correlation with sale price. Neighborhoods with higher bike scores also tend to have higher home prices (see Figure 4.14).

Finally, Figure 4.15 plots the relationship between sale price and school quality, which is measured on a scale of 1 to 10. There are only five school districts in Ames, so there is limited variation in school quality across neighborhoods. Nevertheless, the highest-rated school district contains the most expensive homes, but in the other school districts, there is no observable relationship between sale price and school quality.

4.2. Transforming the data and visualizing high-dimensional features

The first step in preparing our data for modeling is to one-hot encode the categorical variables. The resulting dataset has 364 features. Next, we randomly split the data into training and test sets, assigning 80% of the observations to the training set and 20% of the observations to the test set. Subsequently, we impute missing values using the median value in the training set. Only three features (lot frontage, masonry veneer area, and bike score) are missing values, and no one feature is missing more than 18% of its observations. Finally, we normalize the features in the training and test sets so that the features in the training set have a mean of zero and a standard deviation of one.

With 364 features, it was difficult to visualize the dataset and this prevented us from gaining intuition about the data points. In order to display the data points, we used the Sklearn's manifold class.

To reduce the high dimensional data into three components, we attempted the following:

1. Principal Component Analysis
2. Spectral Embedding
3. Multidimensional Scaling
4. t-distributed Stochastic Neighbor Embedding (t-SNE)

The results from the first three sets of dimensionality reduction algorithms were not intuitive when plotted, as these techniques did not result in clear groupings by the predicted variable, sale price. t-SNE offered the best visual results amongst all the algorithms. t-SNE converts affinities of data points to probabilities. This way, t-SNE allows the user to display the structure of high-dimensional data on a single plot, and it also crowds similar points together into manifolds/clusters.

For our analysis, we grouped the target variable into six buckets, and then transformed the feature space into three major components using t-SNE. We then plotted the transformed space on a 3D scatterplot (see Figure 4.16). The plot reveals that the expensive properties beyond $200k (marked with red, green, and purple) have a distinct cluster from the properties below $200k (marked with orange, brown, and blue).

5. Machine Learning and Details of Solution

Our predicted variable (sale price) is continuous. Consequently, we will use a suite of regression-based models to predict an exact assessment value for each home. This value can then

be used by property taxing jurisdictions (including state and local governments) to determine the annual property tax bill for a specific home.

Models we will consider include linear models, tree-based models, and support vector regressors. We will assess the performance of these models on the dimensions of accuracy, variance, and fairness. Finally, we will combine the best-performing models in a stacked model in an attempt to further reduce bias.

6. Evaluation and Results

6.1. Evaluation Metrics

6.1.1. Accuracy Evaluation Metrics

We will primarily rely on the mean absolute error (MAE) of the test set when evaluating the accuracy of our models. We chose this measure because its units are in dollars, which is the relevant unit for home appraisals and property tax bills. We chose not to rely primarily on mean squared error (MSE) to assess accuracy, because its units (squared dollars) are far less intuitive, and we saw no reason to further penalize homes with large errors when these errors will already skew the MAE, which is itself a measure of average error. In addition, we chose not to use R-squared as our primary metric, because it is unit-less and because not all of our models use the same number of features (e.g. the basic linear model has fewer features than the linear model with polynomial features) and R-squared always increases in the number of features—regardless of the predictive power of those features. As a consequence, we will primarily rely on MAE to assess the accuracy of our models.

6.1.2. Variance Evaluation Metrics

For each model, we will also report the standard deviation of the MAE statistic across the cross-validation folds. The standard deviation will reveal the sensitivity of a model's MAE to the composition of the training set.

6.1.3. Fairness Evaluation Metrics

For each model, we will also consider performance on fairness to check if low-priced homes are systematically over-assessed and if high-priced homes are systematically under-assessed.

Our target variable is continuous so we cannot use the typical fairness metrics (e.g. precision, recall, etc.) that are used to evaluate classification models. Instead, we will examine the predictive error separately for low- and high-priced homes.

To determine which homes are low-priced and which ones are high-priced, we used k-means clustering to divide the homes into three groups based on the similarity of their sale prices. The resulting groups represent low-price, medium-priced, and high-priced homes, respectively. The maximum sale price among the low-priced homes is $174,000, while the minimum price of the high-priced group is $293,077. The clusters are plotted in Figure 6.1.

We define predictive error for home *i* as:

$$Error_i = SalePrice_i - \widehat{SalePrice}_i$$

where $SalePrice_i$ is the actual sale price for home i and $\widehat{SalePrice}_i$ is the predicted sale price for home i. We will consider the percentage of homes in each pricing group that have a positive (or negative) predictive error to check if low-priced homes are more likely to be overvalued and if high-priced homes are more likely to be undervalued.

6.2. Model Results
6.2.1. Linear Models
6.2.1.1. Basic Linear Model

Our first linear model is a basic OLS linear regression model using all 364 features in our dataset.

The model has a test MAE of $24,133. The standard deviation of the MAE statistic across the five cross-validation folds is $63,557,563,609,965. Clearly, the MAE of the basic linear model is highly sensitive to the data points contained in the training set. In terms of fairness, this model undervalues both low-priced and high-priced homes at approximately the same rate (see Figure 6.2). Therefore, this model seems to perform well according to our fairness metrics.

In Figure 6.3, we plot the predicted sale prices for the test set against the actual observed sale prices.

The five most important features in the basic linear model are proximity to a positive off-site feature (such as a park), the existence of a pool of excellent quality, the existence of a garage in excellent condition, the existence of a membrane roof, and the existence of a garage of excellent quality. Note that garage quality and garage condition are recorded as two separate features.

6.2.1.2. Regularized Linear Model

For our second linear model, we applied Lasso, Ridge, and Elastic Net regression to our dataset. To determine the best alpha hyperparameter, we performed a grid search and used five-fold cross validation to minimize MAE. We first performed a coarse-grained grid search on alpha values ranging from 0.1 to 10,000. Once we narrowed down the approximate location of the best alpha hyperparameter, we performed a finer-grained grid search.

For the Lasso regression model, the optimal alpha hyperparameter is 150. This model drops 267 features and yields a test MAE of $20,449. The standard deviation of the MAE statistic across cross-validation folds is $1,426. In terms of fairness, this model undervalues both high- and low-priced homes, but high-priced homes are undervalued at a much higher rate (see Figure 6.4). This is concerning from a fairness perspective, because it suggests that this model's property assessments may lead to the under-taxation of owners of high-priced homes relative to owners of low-priced homes.

For the Ridge regression model, the optimal alpha hyperparameter is 31. This model drops five features and yields a test MAE of $19,474. The standard deviation of the MAE statistic across cross-validation folds is $1,353. In terms of fairness, like the Lasso model, the Ridge model also undervalues high-priced homes relative to low-priced homes (see Figure 6.5).

Finally, for the Elastic Net regression model, the optimal alpha hyperparameter is 0.1. This model drops five features and yields a test MAE of $19,790. The standard deviation of the MAE

statistic across cross-validation folds is $1,331. In terms of fairness, like the Lasso and Ridge models, the Elastic Net model also undervalues high-priced homes relative to low-priced homes (see Figure 6.6), which is a concern from a fairness perspective.

Of these three models, the Ridge regression model performs the best in terms of mean absolute error—performing even better than the basic linear model. In Figure 6.7, we plot the Ridge model's predicted sale prices for the test set against the actual observed sale prices.

The Ridge regression model drops five features: the existence of severe damages, adjacency to a North-South railroad, proximity to a North-South railroad, the existence of a tennis court, and the existence of a clay or tile roof. None of the homes in the training set has these features, so they are natural candidates to drop.

The five most important predictors for the Ridge model are the existence of a kitchen of excellent quality, the overall house quality, the existence of a basement of excellent quality, above-ground square footage, and proximity to a positive off-site feature.

6.2.1.3. Linear Model with Polynomial Features

For our third linear model, we applied an OLS linear regression model with polynomial features. We performed a grid search and used five-fold cross validation to determine whether the optimal number of degrees for the polynomial features was two or three.

When we first attempted to run this model, it ran unsuccessfully due to inadequate memory. As a result, we decided to exclude the features that were dropped by the Lasso regression model (i.e. the features that the Lasso model determined to be the worst predictors of a home's sale price). We chose to drop features based on the Lasso model, because the Elastic Net and Ridge models only dropped five features each. The Lasso model, on the other hand, dropped a total of 267 features. We opted to drop these same features in order to ensure that our truncated dataset was small enough for Scikit-Learn's PolynomialFeatures() function to handle.

The model with two-degree polynomial features has a test MAE of $25,706. The standard deviation of the MAE statistic across cross-validation folds is $3,409. In terms of fairness, this model undervalues both high- and low-priced homes, but high-priced homes are undervalued at a much higher rate (see Figure 6.8).

