STAT 502 – FINAL PROJECT

HOUSE PRICE PREDICTION IN KING COUNTY, W.A

An T. Nguyen

Table of contents:

1. INTRODUCTION

This is a report of my progress during the final weeks of Spring 2020 for the King County's house price prediction competition on Kaggle.

The coding language is Python 3 and the coding platform is Jupyter Notebook. They allow parallel computing, which vastly improves the calculation speed. They also improve the readability of my workflow compared to R.

The objective for model selection is the 10-fold cross validation of Root Mean Square Log Errors of the training dataset. This is a balance between the LOOCV (less computing time) and using a random set of the training set as testing (more reliable).

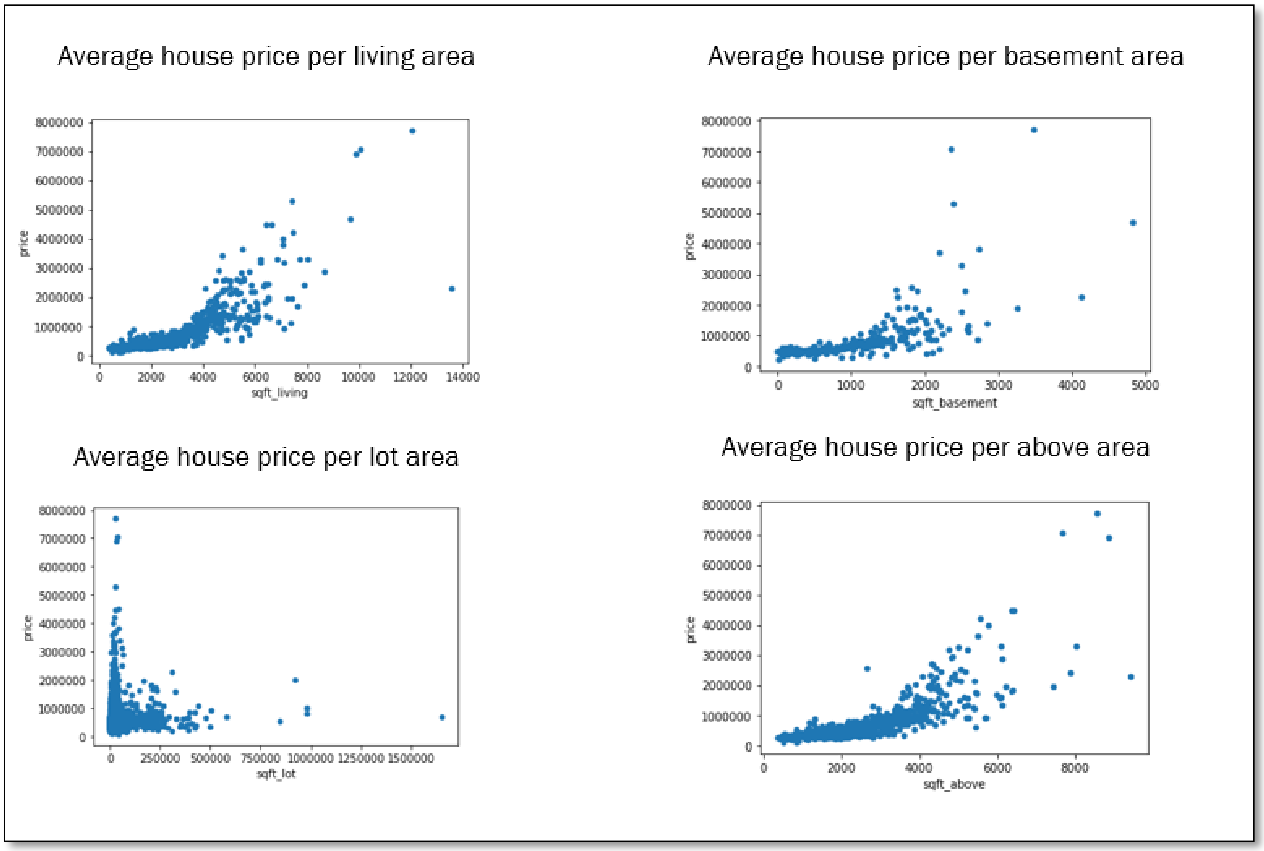
2. DATA VISUALIZATION

a) Relationship between features and target variable:

