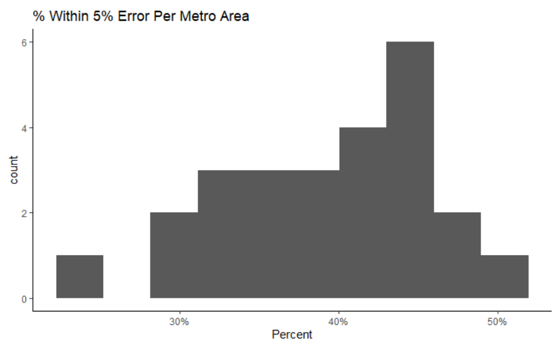
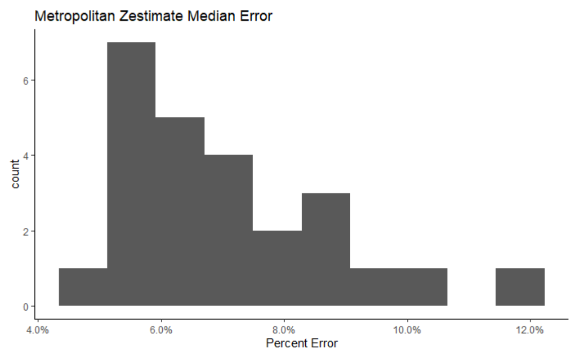
Introduction

Housing and real estate has been on the horizon for data science and analytics for years. As mentioned in our proposal, Zillow, a major player in the housing space, has attempted to create machine learning models to predict housing prices and find arbitrage in prices. Zillow eventually scrapped the project saying large scale market prediction was nearly impossible. The company currently does have a value called ‘zestimate’ which predicts the assets value. These graphs below show Zillow’s public modeling error:





Zillow also has features such as ‘rent estimator’ which looks at previous values and predicts whether the rent is above or below market value. As it currently sits today, the housing market is skyrocketing to unseen heights.

Our group overall disagrees with Zillow and believes that through leveraging the power of Spark and other distributed computing we can account for fluctuations in the market. We are using Kaggle housing price data sampled from the Pacific Northwest of The United States. We believe that starting with a small model and gradually scaling into larger data.

Our data is 515KB in a comma separated value format. The data is 4800 rows and 18 columns in size. The columns include: date, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, waterfront, view, condition, sqft_above, sqft_basement, yr_built, yr_renovated, street, city, state zip, and country. All columns are in numeric format excluding: Date, street, city, state zip, and country. Country was not needed and was dropped as all homes were located within the United States.

The data was relatively clean but had some intricate details which needed addressing. Initially, at first we realized there were duplicate street addresses. Upon further investigation we realized that duplicates either had different dates (rarely), or were apartment complexes. As long as there was some kind of distinction the rows were kept. We also noticed some prices of zero dollars. These values didn’t make any sense in context so they were dropped. In total we only need to drop 49 rows out of the 4800. Another intricacy in the data was that there were asset with no bedrooms but extremely large lots. This was an indication to use that this was not just a housing dataset but a general real estate dataset that contained great swaths of land.

Problem

Our problem is ultimately simple: predict asset prices based on asset attributes and location. We considered many error metrics such as mean squared error or mean absolute error, but ultimately settled on root mean squared error. Root mean squared error provides the most understandable error as this the loss will be in context of dollars. For example, if the root mean squared error loss was ten thousand, that immediately indicates that we are around ten thousand dollars off actual target value. Residuals on an individual basis, likewise, will be understandable. Error measurement is extremely important in the context of this problem. Housing and real estate are most people's great assets. Predictions with even five percent error could have implications on financial stability for whomever is holding the asset. Zillow most likely ran into this problem. Zillow's estimates were by most definitions acceptable, but in the context of this business problem, were completely unacceptable.

Most individuals using Zillow in place of a real estate agent do not have enough background in real estate to make a fully informed decision by themselves. If listed estimates are radically wrong for sellers or buyers, this could have disastrous implications. During the project we continuously return to the question, 'how are we doing compared to zillow?' Zillow put their model performance in the context of percent error from true value, most likely due to the previously mentioned error implications. For the purpose of a baseline and comparing our model, we converted our errors into percent error.

We used Google Colab for data cleaning, feature engineering, and modeling. We believe this was the best course of action as Colab notebooks can be downloaded as .py scripts and ran on cloud servers. For future scaling, having a Python script easily callable and applicable to multiple geographical locations (as long as the data remains the same) is extremely important. This also provides a way to compare geographical regions within a similar context, as they would all have the same script run on them.