The model with three-degree polynomial features has a test MAE of $29,169. The standard deviation of the MAE statistic across cross-validation folds is $3,373. This model performs quite well on fairness, undervaluing both low- and high-priced homes at approximately the same rate (see Figure 6.9).

Because the model with two-degree polynomial features has a lower mean absolute error, we plot this model's predicted sale prices for the test set against the actual observed sale prices in Figure 6.10.

The five most important predictors for the model with two-degree polynomial features are the interaction of wood deck square footage and the fact that the home was sold in the month of February, above-ground square footage, the interaction of open porch square footage and the fact that the house was sold in 2009, the interaction of wood deck square footage and the existence of

a heating system of excellent quality, and the interaction of the existence of a garage built in 2007 and the fact that the home was sold in the month of February.

6.2.1.4. Regularized Linear Model with Polynomial Features

For our fourth and final linear model, we regularized the best-performing linear model with polynomial features (i.e. the model with two degrees). We applied Lasso, Ridge, and Elastic Net regression on the truncated dataset and used five-fold cross validation and grid search based on mean absolute error to determine the best alpha hyperparameter ranging from 0.1 to 10,000. Once we narrowed down the approximate location of the best alpha hyperparameter for each model, we performed a finer-grained grid search.

For the Lasso regression model with two-degree polynomial features, the optimal alpha hyperparameter is 942. This model yields a test MAE of $18,375. The standard deviation of the MAE statistic across cross-validation folds is $1,518. In terms of fairness, this model tends to overvalue low-priced homes and undervalue high-priced homes (see Figure 6.11). This is concerning from a fairness perspective, because it would likely result in over-taxation of owners of low-priced homes and under-taxation of owners of high-priced homes.

For the Ridge regression model with two-degree polynomial features, the optimal alpha hyperparameter is 478. This model yields a test MAE of $17,659. The standard deviation of the MAE statistic across cross-validation folds is $1,754. In terms of fairness, this model also tends to overvalue low-priced homes and undervalue high-priced homes (see Figure 6.12).

Finally, for the Elastic Net regression model with two-degree polynomial features, the optimal alpha hyperparameter is 1. This model yields a test MAE of $17,788. The standard deviation of the MAE statistic across cross-validation folds is $1,751. In terms of fairness, this model also tends to overvalue low-priced homes and undervalue high-priced homes (see Figure 6.13).

Of these three models, the Ridge regression model performs the best in terms of mean absolute error—performing even better than all other linear models considered in this paper. In Figure 6.14, we plot the Ridge model's predicted sale prices for the test set against the actual observed sale prices.

The five most important predictors for the Ridge model are above-ground square footage, overall house quality, basement square footage, the interaction between above-ground square footage and the existence of a garage of average quality, and the number of cars the garage can accommodate.

6.2.2. Support Vector Regressor Model

For our prediction model, amongst the set of models that are good candidates for this dataset included the Support Vector Regressor (SVR). One reason for choosing this algorithm is because SVRs are especially useful in cases where the ratio of the number of features to the number of observations is low, as in our case. Another reason for selecting SVR for modeling is the versatility it offers in using various kernel functions for the decision function.

For our analysis we tried a set of 'polynomial', 'sigmoid' ,'linear' and 'rbf' kernels. Additionally, we tuned the epsilon and regularization parameter, C, in increasing powers of ten. We used five-fold cross validation to evaluate the hyperparameter combinations.

	params	mean_test_score	std_test_score	rank_test_score
	{'C': 1, 'epsilon': 0.1, 'kernel': 'poly'}	-55318.495288	4449.337244	1
	{'C': 1, 'epsilon': 0.01, 'kernel': 'poly'}	-55318.495288	4449.337244	1
	{'C': 1, 'epsilon': 1, 'kernel': 'poly'}	-55318.495288	4449.337244	1

Surprisingly, the plot above shows that SVR performs poorly on our dataset, consistently giving a training MAE of around $56K. Figure 6.15 shows that SVR also performs abysmally on fairness by drastically overvaluing low-priced homes and undervaluing high-priced homes

6.2.3. Tree-Based Models

6.2.3.1. RandomForest

We selected a Random Forest model due to the randomness it offers in each set of trees it grows. This way, we can evaluate how our model behaves on varying distributions of the data. We trained multiple Random Forest models on various cross-validation folds to tune the ccp_alpha parameter, the number of estimators and max_features (i.e. the method used to choose the best split).

The Random Forest model with max_features set to ‘sqrt’ provided the most accurate results, giving an MAE of $17,715 on the test set and a standard deviation of $1,502 across the cross-validation folds. This model uses the square root of the total number of features to select the maximum number of features for the best split. Figure 6.16 shows the optimal parameter combination that yields the lowest MAE and the lowest standard deviation. Figure 6.17 shows how the random forest model severely under-values high-priced homes and over-values low-priced homes, thus performing poorly in terms of fairness.

	params	mean_test_score	std_test_score	rank_test_score
14	{'ccp_alpha': 0.01, 'max_features': 'sqrt', 'n...	-17421.763213	1502.939487	1
5	{'ccp_alpha': 0.001, 'max_features': 'sqrt', '...	-17421.763240	1502.939490	2
23	{'ccp_alpha': 0.1, 'max_features': 'sqrt', 'n_...	-17421.763949	1502.942714	3

6.2.3.2. XGBoost

Amongst the tree algorithms that we wanted to attempt, XGBoost seemed to be an excellent candidate. XGBoost is a more sophisticated extension of the gradient boosting framework and has proven to achieve state-of-the-art results on many machine learning challenges. Another reason for its popularity is that it has been optimized to be highly efficient, flexible and portable.

We followed the same method as we did with SVR and regressed our data on a set of XGBoost parameters, primarily trying the ‘gbtree’, ‘gblinear’ and ‘dart’ booster algorithms. Along with this, we tuned the gamma parameter (which determines minimum loss reduction to make a partition on a leaf node) and the maximum tree-depth parameter.

The graph in Figure 6.18 reveals some sweet spots which not only give a low MAE score, but also a low standard deviation score on the set of folds used for cross validation. The XGBoost

model with a dart boost, gamma of 1 and a max depth of 6 yielded the lowest MAE at $17,282, with a standard deviation of $1,405. For this model, the MAE on the held-out test set is $17,772. As shown in Figure 6.19, XGBoost gives good performance in terms of fairness, undervaluing high-priced homes at just a slightly higher rate than low-priced homes.

	params	mean_test_score	std_test_score	rank_test_score
	{'booster': 'dart', 'gamma': 1, 'max_depth': 6}	-17282.112127	1404.543410	1
	{'booster': 'gbtree', 'gamma': 0.01, 'max_dept...	-17282.112127	1404.543410	1
	{'booster': 'gbtree', 'gamma': 0.1, 'max_depth...	-17282.112127	1404.543410	1

6.2.3.3. LightGBM

LightGBM takes less memory to run and is able to deal with large amounts of data. It is a gradient boosting framework that makes use of tree-based learning algorithms that are very powerful when it comes to computation. LightGBM algorithm grows vertically meaning that it grows leaf-wise, whereas other algorithms grow level-wise. LightGBM chooses the leaf with the largest loss to grow, and it is typically able to decrease the loss more than a level-wise algorithm can when growing the same leaf.

Figure 6.20 shows the optimal parameter set that yields the lowest MAE and standard deviation on the cross-validation folds. The best of these models gives an MAE of $16,773 on the test set and a standard deviation of $1,788 across the cross-validation folds. This best model uses a GBDT (Gradient Boosting Decision Tree), a maximum of 31 leaves and a value of 0.01 for reg_alpha. As shown in Figure 6.21, LightGBMgives performs well on fairness, undervaluing high-priced homes at just a slightly higher rate compared to low-priced homes.

	params	mean_test_score	std_test_score	rank_test_score
	{'boosting_type': 'gbdt', 'num_leaves': 31, 'r...	-16212.575673	1788.358670	1
	{'boosting_type': 'gbdt', 'num_leaves': 31, 'r...	-16217.093093	1773.566965	2
	{'boosting_type': 'gbdt', 'num_leaves': 15, 'r...	-16255.099725	1760.728873	3

6.2.4. Stacking

We use Stacking to blend all of our best-performing models discussed above. Stacking enables us to combine estimators in an attempt to reduce bias. More precisely, the predictions of each individual estimator are stacked together and used as input to a final estimator to compute a final prediction.

