Housing Market Linear Regression

Eric Au

The Business Problem

The King County Development Group (KCDG) wants to look into building a new community of family homes in King County (located in Washington State and near Seattle). Along with the King Contractors (KC), the KCDG needs a better idea on what metrics influence the sale price of a home and would like to get a sense of how to price these homes. KCDG and KC would like to bring on engineers and architects to assist with the design of these homes but need to understand how the sale price of the home will change depending on the design parameters.

The intention is to develop a sale price algorithm to help set a target price for a new housing development in King County.

- The main purpose of this algorithm is predictive, meaning that the model should be able to take in attributes of a home that does not yet have a set price, and to predict a sale price for that home.
- We will also take a look at the model's attributes and explain possible relationships between the attributes of a home and its price.

Stakeholders: The King County Housing Authority (KCHA), King Contractors (KC), prospective architects and engineers.

The Dataset

This project uses the King County House Sales dataset found in the `data` folder in this repository. The description of the column names for the data set can be found in `column_names.md` in the same folder.

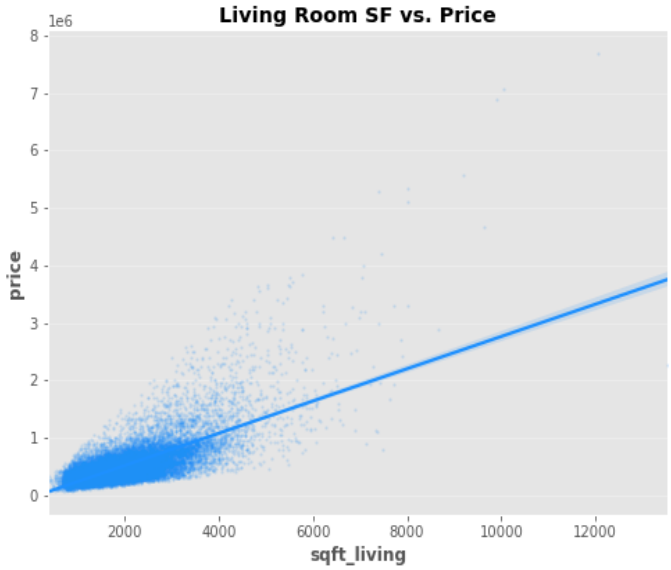
Methods

Baseline Model

The target variable for this project is `price` with all other columns in the dataset chosen as preliminary predictors for this project. A 75%-25% Train-Test split was performed with `price` as the target variable, `y`.

As a baseline, the predictor variable with the highest correlation with `price` was `sqft_living` as it was the highest correlated predictor. However, this simple baseline model did not perform well.

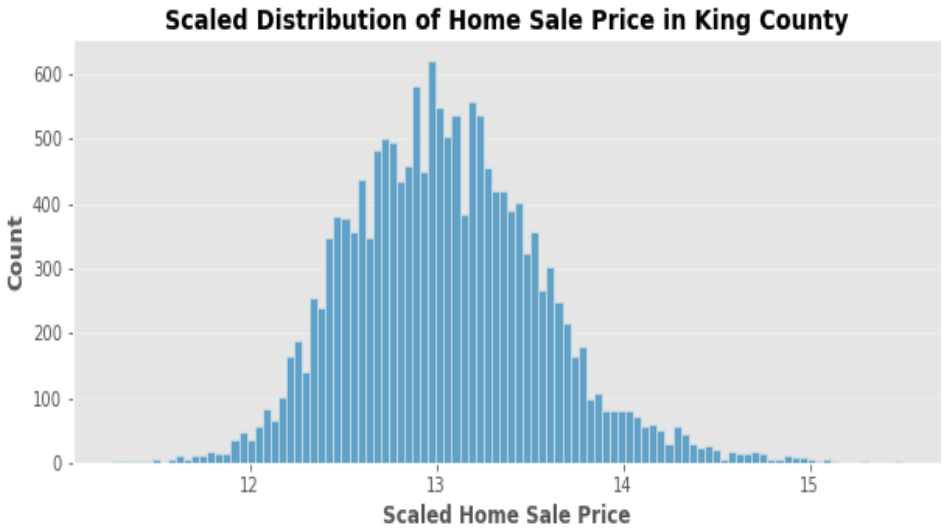
```
Base Training R2: 0.4848
Base Test R2: 0.475
```



However, further exploration into the other predictors was needed to determine an accurate model.

Target Variable

Distribution of the target variable `price` was skewed and transformed using a log function to have a more normalized distribution. The target variable `price` would need to be exponentially scaled back to determine final price after conclusion of modeling.