Feature Engineering

When it comes to feature engineering we decided to first one hot encode the cities and the zip codes. This should be an obvious decision. As we all know location is considered to be the name of the game when it comes to real estate. As such, keeping the zipcode information is very important. We might have to try to move to a different embedding as our data set grows for space constraints. However, for the time being this works well as detailed in the results section We also used hashing to embed the street names. This was at first a controversial choice. However, we decided that if a house is on the same street as another house that we have the price point for it is more likely that those houses would at least be correlated. This can also capture things like neighborhood quality as a latent feature.

Modeling

When modeling we had originally decided to start with using a more simple model for inheritability then more to a more complex model. As such, we decided to use a generalized linear model to minimize the sum of squares. General linear model in particular linear regression works by projecting the vectors in such a way to minimize the least squared error. An interesting outcome was how well the model worked which will be described in more detail in the results section. However, because of this phenomenal performance and the business oriented view of the problem we decided to stay with a simpler model. The reason for this is a more complex model might lead to overfitting and would be less understandable. Also, when training models on large datasets this requires computational time. Given that we were treating this as a real world problem, using more computation would cost the business money for next to no return given the previous outcome.

Our tools that were used to solve this problem were all based in python. We used pyspark for feature engineering. We then used the MLib in pyspark to generate our linear model. These choices were used because as the dataset grows we would want our model to be robust to size and pyspark scales well. Lastly, we used pandas and seaborn/matplotlib in base python for plotting.

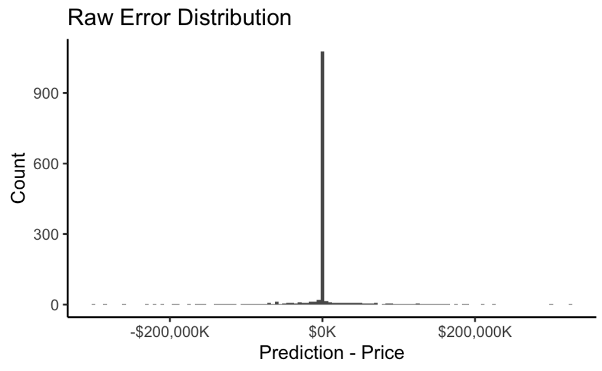
The biggest modeling problem we ran into was deciding if we wanted to run more complex models. This proved to be a conundrum because we were not sure if that made sense from a business or data science perspective because we had already managed to achieved satisfactory results. We ultimately decided not to because of the reasons listed above. The other major problem we had was trying to find Zillow's results to compare our model. We had to compare slightly different results Zillows out of market and our in market price points. However, we believe this to be a close representation.

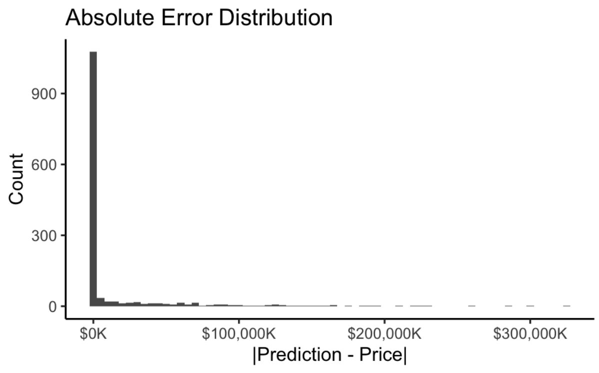
Results

The two metrics we used for evaluation were R2 and RMSE. The R2 value of 0.997 measures almost a perfect fit between our model and the data. This also implies that there is a strong relationship between the model features (independent variable) and the price prediction (dependent variable). The RMSE value of $40,819 seemed like a high value in comparison to the R2 value, so further investigation was needed into the raw and absolute dollar amount errors.

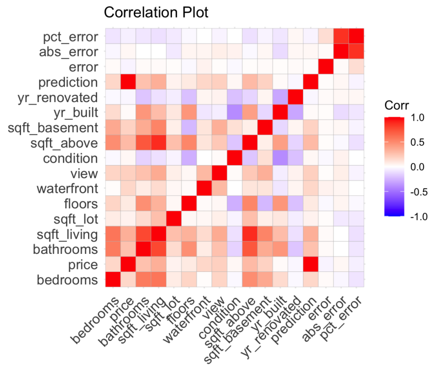
Our raw results were surprisingly very good with 90% of our predictions being within $5,000 of the actual price and 93.44% of our predictions being within 5% of the actual price. The average error in our predictions was $435.80 while the median was -$2.60. Since the median and mean differ by a large amount we assumed that there are some large error values that skew those results. This is represented by minimum error of -$301,100.70 and maximum error of