We stack our models in the following format:

1st layer:
- XGBoost Regressor
- LightGBM Regressor
- Ridge Regression with two-degree polynomial features

2nd (blending) layer:

· Random Forest Regressor

Our stacking model is visualized in Figure 6.22. Our modeling pipeline was able to reduce the test MAE to $16,364. The standard deviation across the cross-validation folds is $1,718. Figure 6.25 plots the predicted sale prices for the test set against the actual sale prices. In terms of fairness, this model undervalues high-priced homes at a somewhat higher rate than low-priced homes (see Figure 6.23). In order to get a better estimate of the impact of the model, we compared how much it over- or under-assesses each property in our test set. These properties are over- or under-assessed by only about 6 percent of their value on average.

Table 6.1 summarizes the results of all the models considered in this paper.

6.3. Discussion and Policy Implications

Our study delivers many useful insights for policymakers and their efforts to reduce property tax regressivity and its role in perpetuating the existing inequities in society.

First and foremost, our results suggest that property tax assessors should adopt more sophisticated models of property valuation in order to achieve better performance on accuracy, variance, and fairness. These models, at the very least, should include the house features identified by our study as some of the most important features such as the size of the basement, the size of the garage, and polynomial features including the interaction terms (which give an idea of the collective importance of the features). Furthermore, subjective measures of a house's value such as overall house quality which are prone to underlying human biases, can be one of the inputs to a well-performing model, but should not be the only input.

Any model for property valuation should also account for the variation in housing prices that is caused by the demographic characteristics of a neighborhood. Inclusion of these neighborhood-level variables helped our models fare better in terms of accuracy compared to conventional methods.

Since our study evaluates each model on the dimensions of accuracy, variance, and fairness, it gives policymakers and other stakeholders the freedom to choose the model that prioritizes their objective(s). Based on our results, if a taxing jurisdiction wants to prioritize accuracy over variance and fairness, it should choose the stacked model. If a jurisdiction wants to prioritize the minimization of variance in order to ensure that its property assessments are not overly sensitive to the data used, it should choose the Elastic Net linear model with no polynomial features. Lastly, if a jurisdiction wants to prioritize fairness over accuracy and variance, it should choose our linear three-degree polynomial model.

Of course, because we only used data from Ames, Iowa, our findings about the optimal model(s) to choose are only applicable to the city of Ames. However, jurisdictions outside of Ames could repeat our analysis using their own datasets in order to identify the best-performing model(s).

A. Appendix

Figure 4.1.

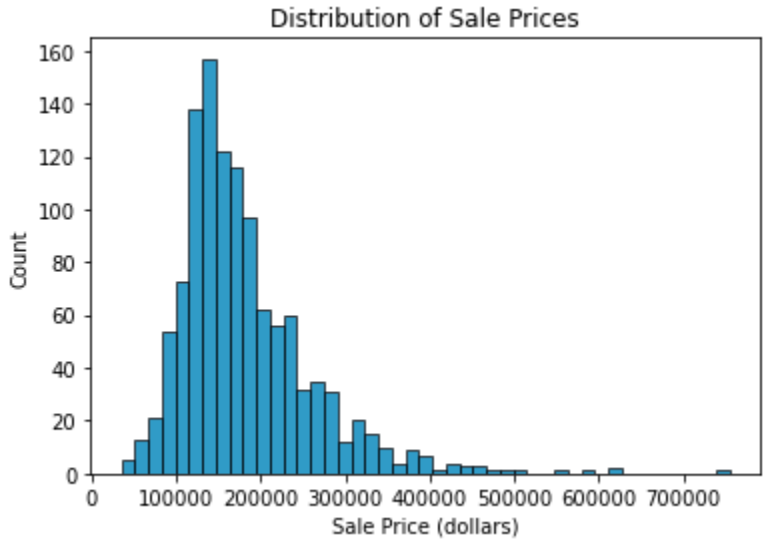

Figure 4.2.

Figure 4.3.

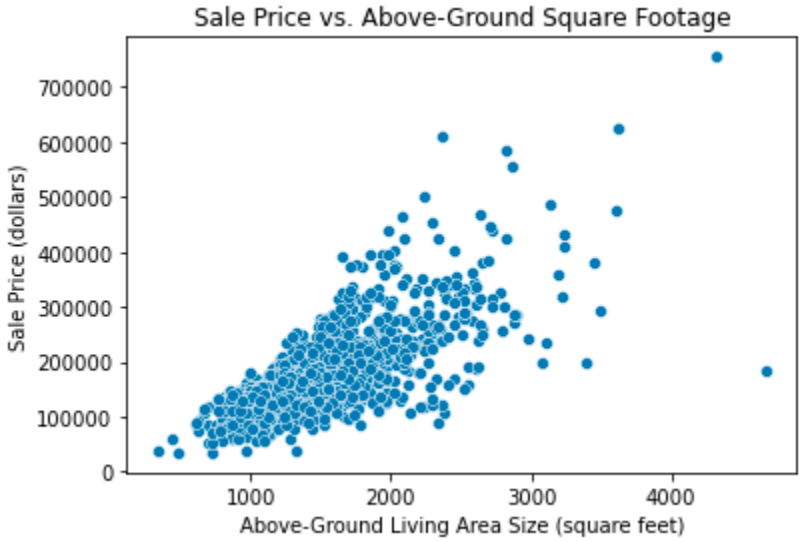

Figure 4.4.

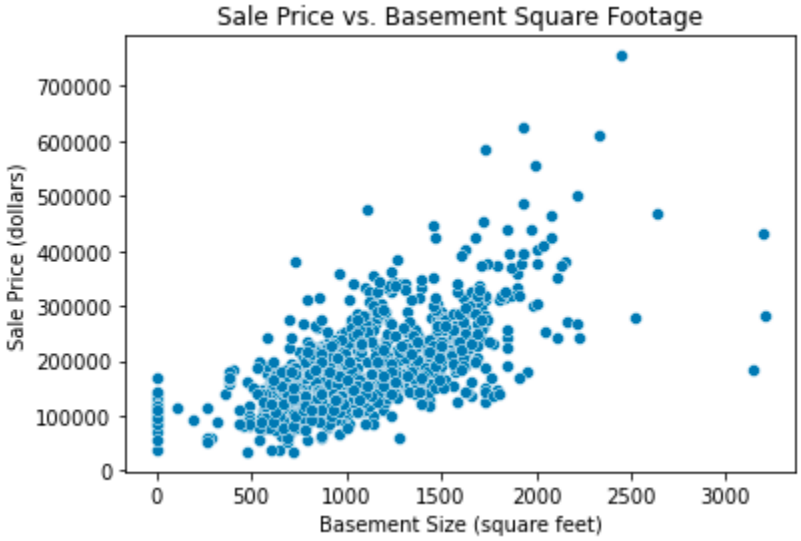

Figure 4.5.

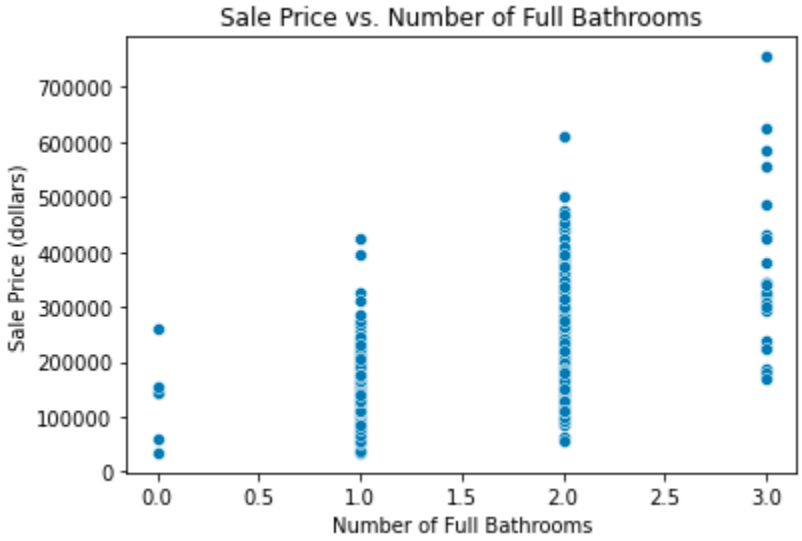

Figure 4.6.

Figure 4.7.

Figure 4.8.