Data Cleaning & Preprocessing

The following transformations summarizes the steps done with regards to cleaning and preprocessing and applied to the `X_test` set as follows:

- Drop unnecessary columns, clean the `date`, `grade`, and `basement` columns to integer values.
- Create encoded nominal values for `waterfront`, `view`, `condition` and `yr_built` columns.
- Log scale the continuous predictors and drop unnecessary columns not logged.
- Encode the zipcodes and concat zipcodes back to the previous dataframe set.
- Apply a standardized scaler to the final dataframe set.

Note: Transformations and scaling of the testing set was not performed until after assessing the R2 scores after each model was test. However, cross-validation was performed on the test set throughout each iterative model, which was a good indicator that the test set will also perform well after trasnformation and scaling.

Modeling & Regression Results

The following summarizes the coefficient of determination scores (R2) on each model after scaling/transforming.

2nd Model (after initial preprocessing)

```
2nd Model Train R2: 0.7721
```

3rd Model (removed excess predictors, log numerical variables, added encoded zipcodes)

```
3rd Model Train R2: 0.8743
```

4th Model (applied Standard Scaler)

```
4th Model Train R2: 0.8743
```

Validation Checks

```
4th Model Train score: 0.875
4th Model Test score: 0.8711

3rd Model Train score: 0.875
3rd Model Test score: 0.8711
```

```
2nd Model Train score: 0.7714
2nd Model Test score: 0.7737

Baseline Model Train score: 0.4833
Baseline Model Test score: 0.4889
```

Alternate: Check for Multicollinearity aka Investigating Inference Variables

While the purposes of this project was to perform a predictive model, I also investigated the variables causing colinearity. Changes in one variable may be associated in huge changes in another variable, thus causing issues interpretting the coefficients associated with the predictors.

	cc
pairs	
(sqft_living_log, sqft_above_log)	0.865190
(sqft_living_log, bathrooms)	0.761335
(sqft_living15_log, sqft_living_log)	0.750639

The original predictor variable, `sqft_living` (aka now transformed to `sqft_living_log`) is highly correlated with other variables and likely leading to multicollinearity in the dataset.

- drop `sqft_above_log` since values are already captured in `sqft_living_log`
- drop `sqft_living15_log` since we only care about the living space SF and not neighbors.

5th Model (removed colinear variables)

```
5th Model Train R2: 0.6096
```

6th Model (reintroduce encoded zipcodes)

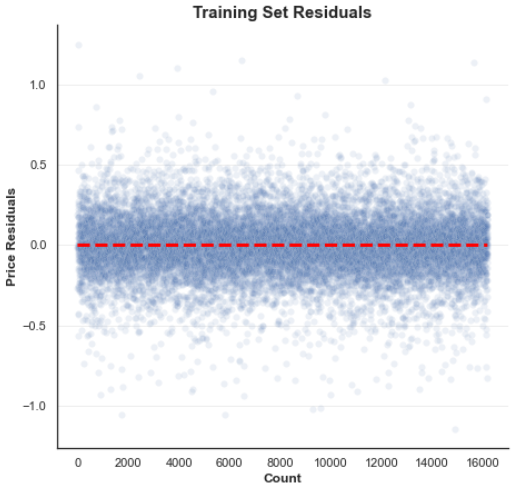
```
6th Model Train R2: 0.8662
```

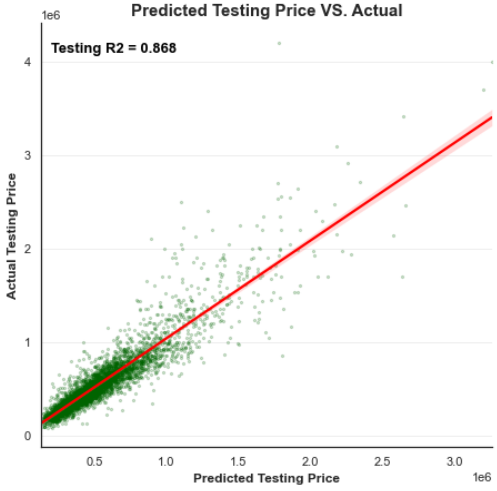
Testing the `X_test` Set with the 4th Model

The final model chosen was the 4th Model since it performed the best and had the most relevant predictors. The `X_test` set was finally tested following the same scaling and transformations performed on the `X_training` set.

```
4th Model Train R2: 0.8743
4th Model Test R2: 0.8684

Train Root Mean Squarred Error: 139905.4892867977
Test Root Mean Squarred Error: 131025.48301270237
Difference in RMSE for Test/Train: 8880.0063
```



