Predicting Home Values in Los Angeles' South Bay

Springboard | Capstone 1 In-Depth Analysis

by: Lauren Broussard

Approach:

With a cleaned dataset, I used machine learning to see how well I could predict housing prices in the South Bay area. As we are looking at a continuous random variable (as opposed to a discrete variable), we'll look at this as a regression problem. Further, since we already have labeled data (features and housing prices), we'll do a Supervised Learning approach. Additionally, we will try to determine which features are most important in predicting home prices in this area.

For this problem, I'll be using Random Forest Regression, an ensemble method that expands on the Decision Tree approach.

Data Pre-Processing & Encoding:

To prepare the data to work with the chosen model, I did data transformation of certain columns and data types. For instance, I dropped columns like 'Address', 'MLS#, and 'City." that were either redundant or could not easily be changed into a numerical value. Further, I broke 'Sold Date' out into its component parts (year, month, day) as separate columns since the model would not take the entire DateTime object. Finally, I create a column called "SEASON" in order to further see whether the time of year of a home sale affected the price of the home.

One Hot Encoding

One-Hot encoding was used to change categorical variables to binary values before putting them into the model.

```
# one hot encoding on all categorical variables
south_bay_f = pd.get_dummies(south_bay)

# Display the first 5 rows of the last columns
south_bay_f.iloc[:,15:].head(5)
```

	PROPERTY TYPE_Condo/Co-op	PROPERTY TYPE_Mobile/Manufactured Home	PROPERTY TYPE_Single Family Residential	PROPERTY TYPE_Townhouse	NEIGHBORHOOD_Alondra Park	NEIGHBORHOOD_Carson	NEIGHBORHOOD_Del Aire	I
0	0	0	1	0	0	0	0	
1	0	0	1	0	0	0	0	
2	0	0	1	0	0	0	0	
3	0	0	0	1	0	0	0	
4	0	0	0	1	0	0	0	

Separate Feature Data from Target (Price) Data

Then, I segmented the data with X being all columns (features) of our DataFrame except for price, and y being the price column.

Training and testing data were separated out, with a 70/30 train/test split. This reserved 9541 homes in the training data and 4090 homes in the testing data. After preprocessing using one-hot encoding, I ended up with 51 feature columns.

```
# split data into training and testing
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3, random_state=42)
```

```
print("Shape of training data:", X_train.shape)
print("Shape of test data:", X_test.shape)
```

```
Shape of training data: (9541, 51)
Shape of test data: (4090, 51)
```

Random Forest Regression:

After running pre-processing steps, I ran a Random Forest Regressor model with the default parameters and all features included.

```
from sklearn.ensemble import RandomForestRegressor

# Instantiate model with default values decision trees
randomforest = RandomForestRegressor(random_state = 42)


# Train the model on training data
randomforest.fit(X_train, y_train)

# predict price based on trained model
y_pred = randomforest.predict(X_test)
```

To establish a baseline metric, I considered what might be a reasonable guess for the price of a home. Without any other knowledge, one might guess the median home price. I set this as the baseline to see if our model could outperform that. The baseline Mean Absolute Error (MAE) using this logic is 485739.85.

Running the initial model above yielded the following preliminary results:

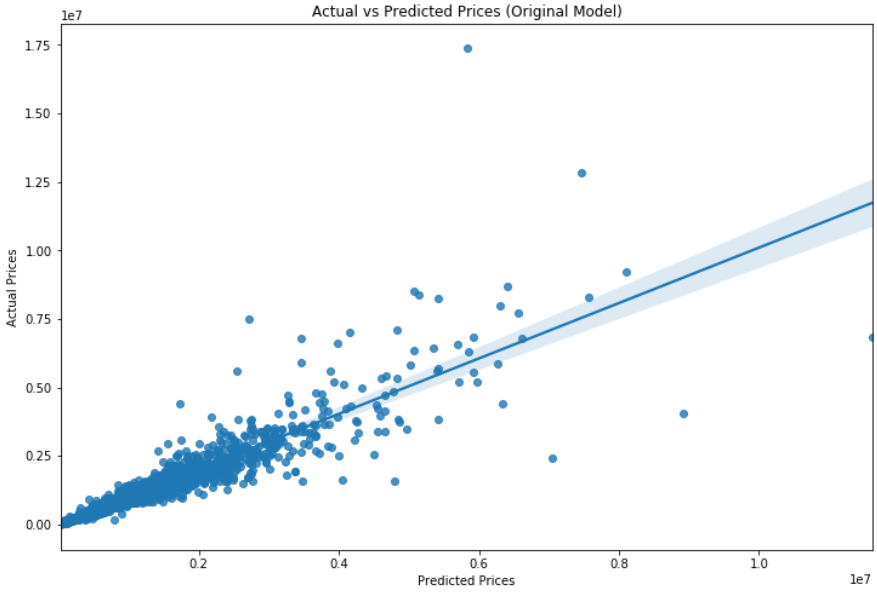
	Original Model
Mean Absolute Error (MAE)	120915
Root Mean Squared Error (RMSE)	363501
R2 Score	0.840562