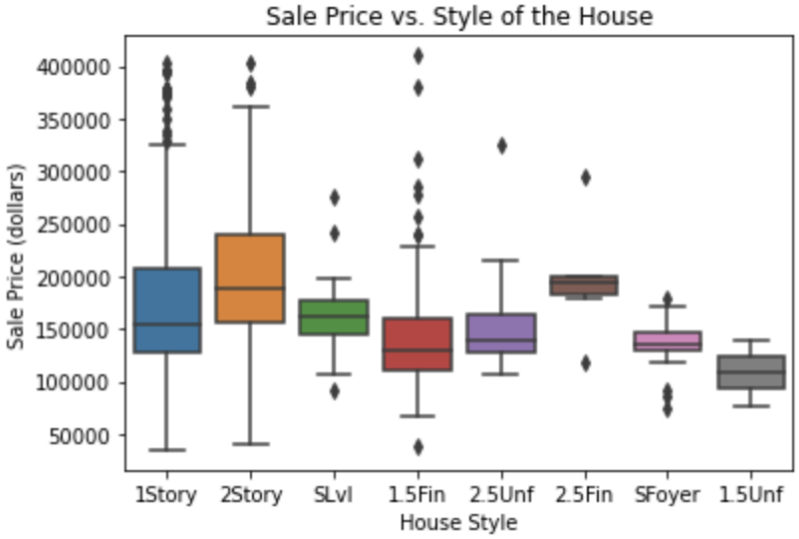

Figure 4.9.

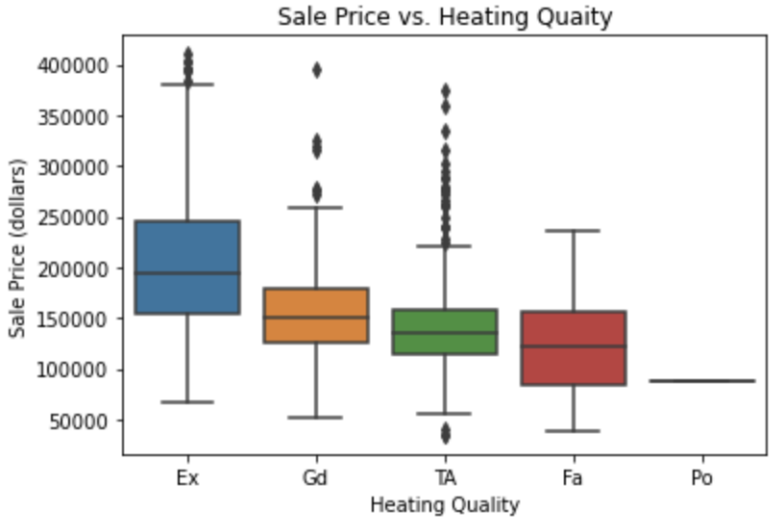

Figure 4.10.

Figure 4.11.

Figure 4.12.

Figure 4.13.

Figure 4.14.

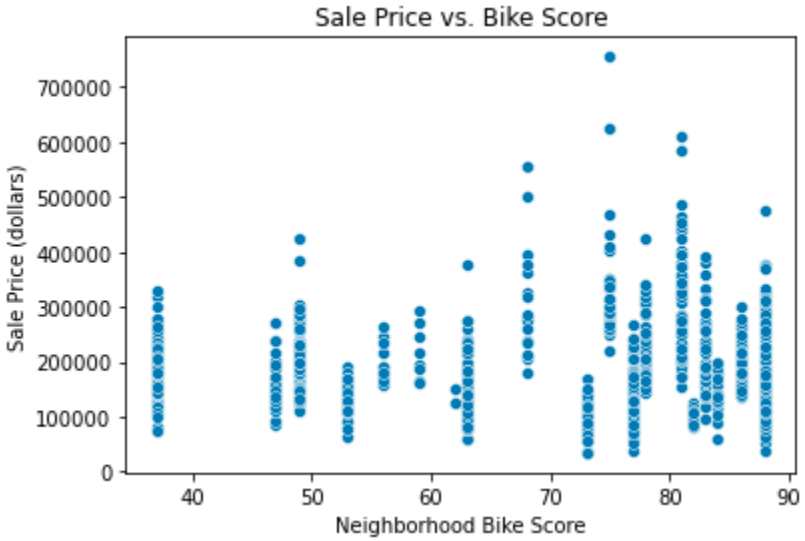

Figure 4.15.

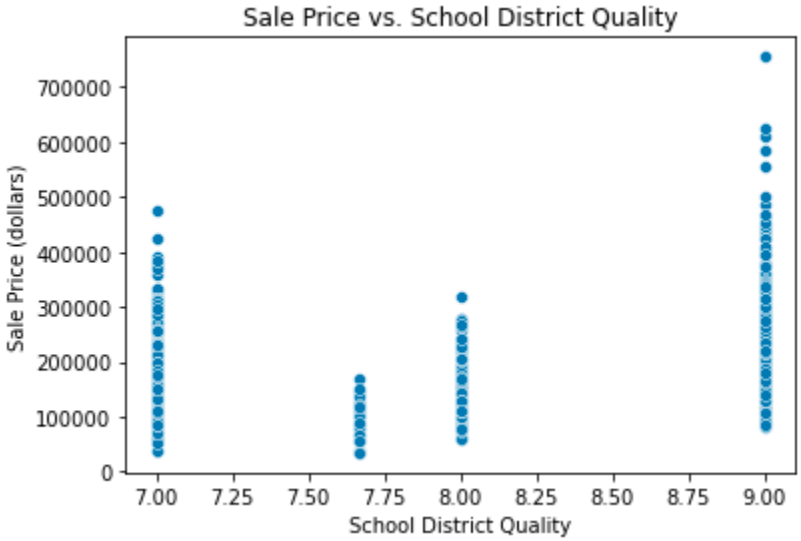

Figure 4.16. Visualization of the Feature Space Using Dimensionality Reduction.

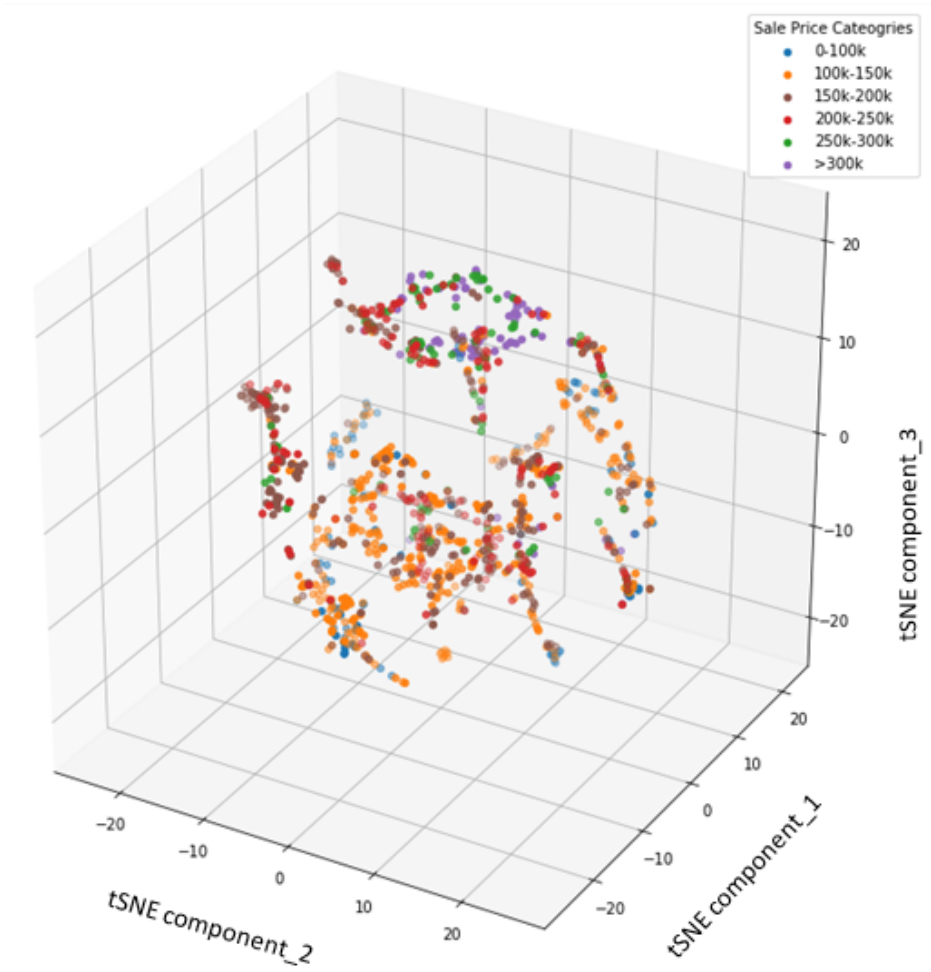

Figure 6.1.

Figure 6.2. Fairness Performance for the Basic Linear Model.

