**King County Housing Price Prediction Project**
TCSS 551 Big Data Analytics

**Group Members**
Robert Wolf

## Problem Formulation/Introduction

The problem I have chosen for this project is prediction of housing prices in King County based on the features of the house. The data set I will be using contains several features of houses. This data analysis would be extremely useful to home buyers, real estate agents, and appraisers. It would greatly help them predict pricing of homes, identify trends, and make informed real estate purchases.

One of the challenges in this project will be the quality of the data. There are some incomplete entries and outliers that will require cleaning. The features included in the data set will need to be analyzed carefully for relevance.

## Data Design/Data Description

The data obtained from https://www.kaggle.com/harlfoxem/housesalesprediction contains housing data from May 2014 to May 2015. The population that this will be applied to is all homes sold in King County in the future. The data obtained is only somewhat representative of this population as it contains all home sales in only the year between May 2014 to May 2015. If we make some assumptions about the data, we can use it. To use this data to solve this problem, we must assume that home prices do not change drastically year to year, and we also have to assume that this data set is complete and accurate. This assumption does not necessarily reflect reality, as past data does not necessarily represent the future house prices.

The dataset used is rectangular with 21 features such as square footage, number of bathrooms, lot size, and date sold. The sales price of homes is very right skewed, with the majority of sales occurring at the lower end of the range. Many other features also display a heavy right skew.

The granularity of the data is high, with a separate data point for each home sale. Its scope is limited to King County only, and is assumed to be complete for this area, and the temporality is limited to May 2014 to May 2015. It should be fairly faithful to reality because of the completeness of the sample, but will likely need to be updated over time because of the limited temporality and the fact that housing markets may fluctuate over time.