The mean absolute error indicates that on average, the model predicts homes within about $120,915. This is better than our baseline, but may be able to be improved.

Additionally, I ran training and testing accuracy scores as found below:

- **Mean Accuracy(Training):** 0.97
- **Mean Accuracy(Testing):** 0.84

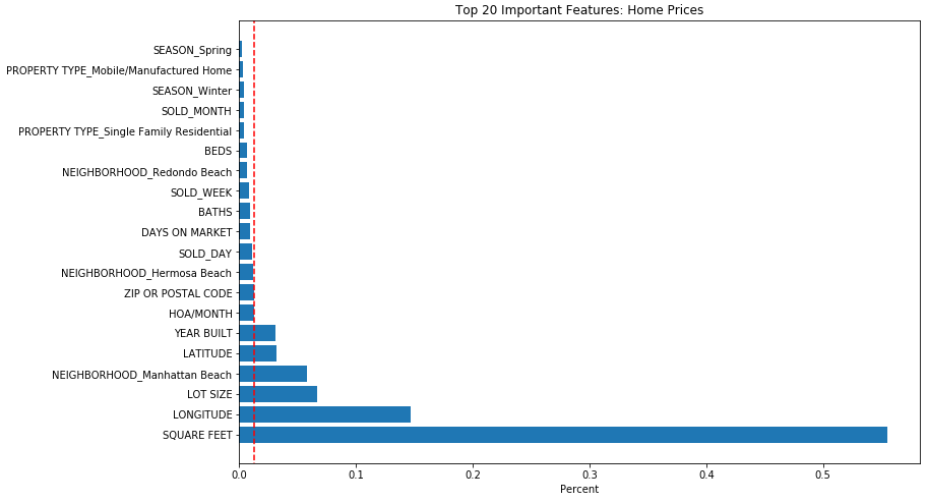
The training data has an accuracy of 97%, but the accuracy goes down to 0.841 with the testing data. This could suggest some overfitting of the training data.



We will try to improve the model with parameter tuning and feature selection.

Parameter Tuning
To see if I could improve the model, I first looked to tune the parameters **n_estimators** and **max_depth**. The parameter n_estimators is the number of trees to be used in the forest, and max_depth tells the model how far down the tree to go. Tuning these parameters suggested that the best values for these two parameters are **n_estimators = 75** and **max_depth = 25 .**

Feature Selection
Using sci-kit learn's Feature Importance method, I was able to list out the top 20 most important features. Looking further at this information, only 8 of those features explained 92.5% of the model. **By far, the two most important features in the model turned out to be the Square Feet of the property and the Longitude** (seemingly, how close



or far a home is to the ocean or inland). The square feet of a property determined about 0.55 of the price prediction.

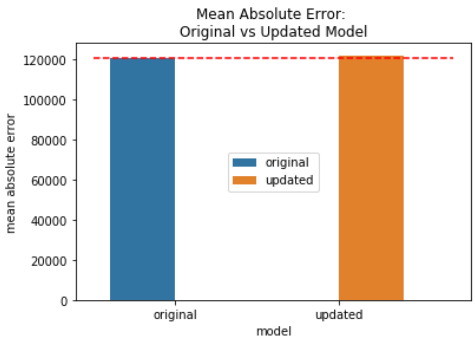
Results + Evaluation of Updated Model:

After implementing the newly tuned parameters and a subset of the original features, I retrained the model and reran the appropriate metrics to see how well it fit the data.