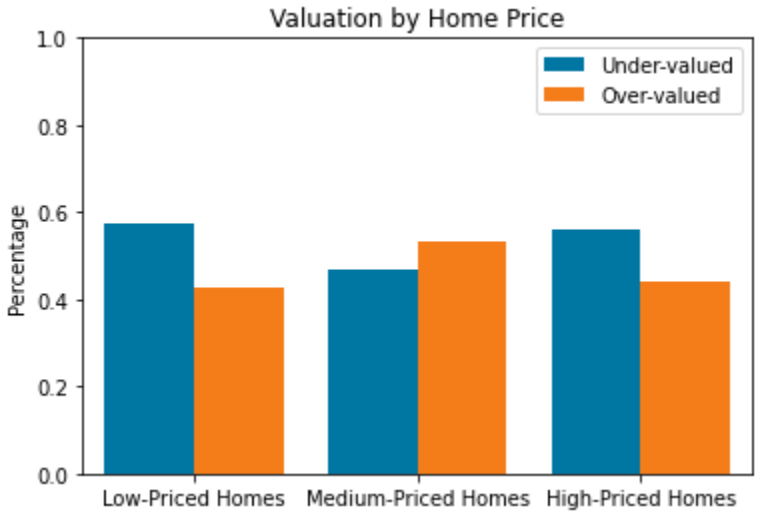

Figure 6.3. Predicted vs. Actual Sale Prices for the Basic Linear Model.

Figure 6.4. Fairness Performance for the Lasso Model.

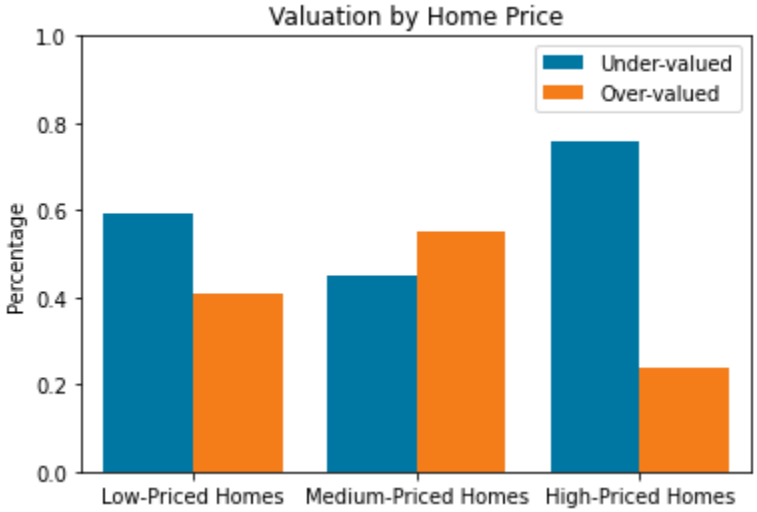

Figure 6.5. Fairness Performance for the Ridge Model.

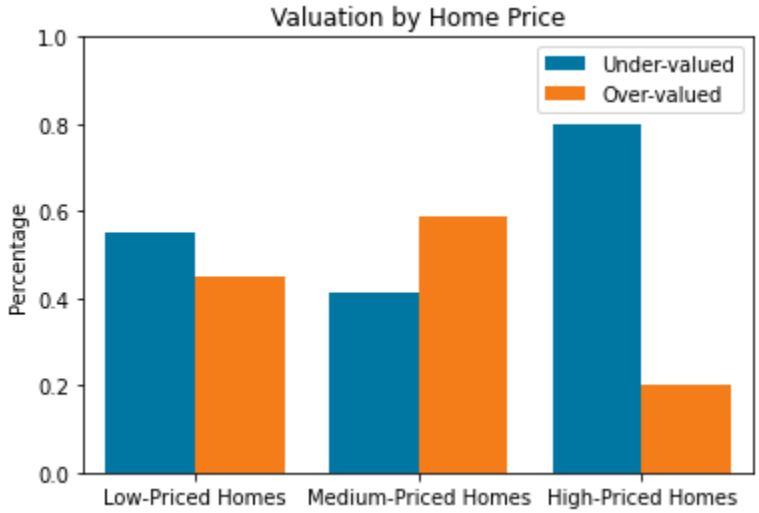

Figure 6.6. Fairness Performance for the Elastic Net Model.

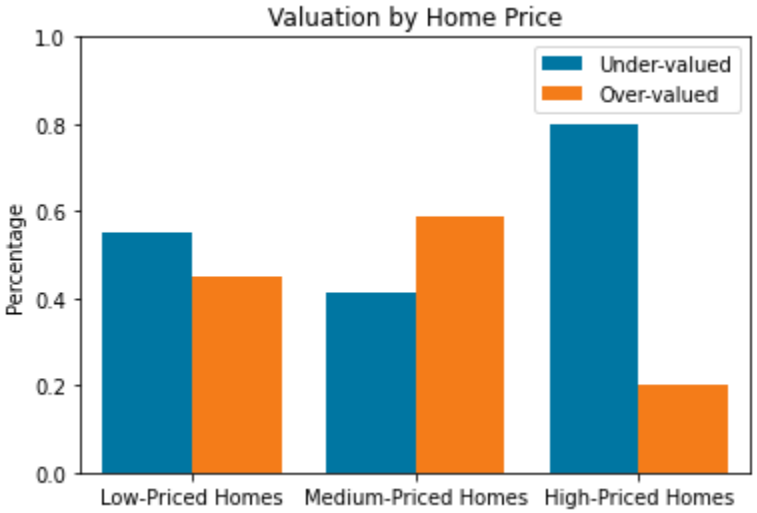

Figure 6.7. Predicted vs. Actual Sale Prices for the Ridge Model.

Figure 6.8. Fairness Performance for the Linear Model with Two-Degree Polynomial Features.

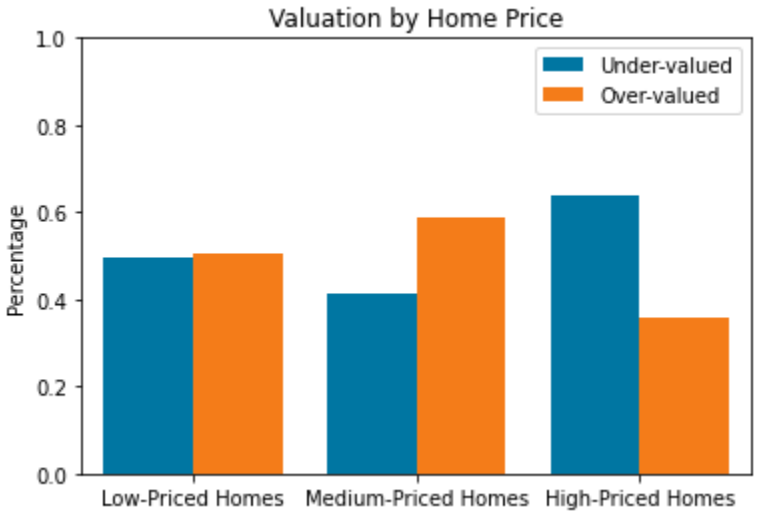

Figure 6.9. Fairness Performance for the Linear Model with Three-Degree Polynomial Features.

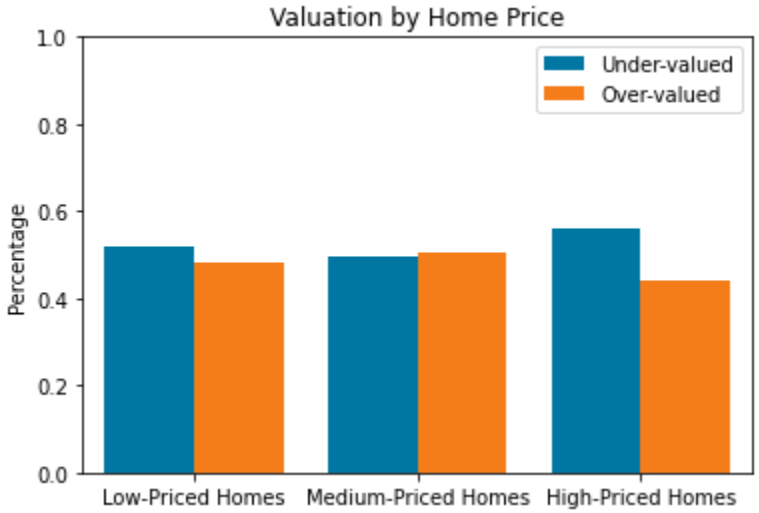

Figure 6.10. Predicted vs. Actual Sale Prices for the Linear Model with Two-Degree Polynomial Features.

Figure 6.11. Fairness Performance for the Lasso Model with Two-Degree Polynomial Features.