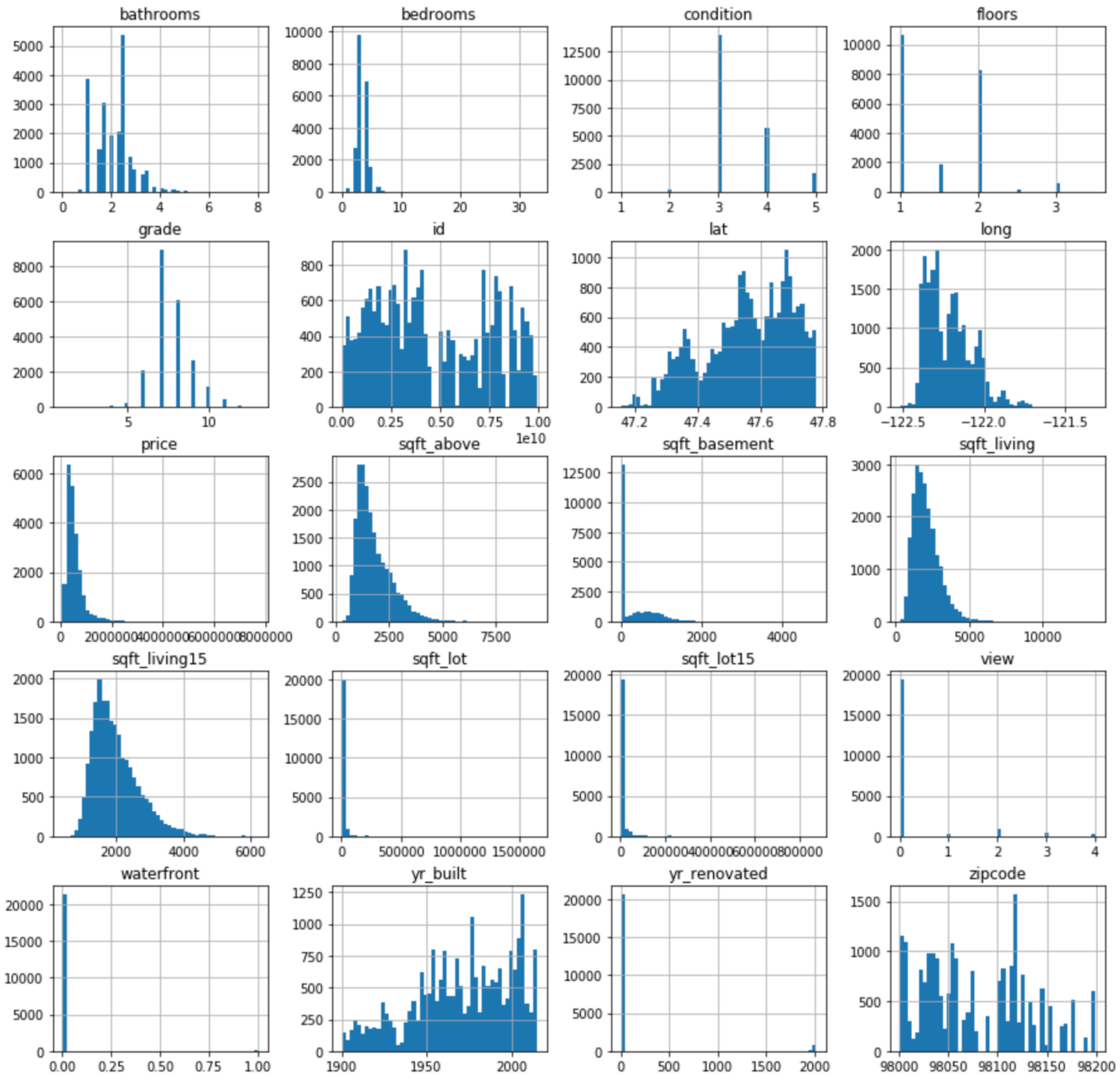
## Related Work

I have found a good example of a very similar project using a different data set. The project is well documented, complete with code snipes and good visualizations. The data differs from my project in that it is summarized for each district, so it is less granular. It does use similar features, though, such as house value, income, bedrooms, and number of rooms. They demonstrated several prediction methods using linear regression, decision tree regressor, and random forest regression. The closest they got their model was within $50,000 of the actual price after fine-tuning, which I don’t think is very good, considering their median house prices ranged from $120,000 to $265,000. Their data set was much less granular than mine, so that could explain the error in their model.

Source: https://medium.com/@gurupratap.matharu/end-to-end-machine-learning-project-on-predicting-housing-prices-using-regression-7ab7832840ab

## Data Transformation and EDA

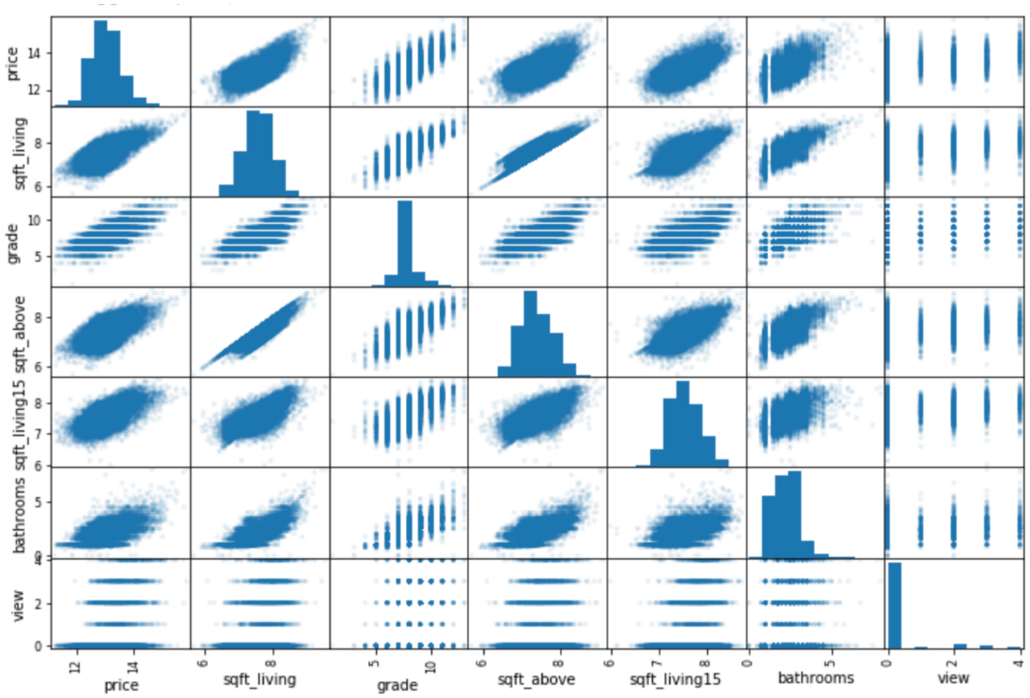
It can be seen from the histograms of the numerical data that many features present a heavy right skew.