For easier interpretation of the results, the features are split into 4 main groups: size, location, conditions, and perks. The splitting is as followed:

Location	Size	Condition	Perks
• Latitude	• Living area	• Grade	• Waterfront
• Longitude	• Basement area	• Condition	• Number of views
• Zip code	• Lot area	• Year built	
	• Total area above ground	• Year renovated	
	• Average living area of 15 nearest neighbors		
	• Average lot area of 15 nearest neighbors		
	• Number of bedrooms		
	• Number of bathrooms		
	• Number of floors		

Note: blue color indicates a quantitative variable and red color indicates a categorical variable.



Average house price per living area

Average house price per basement area

Average house price per lot area

Average house price per above area

Figure 1: Scatter plot of average house price w.r.t areas of the house

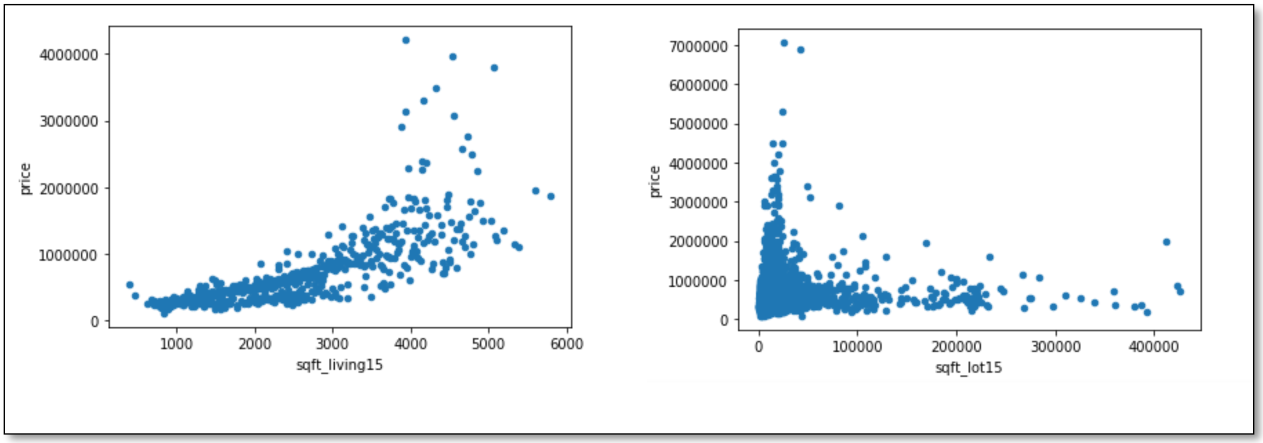


Figure 2: Scatter plot of average house price w.r.t areas of neighboring houses

Since the dataset contains 10000 observations, scatter plot yielded a lot of duplications, which makes it very difficult to identify a potential relationship between each feature and house prices. Luckily enough, all the areas are stored as integer, so I take the average house price per value of area and plot them in figures 1 and 2. This significantly reduces the number of points existed and thus improves readability.

We can see that some of the features demonstrate clear positive correlation with house prices, such as sqft_living and sqft_above, while other features such as sqft_lot and sqft_lọt do not.

Further explorations with respect to other variables are available in Appendix A. In short, the majority of available features exhibit weak linear relationship