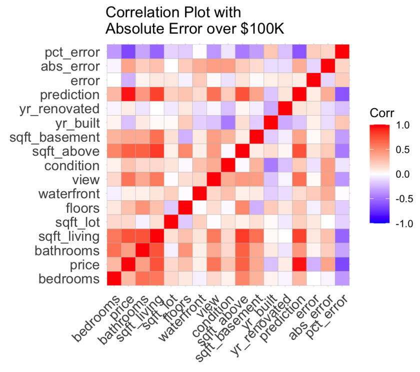
$325,265.80. We also analyzed the absolute error to disregard the direction of the error in our predictions. The median absolute error was $234.40 and the mean absolute error was $13,963.70. Once again, these results are skewed due to the errors over $300,000. Below are histograms displaying both the raw error and absolute error. The bidwidths for both graphs is $5,000.





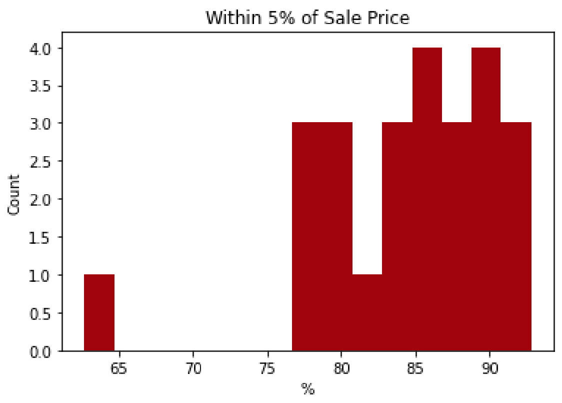
We further investigated our predictions to see if there were specific scenarios where our error spiked. Unfortunately there are no correlations between the characteristics of a house and the error amount. To extend our investigation even further, we filtered on absolute errors over a $100,000 threshold and there were still no significant relationships. In fact, we found that the percent error was negatively correlated with many of the house features when the absolute error amount was over $100,000.



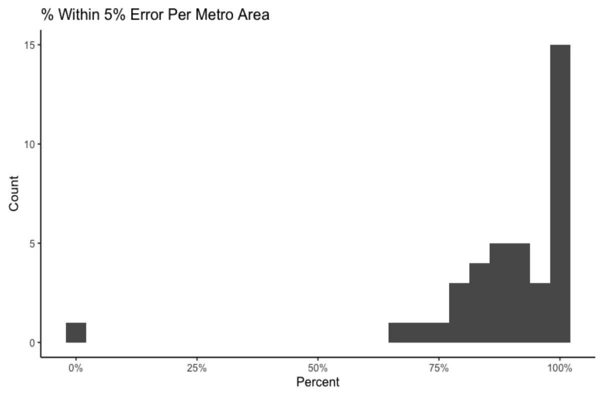


Model	Median Error	% Within 5% of Actual
Zillow Off-Market Model For Seattle Metro Area	6.4%	41.6%
Zillow On-Market Model	2.2%	78.0%

For Seattle Metro Area		
Group 4 Model	0.05%	93.44%



Zillow Error



Group 4 Model Error

In conclusion, our model performed better than the Zillow estimates by 2.15% median error. But we do not know how accurate our model is after the house or asset is actually sold and comes off the market. Upon further investigation and speculation, it looks as if this could be the reason Zillow scrapped their Zestimate project. Zestimates were accurately predicting what a house or asset should sell for with little error, as ours is as well. But, their model is not able to predict the actual price sold once the final deal is made. We have determined, most likely ours will not perform well in this scenario either. There is difficulty in predicting and validating prices that do not yet exist, are driven by human bargaining and individual need. As assets come off the market, prices can be validated, if basic economic principles apply, the new validation prices change the underlying market, making it nearly impossible to get the exact price.

The next steps of this project would be to extend to other metro areas beyond the Pacific Northwest, specifically Seattle. Since our model uses a PySpark data pipeline, our model should be robust enough to scale to a vast amount of data with these additional metro areas. Another consideration would be adding a forecasting type model to the project that does not just predict prices but looks at price changes over time in macro rather than at an individual level. Zillow slowly has hinted in this direction as most of their visualizations are moving to graphs in a timeseries like format. Another possibility, in Zillow's case, is to look at web traffic searching real estate in different areas, as that might be a better indication of popularity and demand in real-time. The key difference being, regression models will always be one step behind as they do not look into the future they simply look at features and existing price.