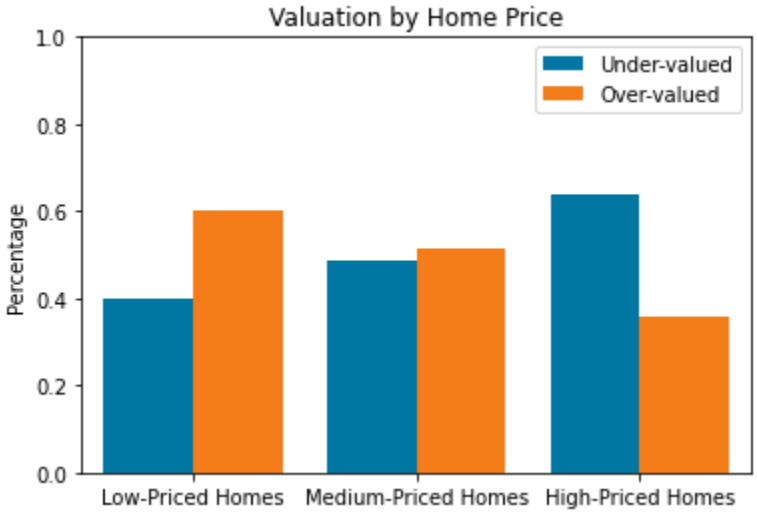

Figure 6.12. Fairness Performance for the Ridge Model with Two-Degree Polynomial Features.

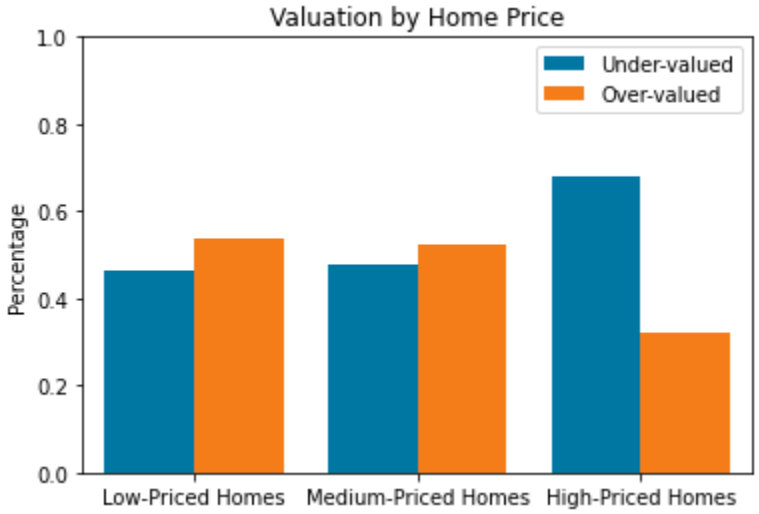

Figure 6.13. Fairness Performance for the Elastic Net Model with Two-Degree Polynomial Features.

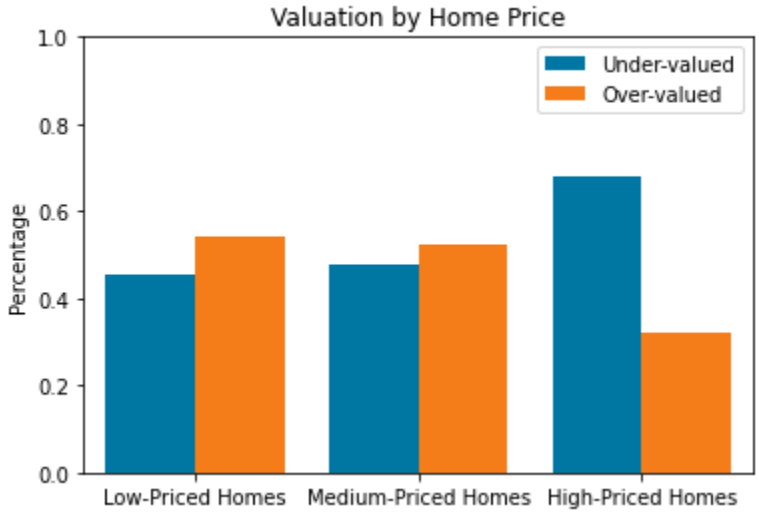

Figure 6.14. Predicted vs. Actual Sale Prices for the Ridge Model with Two-Degree Polynomial Features.

Figure 6.15. Fairness Performance for the Support Vector Regressor.

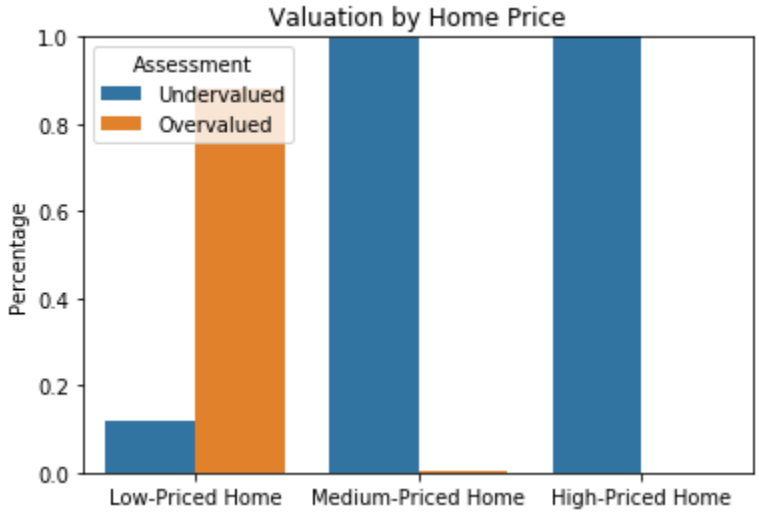

Figure 6.16. MAE and Standard Deviation across Various Parameters for the Random Forest Regressor.

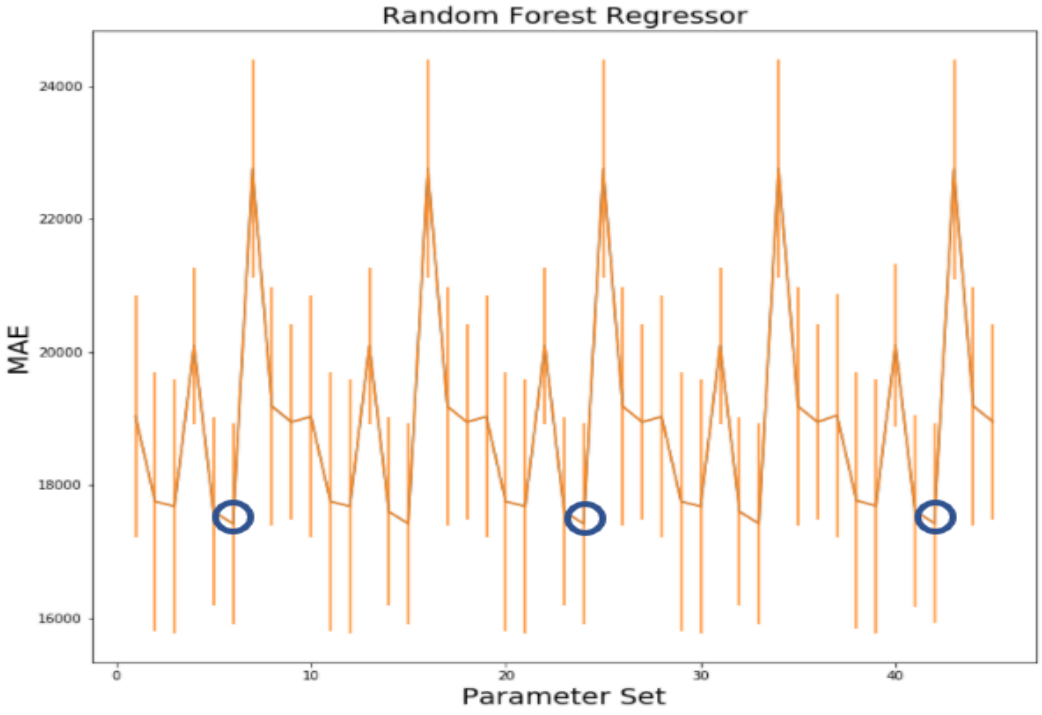

Figure 6.17. Fairness Performance for the Random Forest Regressor.

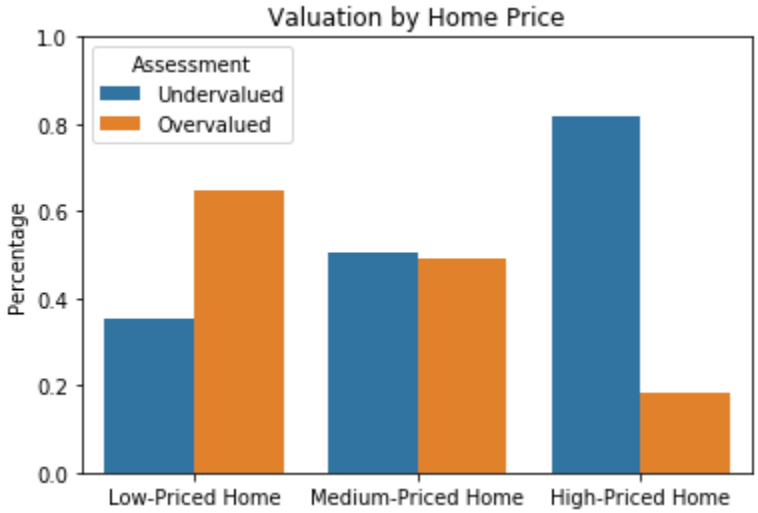

Figure 6.18. MAE and Standard Deviation across Various Parameters for the XGBoost Regressor.