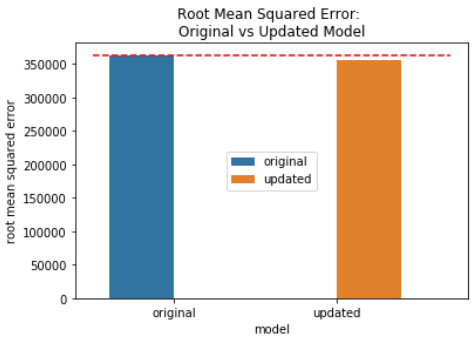
	Original Model	Updated Model	Difference
Mean Absolute Error (MAE)	120915	122162	1246.31
Root Mean Squared Error (RMSE)	363501	356461	-7039.65
R2 Score	0.840562	0.846678	0.00611563

The final tuned model gave only slightly better results than the original model for some of the metrics, and slightly worse results for others.

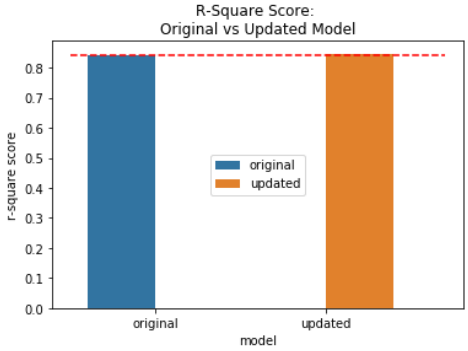
MAE



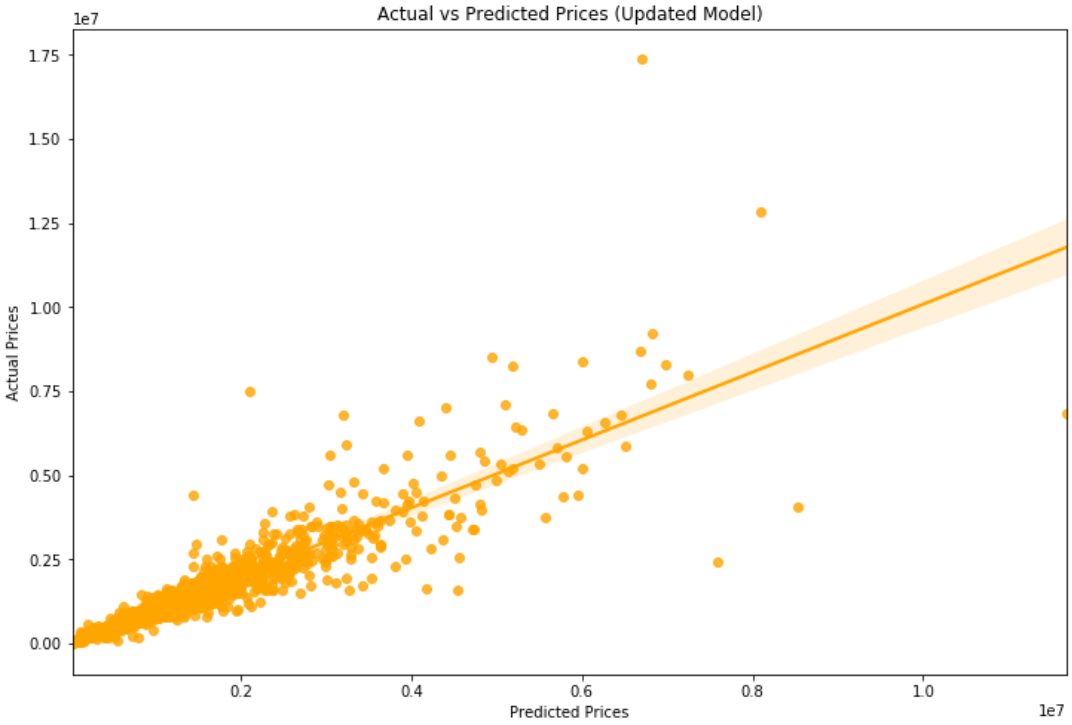
RMSE



R^2 Score



Conclusion and Future Considerations:



While I was able to beat the baseline prediction by $363,577 with the final model, the original model and the tuned model did not seem to perform significantly differently. The model seems to make fairly good predictions up to a certain price point, but does not predict well for others. This may be due to other outliers in the model or other factors. In the future it would be important to look for other parameters to tune, or work to normalize some of the other data.

In terms of some of the other features I was interested in, like season of sale, it does not look like this was a particularly significant predictor for home prices. In the future, other features may be interesting to look at, such as proximity to certain amenities - beaches, gyms, grocery stores, etc. One interesting thing that I found in the data, was how much weight the overall square footage of a home had on its price. This may be important for home builders to consider in giving future home owners the space they need.