# Linear Regression

# Data Cleaning

The first step in preparing our data for linear regression was 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 imputed missing values using the median value of the feature in the training set. Only three features (lot frontage, masonry veneer area, and bike score) were missing values, and no one feature was missing more than 18% of its observations.

Finally, we normalized the features in the training and test sets to have a mean of zero and a standard deviation of one. We normalized the test set based on the normalization parameters of the training set.

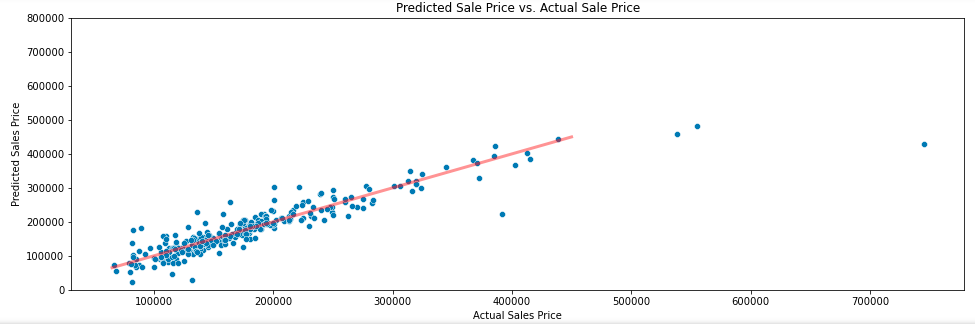
# Results

# Basic Linear Model

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

The model has a mean absolute error of $24,133 and a mean squared error of 3,886,295,539 squared dollars. The R-squared is 43.7 percent.

The predicted sale prices for the test set are plotted below versus 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.

In terms of fairness, among low-priced homes, the 25th percentile of the predictive error is negative $12,021, the median predictive error is $3,828, and the 75th percentile of the predictive error is $13,994. Among the high-priced homes, the 25th percentile of the predictive error is negative $6,313, the median is $7,967, and the 75th percentile is $35,116. Thus, we can see that the predictive error is systematically less negative for high-priced homes. This is evidence that the predicted value of low-priced homes is more likely to be above the actual sale price, whereas the predicted value of high-priced homes is more likely to be below the actual sale price. Thus, low-priced homes are over-valued compared to high-priced homes, which is a concern from a fairness perspective.

# Regularized Linear Model

For our second linear model, we applied Lasso, Ridge, and Elastic Net regression to our dataset. We performed a grid search and used 5-fold cross validation to determine the best alpha hyper-parameter ranging from 0.1 to 10,000.

[INSERT RESULTS]

# 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 5-fold cross validation to determine whether the optimal number of degrees for the polynomial features was two or three degrees.

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 242 features, so we dropped these same features in order to ensure that our pruned dataset was small enough for Jupyter Notebook to handle.

[INSERT RESULTS]

# 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 to this model and used 5-fold cross validation and grid search to determine the best alpha hyper-parameter ranging from 0.1 to 10,000.

[INSERT RESULTS]