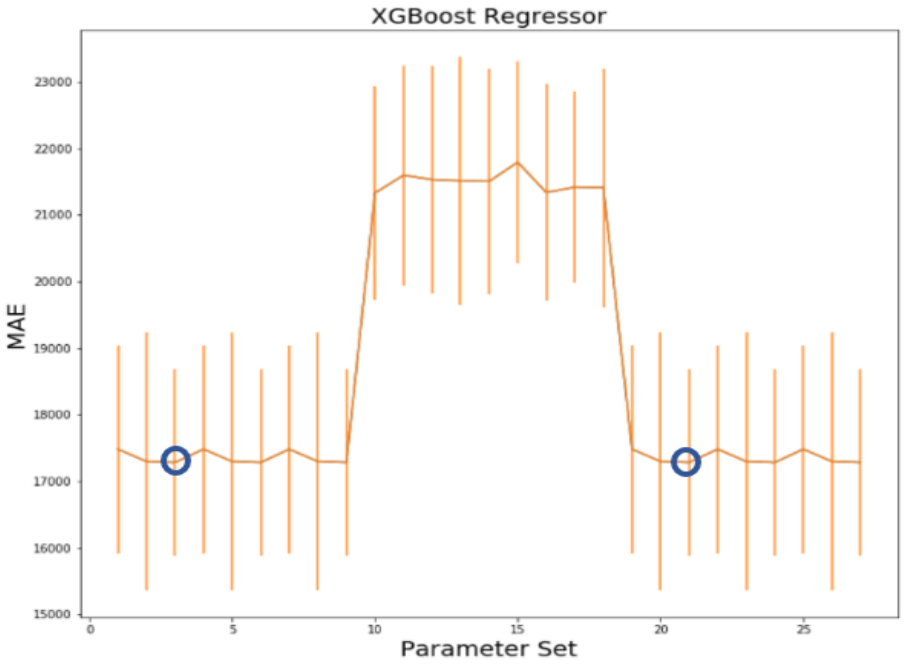

Figure 6.19. Fairness Performance for XGBoost.

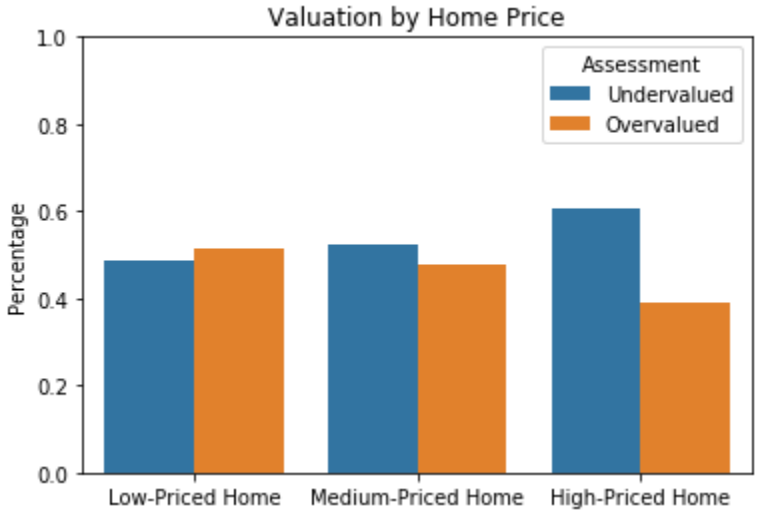

Figure 6.20. MAE and Standard Deviation across Various Parameters for the LightGBM Regressor.

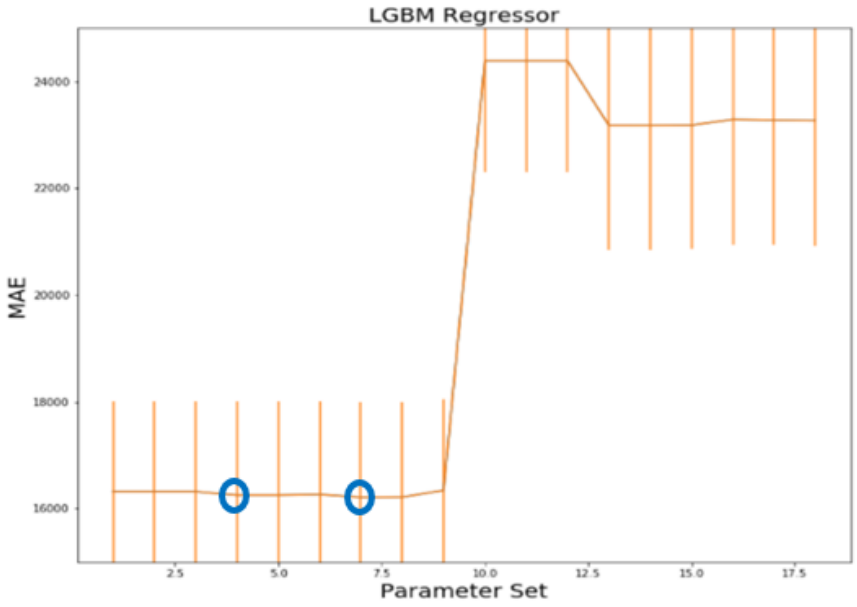

Figure 6.21. Fairness Performance for Light GBM.

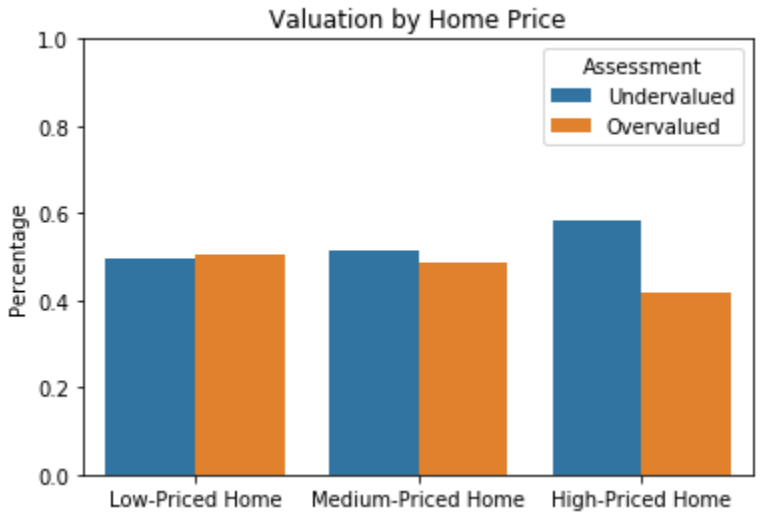

Figure 6.22. Stacked Model Pipeline.

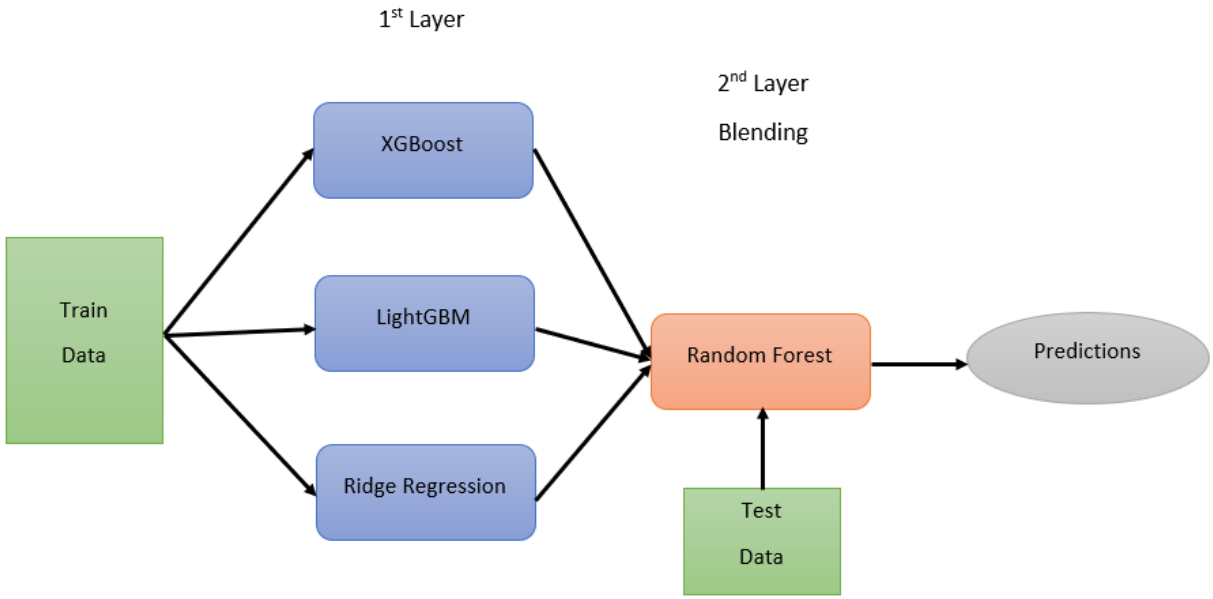

Figure 6.23. Fairness Performance for the Stacked Model.