b) Relationship among features

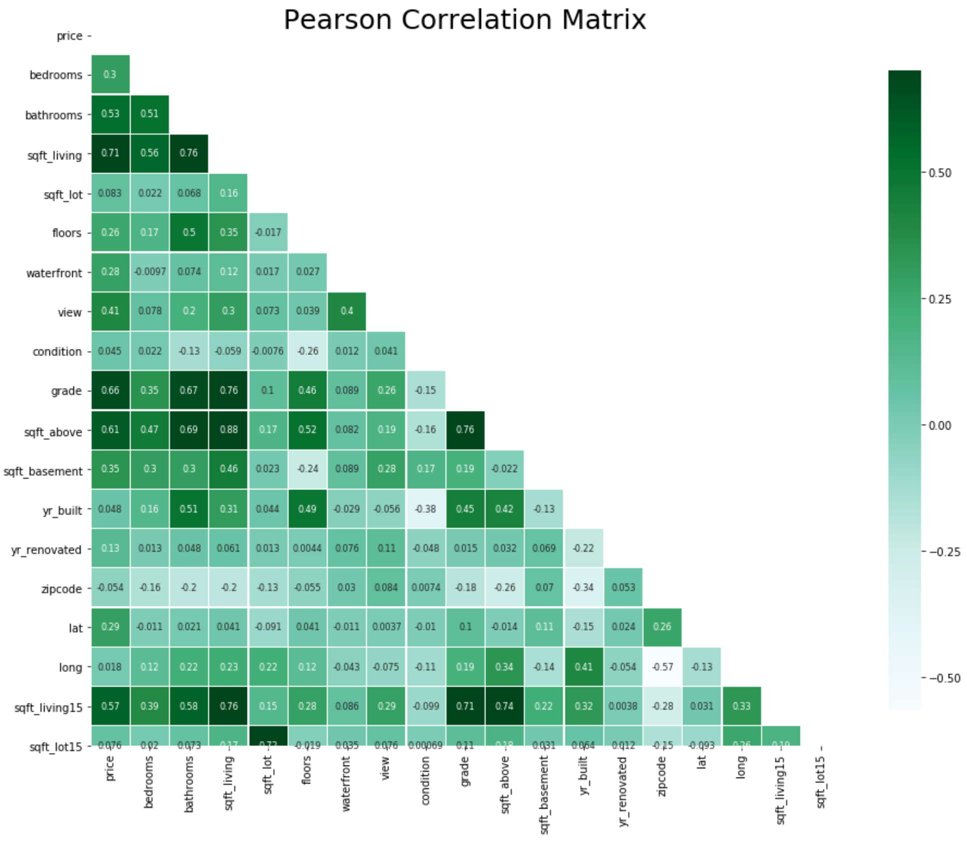


Figure 3: Heatmap of correlation between variables

Many features are indeed correlated, some are completely linearly dependent. For example, the sum of *sqft_above* and *sqft_basement* is *sqft_living*. As correlated features often lead to overfitting the model, sqft_living is removed from the dataset.

3. MODELS SELECTION:

a) Linear regression & regularization

From the data visualization results, it is apparent that many features demonstrate weak to moderate linear relationship with the target variable. A linear model with appropriate choices of variable is thus a reasonable candidate for modelling our data. To address correlation among features, we will apply Principal Component Analysis to project the data onto a subspace of linearly independent components. Regularization (both L1 and L2) is also applied to improve the model's performance.

Since `zip code` is an important variable in identifying a property's neighborhood, but it is not a quantitative variable, special treatment using dummy coding is applied.

The result is as followed:

	Linear Regression	Lasso Regression (L1)	Ridge Regression (L2)
Without PCA	0.4199	0.4907	0.4501
With PCA	0.4650		
Zip code included	0.6787	0.3660	0.2786

The shrinkage parameter for the models are chosen from experimenting. For example, when using Ridge Regression, this plot below shows how the shrinkage parameter (alpha) affects the RMSLE of my prediction.

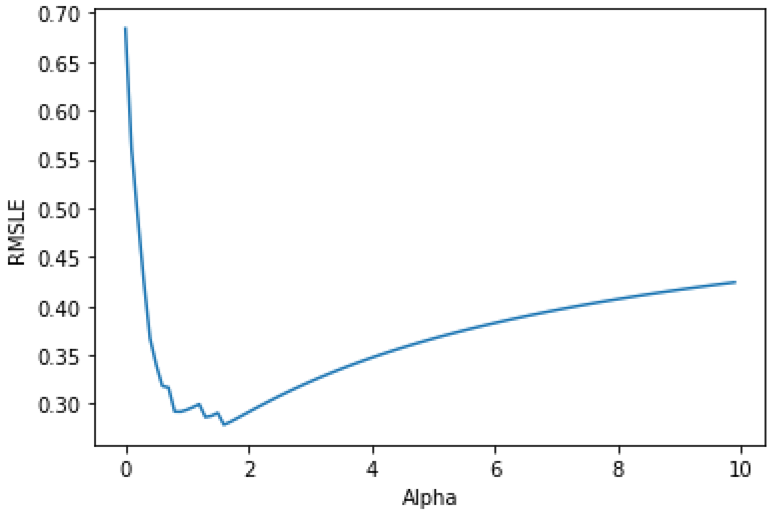


Figure 4: Shrinkage parameter and corresponding 10-fold RMSLE of Ridge Regression

b) Decision Tree Regression:

A disadvantage of linear models is that they cannot account for the interactive terms between features. For example, it is shown that the number of floors is positively correlated with house prices, but given the same living areas, houses with fewer floors are generally more expensive (the cost of land is higher than that of an extra floor). We cannot try to introduce new features without probably inducing correlation to the dataset. Decision Tree Regression is a beautiful solution to address this. The algorithm can also handle categorical variable, which is quite convenient.

For a decision tree, the most important parameters are the **cost function** and **number of samples for splitting a node (m)**. If the value of m is too low, the model will overfit our training data while a high value of m will increase the variance. The following figure shows the CV-RMSLE for each value of the latter parameter.

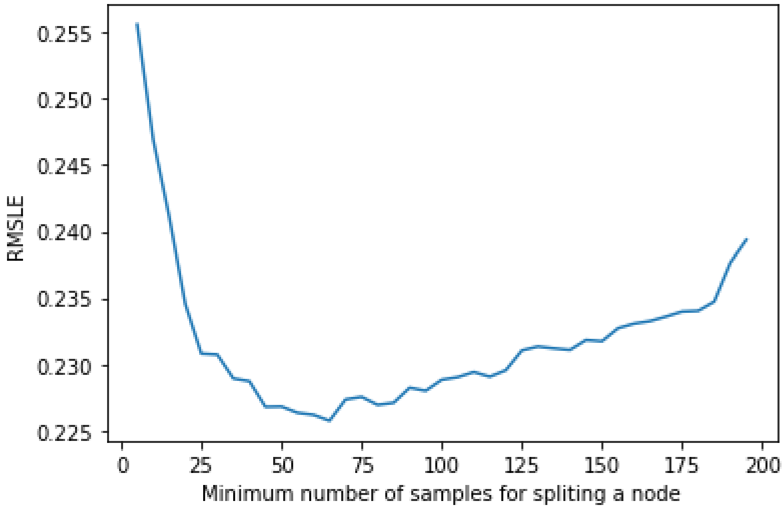


Figure 5: Choosing the minimum number of samples for splitting a node

The minimum RMSLE of 0.2257 is achieved at m = 60

c) Boosting Algorithms:

There are a variety of boosting algorithms / methods that can be used to improve the prediction performance of a decision tree model. The most common ones are:

- Adaptive Boosting (Ada Boost): aims to put more weights in instances that have a higher prediction error.

- Bagging: construct multiple trees based on randomized subsets of the training data, then aggregate the prediction across built trees.
- Random Forest: similar to bagging, but this time the selection of features is also randomized in constructing regression trees
- For each boosting algorithm / method, there are different parameters to be adjusted. Similar experiments in the previous subsection are used in determining the best sets of parameters.

	Adaptive Boosting	Bagging	Random Forest
RMSLE	0.1802	0.1955	0.1851

The Decision Tree Regressor with Adaptive Boosting is the model that I ended up using for my final submission on Kaggle. The result is just slightly worse than my cross-validation result.

4. CONCLUSION

This project showcases the difficulty one must face with a real-world problem in modelling. There are many educated guesses to be made, and the results are often disappointing. In hindsight, there are a lot of things that I could have done to potentially improve the result: feature engineering with binning, clustering… or experimenting with neural networks. The best takeaway from this project, in my opinion, is that intuition and insights of the underlying problem are just as important as knowing many statistical techniques in addressing it.