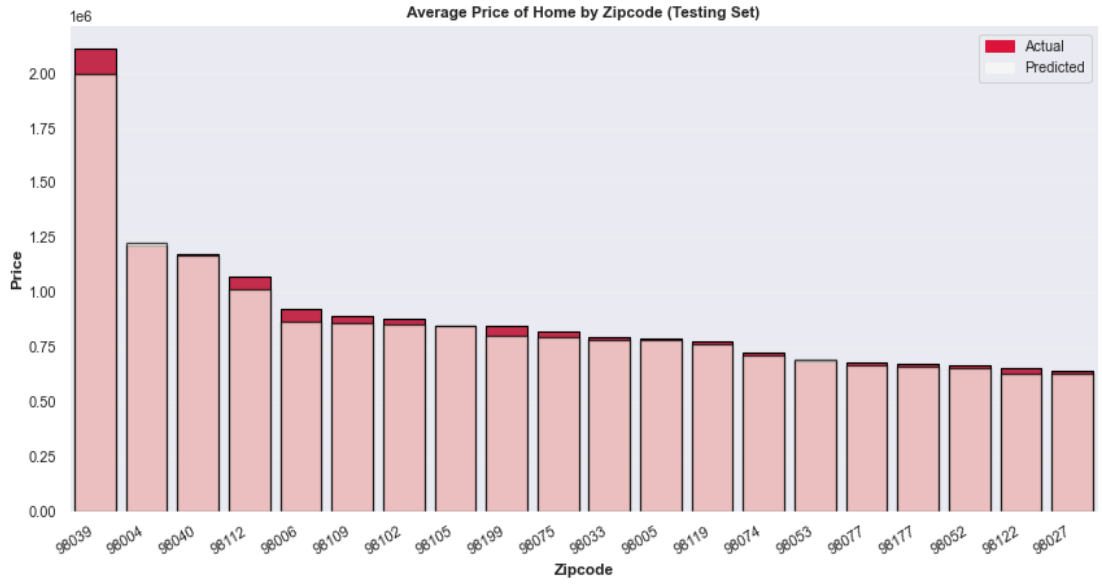


The training R2 and test R2 are very close to one another. This is good, and proves the validation process performed leading up to this point was very accurate.

Additionally, the RMSE difference between the testing and training set is about $9,000. Meaning that the model is about $9,000 off from the testing set.

Highest Priced Zipcode Areas

Since we know that zipcode is a big predictor in this model, I was curious about which zipcodes can we expect to see the highest home prices.



Based on the above, the 4th model is very accurate! There is some variance between the actual and predicted for homes that are more expensive on average. But interesting to note that these are the top highest priced neighborhoods:

- Medina, WA 98039
- Bellevue, WA 98004
- Mercer Island, WA 98040
- Seattle, WA 98112

Creating an input model to predict home prices based on predictors

Recall, the business problem here was to determine sale prices for homes based on an input of parameters. Using the 4th model, a structured input machine was created that takes in input values for the predictors and provides an estimated price for that home.

For example, see below inputs:

SF Living: 4000 SF
SF Above: 3000 SF
SF Living Nearest 15: 3000 SF
Age: 1 (New Building)
Number of Bedrooms: 2
Number of Bathrooms: 2
Number of Floors: 2
View Quality (0-4): 1 (Fair)
Condition Quality (1-5): 4 (Good)
Grade Quality (1-13): 7 (Average grade of construction)
Renovated? No
Zipcode: Seattle, WA 98112

Predicted Sale Price: $ 1,369,146

Conclusions & Recommendations

- Choose the 4th Model because it had the highest R2, also has more predictors.
- The 6th Model removed many predictors but addressed colinearity between the predictors.
- Zipcode explains a significant amount of variance in the model.
- Positive Predictors: SF Living, Bathrooms, View, Condition, Grade, Renovated
- Negative Predictors: Age, Bedrooms, Floors

Next Steps

While this project examined the housing market in the greater Seattle, Washington region, it was only limited to data from 2015. Thus, it would be interesting to explore more recent data and compare prices of homes up to date. Additionally, the process for creating this model could be performed on similar datasets. Could be an interesting future project to explore housing prices in other markets as well.

Repository Structure

```
├── data
├── images
├── .gitignore
├── Housing Market Analysis Slides.pdf
├── Housing Market Linear Regression.ipynb
└── README.md
```

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%