In order to prepare the data for prediction and inference, I have taken the natural log of several features: Lot square feet, price, square feet of living space, and square feet of above ground living space. I have done this to make for a more normal distribution of the data set, and it seems to have improved the correlation between features and price as seen in the next feature.

**Correlation Values Matrix – Feature "Price"**

| Feature | Before | After |
|---|---|---|
| Price | 1.000000 | 1.000000 |
| sqft_living | 0.702035 | 0.703634 |
| grade | 0.667434 | 0.674913 |
| sqft_above | 0.605567 | 0.607187 |
| sqft_living15 | 0.585379 | 0.586303 |
| bathrooms | 0.525138 | 0.550802 |
| view | 0.397293 | 0.449174 |
| sqft_basement | 0.323816 | 0.346522 |
| bedrooms | 0.308350 | 0.343561 |
| lat | 0.307003 | 0.316970 |
| waterfront | 0.266369 | 0.310558 |
| floors | 0.256794 | 0.174586 |
| yr_renovated | 0.126434 | 0.137727 |
| sqft_lot | 0.089661 | 0.114498 |



Based on the EDA done here, I have chosen to use the following features: Price, sqft_living, grade, sqft_above, sqft_living15, bathrooms, view, sqft_basement, bedrooms, lat, waterfront, floors, yr_renovated, and sqft_lot. They all have strong correlation with price, and I believe the others are not correlated enough to be useful.

## Proposed Methods

I am proposing a linear regression model to solve this problem. The features I have selected appear to be linearly related to price, which leads me to think that linear regression would be very performant. I will adjust the features I use and exclude some if necessary to improve performance of the model. Likely candidates for exclusion in linear regression are view, waterfront, and floors, since those are categorical in nature and I do not believe they will contribute to the performance of linear regression.

A second method that I will test will be random forest of regression. I will be able to include those features that I have excluded from linear regression. I believe it would be beneficial to perform the random forest regression both with and without the logarithmic transformation of the data to see which performs better. I am choosing random forest because random forest tests combinations of features without me having to manually select them, which may yield superior results to linear regression.

## Experiments

Now that the data is prepared for training, I have split it into a train set and a test set, reserving 20% of the data for testing the accuracy of my model (17290 train + 4323 test). I was able to fit a linear regressor model from the sklearn package, as well as a random forest regressor. To evaluate these two models against each other, I have chosen to use a mean squared error metric, also provided by the sklearn package. Using the default parameters on both models, the MSE for the linear regressor was 217,779. Using the random forest regressor, it was 179,286. We can see that the random forest regressor has the lower of the two errors and performed better, so we will use this model. In order to fine-tune this model, I performed a grid search which trains several models with different hyper-parameters and measures their negative MSE in order to determine which is the optimal choice. After tuning the n_estimators and max_features hyper-parameters, the model performs even better with a MSE of 173,047.

## Conclusions

So we now have a model that has been tuned for our data, which we will be able to use to make predictions on housing data. I am not very satisfied with this model, however. Even after the fine-tuning stage, the MSE only got down to 173,047, which is still not very accurate. This has been an interesting learning experience on how important the basics of feature selection and feature engineering is. I believe that is where this project is weak and explains why my model does not perform to a level of where I would be satisfied. I think in the future I would like to explore more methods of feature engineering to improve this model.