5. REFERENCE

Kaggle kernels:

https://www.kaggle.com/burhanykiyakoglu/predicting-house-prices

https://www.kaggle.com/kabure/predicting-house-prices-xgb-rf-bagging-reg-pipe

Coding notebooks:

APPENDIX A - Additional data visualization

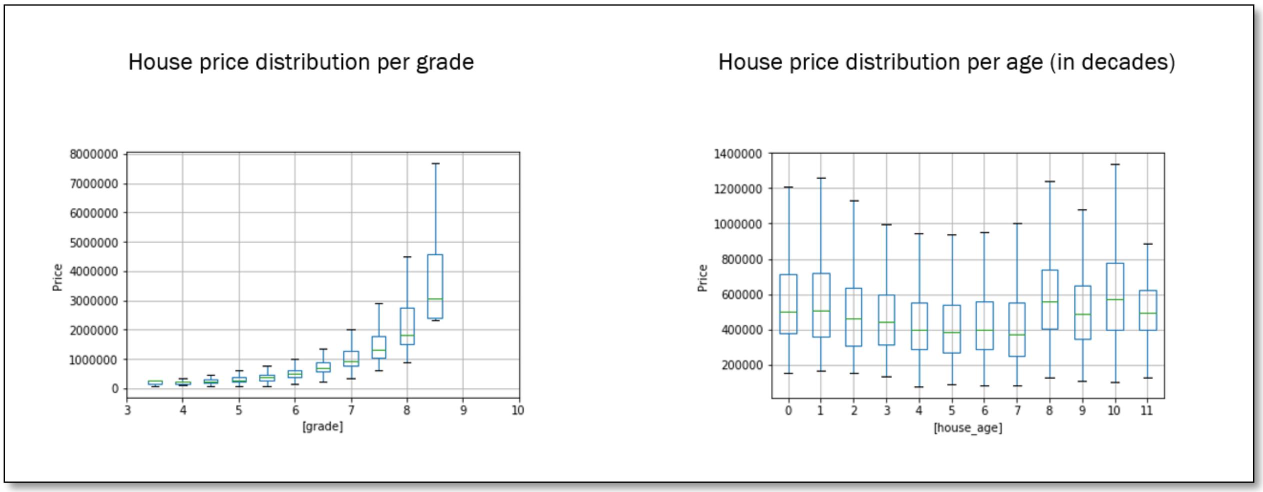


Figure A.1: House price distribution for different `age` and `grade` values

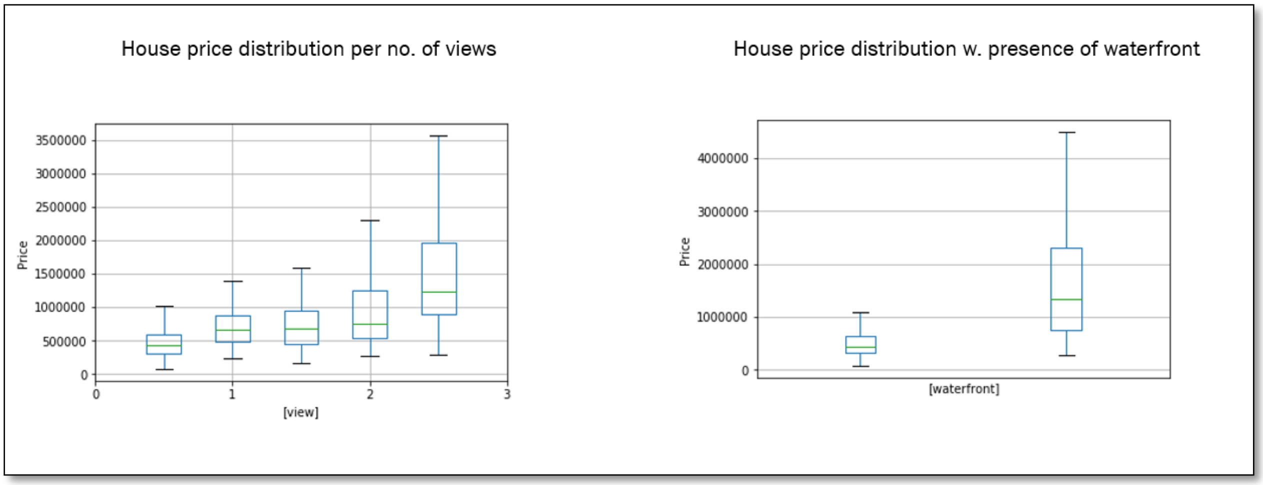


Figure A.2: House price distribution for different `view` and `waterfront` values