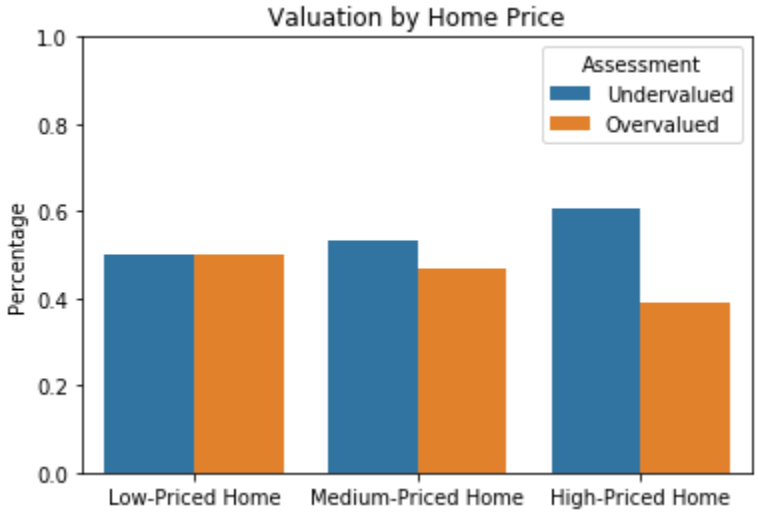

Figure 6.24. More Detailed Fairness Performance for the Stacked Model.

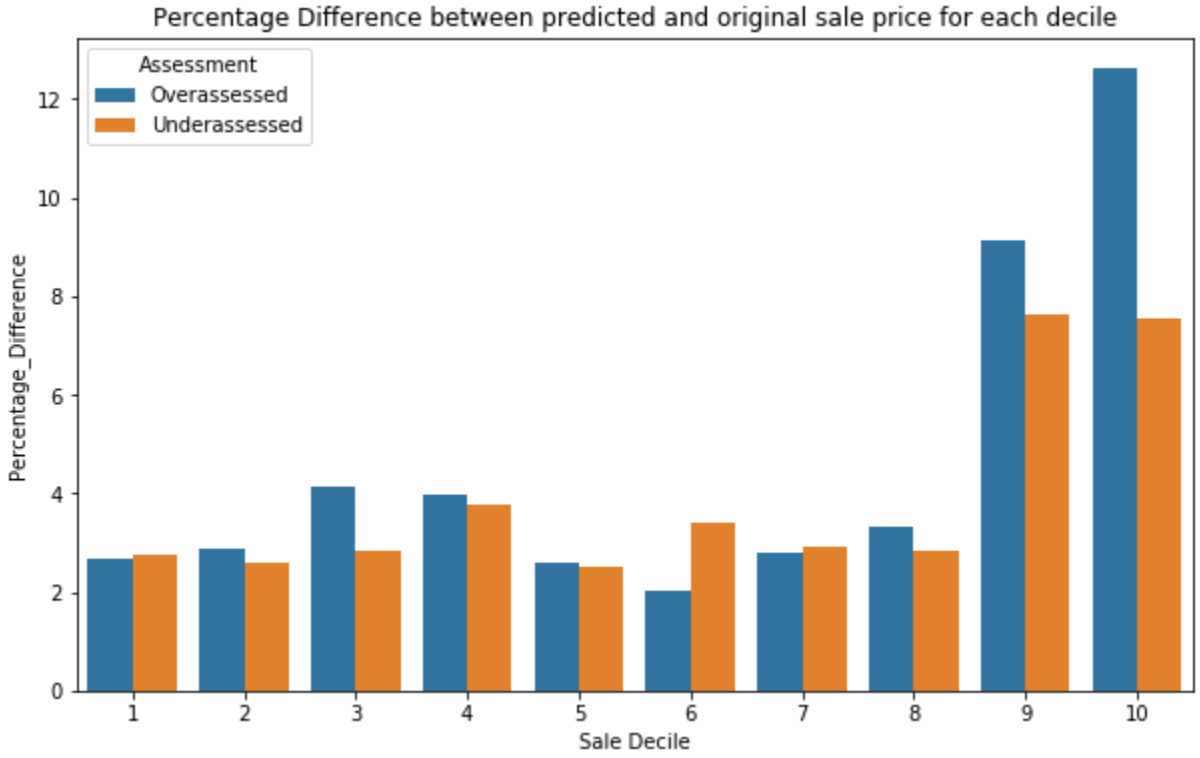

Figure 6.25. Predicted vs. Actual Sale Prices for the Stacked Model.

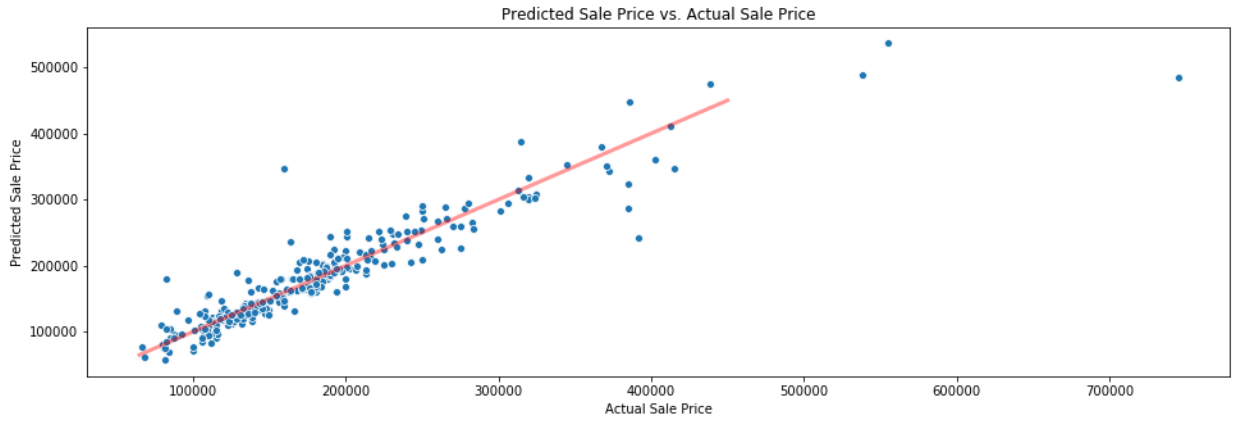

Table 6.1. Summary of Results.

Model	Test MAE (dollars)	Cross-Validation Standard Deviation (dollars)	Performance on Fairness
Basic Linear Model	24,133	63,557,563,609,965	Undervalues both low- and high-priced homes at about the same rate
Lasso Model	20,449	1,426	Undervalues high-priced homes at a higher rate than low-priced homes
Ridge Model	19,474	1,353	Undervalues high-priced homes at a higher rate than low-priced homes
Elastic Net Model	19,790	1,331	Undervalues high-priced homes at a higher rate than low-priced homes
Linear Model with Two-Degree P.F.*	25,706	3,409	Undervalues high-priced homes at a higher rate than low-priced homes
Linear Model with Three-Degree P.F.*	29,169	3,373	Undervalues both low- and high-priced homes at about the same rate
Lasso Model with Two-Degree P.F.*	18,375	1,518	Overvalues low-priced homes and undervalues high-priced homes

Ridge Model with Two-Degree P.F.*	17,659	1,754	Overvalues low-priced homes and undervalues high-priced homes
Elastic Net Model with Two-Degree P.F.*	17,788	1,751	Overvalues low-priced homes and undervalues high-priced homes
Support Vector Regressor	45,744	4,449	Overvalues low-priced homes and undervalues high-priced homes
Random Forest	17,715	1,502	Overvalues low-priced homes and undervalues high-priced homes
XG Boost	17,772	1,405	Undervalues high-priced homes at a higher rate than low-priced homes
Light GBM	16,773	1,788	Undervalues high-priced homes at a higher rate than low-priced homes
Stacked Model	16,364	1,718	Undervalues high-priced homes at a higher rate than low-priced homes

* "Polynomial features" is abbreviated as "P.F."
Note: Optimal metrics are highlighted in yellow.