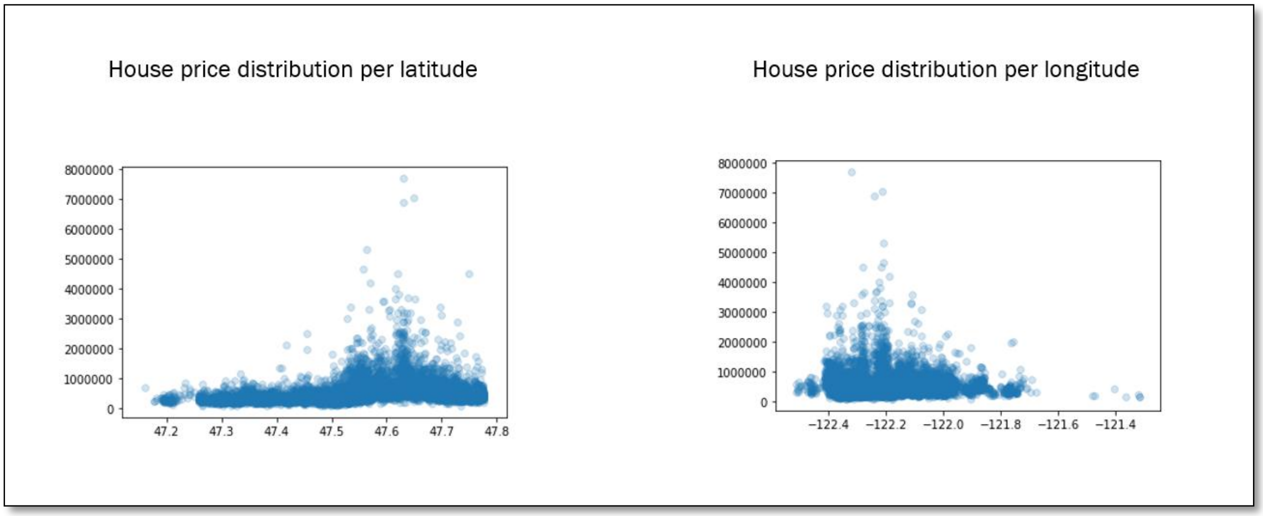


Figure A.3: House price distribution for Latitude and Longitude

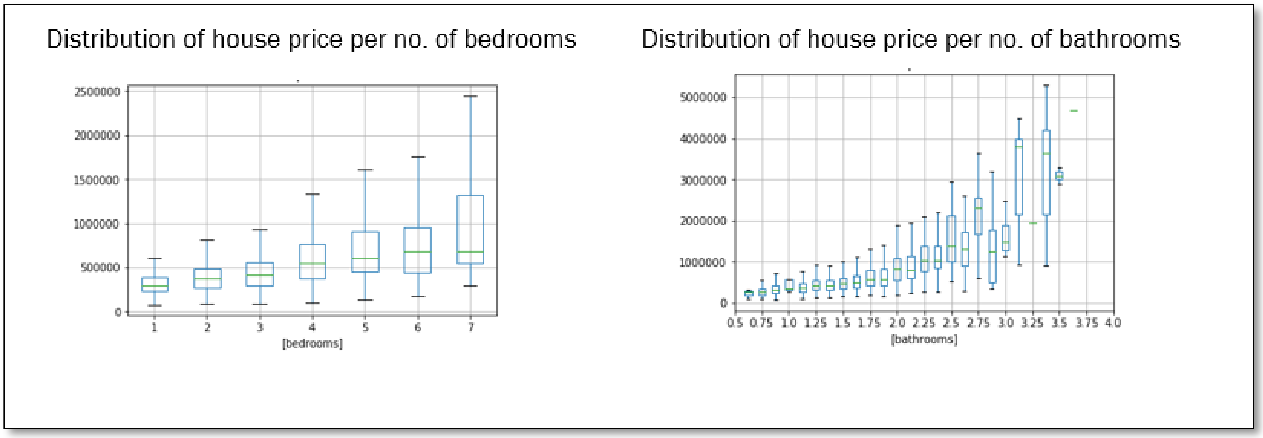


Figure A.4: House price distribution for different number of bedrooms / bathrooms